



Analytical Low-Thrust Trajectory Design

using the Simplified General
Perturbations model

J. G. P. de Jong

Analytical Low-Thrust Trajectory Design

using the Simplified General
Perturbations model

by

J. G. P. de Jong

to obtain the degree of Master of Science
at the Delft University of Technology.
to be defended publicly on Thursday December 20, 2018 at 13:00.

Student number:	4001532	
Project duration:	November 29, 2017 - November 26, 2018	
Supervisor:	Ir. R. Noomen	
Thesis committee:	Dr. Ir. E.J.O. Schrama	TU Delft
	Ir. R. Noomen	TU Delft
	Dr. S. Speretta	TU Delft

November 26, 2018

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

Frontpage picture: NASA.

Preface

Ever since I was a little girl, I knew I was going to study in Delft. Which study exactly remained unknown until the day before my high school graduation. There I was, reading a flyer about aerospace engineering and suddenly I realized: I was going to study aerospace engineering. During the bachelor it soon became clear that space is the best part of the word aerospace and thus the space flight master was chosen. Looking back this should have been clear already years ago: all those books about space I have read when growing up...

After quite some time I have come to the end of my studies. Especially the last years were not an easy journey, but I pulled through and made it to the end. This was not possible without a lot of people and I would like this opportunity to thank them here. First of all, my supervisor Ron Noomen. Thanks for your input and comments. But mostly, thank you for your understanding, patience and kind words when I needed them the most.

The graduation room on the 9th floor was the perfect place for me to work on my thesis. Some students graduated before me and left the room as friends: Laurens, Tim and Alex. And of course Evelyne, my friend throughout the entire studies, during good and bad times. Our trip to Greece was the perfect break during my thesis. The current residents of the graduation room: thank you for your company and support. Amanthla, Corn , Datta, Maaikje, Max, Raam, Sunayna and Rosalie: I am really going to miss the cakes on Wednesday afternoon. My bouldermattie and hockeyteamie Laura, thank you for proofreading parts of this thesis and listening to me during our boulder sessions. I still think that sometimes we actually talked more than we were bouldering.

Lastly, I would like to thank my mother and brother. Thank you for being there for me. For you too, the last years were not easy but both of you always helped me out when I needed it the most.

*J. G. P. de Jong,
Delft, November 2018*

Abstract

It is expected that the use of low-thrust propulsion for spacecraft in orbit around Earth will increase in the upcoming years. Compared to conventional chemical propulsion, low-thrust propulsion has a higher energy density per unit of mass, resulting in a significant reduction of propellant mass required for a given velocity increment to be gained. Due to the small acceleration force of low-thrust propulsion and the long travel time, perturbations have a large influence on a low-thrust transfer orbit. Numerical methods are required to obtain a feasible and precise solution of a trajectory problem, but these methods can be computationally time consuming. An analytical first-guess approximation, which takes perturbations into account, is thus desired. This can then be used as a base-line for the numerical optimization.

The perturbations acting on a spacecraft in an orbit around Earth are currently modelled analytically in the Simplified General Perturbations Model 4 (SGP4) model, which is used to propagate spacecraft forward in time. In this thesis research, the SGP4 method is combined with an existing analytical method for a low-thrust trajectory. This is the solution suggested by Edelbaum and improved by (Kechichian, 1997). This resulted in the SGP4-LT tool. An iterative SGP4 version was required. In the development of this tool, various modifications had to be made. In order to successfully implement this low-thrust solution, an iterative version of SGP4 was developed. To convert osculating orbital elements in mean orbital elements, an existing procedure was corrected and improved. Originally, this solution had poor convergence rates, but after the correction and modifications, a convergence rate of 99.5 % was reached for 17542 objects of the satellite catalogue. Convergence was not reached in case of combinations for inclinations smaller than 0.05° and eccentricities lower than $4 \cdot 10^{-6}$. It is expected that this is caused by floating-point errors in the conversion process.

It was shown that SGP4-LT combines perturbations and low-thrust acceleration correctly. Due to the perturbations, the transfer time will increase slightly and the evaluation of the semi-major axis and inclination showed similar behaviour for a pure Edelbaum solution. When looking at the orientation of the orbit, the right ascension of the ascending node changed with a rate as expected by SGP4 at its corresponding altitude. Normally, a time-consuming numerical integration is required to obtain data to plot the trajectory. SGP4-LT requires only 5 seconds to produce the data required to plot trajectories with flight times of 180 days.

The versatility of the SGP4-LT tool is shown by applying the tool to three different applications. Orbit raising, non-coplanar orbit changes and altitude maintenance in low Earth orbits were considered. SGP4-LT obtained expected results for the orbit raising and non-coplanar orbit change. The altitude maintenance is based on the GOCE spacecraft, which used an ion engine to fly drag-free in a low Earth orbit. It was found that for the lowest thrust acceleration possible, 27.18 kg xenon propellant was required to maintain the altitude during the mission life time. Using historical TLE GOCE data, this was found to be a reasonable first approximation of the propellant mass. Although too low, the values obtained were in the range of the Gravity field and steady-state Ocean Circulation Explorer (GOCE)'s thruster and similar results to GOCE's mission can be obtained when a higher thrust acceleration is used.

In conclusion, a correct functioning tool has been developed which obtains a first approximation solution for transfer trajectories performing whilst orbiting around Earth. SGP4-LT is fast, robust and capable of calculating ΔV budgets for maintaining the altitude in low Earth orbits.

Acronyms

CATO	Computer Algorithm for Trajectory Optimization
DFAC	Drag-Free Attitude Control
DITAN	Direct Interplanetary Trajectory Analysis
DoD	Department of Defense
ECI	Earth-Centered Inertial
EOR	Electric Orbit Raising
ESA	European Space Agency
GEO	Geostationary Earth Orbit
GOCE	Gravity field and steady-state Ocean Circulation Explorer
GPS	Global Positioning System
GSFC	Goddard Space Flight Center
GTO	Geostationary Transfer Orbit
HEO	Highly Elliptical Orbit
IPS	Ion Propulsion System
JAXA	Japan Aerospace Exploration Agency
JD	Julian Date
LEO	Low-Earth Orbit
LOTNAV	Low-Thrust Interplanetary Navigation Tool
MATLAB	MATrix LABoratory
MEO	Medium Earth Orbit
MJD	Modified Julian Date
MMO	Mercury Magnetospheric Orbiter
MPO	Mercury Planetary Orbiter
MTM	Mercury Transfer Module
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NGA	US National Geospatial-Intelligence Agency
NEP	Nuclear Electric Propulsion
NORAD	North American Aerospace Defence Command
NSSCC	National Space Surveillance Control Center

RAAN	Right Ascension of the Ascending Node
REP	Radioisotope Electric Propulsion
RIC	Radial, In-track, Cross-track
RK	Runge-Kutta
RPS	Radioisotope Power Systems
SATCAT	Satellite Catalogue
SDP4	Simplified Deep Space Perturbations
SEP	Solar Electric Propulsion
SEPTOP	Solar Electric Propulsion Trajectory Optimization Program
SGP	Simplified General Perturbations
SGP4	Simplified General Perturbations Model 4
SSN	Space Surveillance Network
STK	Systems Tool Kit
TEME	True Equator and Mean Equinox
TLE	Two Line Elements
TUDAT	TU Delft Astrodynamics Toolbox
USSTRATCOM	United States Strategic Command
XIPS	Xenon Ion Propulsion System
XIPSTOP	Xenon Ion Propulsion System Trajectory Optimization Program

Nomenclature

Greek Symbols

Symbol Description

α	Mean angular position		rad
β	Thrust yaw angle		rad
δ	Angle of velocity vector		rad
ϵ_0	Electric constant	8.854e-12	F/m
η	Engine efficiency		-
γ	Flight path angle		rad
μ	Gravitational parameter		m ³ /s ²
ϕ	Latitude		rad
ρ	Density of atmosphere		kg/m ³
σ	Stefan-Boltzmann constant	5.670367e-8	J/(sm ² K ⁴)
θ	Longitude		rad
θ	True anomaly		rad
ω	Argument of pericenter		rad
Ω	Right ascension of ascending node		rad

Roman Symbols

Symbol Description

a	Semi-major axis		m
a_T	Thrust acceleration		m/s ²
c	Speed of light	299792458	m/s
C_D	Drag coefficient		-
C_R	Reflectivity coefficient		-
e	Eccentricity		-
\mathbf{e}	Unit vector		-
E	Eccentric anomaly		rad
E	Power density		W/m
\mathcal{E}	Total specific energy		m ² /s ²
\mathbf{f}	Perturbing forces acceleration vector		m/s ²

f_{mee}	Modified equinoctial element		-
g	Local gravitational acceleration		m/s ²
$g_{1,0}$	First Gauss geomagnetic coefficient	-2.96e-5	T
g_{mee}	Modified equinoctial element		-
G	Universal gravitational constant	6.670e-11	m ³ /(kgs ²)
h_{ee}	Equinoctial element		-
h_{mee}	Modified equinoctial element		-
i	Inclination angle		rad
I_{sp}	Specific impulse		s
J	Model parameter gravity field potential		-
k_{ee}	Equinoctial element		-
k_{mee}	Modified equinoctial element		-
L	True longitude		rad
m	Mass		kg
\dot{m}	Mass flow		kg/s
M	Mass		kg
M	Mean anomaly		rad
n	Mean angular motion		rad/s
p	Semi-latus rectum		m
P	Input power		W
$P(x)$	Legendre polynomial		-
P_0	Solar constant		N/m ²
q	Electric charge		C
q_{ee}	Equinoctial element		
r	Radial distance		m
\mathbf{r}	Position vector		m
R	Perturbing potential		m ² /s ²
R	Radius of the Earth	6378.135e3	m
S	Effective cross-sectional area		m ²
t	Time		s
\mathbf{T}	Thrust vector		N
T	Temperature		K
T	Thrust		N
U	Gravitational potential		m ² /s ²

U^*	Electrical potential difference	V
V	Velocity	m/s
V_e	Exit velocity	m/s
W	Power density	W/m ²
x	Coordinate in x-direction	m
\mathbf{X}	State vector	m, m/s
y	Coordinate in y-direction	m
z	Coordinate in z-direction	m

Superscripts

Symbol Description

\square'	First derivative w.r.t. independent variable
\square''	Second derivative w.r.t. independent variable
$\dot{\square}$	First derivative w.r.t. time
$\ddot{\square}$	Second derivative w.r.t. time

Subscripts

Symbol Description

0	Initial value
θ	Component along normal direction
a	Apocenter
c_1	Initial circular orbit
d	body d
E	Earth
f	Final
h	Component along angular momentum vector
i	Body i
i	Iteration number
j	Iteration number
m	Degree
max	Maximum value
min	Minimum value
n	Degree
p	Pericenter
r	Component along radial vector
tot	Total
z	Component in axial direction

Contents

Abstract	v
1 Introduction	1
1.1 Research Questions and Objective	2
1.2 Structure	2
2 Orbital Mechanics	3
2.1 Reference Frames	3
2.1.1 Inertial Reference Frame	3
2.1.2 Heliocentric Ecliptic Reference Frame	4
2.1.3 Planetary Reference Frame.	4
2.1.4 Body-fixed Reference Frame	5
2.1.5 Time Concepts.	5
2.2 Coordinate Systems	5
2.2.1 Cartesian Coordinates	5
2.2.2 Spherical Coordinates	6
2.2.3 Polar Coordinates	6
2.2.4 Cylindrical Coordinates	6
2.2.5 Kepler Coordinates.	6
2.3 Newton's Laws	6
2.4 Equations of Motion	8
2.5 Keplerian Orbits	8
2.6 Perturbations	11
2.6.1 Perturbed Orbits	11
2.6.2 Gravity Field Forces	13
2.6.3 Aerodynamic Forces	14
2.6.4 Third-Body Perturbations	15
2.6.5 Radiation Pressure	16
2.6.6 Electromagnetic forces.	17
2.6.7 Comparison	18
2.7 Transfer Orbits	18
2.8 Transfer between two non-coplanar orbits	19
3 Low-Thrust Propulsion	23
3.1 Heritage.	23
3.1.1 Deep Space 1.	23
3.1.2 SMART-1	24
3.1.3 BepiColombo	26
3.1.4 Commercial Satellites	28
3.1.5 GOCE	28
3.2 Electric Propulsion Systems.	29
3.2.1 Characteristics.	31
3.3 Dynamics	31
3.3.1 Thrust Profiles	32
4 Simplified Perturbations Model	33
4.1 Orbital Propagation	33
4.2 History	34
4.3 Two Line Elements	36
4.3.1 Retrieving TLE Data	36

4.4	Algorithm	37
4.5	Behaviour of orbital elements	38
4.6	Software	40
4.7	Conclusions.	40
5	Methodology	43
5.1	Edelbaum's low-thrust transfer orbit solutions	44
5.1.1	Edelbaum's solution	44
5.1.2	Edelbaum's solution with Earth-shadow eclipses.	46
5.2	Low-Thrust Module.	46
5.3	Iterative SGP4 Algorithm	48
5.3.1	Corrected Andersen's Method	50
5.3.2	Modified Walter's Method	51
5.4	SGP4-LT.	53
5.4.1	Delta V.	54
5.4.2	Limitations of SGP4-LT	54
5.5	Conclusions.	54
6	Numerical Methods	57
6.1	Integrators	57
6.1.1	Euler Method	57
6.1.2	Higher-order Runge-Kutta Methods	58
6.2	Root-finding Algorithms	59
6.2.1	Bisection Method	59
6.2.2	Newton-Raphson method	59
6.2.3	Secant Method.	60
6.2.4	Regula Falsi Method	60
6.2.5	Comparison and Discussion	61
7	Verification and Validation	63
7.1	Edelbaum's Solution	63
7.1.1	Numerical Integration of Edelbaum's Solution	64
7.2	SGP4	67
7.3	Validation of the Modified Walter Method.	67
7.3.1	Implementation of the Satellite Catalogue	68
7.3.2	Conclusions	69
7.4	Iterative SGP4 behaviour	74
7.5	SGP4-LT: comparison to analytical solution.	77
7.6	SGP4-LT: Comparison to numerical results	78
7.7	Conclusions.	81
8	Practical Applications	85
8.1	Orbit Raising	85
8.2	Transfer between two non-coplanar orbits	86
8.3	Station-keeping in Low Earth Orbit	88
8.4	Conclusion	90
9	Conclusions and Recommendations	95
9.1	Conclusions.	95
9.1.1	Simplified General Perturbations Model	96
9.1.2	Analytical Low-Thrust Solution	96
9.1.3	Iterative SGP4	96
9.1.4	SGP4-LT	96
9.1.5	Verification	97
9.1.6	Practical Application	97
9.1.7	Research Objective.	98
9.2	Recommendations and Future Research	98

A	Transformations	99
A.1	Frame Transformations	99
A.2	Coordinate Transformations	99
B	Gauss' Form of Lagrange Planetary Equations for near-circular orbits	103
C	SGP4-LT Version 1	105
D	Used Two Line Elements Sets	109
E	Results of validation of iterative SGP4	111
	Bibliography	115



Introduction

"That's one small step for man, one giant leap for mankind"

—Neil Armstrong, 1930-2012

Since the dawn of time, humankind has always been fascinated by planets, the Sun, and other stars. In the 1700's, humankind started to explore the atmosphere and in 1903, the first aircraft had lifted-off. In 1957 Sputnik-1, the first artificial satellite, was launched and the age of space exploration truly began. Not only were spacecraft launched on a journey to the outer edges of the Solar System and beyond, but the number of spacecraft launched to stay in orbit around Earth increased rapidly. In orbit, the Earth, as well as the universe, could be studied in ways that were not possible before.

To get a satellite in the correct orbit, and to maintain this orbit, initially conventional chemical propulsion was mostly used. However, over the past two decades the development and use of spacecraft with low-thrust engines has increased. Low-thrust engines have a higher energy density per unit of mass, resulting in a higher exhaust velocity compared to conventional high-thrust engines. Despite the higher exhaust velocities that can be reached, the actual thrusting force is low compared to conventional chemical propulsion, hence the name *low-thrust propulsion*. Consequently, a longer time of thrusting is required before specific required high velocity changes are reached and an other alternative name is used *continuous thrust propulsion*. Due to these higher exhaust velocities, significant mass savings can be achieved, which results in a reduction of costs (Jahn, 2006).

Recently, low-thrust propulsion was mostly used in interplanetary scientific missions as travel time would be less critical than it would be for commercial satellites around Earth. However, an increasing number of benefits for this type of propulsion has been found for orbits around Earth. Currently, the first spacecraft that performed a orbit transfer using pure low-thrust propulsion are in orbit. The GOCE spacecraft used a low-thrust engine to maintain its altitude in its low Earth orbit (Wallace et al., 2011). Due to the mentioned advantages of low-thrust propulsion over chemical propulsion, it is expected that its use will increase in the future (Jahn, 2006).

The thrusting over long periods of time complicates the dynamics of spacecraft using low-thrust propulsion. Numerical methods are required to obtain a feasible and precise solution of a trajectory problem, but these methods are computationally time consuming. It is thus preferred to obtain a fast yet reliable first-guess approximation of a trajectory, which is then used as a baseline for the fully optimized trajectory. Around Earth, perturbations act on a spacecraft. Due to the small thrust acceleration, these perturbations can have a large influence on the trajectory. Including these perturbations in an analytical low-thrust trajectory design tool would increase the usability of such a tool.

The Simplified General Perturbations 4 (SGP4) model analytically propagates the state of an object in orbit around Earth while taking perturbations into account. It was realized that the low-thrust acceleration can actually be seen as a perturbation and thus it should be possible to include this 'perturbation' in the SGP4 model. This was succeeded already by (Shen et al., 2015), but their method of successful implementation was

not released publicly, probably due to the fact that these authors were employees of the Air Force Research Laboratory of the United States of America. Thus, there was no SGP4 algorithm with low-thrust propulsion available. Therefore the SGP4-LT tool is developed which is presented in this thesis report.

1.1. Research Questions and Objective

The main research question to be answered during this thesis research is:

To what extent would the SGP4 model adapted for low-thrust propulsion perform, compared to numerically low-thrust Earth-orbit transfer trajectory design methods?

This research question can be divided into two subquestions, each of which contains further subquestions.

1. How does SGP4 work?
 - (a) How is the SGP4 algorithm structured?
 - (b) How do orbital perturbations affect the motion of a spacecraft modelled in SGP4?
 - (c) What are the main sources of propagation uncertainty using SGP4?
2. How can SGP4 be used to design low-thrust trajectories?
 - (a) What kind of low-thrust trajectory design method is required to work in cohesion with SGP4?
 - (b) Which modifications have to be made to the original SGP4 algorithm in order to be able to implement a low-thrust module?
3. Are the results obtained by SGP4-LT representative for a first approximation trajectory design tool?
 - (a) How do small changes in (the direction of) the low-thrust force affect the performance of the optimization tool?
 - (b) What are the limitations of SGP4-LT?
 - (c) What is the accuracy of SGP4-LT compared to other methods?
 - (d) For which purposes can SGP4-LT be used?

Answering the research question and its subquestions leads to the research objective which reads as follows.

Extend the existing SGP4 model with a low-thrust propulsion module and develop a fast, reliable and robust optimization tool which uses this extended model.

1.2. Structure

This report is the result of all steps taken in order to answer the research question and its subquestions.

In Chapter 2, basic astrodynamics is introduced, to become familiar with the concepts used throughout this thesis research. Next to basic orbital mechanics, it consists of perturbation theory in order to make the correct assumptions during this research. Chapter 3 provides basic knowledge on low-thrust propulsion, to get an understanding of the values and limits of low-thrust propulsion. The SGP4 model is described in Chapter 4. To add a low-thrust module to SGP4, the algorithm must be completely understood.

The basics are now covered and the methodology/design of the SGP4-LT tool is explained step-by-step in Chapter 5. Then, in Chapter 6 the numerical methods are explained that are required for validation purposes. This validation is performed and its results are discussed in Chapter 7. The SGP4-LT tool is used for three different applications which is extensively described in Chapter 8. Finally, in Chapter 9 the conclusions of this thesis research are presented and recommendations for further research are given.

2

Orbital Mechanics

"For if anyone should come up to the top of the air or should get wings and fly up, he could lift his head above it and see, as fishes lift their heads out of the water and see the things in our world, so he would see things in that upper world; and, if his nature were strong enough to bear the sight, he would recognize that that is the real heaven."

—Socrates, ca. 469 - 399 BC, in Plato's Phaedo

Plato, an ancient Greek philosopher who lived around 400 BC, and a student of Socrates, already had thoughts about the movement of celestial bodies. He believed Earth was at the center of a celestial sphere and other celestial bodies moved in a uniform circular motion around it. It took almost 2000 years before Copernicus (1473 - 1543) came up with the theory of a heliocentric system. Galileo Galilei (1564 - 1642) confirmed this theory with his observations through the just-invented telescope. Meanwhile, Johannes Kepler (1571- 1630) determined his famous three empirical laws on planetary motion, which he derived from the precise celestial observations by Tycho Brahe (1546 - 1601). Finally, Isaac Newton (1643 - 1727) was able to explain the physics of *why* the celestial bodies are moving as they do in the heliocentric system. The findings of these scientists formed the foundation for the orbital mechanics as we know it today.

This chapter gives an insight into (the basics of) orbital mechanics to understand its principles and implications. This knowledge is required to establish proper assumptions for the thesis research. First, reference frames and various coordinate systems are discussed in Sections 2.1 and 2.2. This is followed by Newton's Laws and the equations of motion in Sections 2.3 and 2.4. Kepler orbits and its elements are discussed in Section 2.5. Section 2.6 contains the discussion on perturbed orbits and is followed by a brief explanation of coplanar transfers in Section 2.7. This chapter concludes on non-coplanar transfers in Section 2.8.

2.1. Reference Frames

To describe and obtain the position and velocity of an object in space, the definition of the used reference frame is required. Multiple reference frames exist and depending on the field of interest a certain reference frame can be used. For example, a geodetic reference frame is typically used for satellites orbiting Earth. For interplanetary travel heliocentric reference frames are considered a more convenient option and body-centered reference frames are used for projects which focus solely on a body (like the investigation of thermal loads on a re-entry vehicle).

Several (inertial) reference frames are discussed in this section.

2.1.1. Inertial Reference Frame

The definition of an inertial reference frame can directly be derived from Newton's first Law: these are reference frames with respect to which a particle remains in a state of rest or of uniform motion in a straight line if no external force acts on it (see Section 2.3 for an explanation of Newton's Laws). Of course, in reality every reference frame placed e.g. in the center of the Sun would undergo certain accelerations and rotations and

is thus not truly inertial. However, these motions are significantly smaller than the motion of the spacecraft. Therefore the reference frame is assumed to be inertial.

2.1.2. Heliocentric Ecliptic Reference Frame

In the case of interplanetary travel, a heliocentric non-rotating reference frame can be used to describe the motion of celestial bodies and spacecraft around the Sun. This frame has its origin at the center of the Sun and its XY -plane is the ecliptic plane. The corresponding Z -axis points perpendicular to the ecliptic plane (positive North). The X -axis is aligned with the vernal equinox, which is the intersection of the ecliptic plane of the Sun and the equatorial plane of the Earth. Since the location of this intersection varies due to the precession and nutation of Earth's rotation axis, the vernal equinox at January 1, 2000 at noon (J2000) is often chosen as a reference.

A rotating heliocentric reference frame would be a complex option for orbits around Earth and is therefore not considered any further. Figure 2.1 shows a heliocentric ecliptic reference frame.

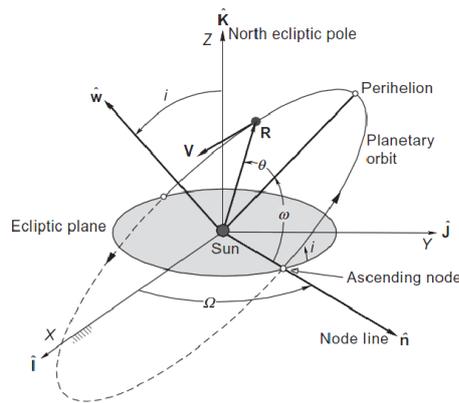


Figure 2.1: Orbital path in a heliocentric ecliptic reference frame (Conte and Spencer, 2015).

2.1.3. Planetary Reference Frame

A planetary reference frame, e.g. Earth Fixed reference frame for Earth, can be used for describing the motion of a spacecraft around a planet. Using this reference frame for interplanetary flight will lead to increased complexity due to the rotation of the planet around its star.

For example, the origin of a non-rotating planetary ecliptic reference frame is the center of mass of the planet. The XY -plane lies in the orbital plane of the planet around its star. In the case of Earth, this plane is known as the ecliptic. The Z -axis points perpendicular to the XY -plane and to the north, and the X -axis points towards the equinox.

A non-rotating planetary reference frame is often used to describe the motion of a spacecraft around a planet and when the motion of an object with respect to the surface is important. This reference frame is shown in Figure 2.2.

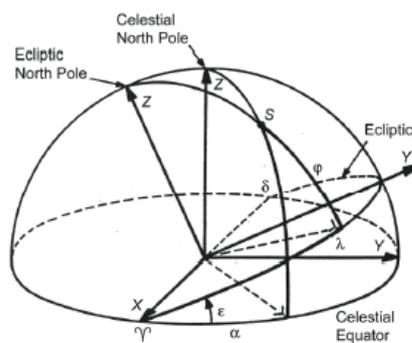


Figure 2.2: The geocentric ecliptic equatorial reference frame and coordinates (Wakker, 2010).

Examples of an Earth-Centered Inertial (ECI) reference frame are J2000 and TEME. J2000 is defined with the Earth's mean equator and Equinox at 12:00 Terrestrial Time on January 1, 2000, with its x-axis aligned with the mean equinox. The z-axis is aligned with the Earth's spin axis. TEME is used for the North American Aerospace Defence Command (NORAD) TLE, which is relative to the true equator and mean equinox of the element set.

2.1.4. Body-fixed Reference Frame

Body-fixed reference frames, like the satellite-based coordinate system, are used to describe the relative motion of spacecraft or predicted errors. The RSW reference system is discussed in this section (L. et al., 2017). The R -axis always points away from the Earth along the radius vector towards the satellite. The S -axis (along-track) is perpendicular to the radius vector and is positive in the direction of the velocity vector. The W -axis (cross-track) is normal to the orbital plane. Figure 2.3 shows the unit vectors of the RSW reference system of a spacecraft in orbit around Earth.

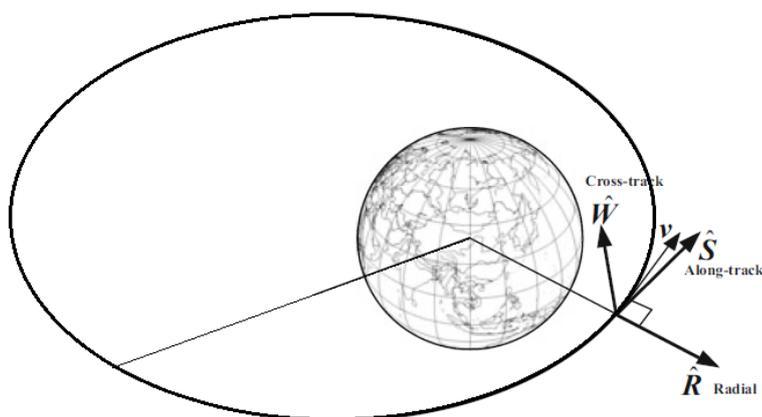


Figure 2.3: The body-fixed reference system RSW, after (L. et al., 2017).

2.1.5. Time Concepts

Next to its position, also the moment in time is an important factor to obtain the necessary information on the orbit of a spacecraft. Time is often represented in Modified Julian Date (MJD).

The Julian Date (JD) was introduced by J.J. Scaliger in 1583. Julian Date counts onwards from day 0 and therefore it is useful in calculations. 4713 BC, January 1 at noon was chosen as starting point. The Julian period is a combination of the solar cycle (solar cycle of the Julian Calendar, 28 years), the Metonic cycle (19 years) and the Roman indication cycle (15 years). It consists of 7980 Julian Years of exactly 365.25 days. Since our current time consists of rather large numbers, MJD was introduced:

$$MJD = JD - 240000.5 \quad (2.1)$$

Another often used variant of the JD is MJD2000. This concept of time starts at the astronomical epoch January 1, 2000, 12:00h:

$$MJD2000 = MJD - 51544.5 \quad (2.2)$$

2.2. Coordinate Systems

To describe positions, velocities, and accelerations with respect to the reference frames considered in the previous section, coordinate systems are required. This section discusses several options.

2.2.1. Cartesian Coordinates

The Cartesian coordinate system uses the axes as defined in the explained reference frames in Section 2.1, called x -, y - and z - axis. The coordinates x , y and z can lie anywhere in the interval $(-\infty, +\infty)$.

Velocity components are also directed along the x , y , z - axes and are expressed in the form \dot{x} , \dot{y} , \dot{z} . The state

vector in Cartesian coordinates then becomes:

$$\mathbf{X}_{cart} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \quad (2.3)$$

Figure 2.4a shows the Cartesian coordinate system with the unit vectors \mathbf{e}_x , \mathbf{e}_y and \mathbf{e}_z .

2.2.2. Spherical Coordinates

Spherical coordinates are typically used for 3D applications. This system consists of the radial distance r , the azimuth angle θ in the xy -plane measured counter-clockwise from the x -axis, and the latitude ϕ . This angle is positive when z is positive and ranges from $-\pi/2$ to $\pi/2$. Figure 2.4b presents an example of the spherical coordinate system. The unit vectors are \mathbf{e}_r , \mathbf{e}_θ and \mathbf{e}_ϕ .

The state vector in spherical coordinates is expressed as:

$$\mathbf{X}_{spherical} = [r \ \theta \ \phi \ \dot{r} \ \dot{\theta} \ \dot{\phi}]^T \quad (2.4)$$

2.2.3. Polar Coordinates

Polar coordinates are used in a 2D space where the position of an object is described by radius r and polar angle θ . The radius r is the radial distance from the origin to the object and θ is the counter-clockwise angle from the positive x -axis to the radius vector. r has to range between $\in [0, +\infty)$, and $\theta \in [0, 2\pi)$. Figure 2.4c shows the polar coordinate system. The state vector in polar coordinates is shown in Equation 2.5.

$$\mathbf{X}_{polar} = [r \ \theta \ \dot{r} \ \dot{\theta}]^T \quad (2.5)$$

2.2.4. Cylindrical Coordinates

Cylindrical coordinates can be used for 3D systems and consist of the angle θ and the radial distance r as explained in Section 2.2.3 and a height z is added to obtain a 3D coordinate system. z ranges from $(-\infty, +\infty)$. The state vector in cylindrical coordinates can be written as:

$$\mathbf{X}_{cylindrical} = [r \ \theta \ z \ \dot{r} \ \dot{\theta} \ \dot{z}]^T \quad (2.6)$$

In Figure 2.4d the cylindrical coordinate system is shown.

2.2.5. Kepler Coordinates

The position and motion of a spacecraft are often described using Kepler elements. In the previously explained coordinate systems, the position of the spacecraft is defined by two or three coordinates in 2D or 3D, respectively. The state in Kepler coordinates is described by six elements. Two describe the orbit of the spacecraft, three elements describe the orientation of the orbit relative to the equator and vernal equinox, and the last element describes the position of the spacecraft in the orbit. This combination gives a full description of both position and velocity. For an extensive explanation the reader is referred to Section 2.5.

2.3. Newton's Laws

Orbital mechanics is based on four laws, of which the three laws of motion, and the Law of Gravitation, formulated by Sir Isaac Newton (1643-1727) in *Philosophiae Naturalis Principia Mathematica*, 1687. The laws read as follows (Wakker, 2010).

- *First Law*: Every particle continues in its state of rest, or of uniform motion in a straight line, unless it is compelled to change that state by forces acting on 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 reference frame 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.
- *Law of 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.

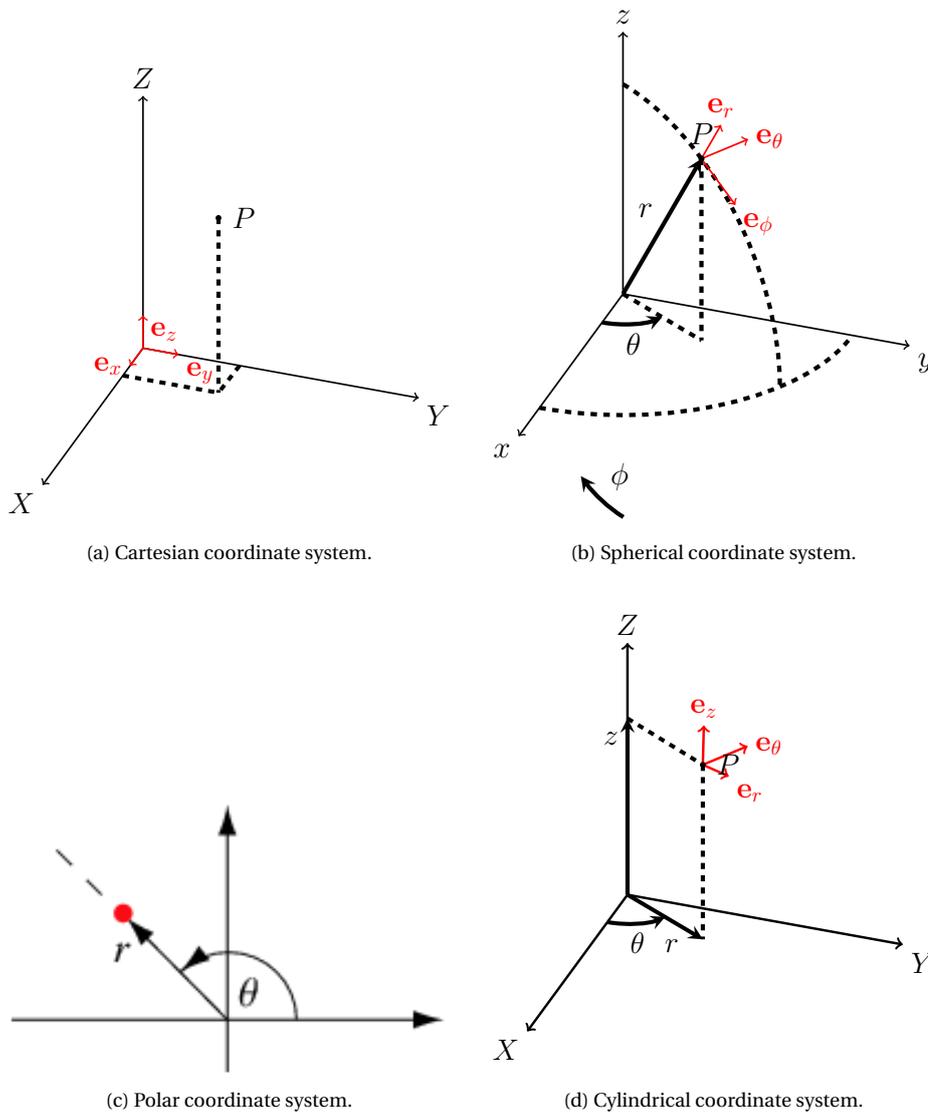


Figure 2.4: Various coordinate systems

When applying Newton's laws, it is assumed that all bodies act as point masses. The second law can be expressed as:

$$\mathbf{F} = m\ddot{\mathbf{r}} \quad (2.7)$$

In this equation, \mathbf{F} is the sum of all external forces acting on a particle, m is the mass of the particle and $\ddot{\mathbf{r}}$ is the acceleration of the particle. The law of gravitation can be mathematically expressed as:

$$\mathbf{F} = -G \frac{m_1 m_2}{r^3} \mathbf{r} \quad (2.8)$$

where \mathbf{F} is the force acting on particle m_2 generated by m_1 and is acting along a straight line connecting the two masses. G is the universal gravitational constant, r is the distance between the two masses and \mathbf{r} is the vector from mass m_1 to m_2 . A graphical interpretation of the gravitational law can be found in Figure 2.5.

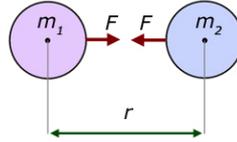


Figure 2.5: Newton's gravitational law (credit: a-levelphysicstutor.com).

2.4. Equations of Motion

With Newton's Laws explained in the previous section, the equations of motion can be obtained. When the distance between the object and primary attracting celestial body is significantly smaller than the distance towards other perturbing bodies, the many-body problem is reduced to a two-body problem. The equation of motion of an object around the primary attracting body is then given by (Wakker, 2010)

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{r^3} \quad (2.9)$$

where the standard gravitational parameter equals $\mu = GM$, with G the universal gravitational constant and M the mass of the primary body.

When thrust is applied Equation 2.9 is expanded to:

$$\ddot{\mathbf{r}} = -\mu \frac{\mathbf{r}}{r^3} + \frac{\mathbf{T}}{m} \quad (2.10)$$

2.5. Keplerian Orbits

As discussed in the previous section a two-body problem is assumed. An orbit is considered purely Keplerian when only the gravitational attraction of the primary body, which is assumed to be symmetrical and much more massive than the orbiting body, exerts a force on the object. These orbits are conic sections and exist in circular, elliptic, parabolic and hyperbolic form, as can be seen in Figure 2.6.

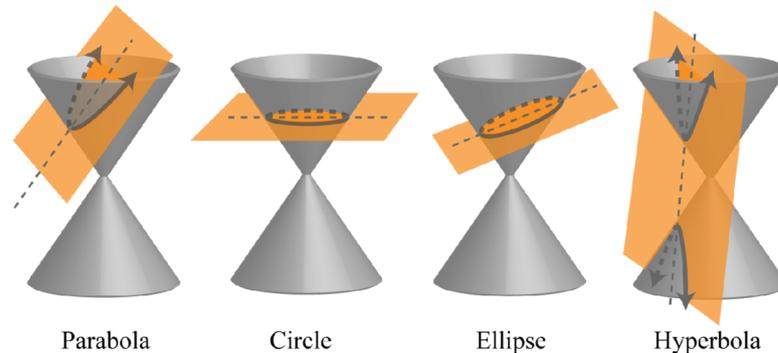


Figure 2.6: Conic sections for Keplerian orbits. (Redden, 2012).

Kepler derived the following three laws (Ley et al., 2008) for the motion of planets around the Sun.

- *First Law:* The orbit of a planet around the Sun is a conic section with one focus which coincides with the center of the Sun.
- *Second Law:* The radius vector from the Sun's center to the planet sweeps over equal areas in equal time intervals.
- *Third Law:* The squares of the orbital periods of two planets are proportional to the cubes of their average distance to the Sun's center.

The shape of the orbit is described by the eccentricity e and the size of the orbit is determined by the semi-major axis a . The right ascension of the ascending node Ω and the inclination i define the attitude of the orbit around the main body. The angle Ω is measured from the vernal equinox to the ascending node of the orbit. The inclination i is the angle between the equatorial plane and orbital plane. The rotational orientation of the semi-major axis is fixed by the argument of periaapsis ω and the exact position of the body in its orbit is given by the true anomaly θ . The shape and attitude of an orbit are shown in Figure 2.7.

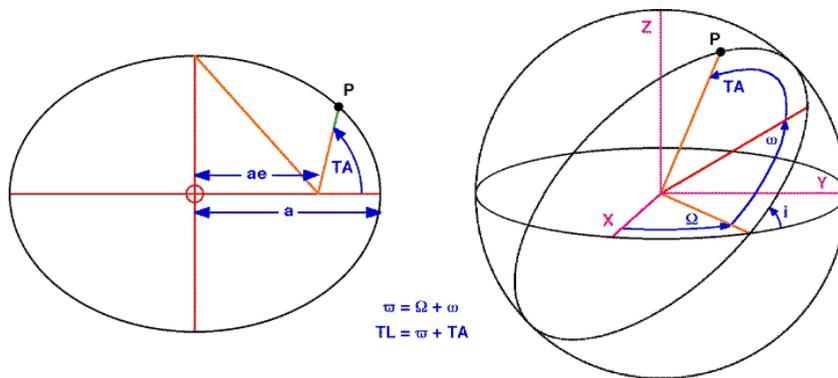


Figure 2.7: Keplerian orbit and elements (credit: astrograv.co.uk)

The state vector for a spacecraft in Keplerian orbital elements is given by:

$$\mathbf{X}_{kepler} = [a \ e \ i \ \omega \ \Omega \ \theta/M]^T \quad (2.11)$$

where θ or mean anomaly M can be used to fix the position of the spacecraft.

The relation of the variation of the velocity of a body in an orbit is known as the vis-viva equation:

$$V^2 = \mu \left(\frac{2}{r} - \frac{1}{a} \right) \quad (2.12)$$

Using orbital elements a relation (Equation 2.13) can be found between the position in the orbit and time. Due to its complicated form it is hard to use in both analytical and numerical analyses.

$$t - \tau = \sqrt{\frac{a^3}{\mu}} \left[2 \arctan \left(\sqrt{\frac{1-e}{1+e}} \tan \frac{\theta}{2} \right) - e \sqrt{1-e^2} \frac{\sin \theta}{1+e \cos \theta} \right] \quad (2.13)$$

A less complicated expression is obtained by introducing the eccentric anomaly E . E is constructed by using a circle around the ellipse, as shown in Figure 2.8. Also the expression for the radius can be written as:

$$r = a(1 - e \cos E) \quad (2.14)$$

From Figure 2.8 one can derive the following expression (Wakker, 2010)

$$\tan \frac{\theta}{2} = \sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \quad (2.15)$$

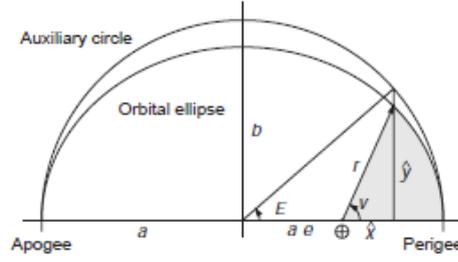


Figure 2.8: Elliptical orbit with auxiliary circle and definition of eccentric anomaly E (Wakker, 2010).

The relation between the eccentric anomaly and time is derived by (Wakker, 2010) and is given by:

$$E - e \sin E = \sqrt{\frac{\mu}{a^3}} (t - \tau) \quad (2.16)$$

Substituting $n = \sqrt{\frac{\mu}{a^3}}$ gives:

$$E - e \sin E = n(t - \tau) = M \quad (2.17)$$

The physical meaning of the orbital elements is a great advantage when using them in general. Ω becomes indeterminate as i tends to zero, which occurs when the orbit and the equatorial plane are aligned. ω becomes indeterminate when e tends to zero. In addition the equations of motion can easily become singular when using these elements. Equinoctial elements, first used by Lagrange, are well adapted to orbits with small eccentricities and inclinations. This makes them a perfect candidate to be used in the integration of orbits under the influence of perturbations. The equinoctial elements are defined as follows (Broucke and Cefola, 1972).

$$a_{ee} = a \quad (2.18)$$

$$h_{ee} = e \sin(\omega + \Omega) \quad (2.19)$$

$$k_{ee} = e \cos(\omega + \Omega) \quad (2.20)$$

$$p_{ee} = \tan(i/2) \sin \Omega \quad (2.21)$$

$$q_{ee} = \tan(i/2) \cos \Omega \quad (2.22)$$

$$\lambda_0 = M_0 + \omega + \Omega \quad (2.23)$$

In certain cases it is desired to use a so-called 'fast variable', or phase angle, as the sixth element. This phase angle varies much more with time compared to the other elements, hence the name 'fast variable'. Besides that, the mean longitude λ_0 is replaced by the true longitude L , which fixes the position in orbit. Last, the semi-major axis is replaced by the semi-latus rectum p , resulting in a set of non-singular equations of motions (with the exception of $i = \pi$). The modified equinoctial elements are expressed by (Walker et al., 1985):

$$p_{mee} = a(1 - e^2) \quad (2.24)$$

$$f_{mee} = e \cos(\omega + \Omega) \quad (2.25)$$

$$g_{mee} = e \sin(\omega + \Omega) \quad (2.26)$$

$$h_{mee} = \tan\left(\frac{i}{2}\right) \cos \Omega \quad (2.27)$$

$$k_{mee} = \tan\left(\frac{i}{2}\right) \sin \Omega \quad (2.28)$$

$$L_{mee} = \Omega + \omega + \theta \quad (2.29)$$

2.6. Perturbations

In the material discussed in the previous sections it was assumed that the two bodies are point masses and the only acting forces are the gravitational forces between the two bodies. This resulted in a perfect Keplerian orbit. In reality however, celestial bodies and spacecraft are not point masses and the motion of the spacecraft is influenced by perturbing forces. These perturbing forces differ in magnitude and direction, and can be neglected depending on the scale of the mission.

This section starts with the discussion on perturbed orbits and is followed by the actual perturbing forces. It is stated whether a perturbing force is taken into account in the SGP4 model (Chapter 4).

2.6.1. Perturbed Orbits

The motion of a satellite relative to a non-rotating, geocentric equatorial reference frame undergoing perturbing forces can be written as (Wakker, 2010)

$$\frac{d^2 \mathbf{r}}{dt^2} + \frac{\mu}{r^3} \mathbf{r} = -\nabla R + \mathbf{f} \quad (2.30)$$

In the above equation, R represents the perturbing potential and describes all perturbing forces that can be expressed by a potential function. \mathbf{f} describes all other perturbing forces.

Looking at Equation 2.30, the gravitational attraction of the Earth will be the main force acting on an object which is near Earth. This gravitational acceleration, which is on the left-hand side of the equation, represents a spherical, homogeneous Earth. The perturbations due to Earth's asymmetry and non-homogeneity are represented by ∇R or \mathbf{f} . Other perturbations that can be included in the right-hand side of the equation are the gravitational attraction by third-body objects, the drag force due to the atmosphere of the Earth, and solar radiation pressure.

These forces can have a secular, short-periodic or long-periodic effect on an orbital element. When the perturbing effect on an orbital element is constant in time this is called secular motion. Short-periodic effects can be averaged out after only a short period of time, whereas long-period effects require a longer period of time to be averaged out. An example of the three different effects on an orbital element can be found in Figure 2.9.

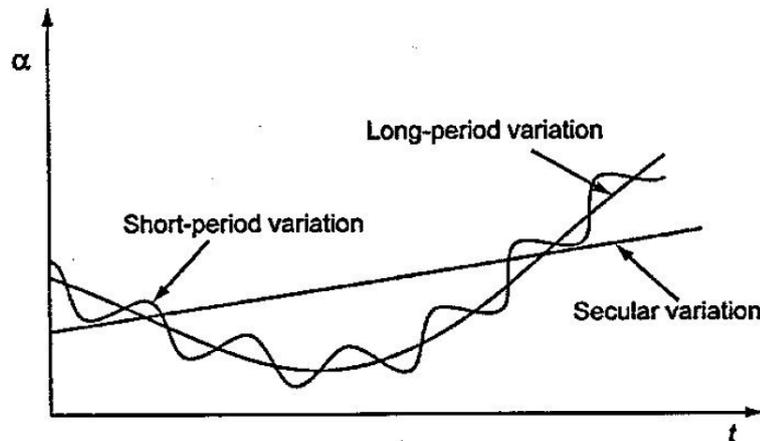


Figure 2.9: Secular, short-period and long-period effects on an orbital element α (Wakker, 2010).

Equation 2.30 cannot be solved analytically and in general there are three classical methods known to approximate the solutions. These methods are the method of Cowell, the method of Encke, and the main subject of this section, the method of variation of parameters, developed by J.L. Lagrange.

When at a certain time the position and velocity are known, the orbit can be completely represented in any coordinate system as introduced in Section 2.2. In the case of the absence of perturbations, a , e , i , Ω , ω and τ are constant for the entire orbit. In the case of a perturbed orbit, each position in this orbit has its own fictitious momentary Kepler orbit. This orbit is called an *osculating orbit* and for each moment in time, a different osculating orbit can be found. Figure 2.10 shows a true orbit and two osculating orbits to clarify this concept.

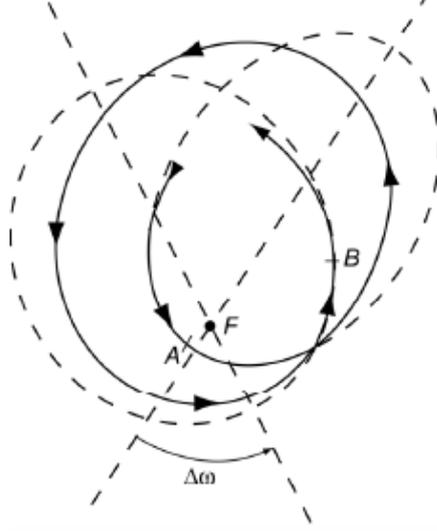


Figure 2.10: True orbit (solid line) with two osculating orbits (dashed lines) at locations A and B (Wakker, 2010).

In a perturbed orbit, the motion of the satellite is considered as a continuous transition between osculating orbits, meaning that the orbital elements show a variation over time. The possibility of obtaining approximate analytical solutions is a main advantage for this method.

After various derivations as shown in (Wakker, 2010), a set of differential equations is found that expresses the effect of a perturbing force on the osculating elements. This set is called Lagrange's Planetary Equations and is only valid for all perturbing forces that can be described through a perturbing potential. J.C.F. Gauss developed Lagrange's planetary equations further and obtained the now so-called Gauss' form of the Lagrange Planetary Equations, which can be found in Equations 2.31 - 2.36 (Wakker, 2010).

$$\frac{da}{dt} = 2 \frac{a^2}{\sqrt{\mu p}} \left[f_S e \sin \theta + f_N \frac{p}{r} \right] \quad (2.31)$$

$$\frac{de}{dt} = \sqrt{\frac{p}{\mu}} \left[f_S \sin \theta + f_N (\cos E + \cos \theta) \right] \quad (2.32)$$

$$\frac{di}{dt} = f_W \frac{r}{\sqrt{\mu p}} \cos u \quad (2.33)$$

$$\frac{d\omega}{dt} = -\sqrt{\frac{p}{\mu}} \left[f_W \frac{r}{p} \cot i \sin u + \frac{1}{e} \{ f_S \cos \theta - f_N \left(1 + \frac{r}{p} \right) \sin \theta \} \right] \quad (2.34)$$

$$\frac{d\Omega}{dt} = f_W \frac{r}{\sqrt{\mu p} \sin i} \sin u \quad (2.35)$$

$$\frac{dM}{dt} = n - f_S \left[\frac{2r}{\sqrt{\mu a}} - \frac{1-e^2}{e} \sqrt{\frac{a}{\mu}} \cos \theta \right] - f_N \frac{1-e^2}{e} \sqrt{\frac{a}{\mu}} \left(1 + \frac{r}{p} \right) \sin \theta \quad (2.36)$$

In the above equations, f_S , f_N and f_W are the components of the perturbation acceleration in respectively the radial, normal and out-of-plane direction.

Walker et al. also obtained the Gauss' form of the Lagrange Planetary Equations in modified equinoctial elements (Walker et al., 1985) and these are shown in Equations 2.37-2.42. The subscripts used to indicate a modified equinoctial element are removed for improved readability.

$$\frac{dp}{dt} = \frac{2p}{w} \sqrt{\frac{p}{\mu}} f_N \quad (2.37)$$

$$\frac{df}{dt} = \sqrt{\frac{p}{\mu}} \left[f_S \sin L + \frac{[(w+1) \cos L + f_1] f_N}{w} - \frac{g(h \sin L - k \cos L) f_W}{w} \right] \quad (2.38)$$

$$\frac{dg}{dt} = \sqrt{\frac{p}{\mu}} \left[-f_s \cos L + \frac{[(w+1) \cos L + g] f_N}{w} - \frac{f(h \sin L - k \cos L) f_W}{w} \right] \quad (2.39)$$

$$\frac{dh}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 f_W}{2w} \cos L \quad (2.40)$$

$$\frac{dk}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 f_W}{2w} \sin L \quad (2.41)$$

$$\frac{dL}{dt} = \sqrt{\mu p} \left(\frac{w}{p} \right)^2 + \sqrt{\frac{p}{\mu}} \frac{(h \sin L - k \cos L) f_W}{w} \quad (2.42)$$

with:

$$s^2 = 1 + h^2 + k^2 \quad (2.43)$$

$$w = 1 + f \cos L + g \sin L \quad (2.44)$$

2.6.2. Gravity Field Forces

In case of a radially symmetric mass distribution of the central body, it can be assumed that the body is a point mass. In reality any body has a mass density distribution. For Earth the gravitational potential at a point outside the Earth can be written as (Wakker, 2010):

$$U = -\frac{\mu}{r} \left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R}{r} \right)^n P_n(\sin \phi) + \sum_{n=2}^{\infty} \sum_{m=1}^n J_{n,m} \left(\frac{R}{r} \right)^n P_{n,m}(\sin \phi) [\cos m(\Lambda - \Lambda_{n,m})] \right] \quad (2.45)$$

In this equation, r , ϕ and Λ are spherical coordinates of the considered point in a geocentric reference frame. The parameters J_n and $J_{n,m}$ and $\Lambda_{n,m}$ are model parameters. Furthermore, $P_{n,m}(\sin \phi)$ is a Legendre polynomial of degree n .

The most important term is J_2 , which scales the Earth flattening, and has a value of $1.083 \cdot 10^{-3}$; all other terms are about a thousand times smaller (Wakker, 2010) and therefore J_2 must be included in first-order orbit computations around Earth. The magnitude of the perturbing acceleration due to J_2 can be written as (Wakker, 2010):

$$\mathbf{f} = -\nabla \left[\frac{\mu}{r} J_2 \left(\frac{R}{r} \right)^2 P_2(\sin \phi) \right] = -\nabla \left[\frac{1}{2} \mu J_2 \frac{R^2}{r^3} (3 \sin^2 \phi - 1) \right] \quad (2.46)$$

The maximum values of the perturbing acceleration due to J_2 occur at $\phi = -90^\circ, 90^\circ$ and therefore equal to:

$$|f_{max}| = |f_r| = 3\mu J_2 \frac{R^2}{r^4} \quad (2.47)$$

It can be observed that the perturbing acceleration is inversely proportional to the radial distance and thus this perturbation should be taken into account in the vicinity of Earth. At Low Earth Orbits, with altitudes below 800 km, a series of J_n and $J_{n,m}$ terms have to be included.

The variations in elements due to the J_2 -term are derived by (Wakker, 2010):

$$\Delta a = -3J_2 \frac{R^2}{r_0} \sin^2 i_0 \sin^2 u_e \quad (2.48)$$

$$\Delta i = -\frac{3}{4} J_2 \frac{R^2}{r_0^2} \sin 2i_0 \sin^2 u_e \quad (2.49)$$

$$\Delta \Omega = -\frac{3}{2} J_2 \frac{R^2}{r_0^2} \cos i_0 \left(u_e - \frac{1}{2} \sin 2u_e \right) \quad (2.50)$$

with u_e the argument of latitude. Δa shows a short-periodic variation of half an orbital period as period: $\Delta a = 0$ when $u_e = k\pi$ with k an integer. As a first-order approximation, it can thus be concluded that the J_2 effect has no secular or long-period effect on the semi-major axis and thus orbital energy.

The inclination i shows a short-period variation. Only $\Delta \Omega$ shows a secular trend which is superimposed on a short-period variation. It then follows that the value of $\Delta \Omega$ after one revolution ($u_e = 2\pi$) equals:

$$\Delta_{2\pi} \Omega = -3\pi J_2 \frac{R^2}{r_0^2} \cos i_0 \quad (2.51)$$

The mean regression rate can thus be written as:

$$\dot{\Omega}_{mean} = \frac{\Delta_{2\pi}\Omega}{T} \quad (2.52)$$

with T the orbital period of the unperturbed orbit.

A spacecraft in a circular orbit around Earth at an altitude of 500 km, has the following variation in orbital elements (Wakker, 2010)

$$\begin{aligned} |\Delta a|_{max} &= 19.2\text{km} \\ |\Delta i|_{max} &= 2.4' \\ |\Delta\Omega|_{max} &= 1.7' \end{aligned} \quad (2.53)$$

The maximum change in semi-major axis is obtained at $i_0 = 90^\circ$ whereas the maximum changes in inclination and Ω are obtained at $i_0 = 45^\circ$. For a spacecraft in a geostationary orbit at $i_0 = 45^\circ$ the changes are:

$$\begin{aligned} |\Delta a_{max}| &= 19.2\text{km} \\ |\Delta i|_{max} &= 4.0'' \\ |\Delta\Omega|_{max} &= 2.7'' \end{aligned} \quad (2.54)$$

In the SGP4 model J_2 , J_3 and J_4 are included and their values are, following the WGS-72 model (Vallado et al., 2006):

$$\begin{aligned} J_2 &= 1082.616\text{e-}6 \\ J_3 &= -2.53881\text{e-}6 \\ J_4 &= -1.65597\text{e-}6 \end{aligned} \quad (2.55)$$

2.6.3. Aerodynamic Forces

The density of Earth's atmosphere decreases by increasing altitude but can still exert a perturbing force on a satellite in orbit around Earth. The acceleration exerted on the spacecraft is described by (Wakker, 2010):

$$\mathbf{f} = -C_D \frac{1}{2} \rho \frac{S}{M} |\mathbf{V}| \mathbf{V} \quad (2.56)$$

In this equation, C_D is the drag coefficient, ρ is the density of the atmosphere at the location of the spacecraft, S is the cross-sectional area of the satellite perpendicular to the velocity vector and M equals the mass of the spacecraft. In general C_D ranges from 2 to 3 and tends to be closer to 2 at lower altitudes and approaches 3 at higher altitudes. The density of the atmosphere at high altitudes is currently determined by satellites in orbit, but is not constant in time due to solar radiation. Below 200 km the aerodynamic drag is the most important perturbing force, whereas above 1000 km this perturbing acceleration can be neglected.

In the first-order approximation of the perturbation due to atmospheric drag, it is assumed that the atmosphere rotates with the same angular velocity as the Earth, $\dot{\theta}$. Again, the change in orbital elements is derived in (Wakker, 2010) and the results are shown here. The semi-major axis decreases secularly during an orbital revolution:

$$\Delta_{2\pi} a = -2\pi \frac{C_D S}{M} \rho_0 r_0^2 \left(1 - 2 \frac{\dot{\theta}}{n_0} \cos i_0 \right) \quad (2.57)$$

In this equation, it is assumed that $C_D S/M$ is constant during one revolution, ρ_0 remains constant and n_0 is the mean motion in radians per second. The secular change in inclination per revolution equals:

$$\Delta_{2\pi} i = -\frac{1}{2} \pi \frac{C_D S}{M} \rho_0 r_0 \frac{\dot{\theta}}{n_0} \sin i_0 \quad (2.58)$$

The change per revolution for Ω equals zero:

$$\Delta_{2\pi} \Omega = 0 \quad (2.59)$$

For a satellite at 300 km altitude with $i_0 = 45^\circ$ the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta_{2\pi} a|_{max} &= -2.55 \text{ km} \\ |\Delta_{2\pi} i|_{max} &= -0.97'' \end{aligned} \quad (2.60)$$

For a satellite at 500 km altitude with $i_0 = 45^\circ$ the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta_{2\pi} a|_{max} &= -135 \text{ m} \\ |\Delta_{2\pi} i|_{max} &= -0.050'' \end{aligned} \quad (2.61)$$

In SGP4, the drag is modelled using a power-law density function (Hoots et al., 2004a).

2.6.4. Third-Body Perturbations

A spacecraft also experiences attraction by celestial bodies other than the body it orbits. This perturbing force depends on the distances and masses of the perturbing bodies. The gravitational perturbations of bodies j on the motion of spacecraft i are expressed by the following perturbing potential (Wakker, 2010):

$$R = -G \sum_{j \neq k, i} m_j \left(\frac{1}{r_{ij}} - \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{r_j^3} \right) \quad (2.62)$$

For the case of a satellite in orbit around Earth, the maximum value of the ratio between the magnitude of the perturbing acceleration and the magnitude of the acceleration due to Earth can be approximated with (Wakker, 2010):

$$\left(\frac{f_d}{f_E} \right)_{max} \approx 2 \frac{m_d}{m_E} \left(\frac{r_i}{r_d} \right)^3 \quad (2.63)$$

In the above equation, f_d and f_E represent the gravitational perturbing acceleration by respectively the perturbing body d and Earth. From Equation 2.63 it follows that the perturbing acceleration of body d increases when the orbital altitude of the satellite increases. Around Earth, an orbit of a spacecraft is mainly influenced due to lunar and solar attraction.

The change in semi-major axis due to a perturbing body is (Wakker, 2010)

$$\Delta a = \frac{3}{2} \frac{\mu_d}{\mu} \frac{r_0^4}{r_d^3} [\sin 2\alpha_0 \cos i \sin 2u_e + 2(\sin^2 \alpha_0 \cos^2 i_0 - \cos^2 \alpha_0) \sin^2 u_e] \quad (2.64)$$

with α_0 the right ascension of the perturbing body at $u = 0^\circ$. From this equation, it can be concluded that Δa varies periodically over an orbital revolution and Δa equals zero when $u_e = k\pi$ with k an integer. A fixed celestial body has thus no long-periodic effect on the total orbital energy of the satellite. The Δa is equal to zero when $i = 90^\circ$ and $\alpha_0 = 90^\circ, 270^\circ$. The maximum value for Δa occurs when $i = 0^\circ$:

$$\Delta a = 3 \frac{\mu_d}{\mu} \frac{r_0^4}{r_d^3} \sin u_e \sin(2\alpha_0 - u_e) \quad (2.65)$$

The variation in inclination are short-periodic and long-periodic:

$$\Delta_{2\pi} i = -\frac{3}{2} \pi \frac{\mu_d}{\mu} \left(\frac{r_0}{r_d} \right)^3 \sin i_0 \sin 2\alpha_0 \quad (2.66)$$

The change in Ω is secular:

$$\Delta_{2\pi} \Omega = -3\pi \frac{\mu_d}{\mu} \left(\frac{r_0}{r_d} \right)^3 \cos i_0 \sin^2 \alpha_0 \quad (2.67)$$

When averaging $\sin^2 \alpha_0$ over the range $\alpha_0 = 0$ to $\alpha_0 = 2\pi$ the long-term mean rate of precession then equals:

$$\dot{\Omega}_{mean} = \frac{\Delta \Omega_{avg}}{T} = -\frac{3}{4} \frac{\mu_d}{\mu} \sqrt{\frac{r_0^3}{\mu}} \cos i_0 \quad (2.68)$$

For a satellite at 500 km altitude with $i_0 = 10^\circ$ the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta_{2\pi} i|_{max} &= 0.012'' \\ |\Delta_{2\pi} \Omega|_{max} &= -0.13'' \\ \dot{\Omega} &= -1.02''/\text{day} \end{aligned} \quad (2.69)$$

For a satellite at GEO altitude with $i_0 = 10^\circ$ the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta_{2\pi} i|_{max} &= -2.7'' \\ |\Delta_{2\pi} \Omega|_{max} &= 31'' \\ \dot{\Omega} &= -15.6''/\text{day} \end{aligned} \quad (2.70)$$

The SGP4 takes the perturbations due to the gravitational attraction of the Sun and Moon into account.

2.6.5. Radiation Pressure

A spacecraft in orbit experiences the radiation force of the Sun which can be expressed as:

$$F = C_R \frac{WS}{c} \quad (2.71)$$

The acceleration on the satellite then becomes:

$$\mathbf{f} = -C_R \frac{WS}{Mc} \mathbf{e}_S \quad (2.72)$$

with \mathbf{e}_S the unit vector from the satellite in the direction of the Sun, W the power density of the incoming radiation and c the speed of light.

The power density can be calculated with (Ley et al., 2008):

$$W = P_0 \left(\frac{1AU}{r_{AU}} \right)^2 \quad (2.73)$$

where P_0 is the solar constant, $1AU$ the distance from Earth to the Sun and r equals the distance from the spacecraft to the Sun in AU .

For satellites in orbit around Earth, solar radiation pressure is the largest contribution to the radiation force. The Earth's albedo and its infra-red radiation produce a smaller amount of force. The power density depends on the distance to the Sun and can be used as a constant for missions around a planet, but eclipses should be taken into account. Furthermore, for satellites with a large area due to e.g. solar panels, radiation pressure is one of the largest perturbing forces (Ley et al., 2008).

The radiation pressure acceleration f thus depends on the position of the spacecraft with respect to the Sun. In the shadow cone of the Earth, the acceleration is logically equal to zero. The shadow cone is specified by angles u_1 and u_2 as can be seen in Figure 2.11. For the derivations of the change in orbital elements, it is assumed that f is constant during an orbital revolution.

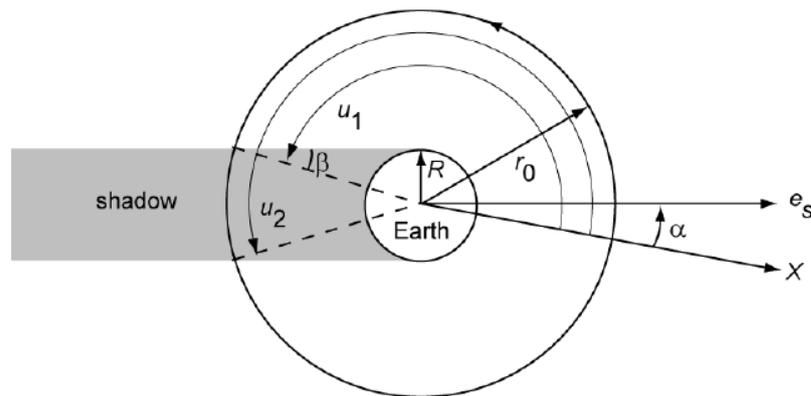


Figure 2.11: Geometry of an eclipse when the Sun is in the satellite's orbital plane (Wakker, 2010).

The change per revolution for the semi-major axis then equals:

$$\Delta_{2\pi} a = 2 \frac{f}{n_0^2} [-\cos \alpha_0 (\cos u_2 - \cos u_1) + \cos i_0 \sin \alpha_0 (\sin u_2 - \sin u_1)] \quad (2.74)$$

Further derivations result in the change for inclination and right ascension of the ascending node:

$$\Delta_{2\pi} i = -\frac{f}{n_0^2 r_0} \sin i_0 \sin \alpha_0 (\sin u_2 - \sin u_1) \quad (2.75)$$

$$\Delta_{2\pi} \Omega = \frac{f}{n_0^2 r_0} \sin \alpha_0 (\cos u_2 - \cos u_1) \quad (2.76)$$

It can be observed that if the satellite never experiences an eclipse during a revolution (i.e. $u_2 = u_1$), the change in semi-major axis, inclination and right ascension of the ascending node is equal to zero. In this case the variations are short-periodic.

For a satellite at 500 km altitude the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta a|_{max} &= 0.91 \text{ m} \\ |\Delta i|_{max} &= 0.037'' \\ |\Delta \Omega|_{max} &= -0.075'' \end{aligned} \quad (2.77)$$

For a satellite at GEO altitude the effect of the perturbation is:

$$\begin{aligned} |\Delta a|_{max} &= 210 \text{ m} \\ |\Delta i|_{max} &= 1.41'' \\ |\Delta \Omega|_{max} &= 2.82'' \end{aligned} \quad (2.78)$$

SGP4 includes solar radiation pressure in its deep-space segment.

2.6.6. Electromagnetic forces

A satellite can obtain a negative electrical potential when in orbit around a planet in the higher parts of the atmosphere since that region is partly ionized. The Lorentz force is the result of an electrically charged particle, or in this case a spacecraft, interacting with the magnetic field of the Earth.

The first-order expression for the perturbing acceleration of an electrically charged satellite due to its motion through the magnetic field of the Earth is derived in (Wakker, 2010):

$$\mathbf{f} = -\frac{q}{M} g_{1,0} R^3 \mathbf{v} \times \bar{\nabla} \left(\frac{\sin \phi}{r^2} \right) \quad (2.79)$$

In this equation, q is the electric charge of the satellite, M the mass of the satellite, $g_{1,0}$ is the first Gauss geomagnetic coefficient, R is the radius of the Earth, and $\bar{\nabla}$ is the velocity of the satellite relative to the rotating geomagnetic field, ϕ is the satellite's latitude, and r is the distance from the satellite to the center of the Earth. The magnitude of this acceleration is found with the use of Equation 2.80, which is derived from Equation 2.79 (Wakker, 2010):

$$|f| = 4\pi\epsilon_0 g_{1,0} \frac{U^* R_S}{M} \frac{R^3}{\sqrt{\mu r^3}} \quad (2.80)$$

where ϵ_0 is the electric constant, R_S the satellite radius and U^* is the electrical potential difference between the satellite and the surrounding plasma.

The semi-major axis shows a short-periodic variation during an orbital revolution:

$$\Delta a = 2 \frac{q g_{1,0}}{M n_0} \frac{\dot{\theta}}{n_0} \frac{R^3}{r_0^2} \sin^2 i_0 \sin^2 u_e \quad (2.81)$$

The change in inclination shows a short-periodic trend as well.

$$\Delta i = -\frac{q g_{1,0}}{M n_0} \left(\frac{R}{r_0} \right)^3 \sin i_0 \left(1 - \frac{\dot{\theta}}{n_0} \cos i_0 \right) \sin^2 u_e \quad (2.82)$$

The right ascension of the ascending node shows a secular perturbation:

$$\Delta_{2\pi} \Omega = -2\pi \frac{q g_{1,0}}{M n_0} \left(\frac{R}{r_0} \right)^3 \left(1 - \frac{\dot{\theta}}{n_0} \cos i_0 \right) \quad (2.83)$$

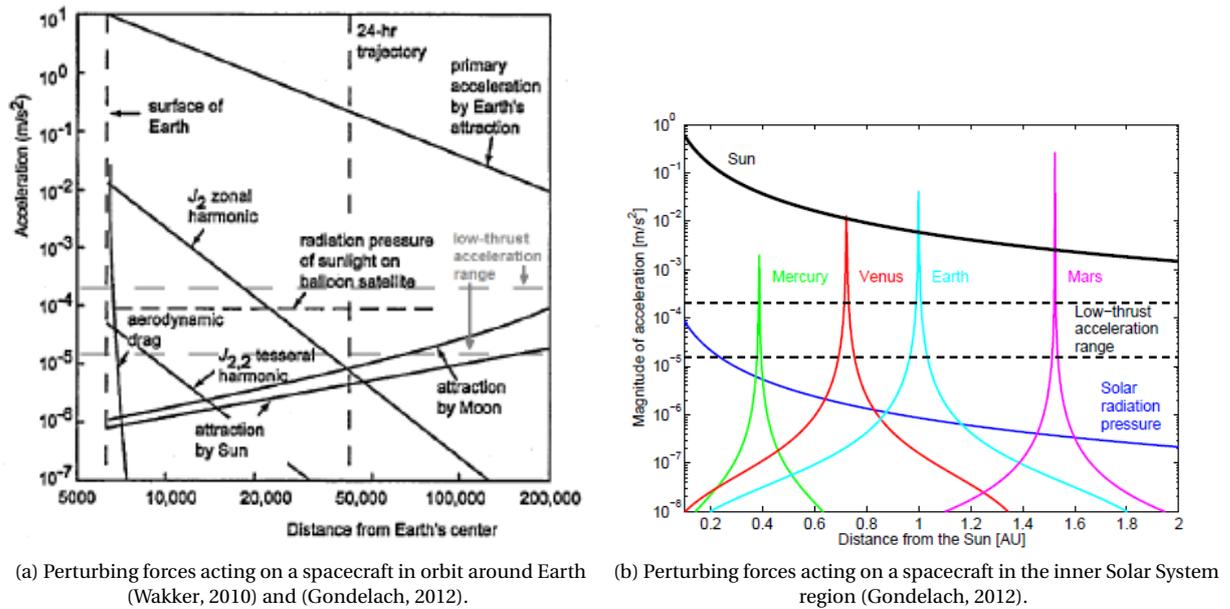


Figure 2.12: Magnitude of perturbing forces.

For a satellite at 500 km altitude the effect of the perturbation is (Wakker, 2010)

$$\begin{aligned} |\Delta a|_{max} &= 11 \mu\text{m} \\ |\Delta i|_{max} &= 3.3e-6'' \end{aligned} \quad (2.84)$$

Thus, according to (Wakker, 2010), the effect on the orbital elements due to the geomagnetic field are extremely small.

In SGP4, this disturbance is not taken into account.

2.6.7. Comparison

In this section the different perturbations that can act on a spacecraft have been discussed. Figures 2.12a and 2.12b show the magnitudes of the perturbing forces near Earth and in the inner Solar System. For spacecraft in low Earth orbit the J_2 zonal harmonic is a dominant perturbing force, at higher altitudes the attraction by Sun and Moon can no longer be neglected. In Figure 2.12a, the low-thrust acceleration range is also shown. It can be concluded that, especially at low altitudes, the perturbations acting on the spacecraft can overcome an applied low-thrust acceleration.

Looking at the inner Solar System region, it can be seen that solar radiation pressure is a large perturbing force at most inner regions and should be taken into account. It is however dependent on the surface area on the spacecraft. Furthermore, the perturbing forces of the planets are only dominant near these planets.

2.7. Transfer Orbits

In the previous sections orbits were discussed: it was assumed the spacecraft was already in a particular orbit, that was either purely Keplerian or subjected to perturbations. This section discusses the transfer trajectory needed to get from an initial orbit to the desired orbit, when using conventional high-thrust propulsion. In Chapter 8 the methods described in this section are used to compare the results of the designed SGP4-LT tool. For simplicity it is assumed that the initial and final orbits are coplanar and circular.

The design of the trajectory depends on the needs or limitations of the mission. This leads to the following considerations.

- The transfer trajectory can be optimized for minimum Delta-V.
- Faster transfer trajectories are a possibility if time is a leading factor, but the physical constraints of the spacecraft must be taken into account.

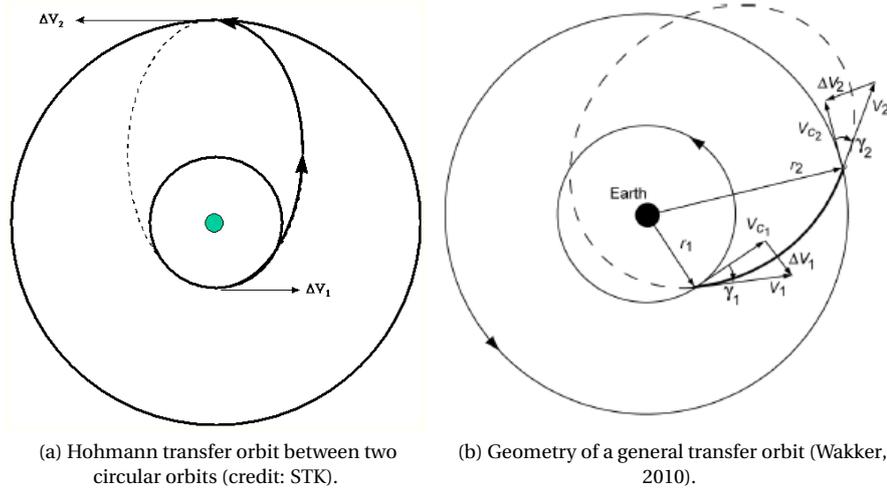


Figure 2.13: Transfer Orbits.

- In case of a rendezvous or capturing by a planet, the synodic period should be taken into account.

Using conventional chemical propulsion, a Hohmann transfer was found to be the most efficient one in terms of Delta-V (Wakker, 2010). Figure 2.13a shows a Hohmann transfer between two circular orbits.

The required total Delta-V for this trajectory is derived in (Wakker, 2010) and equals:

$$\frac{\Delta V_{tot}}{V_{c1}} = (n - 1) \sqrt{\frac{2}{n(n + 1)}} + \sqrt{\frac{1}{n} - 1} \tag{2.85}$$

where

$$n = \frac{r_2}{r_1} \tag{2.86}$$

Figure 2.13b shows the geometry of a general transfer orbit. Although not as efficient in Delta-V as a Hohmann transfer, this is a faster trajectory which could be preferable depending on the mission.

Performing a transfer trajectory with the use of low-thrust propulsion can be even more energy-efficient than performing a Hohmann transfer, although this will be at the expense of travel time. Low-thrust propulsion is discussed extensively in Chapter 3.

2.8. Transfer between two non-coplanar orbits

Using conventional propulsion, the geometry of a general transfer between two non-coplanar orbits is shown in Figure 2.14. An impulsive shot is fired in point P, which is determined by the argument of latitude u_1 . This results in a shift from the initial orbit with i_1, Ω_1 and \bar{V}_1 to the final orbit with i_2, Ω_2 and \bar{V}_2 .

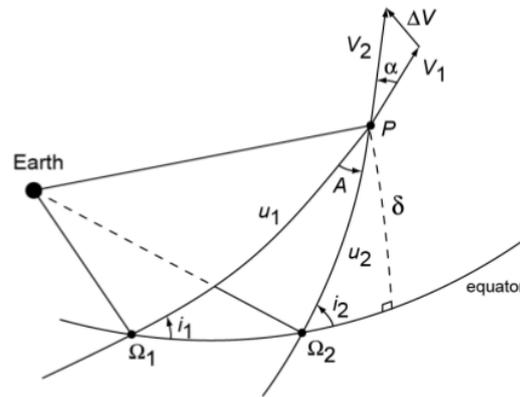


Figure 2.14: Geometry of a general orbital plane change manoeuvre (Wakker, 2010)

In certain cases it is not preferred to change Ω and only a change in inclination i is desired. This implies that the orbital plane only rotates about the line of nodes and therefore:

$$\alpha \neq 0, \quad ; \quad u_1 = u_2 \quad ; \quad V_1 = V_2 \quad ; \quad \Omega_1 = \Omega_2 \quad (2.87)$$

Using the cotangent rule and Figure 2.14, it can be concluded that (Wakker, 2010):

$$\tan \Delta\Omega = \frac{\sin u_1}{\cot \alpha + \cos u_1 \cos i_1} \quad (2.88)$$

For a change in i only, it follows from Equation 2.88 that $\Delta\Omega = 0$ if:

$$u_1 = k\pi \quad , \quad k = 0, 1, 2, \dots \quad (2.89)$$

This indicates that the manoeuvre has to be executed at the ascending or descending node. Figure 2.15 shows the geometry of an inclination change at the nodes.

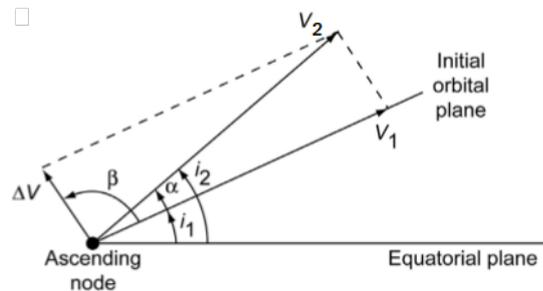


Figure 2.15: Geometry of an inclination change maneuver (Wakker, 2010)

In Figure 2.15, β is the angle between the direction of the impulsive shot and the original orbital plane. In this case, where inclination is the only element to change, $\Delta i = \pm\alpha$. Using the cosine rule, the velocity change required is calculated using Equation 2.90.

$$\Delta V^2 = V_1^2 + V_2^2 - 2V_1V_2 \cos \alpha \quad (2.90)$$

Combining Equation 2.90 with $\Delta i = \alpha$ eventually leads to an equation for the ΔV required for the known V_1 and inclination change:

$$\frac{\Delta V}{V_1} = 2 \sin \frac{1}{2} \Delta i \quad (2.91)$$

It can thus be concluded that less propellant will be consumed for an inclination change at a smaller initial velocity. An inclination change can therefore be most efficiently executed at apogee of the orbit when said

apogee (nearly) coincides with the nodes.

The angle β , which is the angle between the impulsive shot and the initial orbital plane, or the direction of the thrust vector, is derived from Figure 2.15 and can be written as:

$$\beta = \frac{1}{2} (\pi + \Delta i) \quad (2.92)$$

This chapter has provided an overview of basic concepts of orbital mechanics needed to grasp a better understanding of the material discussed in the upcoming chapters. Furthermore, it was discussed which perturbation act on a spacecraft in an orbit around Earth and what the influences of these perturbations are on the orbital elements.

3

Low-Thrust Propulsion

"The Force is strong in this one."

—Darth Vader (The Empire strikes back), 1980

Over the past two decades, there has been an increase in designing and using spacecraft with low-thrust engines. Low-thrust engines have a higher energy density per unit of mass, resulting in a higher exhaust velocity. Despite the higher exhaust velocities that can be reached, the actual thrusting force is low compared to that of conventional chemical propulsion, hence the name *low thrust*. Consequently, a longer time of thrusting is needed before high velocity changes are reached and an other alternative name is therefore *continuous thrusting*.

Today, low-thrust propulsion is mostly used in interplanetary scientific missions, where travel time is less critical than it would be for commercial satellites although low-thrust propulsion results in significant mass savings.

To be able to make the right assumptions for designing low-thrust trajectories a basic understanding of low thrust is required. An increasing number of satellites around Earth use low-thrust propulsion for station-keeping and/or orbit raising.

This chapter gives a brief introduction of low-thrust propulsion, starting with an overview in Section 3.1 of deep space missions that were already successful or are in operation . Next, the different types of low-thrust engines are discussed in Section 3.2. The chapter is concluded with the dynamics of low-thrust propulsion in Section 3.3.

3.1. Heritage

Although low thrust is a still not widely used way of propulsion, spacecraft have already successfully been propelled with a low-thrust engine in the past and are currently in use or in development. This section gives a brief overview of these missions which are presented in chronological order based on the launch date of the spacecraft.

3.1.1. Deep Space 1

On October 24, 1998, Deep Space 1 was launched from Cape Canaveral in the United States of America. This spacecraft, developed by National Aeronautics and Space Administration (NASA), was the first spacecraft to be propelled by Solar Electric Propulsion (SEP) and its mission was to demonstrate and evaluate advanced technologies (Rayman et al., 1999). One of these technologies was an Ion Propulsion System (IPS) which is an implementation of SEP. An artist impression of this spacecraft is shown in Figure 3.1.

Deep Space 1 had an initial mass of 486.3 kg and was equipped with an IPS fuelled by 82 kg xenon, a gas more than 4.5 times heavier than air. Next to xenon, Deep Space 1 was also fuelled by hydrogen which was used by the reaction control system. The spacecraft was delivered in an escape trajectory en route to asteroid 1992KD (Rayman et al., 1999). After a successful encounter with 1992KD, the mission was extended to two

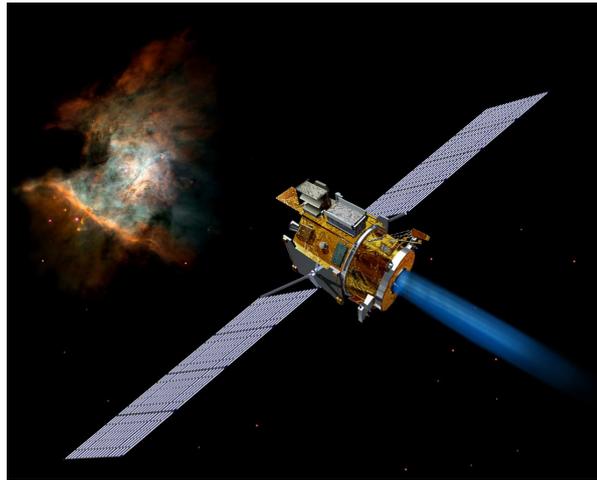


Figure 3.1: Artist impression of Deep Space 1 (credit: NASA).

comets, Wilson-Harrington in January 2001 and Borrelly in September 2001. Due to a star sensor failure in November 1999, the spacecraft had to go into a Sun-safe-hold configuration until a solution was found. It became clear that Deep Space 1 could not encounter both comets and eventually it was decided that the trajectory would be redesigned in such a way that only Borrelly was encountered in September 2001. After this successful encounter, IPS performance tests were conducted and on December 18, 2001 Deep Space had its last contact with Earth.

Trajectory

The primary trajectory was to have a fly-by with asteroid 1992KD with a possible extension to encounters with Wilson-Harrington and Borrelly comets. SEPTOP was used to design an initial estimate of the Deep Space 1 trajectory. It optimized the xenon consumption, the time of arrival, the IPS thrust and time of flight, as a function of time. Furthermore it took the constraints due to the adjustments of the use of hydrogen, forced coasting, forced thrusting in specific directions, and constraints on orientation of the spacecraft into account (Kennedy et al., 2003), (Rayman and Williams, 2002). NAVTRAJ was used as the second program and contains higher fidelity models of small forces to make adjustments to the thrust profile, and these adjustments were directly uploaded to the spacecraft. This is a big advantage of the use of SEP; small errors can be corrected in-flight, although one should bear in mind that this could require extra propellant (Rayman et al., 1999).

Computer Algorithm for Trajectory Optimization (CATO) was used to fine-tune the trajectory for the extended missions, after the encounter with asteroid 1992KD. The expected trajectory for Deep Space 1 with both primary and secondary mission targets is shown in Figure 3.2. The dotted part of the line indicates coasting, at the solid line IPS-thrusting is on.

3.1.2. SMART-1

SMART-1 was developed by the European Space Agency (ESA) and launched on September 23, 2003. It piggy-backed on an Ariane V launch and was injected in a GTO. In this orbit the ion thrusters were turned on, making it the first spacecraft to escape Earth orbit with electric propulsion (Foing et al., 2007).

The spacecraft was a 1 m^3 box with two solar panels which could deliver 1850 W in total. It had a total maximum launch mass of 370 kg, of which 82.5 kg of xenon. After the last moment of SEP thrusting, the spacecraft used 4958.3 hours of thrust and went through the thrust on/off cycle 844 times (Estublier et al., 2007). An acceleration of $1.81 \cdot 10^{-4} \text{ m/s}^2$ was possible for this spacecraft. An artist impression of SMART-1 is given in Figure 3.3. In February 2015 SMART-1 reached its final operational orbit and performed various measurements of the Moon, before its impact on the Moon on September 3, 2005.

Trajectory

Before launch, the trajectory was not completely defined due to uncertainties in launch date and lift-off time. A worst-case scenario was therefore the basis and in-flight adjustments to the orbit were made. After launch, the orbit was continuously raised by thrusting along the velocity vector, except during eclipse.

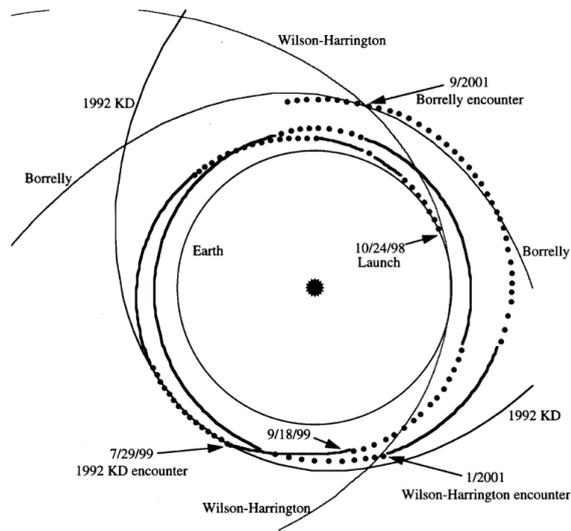


Figure 3.2: Trajectory of Deep Space 1 (Rayman, 1999).



Figure 3.3: Artist impression of Smart-1 (credit: ESA).

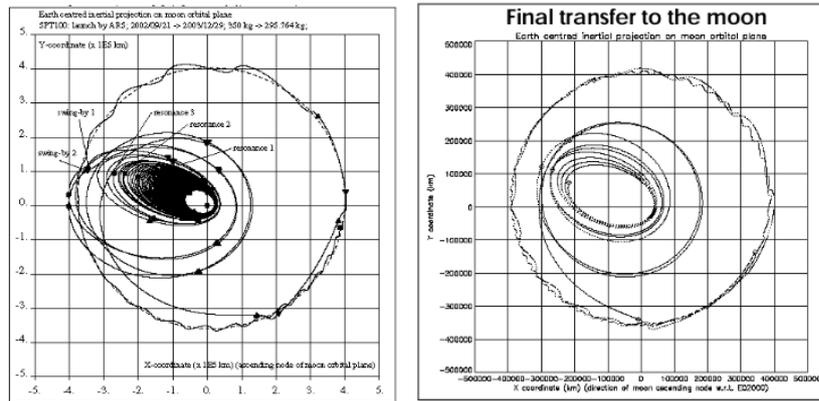


Figure 3.4: Designed trajectory (left) and in-flight executed trajectory of SMART-1 (right) (Milligan et al., 2005).

Near the end of the escape phase, the thrust direction was changed to being perpendicular to the position vector. In this way, the orientation of the orbit was changed to avoid the long eclipse period at this altitude. Later the orbit rotated back to the ecliptic in order to reach the Moon. The pre-launch orbit was designed with three fly-by's at the Moon but these were removed which resulted in a reduction of transfer time. Figure 3.4 shows the initial, worst-case scenario trajectory at the left, and the final trajectory on the right.

3.1.3. BepiColombo

BepiColombo, named after the Italian professor Giuseppe Colombo (1920-1984), is a joint dual-mission between ESA and Japan Aerospace Exploration Agency (JAXA) and was launched October 20, 2018 to investigate Mercury, the planet closest to the Sun. Still little is known of Mercury due to its proximity to the Sun. It is hard to observe the planet from Earth and it requires a large amount of energy to decelerate a spacecraft that moves inwards to a target closer to the Sun. In comparison, a mission to Pluto would require less energy (Benkhoff et al., 2010).

The mission employs two spacecraft: ESA's Mercury Planetary Orbiter (MPO) and JAXA's Mercury Magnetospheric Orbiter (MMO). These spacecraft have been launched together and travel to Mercury with the use of a separate SEP unit, the Mercury Transfer Module (MTM). It consists of two solar arrays, a SEP system and a bi-propellant propulsion system for attitude control during cruise. At arrival at Mercury, the two spacecraft will part ways and will both enter their own polar orbit around Mercury. This insertion will be performed with the use of chemical propulsion.

The launch mass of all vehicles will be 4100 kg of which 580 kg of xenon propellant, capable of providing a total Delta-V of 5.4 km/s. One ion engine has a maximum thrust level of 145 mN, of which four are present. During cruise two thrusters are on for the majority of the time (Yarnoz et al., 2006). An artist's impression of the spacecraft can be found in Figure 3.5.

Trajectory

The original launch date of BepiColombo was July 18, 2014 but has been postponed for several times before the launch date as set at October 20, 2018. This resulted in various trajectory designs. One of the latest trajectory designs, with a launch date of April 17 2018, is shown in Figure 3.6. The trajectory of BepiColombo has many revolutions and therefore this trajectory is shown as a function of time, rather than a plot in a fixed heliocentric frame. A better view of the trajectory can be understood with an animation for the BepiColombo trajectory published on *Youtube*¹.

The trajectory includes multiple flybys at Venus and Mercury before slowing down to be able to be captured by Mercury. The trajectory was designed with the software tools Direct Interplanetary Trajectory Analysis (DITAN) and MANTRA. Next, it was re-optimized with the use of Low-Thrust Interplanetary Navigation Tool (LOTNAV) the core tool used for the trajectory design. Differences between the two trajectories were verified to be below than the acceptable level. A full trajectory simulation was performed with LOTNAV, followed by iterations to compare different options of guidance targeting or schedule (Rocchi and Jehn, 2016).

¹<https://youtu.be/0Yp-q1wqig>, viewed on October 19, 2018



Figure 3.5: BepiColombo in cruise (top), exploded view during cruise (bottom left) and MPO at Mercury (bottom right) (credit: ESA)

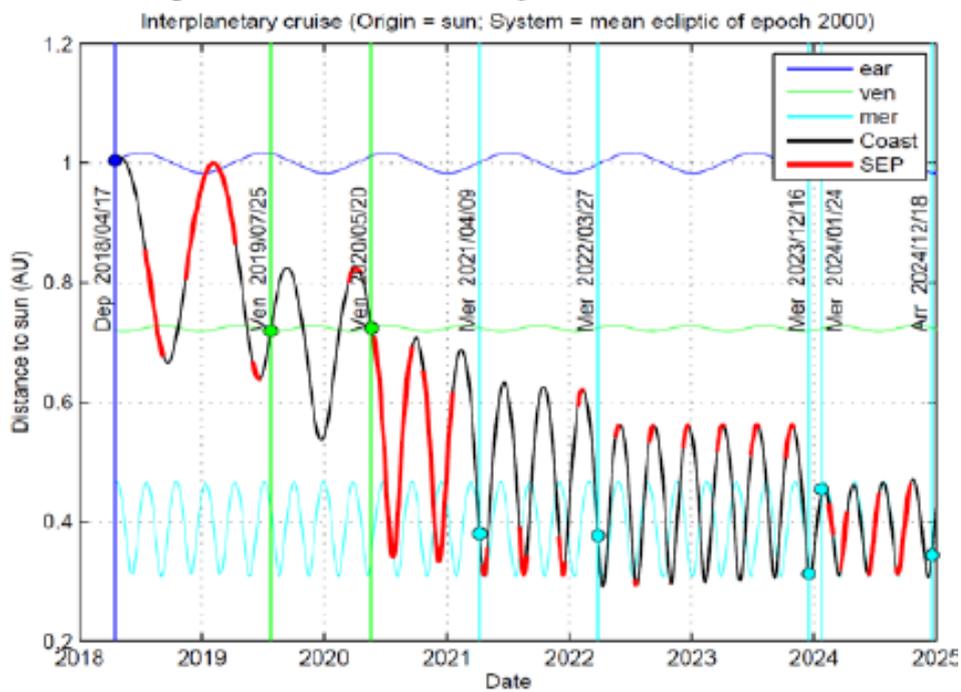


Figure 3.6: BepiColombo trajectory versus time with launch in April, 2018 (Steiger et al., 2016).

3.1.4. Commercial Satellites

Electric Orbit Raising (EOR) is the term used for specifying the use of low-thrust propulsion in transfer orbits around Earth. Low-thrust propulsion has already proven itself at interplanetary missions, as discussed in the previous sections, but for Earth focussed missions hesitation is still present. This is mainly due to the long transfer times which is undesirable for commercial satellites, and the complexity due to the presence of space debris and operational satellites in orbits around Earth. However, the possibility of reducing costs is an interesting aspect for companies and a family of commercial all-electric satellites, the 702 class, is already developed by Boeing (Poole and Ho, 2007)(Goebel et al., 2009). Eutelsat 172B, an all-electric telecommunications satellite built by Airbus Defence and Space, was launched on June 2, 2017 (Ansedé, 2017). Lockheed Martin develops Hall Current thrusters to implement on Geostationary Earth Orbit (GEO) satellites (Khayms et al., 2001). In this subsection, the 702 class by Boeing is briefly discussed.

The 702-class satellites use a 25-cm Xenon Ion Propulsion System (XIPS) which has a specific impulse of 3450 seconds and an average thrust of 10 mN (Poole and Ho, 2007).

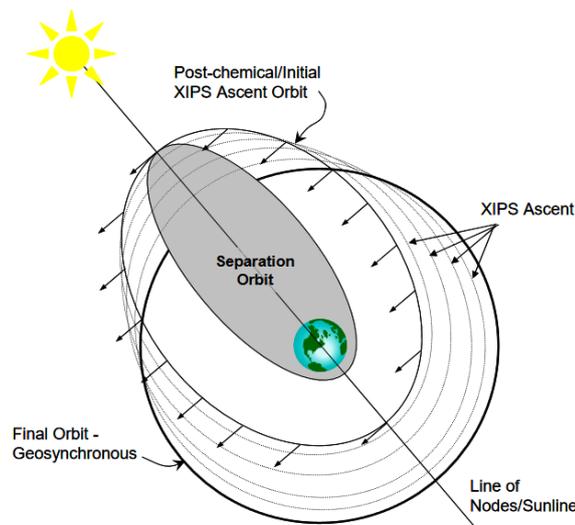


Figure 2-1: XIPS Ascent

Figure 3.7: Orbit Raising Manoeuvres (Poole and Ho, 2007).

Trajectory

The XIPS manoeuvres are calculated by the Xenon Ion Propulsion System Trajectory Optimization Program (XIPSTOP), an optimization algorithm developed by Boeing. It first obtains an initial guess by calculating the thrust vectors and burn times for the optimal two-burn transfer and this is followed by the optimum transfer orbit from initial orbit to the final orbit. It takes eclipse and sensor constraints into account, together with the constraint on Sun angle: Sun shining directly on the XIPS engines must be avoided. Furthermore, close approaches with other satellites are avoided.

The satellite is injected by the launcher in an initial 24-hour synchronous orbit with a semi-major axis of approximately 42164 km. Figure 3.7 shows the orbit raising manoeuvres of the Boeing 702. Note that the initial apogee of the Geostationary Transfer Orbit (GTO) is located beyond the targeted GEO altitude.

3.1.5. GOCE

GOCE is a spacecraft launched by ESA on March 17, 2009 from Plesetsk, Russia. It was the first of ESA's Living Planet Programme satellites and its mission was focussed on the gravity field of the Earth. It was active at a relatively low altitude of 255 km where the drag due to the atmosphere was large enough to require an aerodynamic design. An artist impression of GOCE is shown in Figure 3.8. The gravity field measurements made by GOCE requires the spacecraft to fly in a near-circular, sun-synchronous, dawn-dusk orbit. At this low altitude constant drag compensation is required to maintain this orbit. This was achieved by using a highly controllable ion propulsion system with a QinetiQ T5 Kaufman-type ion thruster, with a total propellant of 40 kg of Xenon. A second T5 engine was installed for redundancy. Due to this propulsion system, the spacecraft flew drag-free, and the only acceleration experienced by GOCE was due to the gravity of the Earth.



Figure 3.8: Artist impression of the GOCE spacecraft (credit: ESA).

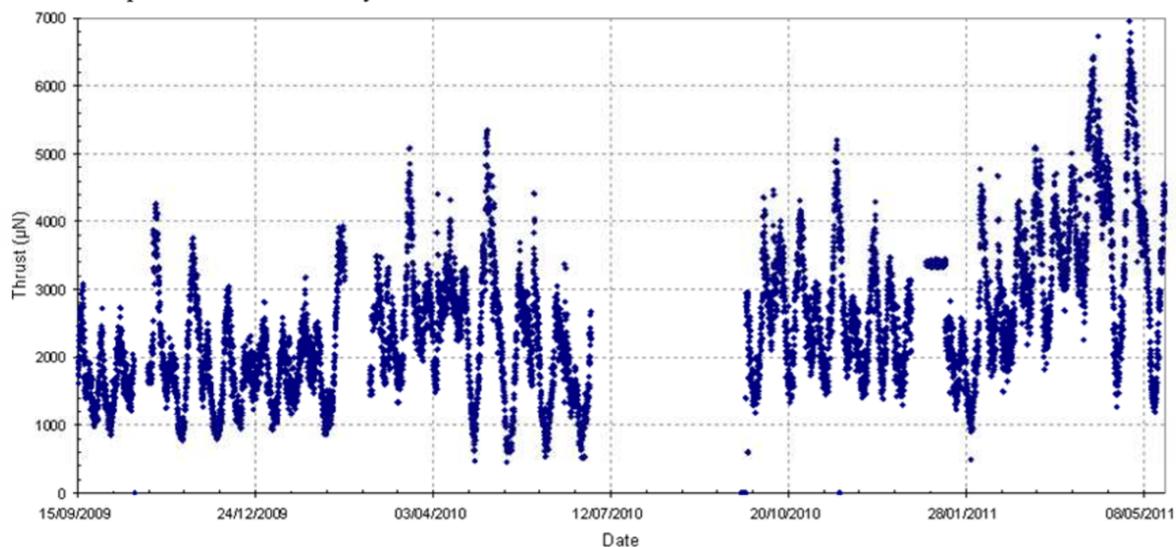


Figure 3.9: Thrust history of GOCE since the start of the science operations (Wallace et al., 2011).

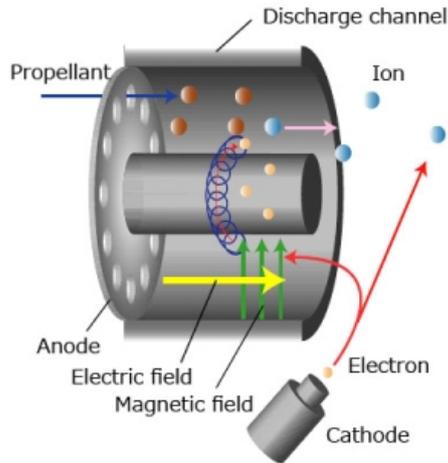
The propulsion system compensated real-time for the drag force and this required a continuously variable thrust between 0.6 and 20.6 mN. The Drag-Free Attitude Control (DFAC) sub-system sensed drag perturbation accelerations on the body of the spacecraft and commanded the propulsion assembly unit to produce the correct amount of thrust to counteract this drag acceleration (Wallace et al., 2011). The thrust history for almost 2.5 years since the start of the science operations can be seen in Figure 3.9. The thrust varies greatly within its small thrust range, with an average around 2 mN. The large gap between July 2010 and September 2010 occurred due to a communications malfunction which resulted in no downlink of scientific data.

November 2010 was the expected completion date but it was decided to extend the mission until the end of 2012. In November 2012, the orbit was lowered to 235 km to get higher resolution data. It was followed in May 2013 by a further lowering to 229 km. Finally, on October 21 2013 GOCE ran out of its xenon propellant and disintegrated on November 11.

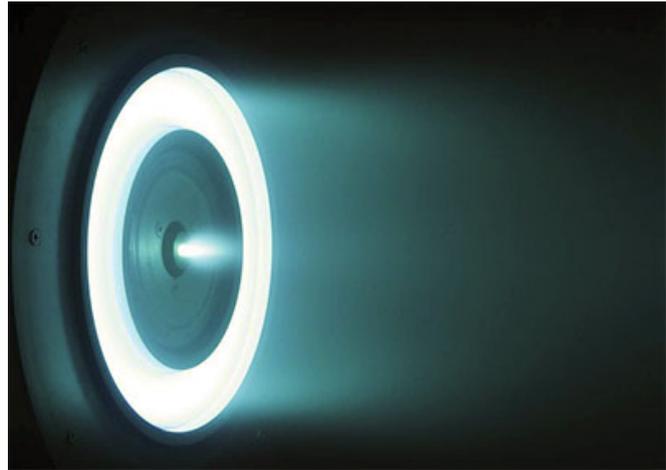
3.2. Electric Propulsion Systems

As stated before, low-thrust propulsion actually refers to electric propulsion. Conventional chemical propulsion is limited by its specific impulse as the reaction-energy per mass unit is stored in the chemical propellant itself. Electric propulsion systems do not have this limitation, since the power used to accelerate the propellant is supplied by an external energy source and therefore high specific impulses can be achieved (Ley et al., 2008).

Three types of electric propulsion can be identified and are discussed briefly. An overview of the different



(a) Working principle of a Hall-effect thruster (credits: University of Tokyo).



(b) Operational Hall-effect thruster (Reid, 2009).

Figure 3.10: Hall-effect thruster.

low-thrust propulsion systems can be found in Table 3.1.

Electrothermal Propulsion

This propulsion system relieves some of the limitations of conventional chemical propulsion systems. The propellant is electrically heated in a chamber and then expanded through a conventional nozzle. Due to the heating, more energy is available per unit of mass of propellant but the exhaust velocity is limited by the maximum allowable temperature in the engine. Examples of electrothermal propulsion engines are resistojets, arcjets, and inductively and radiatively heated engines (Ley et al., 2008). Electrothermal propulsion is mostly used for orbit insertion and attitude control (Jahn and Choueiri, 2001).

Electrostatic Propulsion

An ionized propellant, such as cesium, mercury, argon, krypton and mostly used xenon, is accelerated by an electric field. The exhaust velocity of the ionized particles depends on their mass, charge and the potential difference across the thrust chamber. High exhaust velocities can be achieved without high temperatures. When the particles are ejected, neutralization is required to prevent charging the spacecraft and this is achieved by adding electrons to the ion beam outside the nozzle.

Examples of electrostatic engines are the Hall-effect thruster, Kaufman thruster and radio-frequency thruster. Figure 3.10 shows the schematic working of a Hall-effect thruster (Subfigure 3.10a) and an image of a Hall effect thruster in operation (Subfigure 3.10b).

Electromagnetic Propulsion

Electromagnetic propulsion uses electric and magnetic fields which are orthogonal to the thrust vector. Due to these fields, the ionized particles are accelerated by a Lorentz force. The exhaust velocities achieved with electromagnetic propulsion are even higher than the exhaust velocities produced by electrostatic or electrothermal propulsion, but it requires a much higher power consumption, making it a non-feasible propulsion option for now (NASA, 2010).

Table 3.1: Characteristics of various electric propulsion systems.

Propulsion type	Specific impulse [s]	Max Thrust [N]
Arcjet (Litchford, 2012)	1000-2000	5
Hall Effect Thruster (Szabo and Azziz, 2005)	3200	3
Magnetoplasmadynamic (NASA, 2010)	10000	100

3.2.1. Characteristics

The engines of electric propulsion systems need a separate power source. Neglecting the lifetime of the power source, the power supply could be theoretically unlimited, leading to high exhaust velocities and specific impulses. The power is however limited by the power source, resulting in a limited thrust level. Currently, the power for an electric engine is delivered by a solar array i.e. SEP or a nuclear reactor i.e. Nuclear Electric Propulsion (NEP).

This section briefly discusses these two power suppliers.

Solar Electric Propulsion

A spacecraft with a SEP gains its power from the Sun through a solar array. The intensity of the solar radiation lowers with increasing distance from the Sun, resulting in a decreasing maximum achievable thrust. The total power emitted by the Sun, assuming a perfect black body equals (Franzen, 2016):

$$P = 4\pi R_s^2 \sigma T^4 \quad (3.1)$$

with R_s as the radius of the Sun, σ the Stefan-Boltzmann constant and T as the effective temperature of the Sun, resulting in $P = 3.826 \cdot 10^{26}$ W.

The power density that is received by the solar array at a distance r from the Sun is, using the simplest model (Franzen, 2016):

$$E = \frac{P}{4\pi r^2} \quad (3.2)$$

This results in a quadratic decrease in available power with increasing distance from the Sun. This could lead to a point where the distance has become too large to have functional solar arrays with current technologies. NASA's spacecraft Juno broke the record of most distant solar-powered spacecraft from ESA's Rosetta, at 793 million km (5.3 AU) from the Sun and is currently orbiting Jupiter (NASA, 2016).

When orbiting closer to the Sun, a faster degradation of the solar panels due to solar radiation should be taken into account.

Nuclear Electric Propulsion

Robotic Solar System exploration missions can be equipped with a NEP system to generate power. Unlike SEP, a NEP unit is power-independent of the distance it has to the Sun. Ignoring degradation, the available power throughout the mission is constant and NEP systems have a high power density and long lifetime compared to conventional chemical propulsion. Several missions used NEP as power system including Voyager, Cassini-Huygens and New Horizon. Their main propulsion system is chemical-based and NEP is used to generate power for the internal spacecraft operations and instruments.

Originally, NEP is reactor-based but recent studies have shown the advantages of Radioisotope Power Systems (RPS) over reactor power systems (Schmidt et al., 2010). Radioisotope-based nuclear propulsion is also known as Radioisotope Electric Propulsion (REP) and combining REP with Hall-effect thrusters sufficient specific impulses can be reached (Schmidt et al., 2010). The energy produced is in heat and therefore a conversion unit is needed to convert the heat into electricity. This heat is emitted during the decay of a radioactive plutonium isotope.

At this moment, the thrust and power achievable with REP is approximately ten times smaller compared to SEP but REP-based spacecraft would be much smaller and lighter compared to reactor-based NEP, making REP a good candidate for interplanetary travel where chemical and solar electric propulsion systems are difficult or even not feasible (Schmidt et al., 2010).

3.3. Dynamics

The basic rocket theory discussed in this section is valid for both conventional propulsion and electrical propulsion.

The specific impulse I_{sp} is a parameter used to indicate the efficiency of an engine and it shows how much impulse can be obtained from a unit mass of propellant (Cornelisse et al., 1979). As seen in Table 3.1 electric engines have high specific impulses. With Equation 3.3 the specific impulse can be computed.

$$I_{sp} = \frac{T}{\dot{m}g_0} = \frac{V_e}{g_0} \quad (3.3)$$

In the above equation, T equals the thrust, \dot{m} the mass flow, g_0 the standard acceleration due to gravity at sea level on Earth, and V_e the exhaust velocity. The thrust of a rocket engine is computed with:

$$T = \dot{m}V_e \quad (3.4)$$

The mass flow \dot{m} can be obtained with

$$\dot{m} = -\frac{dM}{dt} = \frac{T}{V_e} = \frac{fM}{V_e} \quad (3.5)$$

where M equals the spacecraft mass and f the thrust acceleration. Using Equation 3.5, Tsiolkovsky's Equation can be derived:

$$\Delta V = V_e \ln \frac{M_0}{M} \quad (3.6)$$

Integrating Equation 3.5 and realizing that the mass flow of a spacecraft is actually negative, one obtains:

$$-\int_{M_0}^{M_e} \frac{dM}{M} = \int_0^{t_f} \frac{f}{V_e} dt \quad (3.7)$$

With a constant specific impulse and thus constant exhaust velocity and combining Equation 3.6, this becomes:

$$\Delta V = \int_0^{t_f} f dt \quad (3.8)$$

If the thrust profile f is known, the corresponding Delta-V can be computed using Equation 3.8.

3.3.1. Thrust Profiles

The trajectory of a spacecraft with a low-thrust propulsion system is greatly dependent on the magnitude, direction, and duration of the thrust force. As discussed in Subsection 3.2.1, there are two different thrust profiles regarding the behaviour of their magnitude and they are discussed first, followed by two thrusting profiles regarding direction. It should be mentioned that one of the thrust profiles for magnitude and direction must be combined to obtain a full thrust profile.

Constant Thrust

Constant thrust can be obtained with a NEP, or keeping the thrust levels of a SEP constant by increasing the power percentage available for propulsion with increasing distance from the Sun. This is, however, limited by the maximum amount of generated power of the SEP.

Thrust varying with radial distance

This applies to a SEP system. As previously discussed, the power generated by the solar arrays decreases by increasing distance from the Sun, resulting in a lower maximum thrust level.

Thrusting in tangential direction

It was found that thrusting with constant thrust tangentially to the velocity vector is close to the optimum thrust profile, i.e. $\delta = \gamma$. In this case the rate of change of the orbital energy reaches its maximum (Wakker, 2010).

Thrusting in radial direction

Thrust acting continuously in the radial direction is another option. However, research indicated it is not a feasible solution for interplanetary missions with the current technology (Wakker, 2010).

This chapter gave a brief overview of low-thrust propulsion. Various heritage missions were summarized to familiarize with the magnitude of thrusting force, accelerations and the corresponding trajectories. The numerical programs used to design the trajectories of the heritage missions were named and their most important design criteria were given. The technical aspects of low-thrust propulsion were named, resulting in a good understanding of low-thrust propulsion.

4

Simplified Perturbations Model

"Prediction is very difficult, especially if it's about the future."

—Nils Bohr, 1885-1962

NORAD saw the need of a space surveillance program in the early 1960's to keep track of the increasing number of satellites in orbits around Earth. At that time, satellites were observed at more than 150 locations throughout the United States. These observations were obtained using radar, telescopes and radio receivers and even visual sightings. Via personal messengers, telephone, mail and teletype, these observations were delivered at the National Space Surveillance Control Center (NSSCC). At this facility, the data was reduced and corrections were made by a data analyst before feeding the data to an IBM-709 computer, which was able to process the data within days. To get an idea of the computing power of this machine: it stored data on magnetic tapes and is easily overpowered by the smallest modern-day mobile phones, although it sold at that time for around 2.6 million dollars (Gjerlov, 2015).

Not only was it need-to-know where a spacecraft was at that moment in time, but it was even more important to know where the spacecraft was going to be in the next days. This processing evolved into the SGP4 algorithm as we know it today and this is discussed in this chapter. First, orbital propagation in general is discussed. This is followed by the history of SGP4 in Section 4.2. The input of SGP4, the TLE, is considered in Section 4.3. The original SGP4 algorithm is presented in Section 4.4 and this is followed by Section 4.5, where the behaviour of the orbital elements due to the perturbations added by SGP4 is investigated. Finally, the software required to run SGP4 and obtain plots of the results is mentioned and the chapter is concluded with a brief summary and conclusions.

4.1. Orbital Propagation

Not only is it desired to know the location of a satellite at the current time, but even more is it desired to know where the spacecraft will be in the near future. The determination of a spacecraft's future location, by making use of its initial conditions at a certain time, is known as propagation. The state of a spacecraft is often described as a vector in a certain coordinate frame, e.g. Cartesian coordinates with positions x, y, z and velocities $\dot{x}, \dot{y}, \dot{z}$, or orbital elements $a, e, i, \omega, \Omega, M$.

If all forces that act on the spacecraft are known, one is able to calculate the position forward in time. According to Newton's Laws, the equation of motion including perturbations for a satellite is (Wakker, 2010):

$$\frac{d^2\mathbf{r}}{dt^2} + \frac{\mu}{r^3} = -\nabla R + \mathbf{f} \quad (4.1)$$

where the disturbing potential R includes all perturbing accelerations that can be written as a potential function and all perturbing accelerations that cannot be written as the gradient of a scalar function are described by \mathbf{f} . With the computing power of computers today, Equation 4.1 can be solved using numerical integration techniques.

In the 1950's and 1960's computers were not powerful enough and thus approximate analytical methods were

developed. SGP4 is an example of such an analytical method.

The most simple and fastest approximation of the motion of a satellite around Earth would be if the spacecraft were in a perfect Kepler orbit: only the gravitational attraction of a perfectly spherical homogeneous Earth is taken into account. This results in the right-hand side of Equation 4.1 to become zero. The semi-major axis, eccentricity, inclination, argument of perigee and right ascension of ascending node would remain constant at all time. Of course, in reality perturbations do act and for accurate propagation these perturbations should be taken into account. When propagating a spacecraft in the far future, the highest accuracy is achieved when dividing this large time step in small time intervals and calculate the new state vector per time step. This would require a large number of calculations and a numerical integrator to solve Equation 4.1 to propagate the orbit of the spacecraft, resulting in long computation times.

Using an analytical method to propagate the orbit of the spacecraft will reduce the computation time significantly but it the accuracy could decrease as well. It also often has a domain: if the propagated time exceeds this domain, the accuracy will decrease more rapidly. Even today, the numerical propagation of a satellite orbit is time-consuming and for the entire satellite catalogue that is an impossible task. Even though the number of spacecraft has increased enormously compared to the 1960's, this was also impossible in that time. Not only for the computing power, but for the lack of efficient numerical integration methods as well. Thus approximate analytical methods were developed, such as the predecessor of SGP4, which is still widely used today.

4.2. History

The development of SGP4 started in 1959, when Dirk Brouwer suggested a solution for the motion of a near-Earth satellite under the influence of the zonal harmonics J_2, J_3, J_4 and J_5 . At the same time, Yoshide Kozai published a solution to the same problem using a different approach. In 1960, the NSSCC developed a new analytic prediction model, which was based on the work of both Brouwer and Kozai: Simplified General Perturbations (SGP). This model became the main orbital prediction model for NORAD. In the subsequent years, the computational limit of the computers was reached by the increasing number of satellites in the catalogue. A simplified solution as suggested by Lane and Cranford in 1969 was implemented as SGP4 (Hoots et al., 2004a). SGP4 was used alongside SGP during the next decade until 1979 when SGP4 became the leading propagation model.

In the following years, spacecraft became more operative at higher altitudes where other disturbances are present. This resulted in the development of the Simplified Deep Space Perturbations (SDP4) model, which is used for "deep space" missions that typically have an orbital period greater than 225 minutes. SDP4 takes into account the perturbations due to the Sun and Moon, as well as the sectoral and tesseral Earth harmonics.

The original SGP4 code was released in *Spacetrack Report Number 3* (Hoots and Roehrich, 1988) with the intention to make the implementation available to a larger audience. This technical report included the computer source code with most of the equations given and this code became the standard for using TLE sets. In *Spacetrack Report Number 3* five orbital propagation models were presented to the global user community: SGP, SGP4, SDP4, SGP8 and SDP8, which are all 'generally' compatible with the TLE data. SGP8/SDP8 included the special cases of orbital decay and re-entry, but there is no evidence that SDP8/SGP8 was implemented for operational TLE formation and is therefore not as popular as SGP4 is today.

Spacetrack Report Number 6 was released by NORAD in 1986 and although it had little to do with SGP4/SDP4, it confirmed insufficiencies in SGP4 found by amateur satellite trackers and researchers.

In 1990 NASA Goddard Space Flight Center (GSFC) released the unrestricted SGP4 code as part of a study on orbit propagation models for a mission and this was followed by an online release in 1996/1997. This version, the 'GSFC version', again confirmed changes already discovered by researchers and amateur satellite trackers. A complete history of the equations, background, and technical information of SGP4 was published by (Hoots et al., 2004a). (Vallado et al., 2006) noted that all published reports so far only suggested improvements and the code still lacked an implementation of named improvements. Therefore the code used in (Hoots et al., 2004a) was the base of the improved SGP4 code published in (Vallado et al., 2006). It was found that SGP4 and SDP4 share similarities and as a result, both versions are merged into SGP4. The code published in (Vallado et al., 2006) is used as a framework in this thesis research.

Today, the objects in Earth orbit are monitored by the Space Control Center in Cheyenne Mountain, Colorado which is still run by NORAD. They rely on the Space Surveillance Network (SSN), a network of sensors located at sites worldwide which are operated by the US Army (Kelso, 2018b). Figure 4.1 shows the locations of the sites.

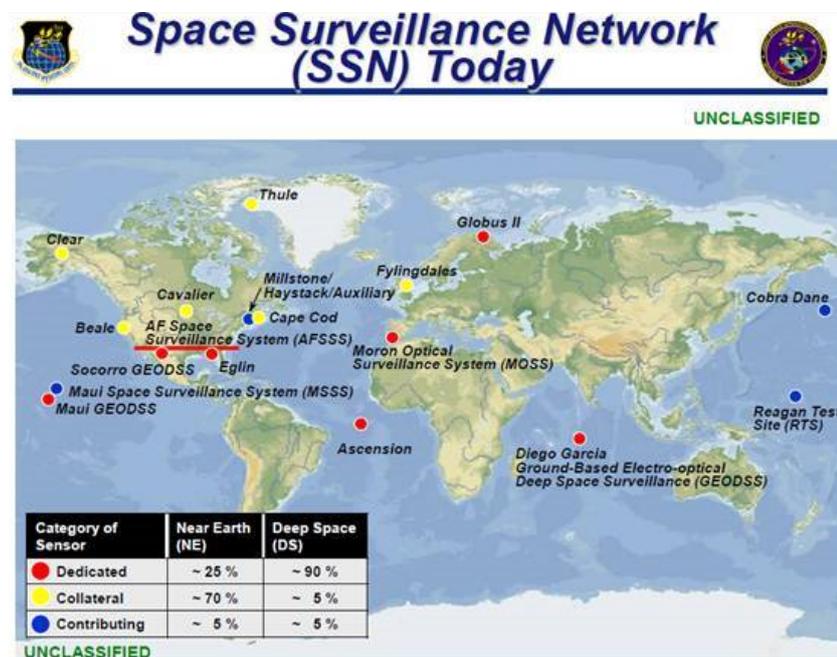


Figure 4.1: Space Surveillance Network Sites in 2013 (credit: spacesafetymagazine.com)

Every object in space that is larger than 10 centimeters is tracked and catalogued. Of all objects detected today the remaining 7 percent are operational satellites, the remaining 93 percent are either rocket bodies, inactive satellites or space debris. This shows the relevance of tracking objects in Earth orbit today: an enormous number of uncontrolled objects must be monitored in order to be able to avoid collisions with active spacecraft. Furthermore, when two objects do collide, the resulting space debris must be monitored as well since an object of 0.5 cm can already cause serious damage to a spacecraft. High-risk objects are constantly tracked by SSN (Kelso, 2018b).

The (initial) state of tracked objects is estimated on a regular basis, and documented in a so-called Two-Line Elements format (TLE). Thus to obtain the location of spacecraft in the near future, the TLE data is used by SGP4. With the increasing computing power available, the use of numerical methods for orbit propagation increases but still SGP4 is widely used today by the scientific and general community: the TLE data still represents the only public source of data of orbiting objects and this analytical method is in general faster than numerical methods. Examples of use are (Vallado et al., 2006):

- Rapid search for satellite visibility for ground stations.
- (Initial) programmed tracking of medium-beamwidth antennas.
- Initial orbit design with low-precision requirements.
- Collision prediction.

The following perturbations are taken into account in the SGP4 model.

- Atmospheric drag
- Earth zonal harmonics
- Lunar and Solar gravity effects
- Earth gravity resonance effects

4.3. Two Line Elements

The input of the SGP4 algorithm are mean orbital elements in the TLE format. This is a two-line ASCII text consisting of mean orbital elements of a spacecraft at a certain time. Mean orbital elements are orbital elements where the periodic variations are removed, and are therefore not equal to osculating orbital elements (Vallado et al., 2006).

In reality a TLE is often a three-line text file, with the top line representing the name of the spacecraft, as can be seen in Figure 4.2.

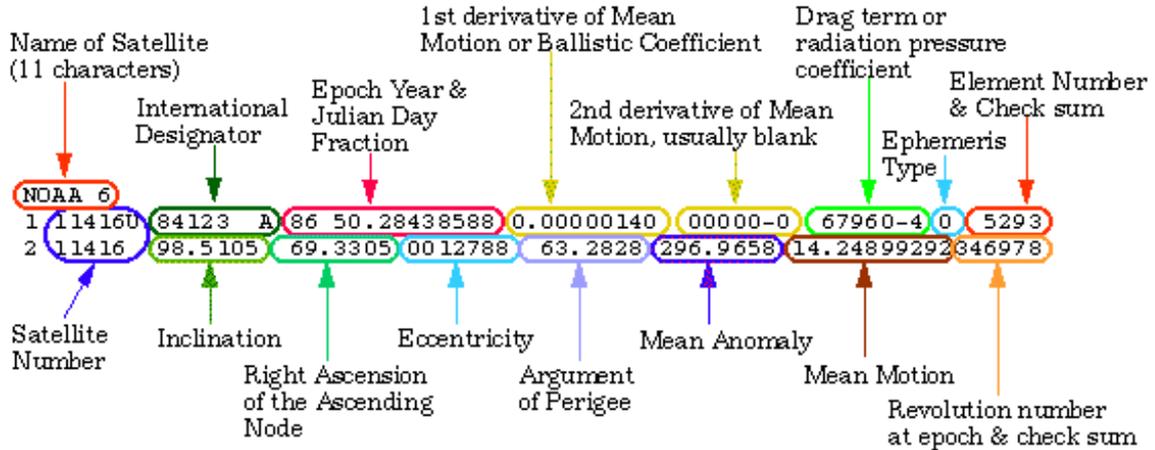


Figure 4.2: Two Line Elements break-down (credit: NASA).

The satellite numbers consist of any number between 0-99999. The International Designator consists of three parts: the first two digits show the last two digits of the launch year, followed by three digits for the launch number of that year, and the piece in the launch (3 digits).

The first two digits of the 'epoch year and Julian day fraction' show the two last digits of the epoch year. The remaining digits are the Julian day fraction, which consists of the day of the year and the fraction of the day.

The derivatives of the mean motions are not used by SGP4 and are only used in older SGP models.

On the bottom line, mean inclination, mean right ascension of the ascending node and mean argument of perigee are shown in degrees. For the mean eccentricity and the drag term B^* one should assume a leading decimal point.

Normally, ballistic coefficients (BC) or drag terms are used in aerodynamic theory but SGP4 uses B^* , which is an adjusted value of BC using the reference value of atmospheric density at one Earth radius (Vallado et al., 2006):

$$BC = R_e \rho_0 / (2B^*) \quad (4.2)$$

4.3.1. Retrieving TLE Data

TLE data can be obtained online from celestrak.org and space-track.org. Celestrak.org is operated by Dr. T. S. Kelso and was the first online source for TLE data. No registration is required to download TLE data (Kelso, 2018a) and data not published can be accessed on request. On space-track.org, United States Strategic Command (USSTRATCOM) publishes the TLE data of all objects in Earth orbit as well, but it requires the registration of an account to access the data. space-track.org is used in this thesis research to download the TLE data.

The format can be set to two line or three line data. In this thesis, the three line variant is used to obtain the name of the satellite as well. TLE's can be retrieved by satellite catalogue number, for a given data range, the last 5 available TLE's or the most recent. All TLE's combined of all tracked objects, the Satellite Catalogue (SATCAT), is also available as bulk download. This data consists of active spacecraft, but is mostly

space debris and rocket bodies. The latest SATCAT used in this thesis research dated October 24, 2018, and consisted of 17542 objects.

The SATCAT is updated twice per day, but not all spacecraft in the SATCAT are updated that frequently and thus remain unchanged. Subsets of the SATCAT can also be selected i.e. weather satellites or navigation satellites.

An interesting and extensive analysis of all bodies in orbit on January 10, 2011 and April 10, 2011 was performed by (Leloux, 2012). It would be interesting to perform the same analysis to compare the results after more than 7 years, but this is beyond the scope of this research.

The downloaded data from space-track.org is stored in an ASCII text file with .TLE extension and is ready to be used. In the standard SGP4 algorithm, which is discussed in the following section, the TLE data is loaded by the TLE reader developed by (Vallado et al., 2006).

4.4. Algorithm

The SGP4 algorithm described in this section is the algorithm as published in (Vallado et al., 2006). It consists of analytical expressions and returns a predicted state of a satellite at a certain moment in time. The equations used in SGP4 for the initializations and updates are the result of extensive analytical calculations dated from the 1950's-1960's and it is not relevant to show them here as these calculations are treated as black boxes. For the complete algorithm, the reader is referred to (Vallado et al., 2006).

It should be noted here that the TLE data is generated using SGP4 (or the equivalent used by NORAD) and therefore propagation is only valid when carried out by SGP4. Using another propagation tool will lead to incorrect results.

To initialize the SGP4 algorithm as provided by (Vallado et al., 2006) the following user inputs are required.

1. Choose between operation modes a and i . Operation mode a follows the conventional SGP4 code, whereas mode i follows an improved SGP4 behaviour. Since (Vallado et al., 2006) uses operation mode a for its validation, this operation mode is used for the thesis research as well.
2. Choose between different running options c , v and m . Option c compares one year of data from the satellite catalogue. Option v is a verification run and requires modified files as provided by (Vallado et al., 2006). Option m is the manual option and requires additional input. The start time and end time can either be given in minutes, calendar date, or Julian date.
3. Choose the World Geodetic System. The options are WGS-721, WGS-72 and WGS-84. The WGS-72 is standard in SGP4 and therefore used in the thesis research.
4. Give the input file name. This file contains one or more TLE.

The algorithm then takes the following steps.

1. The TLE is loaded and its elements are converted to the usable SGP4 units.
2. The orbit is initialized using `sgp4unit()` and all satellite properties and variables are stored in the `struct satrec`:
 - (a) For both near Earth and deep space: All variables set to zero.
 - (b) Identify Earth constants.
 - (c) In case of deep space perform `DSCOM()` for secular and periodic contribution:
 - i. Initialize lunar terms.
 - ii. Perform lunar and solar terms contributions:
 - (d) In case of deep space perform `DPPER()` for deep space long period periodic contributions.
 - i. Calculate time varying periodics:
 - ii. Apply periodics.

- (e) In case of deep space perform DSINIT() which provides deep space contributions to the derivative of the mean motion due to geopotential resonance with half day and one day orbits:
 - i. Initialize deep space constants.
 - ii. Perform solar and lunar contributions.
 - iii. Calculate deep space resonance effects.
 - iv. Initialize resonance effects.
 - v. Calculate geopotential resonances contributions .
 - (f) If not deep space: Initialize near Earth variables.
 - (g) Run the SGP4 algorithm at zero epoch to obtain the initial state vector.
3. Perform the propagation using sgp4() until $t = t_f$:
- (a) Update for secular gravity and atmospheric drag.
 - (b) Compute extra mean quantities.
 - (c) Add lunar and solar periodics (deep space: use DPPER() and DSCOM).
 - (d) Add long period periodics.
 - (e) Solve Kepler's Equation.
 - (f) Short period preliminary quantities.
 - (g) Compute orientation vectors.
 - (h) Compute position and velocity vectors.

The output position and velocity vectors are in the True Equator and Mean Equinox (TEME) coordinate system. Figure 4.3 shows the flow chart of the SGP4 algorithm. Note that for simplicity, only the deep-space options are shown. The near-Earth mode are always run, whereas the deep-space only runs when the deep-space conditions are met.

4.5. Behaviour of orbital elements

The behaviour of the orbital elements of a spacecraft in Earth orbit are plotted versus time to obtain a better understanding of the change in orbital elements and by which perturbations they are caused. Two different spacecraft are selected which have a very low eccentricity and both act in the 'near-Earth' or 'deep-space' regions as specified by SGP4, respectively. The first spacecraft is SL-16 R/B, which is a rocket body of a Zenith rocket with a semi-major axis of around 7220 km and an eccentricity of 0.00014. The SGP4 algorithm considers LEO a 'near-Earth' orbit and GEO a 'deep-space' orbit. The output Cartesian state vectors are converted to osculating orbital elements and are plotted versus the time. The SGP4 algorithm was run for one day with a step size of one minute and the results are shown in Figures 4.4 and 4.5. The spacecraft in GEO is Intelsat 4A-F1 with an initial mean inclination of 13.3105 degrees and mean eccentricity of 0.0003125. Intelsat is propagated for 14 days to obtain better comparable data: the number of revolutions is the same as for the Low-Earth Orbit (LEO) spacecraft. Both TLE's can be found in Appendix D.

The change in semi-major axis for the LEO case represents a sinusoid with an amplitude of around 8 km and shows no significant change over time (Figure 4.4a). Looking at the semi-major axis of the GEO spacecraft in Subfigure 4.5a, the amplitude is only 0.1 km and the mean of the semi-major axis decreases as well. The LEO eccentricity has a constant mean value, but has large negative and positive peaks. The GEO eccentricity has a constant mean value and a smaller amplitude. The inclination of the LEO spacecraft shows a similar pattern

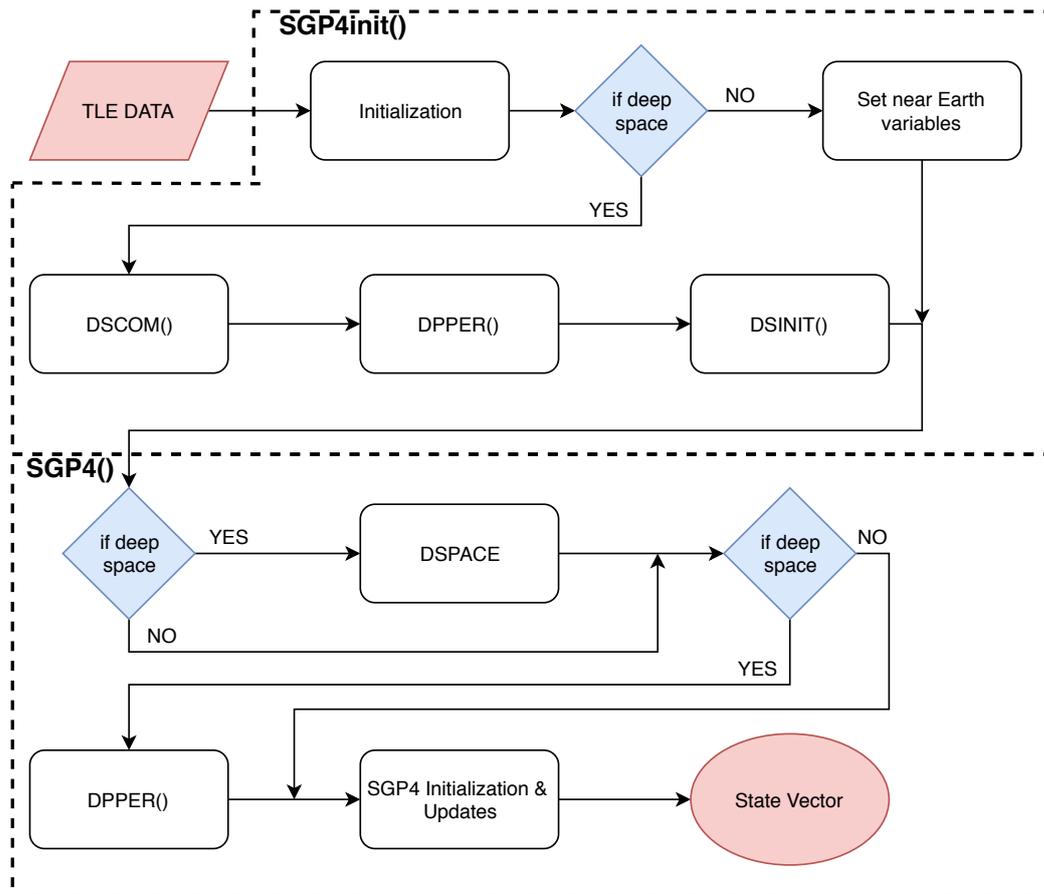


Figure 4.3: SGP4 algorithm structural organization, elaborated and based on (Vallado et al., 2006). For simplicity, only the deep space modules are shown. In the SGP4 algorithm, the near-Earth part is always run for both near-Earth and deep-space cases.

with the semi-major axis: a constant mean value and constant amplitude of around 0.01 degrees. The inclination in GEO shows a completely different pattern. It initially decreases slowly and around day 6 suddenly increases before it flattens out again near day 14. The total change in inclination is however not greater than 0.014 degrees.

The right ascension of the ascending node Ω decreases with time and this slope is steeper at LEO altitude (Figures 4.4d and 4.5d). This change is mainly the result due to gravity irregularities (Section 2.6.1). To illustrate this, the first-order approximation of the effect of this perturbation for the right ascension of the ascending node is, as explained in Section 2.6, (Wakker, 2010):

$$\Delta\Omega_{2\pi} = -3\pi J_2 \left(\frac{R_e}{p}\right)^2 \cos i \quad (4.3)$$

which results in a negative change per revolution for Ω in both cases, but this change decreases with increasing semi-major axis since $p = a(1 - e^2)$.

The argument of perigee shows a similar pattern as the eccentricity in LEO with a constant mean value and shifts between 0 and 360 degrees. In the GEO case however, the amplitude is much smaller, around 8 degrees and is slowly increasing and it thus has a smaller effect on the orbit of the spacecraft.

For circular orbits, the argument of perigee is undefined and in the case of very small eccentricities, ω is poorly defined. This explains the great shifts in ω in time and its similar pattern compared to the eccentricity.

4.6. Software

The described algorithm in the previous sections was published in a number of languages by (Vallado et al., 2006). The C++ version used in this research was provided by Ir. J. Geul who included CMake files, but no changes to the algorithm were made.

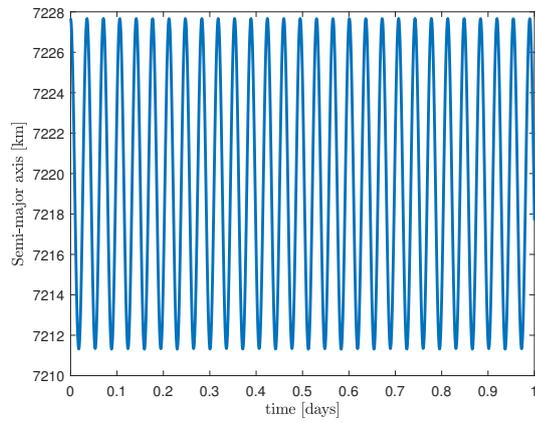
Modifications to the initializations code are required to obtain the baseline software usable for this research, which is further discussed in Chapter 5.

The code is run on a Windows computer using CMake and the Ubuntu 18.04 application for Windows 10. The plots are generated by Matlab. The written software is verified using the example TLE's and their results given in (Vallado et al., 2006). The output of the software tool is a ASCII file containing seven columns with time since start, x , y , x , \dot{x} , \dot{y} , and \dot{z} . This output was successfully compared to the verification data.

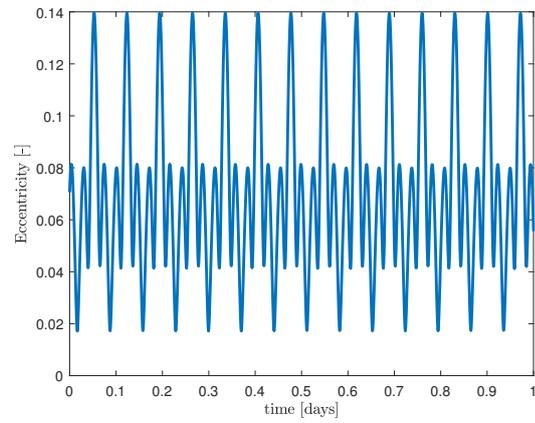
4.7. Conclusions

In this chapter SGP4 was considered. A brief history was given, the algorithm was explained and the behaviour of the orbital elements due to the perturbations was shown. It was found that there are significant differences between 'near-Earth' and 'deep-space' cases and that the changes in orbital elements are larger for the near-Earth case. Especially the change in Ω and ω have a significant influence on the orbit and this is mainly due to the effects of Earth's gravitational field. The issues of the current version of the SGP4 algorithm are discussed and although the latest version had a large number of bug fixes, the accuracy of SGP4 decreases fast after a few days. Modifications were made to the C++ SGP4 code provided by Ir. J. Geul to be able to use this code for this thesis research. The code is run using Cmake and Ubuntu 18.04 on Windows 10.

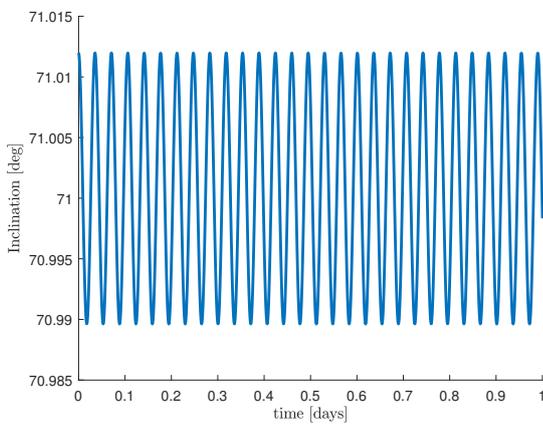
With the secrets of SGP4 now unravelled, the next chapter discusses the methodology of the designed analytical low-thrust trajectory design tool.



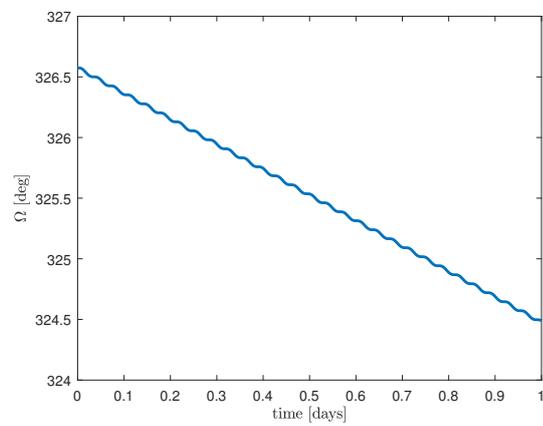
(a) Semi-major axis versus time



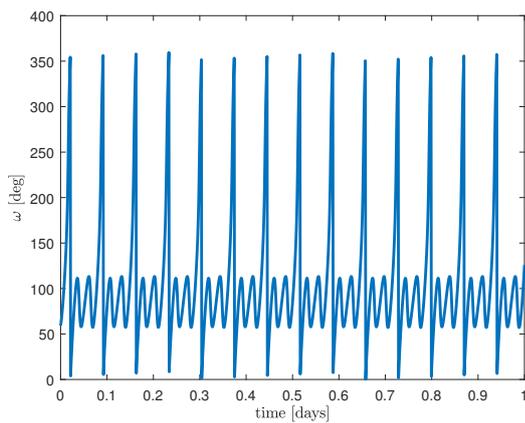
(b) Eccentricity versus time



(c) Inclination versus time.

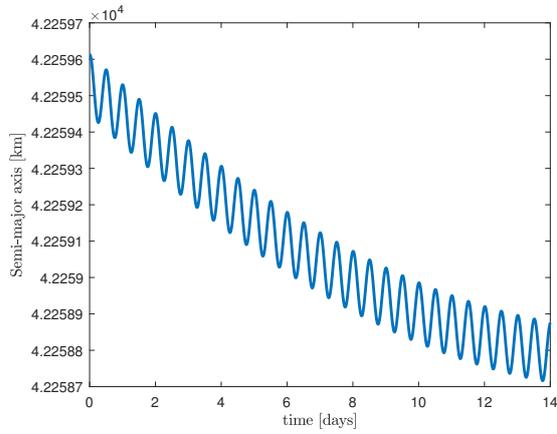


(d) Right ascension of the ascending node versus time.

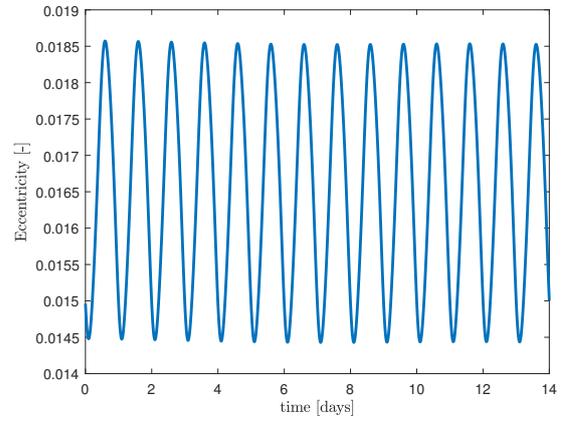


(e) Argument of perigee versus time

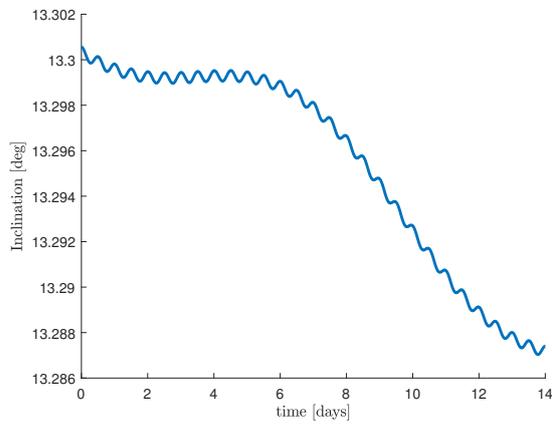
Figure 4.4: The converted osculating orbital elements of LEO spacecraft SL-16 R/S from the Cartesian output vectors of SGP4 for one day.



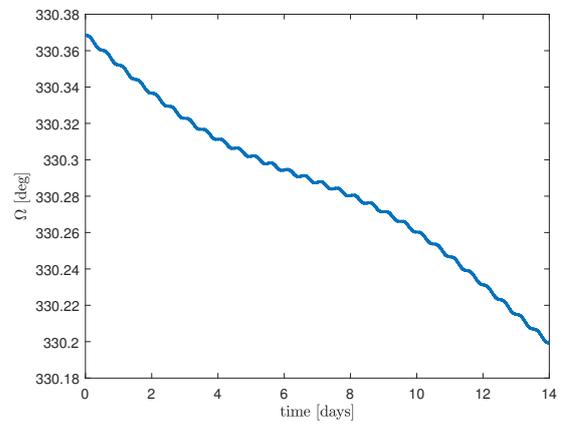
(a) Semi-major axis versus time



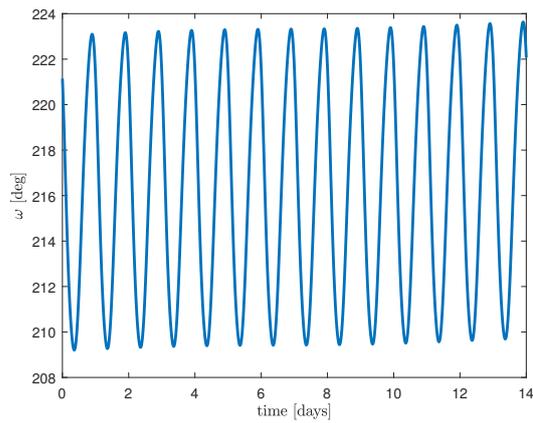
(b) Eccentricity versus time



(c) Inclination versus time.



(d) Right ascension of the ascending node versus time.



(e) Argument of perigee versus time

Figure 4.5: The converted osculating orbital elements of GEO spacecraft Intelsat 4A-F1 from the Cartesian output vectors of SGP4 for 14 days.

5

Methodology

"You could not direct the wind, but you could trim your sail so as to propel your vessel as you pleased, no matter which way the wind blew."

—Cora L.V. Hatch, 1840-1923

With an increasing number of spacecraft in orbit and increasing computer power, new techniques were developed to propagate spacecraft better and faster in time. Today, the increasing use of low-thrust propulsion requires a modified version of such a propagation tool.

The focus of this research is to design a low-thrust trajectory using SGP4. An existing low-thrust method that could interact with the SGP4 algorithm would be beneficial. As was discussed in the previous chapter, SGP4 uses mean orbital elements which are retrieved in the algorithm once.

However, a problem arises when the current SGP4 algorithm is combined with a low-thrust module. To illustrate this, the low-thrust trajectory solution is solely used which calculates a trajectory from orbit A to orbit B . At t_0 , the spacecraft has state \mathbf{X}_0 and after n time steps, the spacecraft arrived in desired location with state \mathbf{X}_f . This trajectory is schematic shown in black in Figure 5.1. Now let us consider the theoretical case where a low-thrust solution is implemented in SGP4 and the corresponding states and times are depicted in red. Due to the perturbations, the state of the spacecraft \mathbf{X}_1 will differ at each time interval than expected when applying solely the low-thrust transfer orbit method. E.g. the state \mathbf{X}_1 is not equal to the state \mathbf{X}_1 at t_1 . However, the low-thrust module is not aware of state \mathbf{X}_1 since it is only initialized at t_0 . This results in incorrect values for total transfer time, ΔV and final state. This problem is graphically explained in Figure 5.1, with the pure low-thrust trajectory depicted in black, and the trajectory obtained with the combination of the current SGP4 and a low-thrust trajectory design. The low-thrust trajectory solution is depicted in red. Thus at each time interval, the low-thrust trajectory design method must be initialized again to include the change in state due to the perturbations. Thus the low thrust solution is initialized again at t_1 using $\mathbf{X}_1 = \tilde{\mathbf{X}}_0$ where the tilde symbol

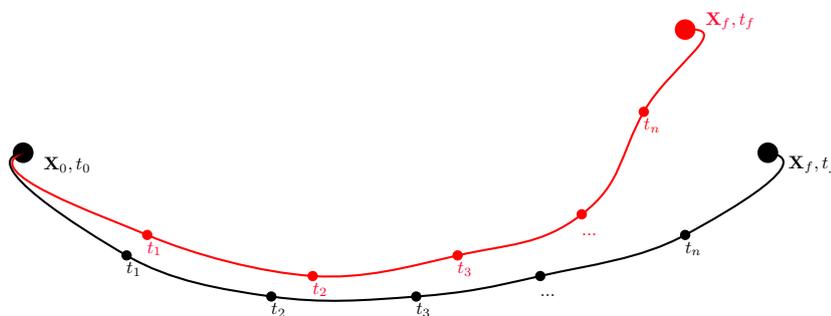


Figure 5.1: Visualization of the problem when the low-thrust design solution is only initialized at t_0 , with the black line and dots the low-thrust trajectory without perturbations. The red line indicates the low-thrust trajectory with perturbations, with the low-thrust solution, which does not take perturbations into account, only initialized at t_0 .

indicates a pseudo-initial state. This can become a time-consuming process when numerical methods are applied. It is therefore chosen to avoid these and choose a purely analytical method. Analytical shape-based methods use a predefined geometrical shape to design a low-thrust trajectory. Unfortunately, they require a numerical optimization to fulfil the boundary conditions such as the initial and final orbit properties and are thus not suitable methods. An extensive explanation of shape-based methods can be found in (J.G.P. de Jong, 2017).

Edelbaum developed an analytical method for inclination changes for two circular orbits of given size and fixed transfer time in the 1960's. This method was further developed by (Kechichian, 1997) and it was extended into a minimum transfer time problem using optimal control theory. This makes it a perfect candidate to be used in combination with the SGP4 algorithm.

This chapter discusses the developed low-thrust mission design tool SGP4-LT. The previous chapter discussed the SGP4 algorithm, which will be combined with the methods discussed in this chapter. First, Edelbaum's approach is discussed in Section 5.1. This is followed by Section 5.2 in which the combination of SGP4 and Edelbaum's solution is discussed. In Section 5.3 an iterative SGP4 version is presented, and is preceded by a novel approach to obtain the mean elements from Cartesian coordinates. Finally, this is followed by a discussion of the final tool in Section 5.4.

5.1. Edelbaum's low-thrust transfer orbit solutions

Edelbaum's solution is discussed in Section 5.1.1. This solution was later adjusted by (Kleuver, 2011) to include Earth-shadow eclipses which is discussed in Section 5.1.2.

5.1.1. Edelbaum's solution

To obtain a solution of an inclination change of a spacecraft using low-thrust propulsion, (Kechichian, 1997) used the Gaussian form of the Lagrange Planetary Equations for near-circular orbits, using the following assumptions: the starting orbit and final orbit are both circular and therefore V_0 and V_f are known.

The Gaussian form of the Lagrange Planetary Equations for near-circular orbits are derived in Appendix B and are (Kechichian, 1997):

$$\frac{da}{dt} = \frac{2af_N}{V} \quad (5.1)$$

$$\frac{de}{dt} = \frac{f_S \sin \theta + 2f_N \cos \theta}{V} \quad (5.2)$$

$$\frac{di}{dt} = \frac{f_W \cos(\omega + \theta)}{V} \quad (5.3)$$

$$\frac{d\Omega}{dt} = \frac{f_W \sin(\omega + \theta)}{Vi} \quad (5.4)$$

$$\frac{d\omega}{dt} = -\frac{d\Omega}{dt} + \frac{1}{eV} (2f_N \sin \theta - f_S \cos \theta) \quad (5.5)$$

For $e = 0$ and $i = 0$ singularities will occur and therefore another set of variables is used:

$$e_x = e \cos(\Omega + \omega) \quad , \quad e_y = e \sin(\Omega + \omega) \quad (5.6)$$

$$i_x = i \cos \Omega \quad , \quad i_y = i \sin \Omega \quad (5.7)$$

In above equations, A_R is the thrust component in the radial direction, A_T tangential to the spacecraft motion and A_W the out-of-plane direction. Using the angle α between the thrust projection on the orbital plane and tangential direction, and angle β between the thrust and orbital plane, the thrust components can be written as:

$$f_S = (T/m) \sin \alpha \cos \beta, \quad f_N = (T/m) \cos \alpha \cos \beta, \quad f_W = (T/m) \sin \beta \quad (5.8)$$

In Edelbaum's method, only out-of-plane and tangential thrust is applied, resulting in $\alpha = 0$ and thus $f_S = 0$. Edelbaum assumed Ω and ω constant. Therefore Equations 5.4 and 5.5 can be neglected. In reality, Ω and ω will vary during the manoeuvre but in Edelbaum's theory only complete revolutions are considered after which it is assumed that Ω and ω will obtain their initial values again.

β is kept piecewise constant and switches sign at the orbital antinodes. Furthermore, it is assumed that the orbit remains circular during transfer: $de_x/dt = de_y/dt = 0$. It is assumed there is only tangential and out-of-plane thrust, therefore setting $f_n = 0$. Gauss' form of the Lagrange planetary equations further reduce to:

$$\frac{da}{dt} = \frac{2af_t}{V} \quad (5.9)$$

$$\frac{di}{dt} = \frac{\cos\theta f_h}{V} \quad (5.10)$$

$$\frac{d\theta}{dt} = n \quad (5.11)$$

Next, the angular position θ is averaged out by integrating with respect to θ while f , β and V are kept constant:

$$\int_0^{2\pi} \left(\frac{di}{dt} \right) d\theta = \frac{2f \sin\beta}{V} \int_{-\pi/2}^{\pi/2} \cos\theta d\theta \quad (5.12)$$

which results in β being now a continuous function of time:

$$\frac{di}{dt} = \frac{2f \sin\beta}{\pi V} \quad (5.13)$$

Using Equations 5.1 and 2.12, and realizing $a = r$ for circular orbits, the time rate of change of velocity is obtained and shown in Equation 5.14.

$$\frac{dV}{dt} = -f \cos\beta \quad (5.14)$$

To make V the independent variable and i and t the dependent variable, Equations 5.13 and 5.14 are replaced with Equations 5.15 and 5.16.

$$\frac{di}{dV} = -\frac{2 \tan\beta}{\pi V} \quad (5.15)$$

$$\frac{dt}{dV} = -\frac{1}{f \cos\beta} \quad (5.16)$$

Further derivation results in Equations 5.22-5.18 with time t as the only variable, which can be implemented in a computer program (Kechichian, 1997)

$$\tan\beta_0 = \frac{\sin(\frac{1}{2}\pi\Delta i)}{\frac{V_0}{V_f} - \cos(\frac{1}{2}\pi\Delta i)} \quad (5.17)$$

$$\Delta i(t) = i_0 - \frac{2}{\pi} \left[\arctan\left(\frac{ft - V_0 \cos\beta_0}{V_0 \sin\beta_0}\right) + \frac{\pi}{2} - \beta_0 \right] \quad (5.18)$$

The beta angle and velocity as a function of time are derived as (Kechichian, 1997):

$$\beta(t) = \arctan\left(\frac{V_0 \sin\beta_0}{V_0 \cos\beta_0 - ft}\right) \quad (5.19)$$

$$V(t) = \sqrt{(V_0^2 - 2V_0ft \cos\beta_0 + f^2t^2)} \quad (5.20)$$

In Equations 5.19 and 5.20 the initial values β_0 and V_0 are obtained, respectively, when $t = 0$ is used. Thrust is continuous and turned on all time, and thus the amount ΔV required equals the the thrust acceleration times the time: $\Delta V = ft$. From Equation 5.18 it follows then:

$$\Delta V = V_0 \cos\beta_0 - \frac{V_0 \sin\beta_0}{\tan(\frac{1}{2}\pi\Delta i + \beta_0)} \quad (5.21)$$

Then the total flight time is calculated using:

$$\Delta t = \Delta V / f \quad (5.22)$$

This method as suggested by Edelbaum and (Kechichian, 1997) has continuous thrust over the complete transfer orbit. Minimizing the transfer time is also equivalent to minimizing the ΔV since thrust is always on during the transfer.

An inclination change is most efficient when it is performed at or near the ascending or descending node (where true longitude $u = 0$ or $u = \pi$) and it is not efficient to thrust near and at the anti-nodes (where true longitude $u = 1/2\pi$ or $u = 3/2\pi$) and therefore thrust is wasted at these locations. Furthermore, in reality both Ω and ω vary during the manoeuvre but Edelbaum's method only considers complete revolutions after which Ω and ω have obtained their initial values again. It follows that when the transfer time is short, e.g. less than one revolution is needed, Edelbaum's method might not give the correct results (Kechichian, 1997).

5.1.2. Edelbaum's solution with Earth-shadow eclipses

(Kleuver, 2011) improved Edelbaum's method by (Kechichian, 1997) further by including Earth-shadow eclipses. SEP spacecraft cannot thrust during eclipse if not specifically designed to carry a dedicated battery and therefore no constant thrust can be applied. To implement discontinuous thrust, first the amount of ΔV needed is written as:

$$\Delta V_{k+1} = \Delta V_k + f \Delta t; \quad k = 0, 1, 2, \dots, N \quad (5.23)$$

with $\Delta t = t_{k+1} - t_k$. It is assumed that the total ΔV required matches the Edelbaum solution for the desired orbit manoeuvre and therefore ΔV_k is determined by evaluating Edelbaum's solution at discrete interval and thus the required time step for a known increase in ΔV can be calculated using:

$$t_{k+1} = t_k + \frac{\Delta V_{k+1} - \Delta V_k}{f}; \quad k = 0, 1, 2, \dots, N \quad (5.24)$$

To implement discontinuous thrust due to the Earth-shadow eclipse, Equation 5.24 is modified to

$$t_{k+1} = t_k + \frac{\Delta V_{k+1} - \Delta V_k}{\bar{f}_k w_k} \quad (5.25)$$

where \bar{f}_k is the average thrust acceleration. This is computed using the thrust magnitude T , which is assumed constant, and the average mass between discrete steps k and $k + 1$. This mass can be calculated using Tsiolkovsky's rocket equation (Eq. 3.6) with I_{sp} and the known ΔV . In Equation 5.25, weighting function w_k represents the percentage of time the spacecraft is thrusting during one revolution. E.g. $w_k = 0.75$ implies that the spacecraft is in sunlight for three quarters of a revolution. This weighing function is calculated using the Earth shadow angle $\Delta\theta_{SH}$. This arc is calculated using the method developed by (Neta and Vallado, 1998).

$$w = 1 - \frac{\Delta\theta_{SH}}{2\pi} \quad (5.26)$$

In reality, discontinuous thrust will eventually lead to an increase in eccentricity but numerically integrated optimal transfers have shown that the maximum eccentricity never exceeds 0.15 and on average not more than 0.1. This variation in eccentricity will thus have a small effect on the transfer orbit compared to the pure Edelbaum solution (Kleuver, 2011).

This version of Edelbaum's method with Earth-shadow eclipses is not yet implemented in the tool and is left as a recommendation.

5.2. Low-Thrust Module

The developed software tool SGP4-LT combines the two previously explained analytical satellite propagation methods. Edelbaum's theory (Section 5.1) is implemented in SGP4 (Chapter 4) resulting in a satellite propagation or design tool for low-thrust propelled spacecraft in Earth orbit. Edelbaum's method gives analytical functions for the required change in orbital elements and the information of the final target orbit can be set while TLE's are used as input data. In order to combine these two methods it has to be checked whether they are compatible. The input of SGP4 is the data from TLE, which are mean orbital elements with the periodic variations removed, whereas Edelbaum's solution uses osculating orbital elements.

Before the correct solution was found, time was spent on the development of an incorrect model. The option of using the osculating orbital elements as input mean elements was tested and it was shown that this is not a

valid option. Using this method, it was found that at time steps smaller than 10 minutes, the spacecraft would crash into Earth. The mean motion, which was directly obtained from the osculating elements, was too large compared to its actual mean value and per time epoch this mean motion became larger until it corresponded to a semi-major axis equal to the radius of the Earth. Further details of this approach and its conclusion can be found in Appendix C.

The position where to implement the low-thrust module is determined. When looking at the SGP4 algorithm, one can distinguish between secular effects, long-period effects and short-period effects, present in the algorithm in this order. As stated in Section 5.1 the thrust force is constant during the transfer, but its direction (the thrust angle β) varies. Figure 7.1b shows the inclination during a simple inclination change and this is clearly not a short-period or long-period movement. It is therefore decided to implement the Low-Thrust Module right after the secular effects in the SGP4 algorithm.

At initialization of SGP4, $\text{SGP4init}()$ is run. The purpose of SGP4init is to obtain the state vector at epoch t . In $\text{SGP4init}()$ no thrust is applied. Next, $\text{SGP4}()$ is evaluated at epoch $t + \Delta t$, with the thrust turned on and a state vector which includes the effect of the thrust is obtained.

An issue arises when using Edelbaum in the middle of the SGP4 algorithm. SGP4 does not use a semi-major axis but uses the Kozai mean motion and converts it into the Brouwer's convention for the mean motion. The Edelbaum method used in the low-thrust module requires the semi-major axis in kilometres. Thus, this semi-major axis needs to be obtained from the stored mean motion in Kozai convention, using Equation 5.27.

$$a = \sqrt[3]{\mu \left(\frac{86400}{2\pi n_0} \right)^2} \quad (5.27)$$

with a in km, μ in km^3/s^2 and n_0 the Kozai mean motion in revolutions per day.

Next, the semi-major axis in Brouwer's convention is also obtained from the stored Kozai mean motion using Equations 5.28 - 5.33 (Hoots et al., 2004a). It is chosen to store the Kozai motion instead of the Brouwer's convention.

$$a_1 = \left(\frac{k_e}{n_0} \right)^{2/3} \quad (5.28)$$

$$\delta_1 = \frac{3}{2} \frac{k_2}{a_1^2} \frac{3 \cos^2 i_0 - 1}{(1 - e_0^2)^{3/2}} \quad (5.29)$$

$$a_2 = a_1 \left(1 - \frac{1}{3} \delta_1 - \delta_1^2 - \frac{134}{81} \delta_1^3 \right) \quad (5.30)$$

$$\delta_0 = \frac{3}{2} \frac{k_2}{a_2^2} \frac{3 \cos^2 i_0 - 1}{(1 - e_0^2)^{3/2}} \quad (5.31)$$

$$n_0'' = \frac{n_0}{1 + \delta_0} \quad (5.32)$$

$$a_0'' = \left(\frac{k_e}{n_0''} \right)^{2/3} \quad (5.33)$$

using

$$k_2 = \frac{1}{2} J_2 a_E^2 \quad [\text{units of (Earth radii)}^2]$$

$$J_2 = 1.082616e - 3$$

$$k_e = \sqrt{GM} = 0.0743669161 \quad [\text{units of (Earth radii)}^{1.5}/\text{minute}]$$

G = universal gravitational constant

M = mass of the Earth

a_E = equatorial radius of the Earth

i_0 = inclination

This stored mean motion is the mean motion at t_{n-1} and has not been altered by perturbations acting at t_n . Let the associated unaltered Brouwer's convention be a_n , and the unaltered inclination i_n .

SGP4 gives at this point in the algorithm updated values for the elements after the secular perturbations and drag effects. Let's call these elements, the Brouwer's convention of mean motion and inclination, the only two elements that change in Edelbaum's analytical method, $a_{SGP4(n+1)}$ and $i_{SGP4(n+1)}$. The semi-major axis is retrieved and the Edelbaum algorithm runs now using the values a_n , i_n . The corresponding $a_{EB(n+1)}$ and $i_{EB(n+1)}$ are calculated. This implies that Edelbaum completely renews for every time epoch taken: it takes a_n and i_n as if it were the orbit parameters of the initial orbit and calculates a new transfer orbit at each time epoch.

Using this approach results at t_{n+1} in two different values for semi-major axis and inclination. The semi-major axis and inclination due to the secular and drag effects are expressed as $a_{SGP4(n+1)}$ and $i_{SGP4(n+1)}$. The semi-major axis and inclination due to the addition of low thrust are expressed as $a_{EB(n+1)}$ and $i_{EB(n+1)}$. Combining these values gives the new, updated values for a_{n+1} and i_{n+1} then become:

$$a_{n+1} = a_{SGP4(n+1)} + (a_{EB(n+1)} - a_n) \quad (5.34)$$

$$i_{n+1} = i_{SGP4(n+1)} + (i_{EB(n+1)} - i_n) \quad (5.35)$$

The updated elements are fed to the remaining part of the SGP4 algorithm. As said before, Edelbaum's solution is not correct for periods of time shorter than one revolution since it assumes Ω and ω to be constant. However by initializing Edelbaum's solution again at every time step, this problem is largely eliminated since the changes in Ω and ω will be small.

The flow diagram of the low-thrust module is shown in Figure 5.2. The complete structural organization of the SGP4-LT routine can be found Figure 5.3.

The software is written in such a way, that an improved analytical method can easily replace Edelbaum's solution inside the `SGP4()` function.

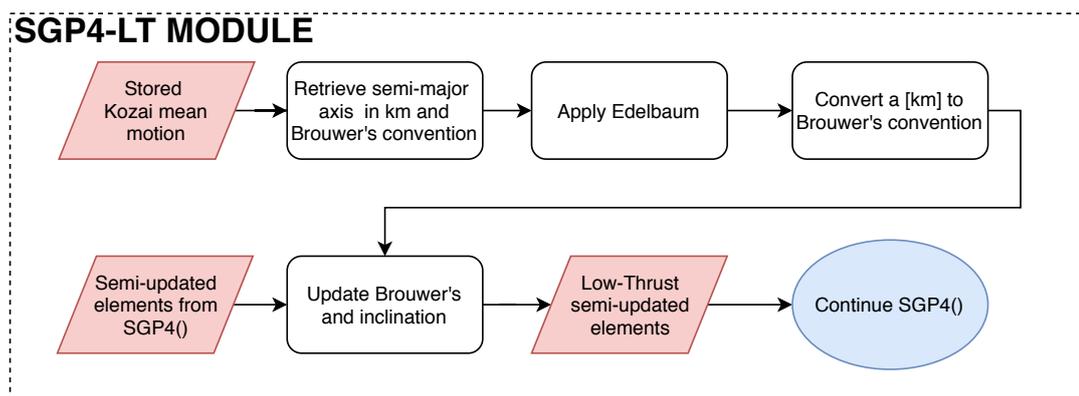


Figure 5.2: SGP4-LT algorithm structural organization as inside the `SGP4()` routine.

The input file of SGP4 is, as explained, a TLE data file and still is for SGP4-LT. It is therefore up to the user to decide if the algorithm runs with or without the low-thrust module. The thrust force is assumed constant and thus it is decided to treat the magnitude of the thrust force acceleration as a user input as well.

5.3. Iterative SGP4 Algorithm

The previous section discussed the SGP4-LT module and the procedure of one time step was explained. As cited before, the SGP4 algorithm only works correctly when the NORAD mean orbital elements are used as input. The output of SGP4 is a state vector with Cartesian coordinates and velocity components. Furthermore, at the start of the algorithm all constants are initialized and stored. When the orbit significantly changes due to the addition of a thrust force these initial values will become inaccurate and a re-initialization is needed. As was explained before, the low-thrust module requires an initialization at every time step.

The output of SGP4 cannot be directly used as input as was shown in Appendix C. It is therefore desired to obtain mean elements from the output of SGP4 to replace the mean elements that were used as input.

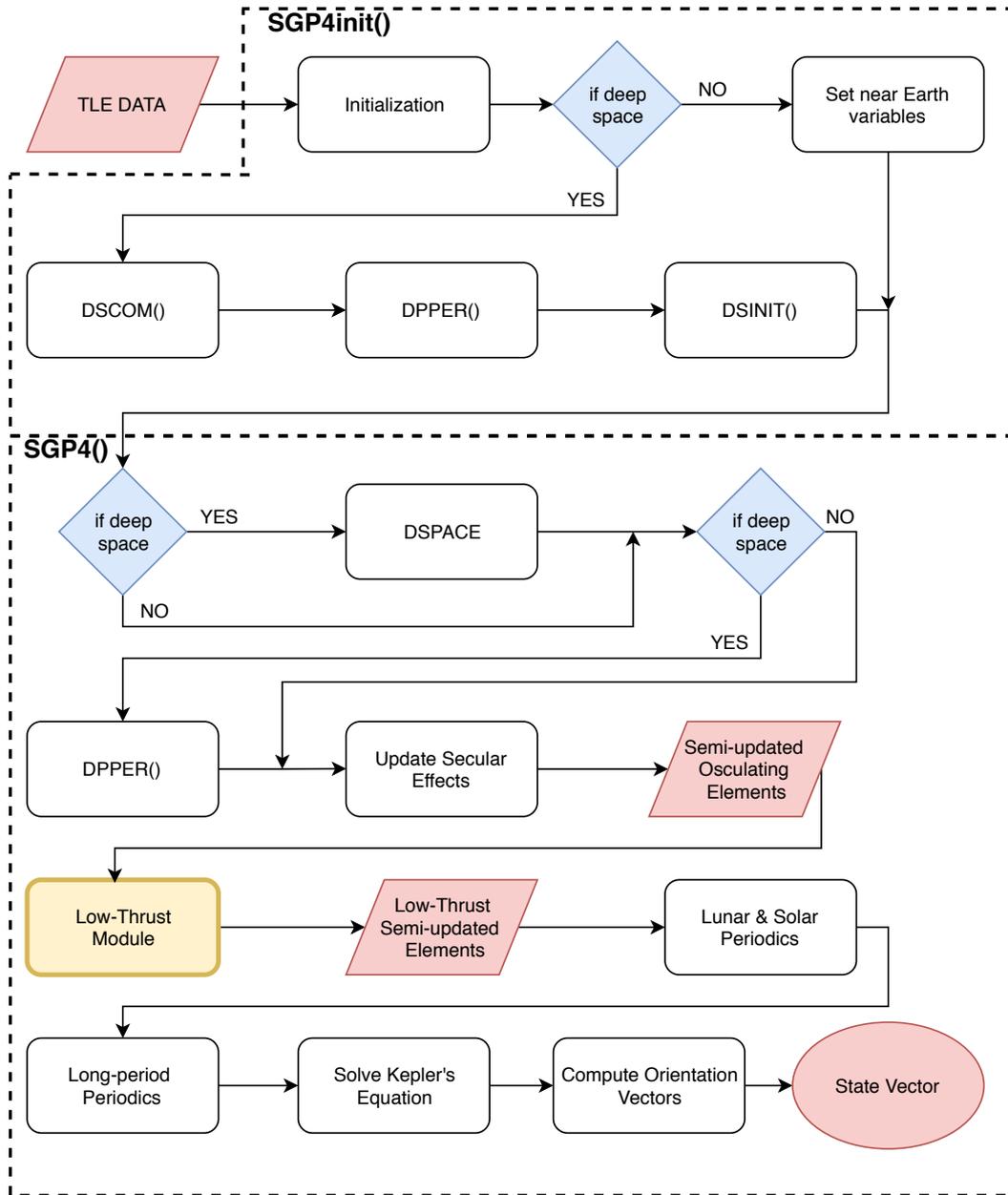


Figure 5.3: Structural overview of SGP4init() and SGP4() with low-thrust module.

In literature various methods are suggested to obtain the mean elements from osculating elements: (Walter, 1967) developed an iterative process based the expansion of Taylor series. (Ely, 2013) suggested a numerical method using a Fast Fourier Transform, (Der and Danchick, 1996) obtains the mean elements using a first-order theory of secular perturbations and a transition matrix with numerical partials and (Lee, 2002) uses another iterative procedure and tried to derive the drag parameter B^* from a set of osculating orbital elements. These numerical methods are ignored, since the focus of this thesis research has an analytical approach. The method of (Andersen, 1994) resembles the work of (Walter, 1967) and gives a good framework to use for this thesis research.

5.3.1. Corrected Andersen's Method

The algorithm developed by (Andersen, 1994), which is clearly based on the theory by (Walter, 1967), treated the SGP4 algorithm as a black box and consists of a direct iteration approach to obtain the mean elements from the output state vector. Furthermore, this algorithm is capable of an addition of an impulsive manoeuvre.

The osculating orbital elements and the mean elements are respectively represented as:

$$\tilde{\mathbf{y}}(t) = [a \ e \ i \ \Omega \ \omega \ M]^T, \quad \mathbf{y}(t) = [\bar{e} \ \bar{i} \ \bar{\Omega} \ \bar{\omega} \ \bar{M} \ \bar{n}_0]^T \quad (5.36)$$

The notation of the mean elements vector follows the notation of the TLE sets where the mean motion in revolutions per day (n_0) is used. The derivatives of the mean motion are not used in SGP4 and are therefore ignored. The ballistic coefficient is assumed to be constant at all times. Then, let

$$\mathbf{f}(\mathbf{y}_0, t_0) = \mathbf{X}_0 \quad (5.37)$$

with \mathbf{f} as the SGP4 algorithm, \mathbf{y}_0 the mean elements at time t_0 , \mathbf{X}_0 the state vector as the output of SGP4. Subtracting the state at both sides of this equations leads to a new function which yields:

$$F \equiv \mathbf{f}(\mathbf{y}, t) - \mathbf{X} = 0 \quad (5.38)$$

When \mathbf{X} is known and an approximation, or when a guess $\tilde{\mathbf{y}}$ is used, the left-hand side of Equation 5.38 is not equal to zero but to an error $\Delta\mathbf{X}$:

$$F \equiv \mathbf{f}(\tilde{\mathbf{y}}, t) - \mathbf{X} = \Delta\mathbf{X} \quad (5.39)$$

The goal of the developed algorithm by Andersen is to obtain a $\Delta\mathbf{X}$ as small as a set tolerance. The mean elements are corrected by approximating the correction with the change in osculating elements between the elements of the first guess and the elements of the last iteration step. The new guess becomes the last guess plus this correction. In iteration notation this leads to:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + (\tilde{\mathbf{y}}_0 - \tilde{\mathbf{y}}_i) \quad (5.40)$$

The mathematical derivation to this equation can be found in (Walter, 1967).

Andersen's algorithm is implemented in C++ and modified to function with the latest version of SGP4. This updated version takes the following steps at each time step. Furthermore, an error was found and corrected. This error resulted in an incorrect calculation of the maximum difference for the velocity components of the state vector.

1. Read in data: TLE data \mathbf{y} , propagation time and tolerance ϵ .
2. Run `SGP4init()` and obtain state vector \mathbf{X}_0 at t_0 .
3. Run `SGP4()` at $t_0 + \Delta t$.
4. Convert output state vector \mathbf{X} to osculating elements $\tilde{\mathbf{y}}_0$, which is the first guess and used throughout the iteration.
5. Let $\mathbf{y}_0 = \tilde{\mathbf{y}}_0$.
6. Run `SGP4init()` using \mathbf{y}_i and obtain state vector \mathbf{X}_i .
7. Compute the difference $\Delta\mathbf{X}_i = \mathbf{X}_0 - \mathbf{X}_i$

8. Convert \mathbf{X}_i to orbital elements $\tilde{\mathbf{y}}_i$.
9. Check for convergence: If $\Delta\mathbf{X}_i < \epsilon$ stop the iteration and let $\mathbf{y} = \tilde{\mathbf{y}}_i$.
10. If convergence is not reached set $\mathbf{y}_{i+1} = \mathbf{y}_i + (\tilde{\mathbf{y}}_0 - \tilde{\mathbf{y}}_i)$ and return to step 7.

Andersen tested his algorithm by applying it to 15 different TLE data sets, with convergence criteria of one centimetre in position and one centimetre per second in velocity. There was no added time step to SGP4 and thus is the output mean elements vector equal to the input mean elements vector. This makes it convenient to check whether the algorithm works correctly. In Table 5.1 the test cases are shown. Cases 1 and 2 are test cases from (Hoots and Roehrich, 1988) with cases *b* are altered versions of these TLE's. The test cases 1b and 2b have changed values for the mean eccentricity, mean argument of perigee, mean mean anomaly and mean mean motion. The cases 1a and 2a consist of an impulsive manoeuvre and are therefore not taken into account. Cases 3 - 11 are TLE's from actual spacecraft. Although some of them have decayed today, the TLE data is still available online and this data was retrieved via space-track.org. The used TLE data of decayed satellites are from 1994-06-01 and 1994-01-01. The most recent TLE's have been used for the still functioning spacecraft. There was no data available for satellite 1994004A. All used TLE's can be found in Appendix D.

Table 5.1 shows the mean elements of these TLE's for each spacecraft. The column right to the mean mean anomaly column with " Σ " shows the summation of the mean argument of perigee and mean mean anomaly. According to Andersen, five cases failed to converge (TDRS-4, Mir, Kvant-1, EUVE and HST), while the other cases required 4 to 175 iterations to converge. It was not succeeded to reproduce the result of TDRS-4 obtained with Anderson's method. The results obtained by Andersen and the results obtained by implementing Andersen's method with the latest version of SGP4 are both shown in column 'I' in Table 5.1.

Two of the four failed cases (Mir and EUVE) did still not converge using the updated C++ code with the previously mentioned correction: the other two cases did converge after the error was found and corrected. They did not converge with both Anderson's method and the corrected method. The test case of TDRS-4 would also not converge using the corrected method.

According to Andersen's conclusions, column I of Table 5.1, non-convergence occurred when the summation equals 360 degrees but removing the error resulted in two converging cases at a summation of 360 degrees, making it a false conclusion. The results of the corrected Andersen's method can be found in column II of Table 5.1.

No convergence criteria were given by (Andersen, 1994) but according to (Walter, 1967) the algorithm converged always for values for eccentricity between 0.01 and 0.30, and between 0.1 and 90.0 degrees for inclination.

A closer look was taken at the values of the elements during the iteration and it was found that the eccentricity at the non-converging cases quickly 'runs away' from the desired value. Furthermore, the cases with high or zero inclination require a large number of iterations compared to the other cases with moderate inclinations. It is concluded that singularity of eccentricity and inclination when close to zero contributes considerable to this non-convergence. This plays a role at the numerical calculation of the updated elements using Equation 5.40.

Since it was clear that Andersen based his research on (Walter, 1967), it is decided to name the solution after H. G. Walter. This solution, the Modified Walter's Method, is based on (Walter, 1967) but uses the corrected version of the algorithm published by (Andersen, 1994) and is presented in the next section.

5.3.2. Modified Walter's Method

In order to obtain a correct and robust SGP4-LT model, the correct mean elements need to be obtained at each time step. Since the transfer orbit changes largely in time, it cannot be guaranteed that the algorithm will converge to the correct mean elements later in time, even when it did converge at the start time. A non-sensitive algorithm is therefore required and a solution to the problems of Andersen's algorithm needs to be found.

In the previous section it was shown that the non-convergence was due to the singularity of the eccentricity and inclination. It was therefore decided to modify Walter's solution using modified equinoctial elements which only has one case of singularity: when the inclination is equal to 90 degrees (Section 2.5). The modified equinoctial elements replace the elements in Equation 5.40, and afterwards the conversion back to orbital

elements is made since these are required as input of the SGP4 algorithm. Andersen's algorithm then changes, with the changes depicted in red, to:

1. Read in data: TLE data \mathbf{y} , propagation time, ΔV and tolerance ϵ .
2. Run `SGP4init()` and obtain state vector \mathbf{X}_0 at t_0 .
3. If required, add ΔV to the state.
4. Run `SGP4()` at $t_0 + \Delta t$.
5. Convert output state vector \mathbf{X} to **modified equinoctial elements $\tilde{\mathbf{y}}_0^{mee}$** , which is the first guess and used throughout the iteration.
6. **Convert $\tilde{\mathbf{y}}_0^{mee}$ to osculating elements $\tilde{\mathbf{y}}_0$.**
7. Let $\mathbf{y}_0 = \tilde{\mathbf{y}}_0$.
8. Run `SGP4init()` using \mathbf{y}_i and obtain state vector \mathbf{X}_i .
9. Compute the difference $\Delta\mathbf{X}_i = \mathbf{X}_0 - \mathbf{X}_i$.
10. Convert \mathbf{X}_i to **modified equinoctial elements $\tilde{\mathbf{y}}_i^{mee}$.**
11. Check for convergence: If $\Delta\mathbf{X}_i < \epsilon$ stop the iteration and let $\mathbf{y}^{mee} = \tilde{\mathbf{y}}_i^{mee}$ and convert \mathbf{y}^{mee} to \mathbf{y} .
12. If convergence is not reached set $\mathbf{y}_{i+1}^{mee} = \mathbf{y}_i^{mee} + (\tilde{\mathbf{y}}_0^{mee} - \tilde{\mathbf{y}}_i^{mee})$. Convert \mathbf{y}_{i+1}^{mee} to \mathbf{y}_{i+1} and return to step 8.

Table 5.1: Results overview for 15 test cases, obtained from (Andersen, 1994) and celestrack.org with the initial mean elements shown. The column with the Σ notation is the summation of the two preceding column entries, the mean argument of perigee and mean mean anomaly. Column I contains the results from Andersen's method, column II contains the results of the corrected Andersen's method, and column III shows the results of the Modified Walter's Method.

Case	Satellite	\bar{i}	$\bar{\Omega}$	\bar{e}	$\bar{\omega}$	\bar{M}	Σ	\bar{n}_0	No. of iterations		
									I	II	III
1		73	116	0.00867	53	111	163	16	8	5	4
1b		73	116	0.20658	161	2	163	11	8	4	4
2		47	230	0.73180	47	10	58	2	4	3	4
2b		47	230	0.57356	29	30	59	5	4	4	4
3	TDRS4	0	252	0.00008	191	311	502	1	114/DNC	DNC	10
4	CRRES	18	246	0.71975	83	347	429	2	4	4	4
5	SPOT1	99	104	0.00006	84	276	360	14	175	62	4
6	Mir	52	171	0.00044	243	117	360	16	DNC	DNC	4
7	Kvant-1	52	176	0.00043	248	112	360	16	DNC	17	4
8	EUVE	28	276	0.00105	240	120	360	15	DNC	DNC	3
9	ProgressM17	52	153	0.00630	192	168	360	16	8	7	4
10	HST (Hubble)	28	77	0.00050	183	177	360	15	DNC	52	4
11	1994004A	67	82	0.00341	164	196	360	16	15/ N/A	N/A	N/A

The flow chart is shown in Figure 5.4. The modified Andersen's algorithm is tested with the test cases shown in Table 5.1. It was observed that all test cases converge within 10 iterations. A correct working algorithm was thus developed that is implemented in the SGP4-LT algorithm. The sensitivity and validation of this method are discussed in Section 7.3.

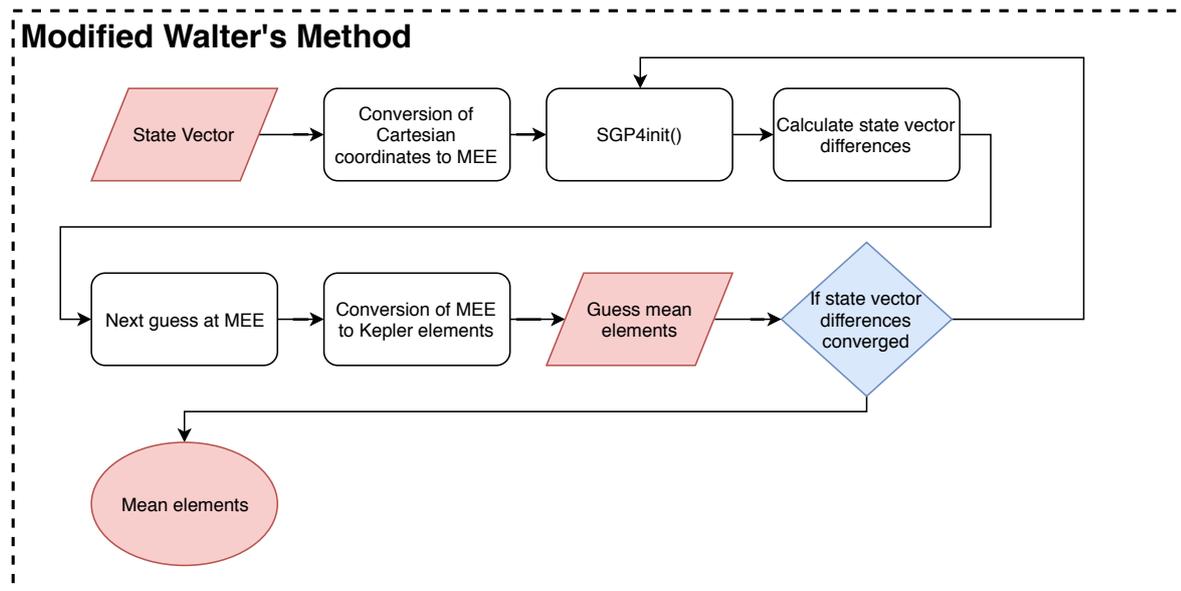


Figure 5.4: Structural overview the Modified Walter's Method.

5.4. SGP4-LT

The complete developed tool, SGP4-LT, uses an iterative approach and is based on SGP4 and implemented Edelbaum's solution for the addition of a low-thrust force. The structured overview for this complete tool is shown in Figure 5.5. The low-thrust module is added to the SGP4() function and is therefore not visible in this structure but was shown in Figure 5.3. The input is TLE data with its mean elements and these are fed to the SGP4init() and SGP4() functions and a state vector with Cartesian elements in the TEME coordinate system is output. Then, it is decided whether the end time is reached. This depends on the current conditions compared to the final orbit: the difference between the current semi-major axis and target semi-major axis should not exceed a certain margin, and at the same time this holds for the inclination. If the margin is not reached yet, the Modified Walter's Method is used to obtain the mean elements from the Cartesian vector output and the pre-set time step is added to the current epoch. This repeats until the final conditions are reached.

A main handle file is written which is used by the user and requires the following inputs:

- `satrec.lowthrust` Low-thrust switch: "1" means the low-thrust engines are turned, when "0" is typed, the low-thrust is turned off and the normal, but still iterative, SGP4 routine is enabled.
- `satrec.facc`: The low-thrust acceleration in km/s^2 .
- `satrec.transfer`: The operation mode for transfer trajectories ("1") or maintaining of altitude ("0"). (This is further discussed in Chapter 8).
- `satrec.afinal` The semi-major axis for the target orbit in km.
- `satrec.ifinal` The inclination for the final orbit in degrees.
- The start -and stop times in minutes: `startmfe` and `stopmfe`, respectively.
- The stepsize `satrec.deltamin` in minutes.
- The input TLE file.
- The output file name.

Furthermore, there are some SGP4-specific options shown in this file, where the standard options are enabled: the use of the WGS72 geodetic system, and the standard 'afspc' for SGP4. It is not recommended to change the standard type. The geodetic system may be changed if required.

Next, the TLE data is loaded and it is checked if the data is valid for SGP4-LT: the values of inclination and

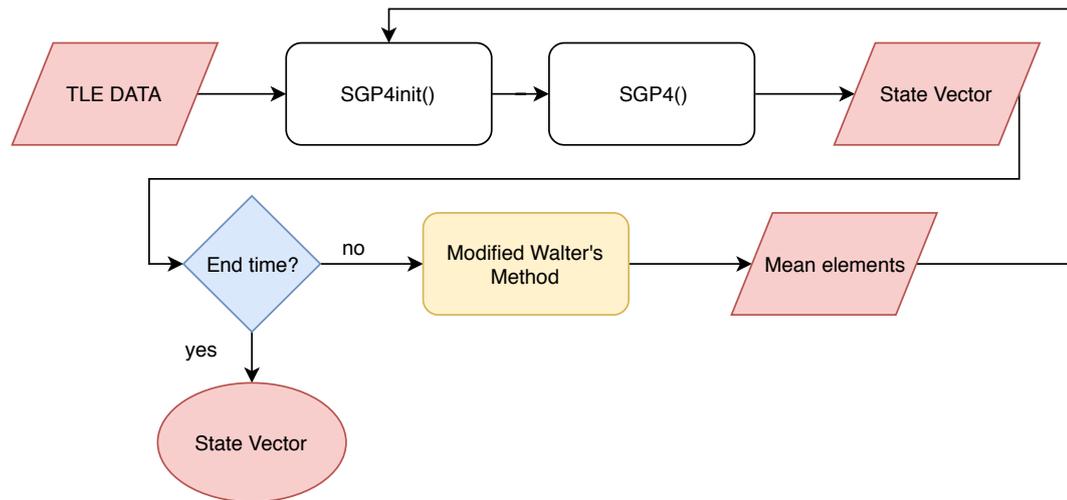


Figure 5.5: Complete structural overview of propagated SGP4-LT, with the low-thrust module inside SGP4().

eccentricity cannot be too small, and the value for eccentricity cannot be too large. In case of too small values for inclination and eccentricity, a set minimum value for inclination and eccentricity is used, for which it was proven SGP4-LT works correctly (See Section 7.3), For initial and final inclination these values are 0.1 degrees, the eccentricity has a minimum value of $4 \cdot 10^{-6}$. These values are still very small and thus these values can be used instead. For a too large eccentricity however, this is not the case. The orbit is then not circular enough to be used in Edelbaum's method. The velocity vector is not perpendicular to the radius vector and SGP4-LT is currently aborted when a TLE is inserted with an eccentricity larger than 0.001, in case the low-thrust propulsion is switched on.

At the start of the program, the given inputs and the TLE data are shown in the terminal. The total flight time and the day when SGP4 enters deep space are shown. The output vectors in Cartesian coordinates are stored in the output file with a name specified by the user. This data file can then be used for further analysis.

5.4.1. Delta V

By definition of Edelbaum's method, the thrust is constant and turned on during the complete transfer. The required ΔV in km/s is thus simply:

$$\Delta V = f_{acc} t_f \quad (5.41)$$

where f_{acc} equals the thrust acceleration in km/s^2 and t_f the total flight time calculated by SGP4-LT in seconds.

5.4.2. Limitations of SGP4-LT

In the above sections limitations of the SGP4-LT algorithm were mentioned and are summarized here.

The limitations partly depend on the analytical low-thrust method chosen. Edelbaum's solution requires the eccentricity of the initial and final orbit to be equal to, or near, zero. Furthermore there is only out-of-plane thrust and this thrust is constant during the whole transfer, which includes near the anti-nodes where thrusting for inclination change is highly inefficient.

The initial and final inclination cannot be equal to zero and must have a value larger than 0.1 degrees. The accuracy of SGP4-LT greatly depends on the accuracy of SGP4: no matter which analytical low-thrust design method is used, the error of SGP4 remains the same which is about a km per day compared to actual satellite data (Vallado et al., 2006).

5.5. Conclusions

In this chapter all aspects of the developed SGP4-LT tool are discussed: the SGP4 tool, the analytical low-thrust solution by Edelbaum and the developed Modified Walter's Method to obtain an iterative version of SGP4. This resulted in a tool which is capable to analytically obtain a low-thrust transfer trajectory including perturbations which are modelled in SGP4.

With the development of SGP4-LT now completed, the tool is to be verified. This process is discussed in the following chapter.

6

Numerical Methods

"All right," said Deep Thought. "The Answer to the Great Question..." "Yes..!" "Of Life, the Universe and Everything..." said Deep Thought. "Yes...!" "Is..." said Deep Thought, and paused. "Yes...!" "Is..." "Yes...!!!" "Forty-two," said Deep Thought, with infinite majesty and calm."

—Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

In many spaceflight problems the state of the spacecraft is propagated in time using numerical methods. The developed tool of this thesis research uses purely 'elegant' analytical methods and does not require such 'brute force' methods. However, a numerical propagator, or numerical integrator, is required to verify Edelbaum's method and every orbital element can be checked, which will be carried out in Chapter 7. Various numerical integrators are briefly discussed in Section 6.1 and the best option for Edelbaum's solution is chosen. Initially, a root-finding algorithm was demanded to find the root of the equation to recover the Kozai mean motion (Appendix C) and thus various root-finding algorithms are discussed in Section 6.2.

6.1. Integrators

Numerical integration is used to numerically approximate the solution of differential equations. Using numerical integration can be beneficial if many points need to be evaluated, but the process could be time consuming. There has to be a trade-off in time/accuracy when the objective function is exactly known.

The methods discussed in this section are all single-step methods. This means these methods do not take previous states into account: such a method only evaluates the current state to obtain the next one. E.g. state X_i is evaluated to obtain X_{i+1} and for the next step it will not use its previous calculated results, or in other words, each time step will be treated as an initial-value problem.

To select the best method, one would have to take into account the truncation error, the speed of the method, the ease of changing the step size, the stability, and the total error that can occur.

Throughout this chapter, the following notation is used for the state derivative $\dot{\mathbf{x}}$.

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \quad (6.1)$$

6.1.1. Euler Method

The Euler method is the simplest methods and a first-order Runge-Kutta method. It is based on a first-order Taylor expansion and the problem can be written as (Noomen, 2016):

$$\mathbf{x}_{i+1} = \mathbf{x}(t_{i+1}) \approx \mathbf{x}_i + h\Phi = \boldsymbol{\eta}(t_{i+1}) \quad (6.2)$$

where

$$\Phi = \dot{\mathbf{x}}_i = \mathbf{f}(t_i, \mathbf{x}_i) \quad (6.3)$$

Looking at Equation 6.2, it can thus be seen that \mathbf{x}_{i+1} , which is \mathbf{x} at t_{i+1} , is approximated by the state at t_i plus the differentiated state at t_i times the step size h , which is $t_{i+1} - t_i$. Or in other words, a time step is taken

along the tangent of \mathbf{x} at \mathbf{x}_i .

As a state vector, the Euler method can be written as:

$$\mathbf{x}(t_{i+1}) = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{pmatrix} + h \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ \ddot{x}_i \\ \ddot{y}_i \\ \ddot{z}_i \end{pmatrix} \quad (6.4)$$

The Euler method is a good starting point to understand numerical integration, but its accuracy is not good enough in order to be used in complex problems.

6.1.2. Higher-order Runge-Kutta Methods

The Euler method is using the slope of \mathbf{x} in \mathbf{x}_i only. The Midpoint method uses this approach but combines it with the slope at $\mathbf{x}(t_0 + 0.5h)$. It is also known as the second-order Runge-Kutta method and is expressed as (Press, 2007):

$$k_1 = h\mathbf{f}(t_i, \mathbf{x}_i) \quad (6.5)$$

$$k_2 = h\mathbf{f}\left(t_i + \frac{1}{2}h, \mathbf{x}_i + \frac{1}{2}k_1\right) \quad (6.6)$$

$$\mathbf{x}(t_{i+1}) \approx \mathbf{x}_i + k_2 \quad (6.7)$$

The fourth-order Runge-Kutta method (RK4) is widely used in engineering. It is fast and gives results with good accuracy. It uses the slopes at four different points, as can be seen in Figure 6.1 and thus requires four evaluations of the function f per step h , resulting in the possibility of taking larger integration steps leading to better efficiency. Solving a differential equation using RK4 is done by using Equations 6.8-6.13 (Press, 2007):

$$\mathbf{x}(t_0 + h) \approx \mathbf{x}_0 + h\Phi_{RK4} = \eta(t_0 + h) \quad (6.8)$$

$$\Phi_{RK4} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6.9)$$

with:

$$k_1 = \mathbf{f}(t_0, \mathbf{x}_0) \quad (6.10)$$

$$k_2 = \mathbf{f}(t_0 + h/2, \mathbf{x}_0 + hk_1/2) \quad (6.11)$$

$$k_3 = \mathbf{f}(t_0 + h/2, \mathbf{x}_0 + hk_2/2) \quad (6.12)$$

$$k_4 = \mathbf{f}(t_0 + h, \mathbf{x}_0 + hk_3) \quad (6.13)$$

In addition to the above described RK4 method there are more Runge-Kutta methods available with a higher order of convergence and better efficiency for better applications. In general, each Runge-Kutta method can be expressed by a so-called Butcher tableau:

Table 6.1: Butcher tableau for Runge-Kutta methods. (Butcher, 2008).

c_1	a_{11}	a_{12}	\cdots	a_{1s}
c_2	a_{21}	a_{22}	\cdots	a_{2s}
\vdots	\vdots	\vdots	\ddots	\vdots
c_s	a_{s1}	a_{s2}	\cdots	a_{ss}
	b_{11}	b_{12}	\cdots	b_{1s}

Then, the solutions are expressed as (Butcher, 2008)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h \sum_{i=1}^s b_i k_i \quad (6.14)$$

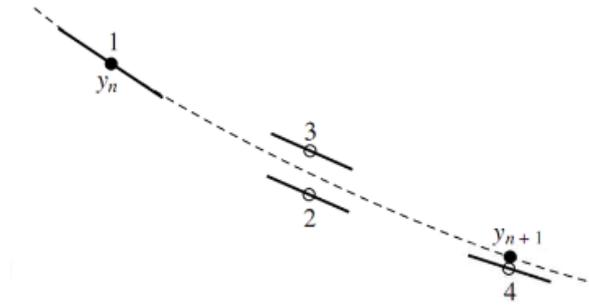


Figure 6.1: Graphical Representation of Runge-Kutta 4 (Press, 2007).

$$k_i = f \left(x_n + h \sum_{j=1}^{i-1} a_{ij} k_j, t_n + c_i h \right) \quad (6.15)$$

Adaptive step size control can be beneficial when there are multiple regions where the function behaves differently. Larger steps can be taken when the function behaves as expected and is smooth, and smaller steps can be taken where more differences in a smaller time frame occur. For example, in an orbit propagation problem with a large eccentricity, adaptive step size control is beneficial. At the perigee of the orbit, one would take small time steps, while at apogee, larger time steps could be taken, since the movement of the spacecraft is much lower with respect to the movement at perigee. This results in increased accuracy and computation speed.

According to (Montenbruck and Gill, 2001), Runge-Kutta-Fehlberg7(8) (RK78), which is a seventh-order method with adaptive step size, is a recommended method for a large range of applications. Since the integrator is used to verify the analytical method a high accuracy is required. Therefore it is decided to use the RK78 method for verification.

6.2. Root-finding Algorithms

In Chapter 5 the methodology of the SGP4-LT algorithm was discussed and it was initially thought that, in order to continue with the Edelbaum Solution, an equation had to be solved. In general, such a problem resembles Equation 6.16:

$$h(x) = g(x) \quad (6.16)$$

which is more convenient to solve when transformed to:

$$f(x) \equiv h(x) - g(x) = 0 \quad (6.17)$$

Various numerical methods are developed for equations that cannot be solved analytically and a selection of these are briefly discussed in this section. All these methods are iterative, c.q. a first guess of the solution is made and is followed by a new (intelligent) guess based on the previous solution and its evaluation. This is repeated as long as the function closeness to zero and solution convergence are not reached.

6.2.1. Bisection Method

The bisection method is one of the simplest root-finding algorithms. It does not use an intelligent guess but always divides the search region in two equal halves, with the exact middle being the next solution. A graphical representation is found in Figure 6.2. The initial range was $[x_{+1}, x_{-1}]$ where function values have opposite signs. The function is evaluated at the exact middle and depending on the sign of the function value, the original section is replaced by the location of the new function value and the end point with opposite sign. This results in the replacement of x_{+1} by x_{x+2} and the next iteration starts by dividing the new range in two, obtaining x_{-3} . The iteration continues until the solution has converged to (near) zero and the set tolerance is met. The bisection method has an order convergence of one.

6.2.2. Newton-Raphson method

The Newton-Raphson method requires only one initial value x_0 , which is the initial guess. It uses the derivative of function $f(x)$ and this is drawn through the guess point. The point where this line crosses the x -axis is

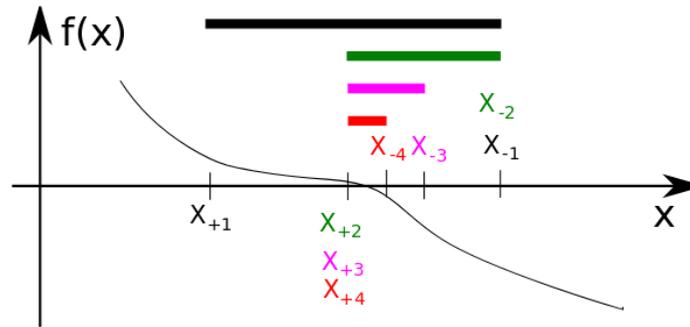


Figure 6.2: Graphical representation of the Bisection method.

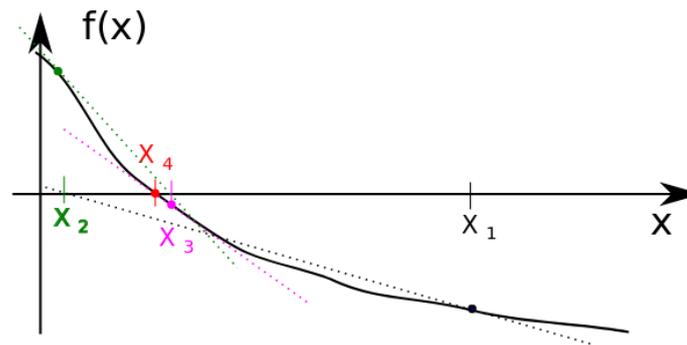


Figure 6.3: Graphical representation of the Newton-Raphson method.

taken as the next guess point. This iterative process continues until the solution converges. The corresponding equation is:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (6.18)$$

A graphical representation is shown in Figure 6.3. The derivative of function f must be known in order to use this method.

6.2.3. Secant Method

The secant method is derived from the Newton-Raphson method but does not require a known derivative. Instead, the derivative is replaced by the finite difference approximation of the derivative:

$$f'(x_i) = \lim_{h \rightarrow 0} \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (6.19)$$

which results in

$$x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} \quad (6.20)$$

The order convergence equals the golden ratio ($\phi = (1 + \sqrt{5})/2 \approx 1.618$). Convergence of this solution is not guaranteed since the root is not bracketed. If the root is completely unknown beforehand and the two initial guesses are not near the root, this method has the possibility of not converging to the root.

6.2.4. Regula Falsi Method

When using this method, it is assumed that the function is smooth and its derivative does not change too widely and it can then be approximated as a line. As the first guess, this line is defined by the ends of the brackets, where in between the root must lie. The point where this approximated line crosses the x -axis is the new guess point and the appropriate end of the bracket is used as the other new guess point. This 'appropriate' end is the end of the bracket where the function value has an opposite sign compared to the

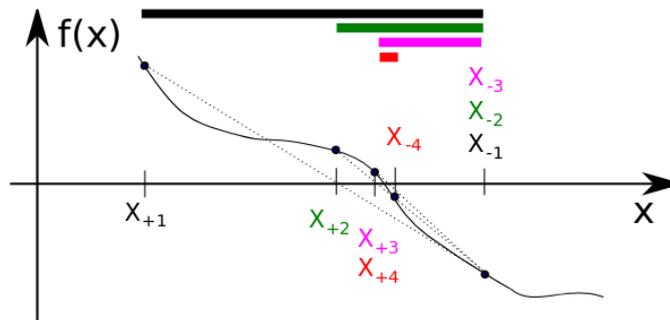


Figure 6.4: Graphical representation of the Regula-Falsi method.

new guess point. The new guess value is determined using:

$$x_0 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)} \quad (6.21)$$

This method will always converge to the root in the set range and has a linear order of convergence. Figure 6.4 shows a graphical representation of the Regula Falsi method.

6.2.5. Comparison and Discussion

In the first SGP4-LT version a root-finding algorithm was required to obtain the semi-major axis in SI units which is an input for the low-thrust module. This semi-major axis is obtained by solving equation C.7. As can be seen in Figure C.1 this function is smooth and simple in the shown range. This range can vary greatly, e.g. EutelSat I-F1 has a n_0 of 0.98 while the ISS has a n_0 of 16. Not knowing where the root may lie in this range leads to the conclusion that the Secant method is not an option.

The derivative of Equation C.7 cannot be obtained analytically and thus the Newton-Rapshon method is not a candidate.

In general, the Regula Falsi method converges faster than the Bisection method and therefore it is decided to use the Regula Falsi method.

7

Verification and Validation

"Remember kids, the only difference between screwing around and science is writing it down."

—Adam Savage (Mythbusters)

To ensure that the developed software tool obtains the correct results, the software tool is checked by verification and validation. The definitions of both are given by (Oberkampf et al., 2002):

- **Verification** The process of determining that a computational model accurately represents the underlying mathematical model and its solution.
- **Validation** The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended use of the model.

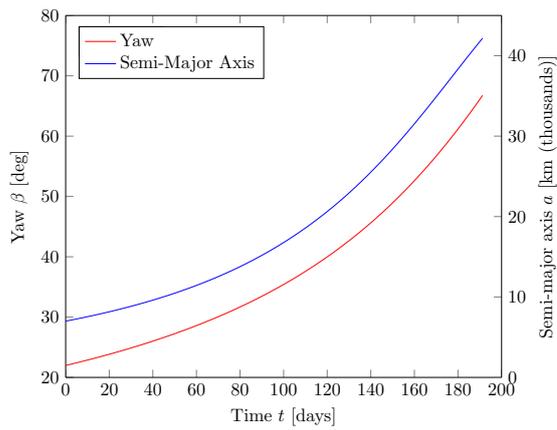
In other words, verification is performed to ensure that there are no bugs present in the code that prevent the models to behave as they are supposed to. When validating, one looks at the result of the model after its verification and compares it to experiments to check whether or not it can be considered realistic.

A bottom-up approach is used when verifying the results of the different subroutines of SGP4-LT, e.g. each subroutine is tested separately using *unit tests*, and then when working in unison. Sanity checks are performed along the way during the creation of the code, preventing major errors.

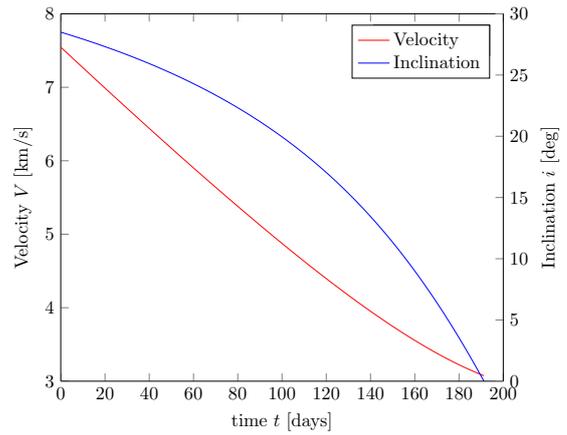
In this chapter, first the verification of all subroutines separately is considered and is followed by a validation of the complete SGP4-LT tool. Edelbaum's solution is verified in Section 7.1, which includes a numerical integration. Next, the Modified Andersen's Method is verified and validated using the entire satellite catalogue obtained from space-track.org. It continues with the verification of the iterative SGP4 model, which requires a correct working Modified Andersen's Method. The complete SGP4-LT model is verified and validated in Sections 7.5 and 7.6. Finally conclusions are drawn in Section 7.7.

7.1. Edelbaum's Solution

The written Edelbaum software code is verified using the results of (Kechichian, 1997) to ensure the written code works as expected. In Figures 7.1 and 7.2 the resulting plots of two different cases are shown. Both cases use a thrust acceleration of $3.5 \cdot 10^{-7} \text{ km/s}^2$. The first case considers a transfer from an initial circular orbit with $a_i = 7000 \text{ km}$ with an inclination of 28.5 degrees to a circular orbit with a radius of 42166 km and an inclination of 0 degrees. Using the implemented method of Edelbaum results in a β_0 of 21.99 degrees and a ΔV of 5.78 km/s. The second case considers the same orbit radii, an initial inclination of 90 degrees and a final inclination of 0 degrees. This leads to a value for β_0 of 10.92 degrees and a total ΔV of 10.13 km/s.

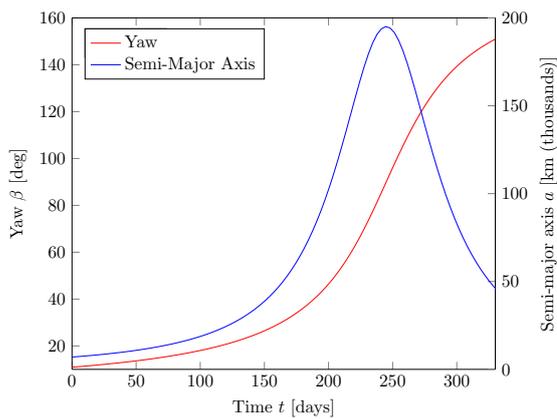


(a) Yaw angle and semi-major axis versus transfer time.

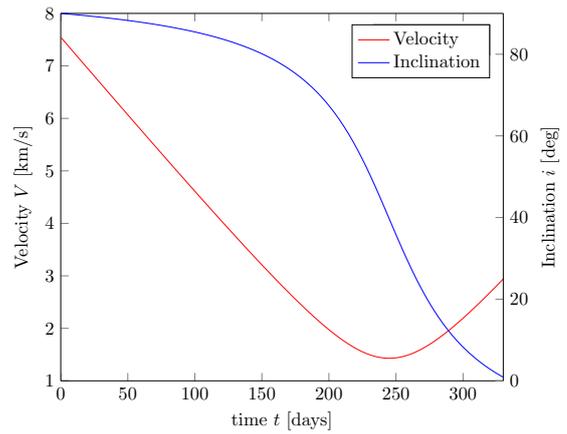


(b) Velocity and inclination angle versus transfer time.

Figure 7.1: Results of Edelbaum's methods for an inclination change of 28.5 degrees.



(a) Yaw angle and semi-major axis versus transfer time.



(b) Velocity and inclination angle versus transfer time.

Figure 7.2: Results of Edelbaum's methods for an inclination change of 90 degrees.

The result of the large inclination change shows a notable change in semi-major axis. As shown in Section 2.8, it is most efficient to change inclination at higher altitudes, where the velocity is lower. Figure 7.2a clearly shows this behavior by obtaining the steepest change in inclination at a high altitude, which is higher than the altitude of the final orbit and thus obtaining a (near-)optimal result.

Next, the software is validated by comparing results of a numerical approach.

7.1.1. Numerical Integration of Edelbaum's Solution

To ensure a correct functional analytical tool, Edelbaum's method is verified using numerical integration. The RK78 procedure as discussed in Chapter 6 is used. The equations of motion used for this case are expressed in Gaussian form of motion in modified equinoctial elements. As discussed in Section 2.6.1 this element set can be used for perturbed orbits. By using modified equinoctial elements, obtaining indeterminate values for Ω and ω when the inclination and eccentricity respectively tend to zero, are avoided.

Edelbaum assumes only thrust with a component in the tangential and out-of-plane direction and thus Equ-

tions 2.37-2.42 are reduced to:

$$\frac{dp}{dt} = \frac{2p}{w} \sqrt{\frac{p}{\mu}} f_{acc} \cos \beta \quad (7.1)$$

$$\frac{df}{dt} = \sqrt{\frac{p}{\mu}} \left[\frac{[(w+1) \cos L + f] f_{acc} \cos \beta}{w} - \frac{g(h \sin L - k \cos L) f_{acc} \sin \beta}{w} \right] \quad (7.2)$$

$$\frac{dg}{dt} = \sqrt{\frac{p}{\mu}} \left[\frac{[(w+1) \cos L + g] f_{acc} \cos \beta}{w} - \frac{f(h \sin L - k \cos L) f_{acc} \sin \beta}{w} \right] \quad (7.3)$$

$$\frac{dh}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 f_{acc} \sin \beta}{2w} \cos L \quad (7.4)$$

$$\frac{dk}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 f_{acc} \sin \beta}{2w} \sin L \quad (7.5)$$

$$\frac{dL}{dt} = \sqrt{\mu p} \left(\frac{w}{p} \right)^2 + \sqrt{\frac{p}{\mu}} \frac{(h \sin L - k \cos L) f_h}{w} \quad (7.6)$$

The state derivative used in the RK78 algorithm is:

$$\dot{\mathbf{x}} = \left[\frac{dp}{dt} \quad \frac{df}{dt} \quad \frac{dg}{dt} \quad \frac{dh}{dt} \quad \frac{dk}{dt} \quad \frac{dL}{dt} \right]^T \quad (7.7)$$

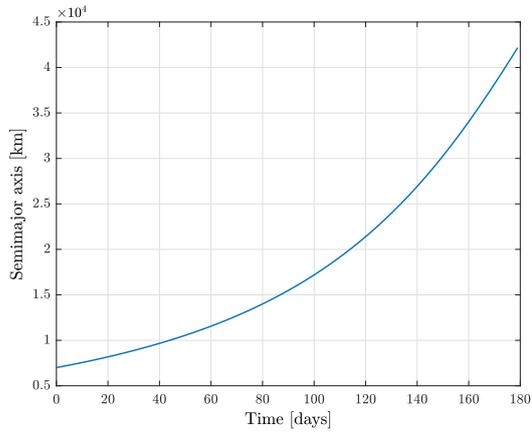
The current β angle is calculated using the analytical equation of Edelbaum's method (Equation 5.19). The sign of the thrust angle β in above equations is based on the current position of the spacecraft in orbit: the sign is flipped at the antinodes by definition (Kechichian, 1997).

Figure 7.3 shows the results of the numerical integration. In Edelbaum's solution, it is assumed that Ω obtains its original value after each revolution and is further not required. The numerical solution however, integrates Ω in time. In this example Ω is set equal to 10 degrees. This way the small change in Ω can be observed, whereas with $\Omega = 0$ the results will jump between 0 and 360 degrees.

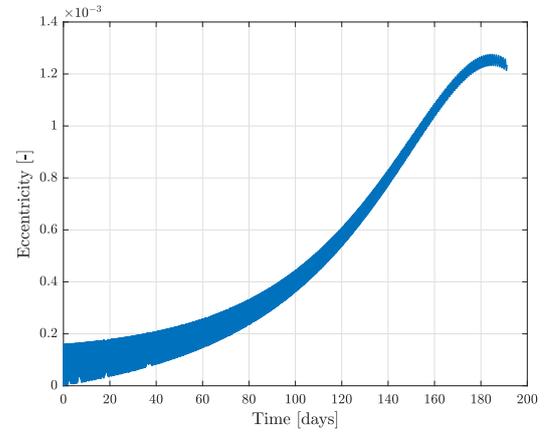
As can be seen in Figures 7.3a and 7.3c, the semi-major axis and inclination obtain their final value in the same transfer time as the analytical solution. Although the eccentricity is assumed zero in the analytical solution, there is a slight variation in eccentricity observable in the numerical calculation (Figure 7.3b). However, with a maximum value of eccentricity of $1.25 \cdot 10^{-3}$ it can be concluded that Edelbaum's assumption of $e = 0$ throughout the transfer is acceptable. When looking at the yaw angle in Figure 7.3d it is clear that the absolute maximum values are the same as of the analytical solution. In the numerical case however, the change of sign at the nodes as defined by Edelbaum is included in order to obtain the correct inclination change. This change in sign is clearly visible in Figure 7.3d. Figure 7.3e shows the change in Ω during the transfer. It can be concluded that these changes are small and thus Edelbaum's assumption is correct. It is noted that at the end of the transfer, the change in Ω becomes slightly larger. This is due to the inclination getting close to zero: at an inclination of zero degrees, the right ascension of the ascending node is poorly defined. At inclinations close to zero, a large change in Ω has little influence on the orbit. This explains the large change in Ω in Figure 7.3e.

It is interesting to investigate the transfer orbit behaviour when increasing the thrust acceleration magnitude. This leads to a better visual understanding of the transfer orbit. To obtain the plots shown in Figure 7.4, the thrust acceleration is increased to $7.0 \cdot 10^{-6} \text{ km/s}^2$. In Figure 7.4a both the numerical and analytical solutions of the inclination versus the transfer time are plotted. It can be clearly seen that the numerical solution has near-horizontal parts. This occurs near the anti-nodes where thrusting for inclination changes is highly ineffective. The analytical solution shows a smooth curve which averages the inclination change with respect to the numerical solution. Figure 7.4b shows the numerical and analytical solutions for the yaw angle versus transfer time. It can be observed that the value the yaw angle changes sign at the nodes, and again the analytical curve matches the positive maximum values of the numerical solution.

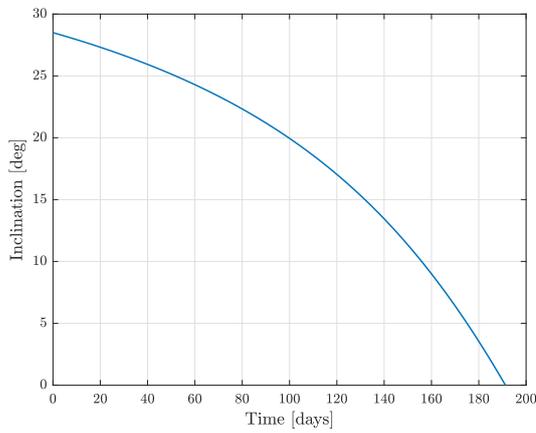
No numerical solution for the Delta-V is available: there is still continuous thrust during the transfer and therefore the Delta-V is only depending on the total transfer time and thrust acceleration. The total ΔV is therefore still analytically calculated.



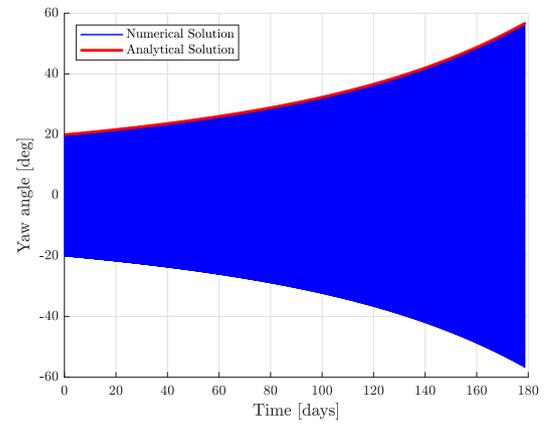
(a) Semi-major axis versus transfer time.



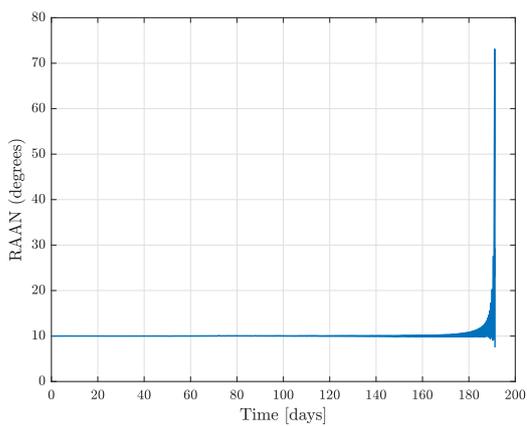
(b) Eccentricity versus transfer time.



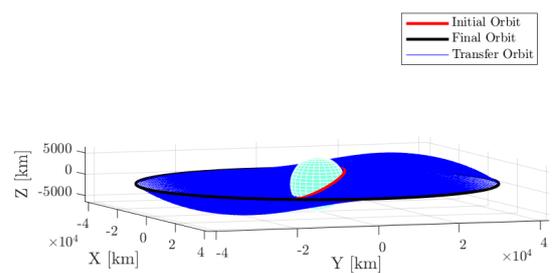
(c) Inclination versus transfer time.



(d) Yaw angle versus transfer time.

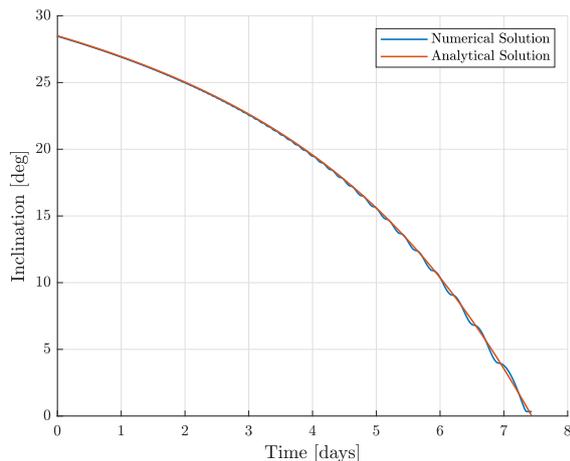
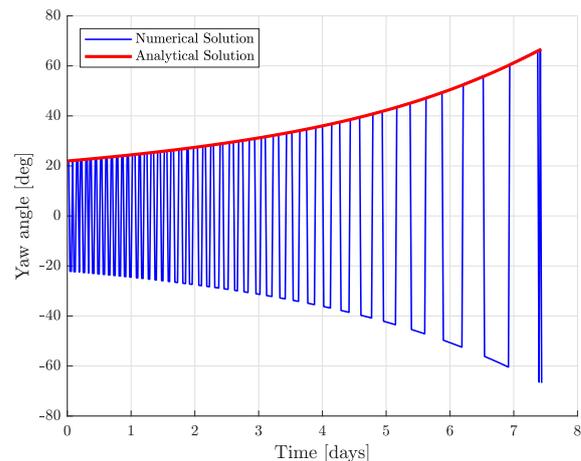


(e) Yaw angle versus transfer time.



(f) Transfer Trajectory

Figure 7.3: Results of the numerical integration of Edelbaum's method for an inclination change of 28.5 degrees.

(a) Inclination versus transfer time with $3.5e-5 \text{ km/s}^2$.

(b) Eccentricity versus transfer time.

Figure 7.4: The numerical and analytical solutions of the inclination and yaw angle, of Edelbaum's method for an inclination change of 28.5 degrees with a thrust acceleration of $7.0 \cdot 10^{-7} \text{ km/s}^2$.

7.2. SGP4

The SGP4 algorithm used in the SGP4-LT tool is obtained from (Vallado et al., 2006) and validated using the validation cases published in this report. It is required to obtain an understanding of the behaviour of the (mean) orbital elements to draw conclusions about the SGP4-LT tool. This behaviour was discussed in Chapter 4.

7.3. Validation of the Modified Walter Method

In Chapter 5 the solution was given to obtain the mean orbital elements from the SGP4 output state vector. The resulting set of elements is inserted as input of the SGP4 algorithm to obtain the position of the spacecraft at the next point in time. It was shown that for each test case convergence was reached. This section discusses the convergence criteria.

It was found that the TDRS-4 case did converge but required more iterations than the other test cases. Next to that, the values found for the mean elements did not exactly match the input mean elements. Compared to the other converging cases, TDRS-4 has an inclination angle of near zero: 0.041 degrees where all other cases have inclinations ranging between 18 and 99 degrees. Other, different test cases have been selected which have inclinations near zero to test the behaviour of the Modified Walter Method. The cases are spacecraft in geostationary orbit, with a mean motion around 1 revolution per day, and 2 spacecraft with a mean motion around 5 revolutions per day. The tolerance is increased to $1 \cdot 10^{-9} \text{ km}$ and $1 \cdot 10^{-9} \text{ km/sec}$ which leads to better results. Table 7.1 shows the results of the Modified Walter's Algorithm applied to the test cases. It can be seen that for inclinations larger than 0.045 degrees the correct result is obtained. The test cases with a smaller value of inclination did not converge, or required a high number of iterations which would still result in an incorrect result. Two cases with an inclination smaller than 0.045 degrees did however converge to the correct result. These two cases, HISPASAT-1D and BSAT-3A had larger eccentricities compared to the other cases and therefore managed to converge to the correct result.

It can be concluded that the convergence depends mainly on the size of the inclination angle. At angles below 0.045 degrees, the rate of convergence becomes slower and the success of convergence starts to depend on the values for eccentricity: the greater the eccentricity, the faster it converges with a correct result.

Table 7.1: Results overview for 12 test cases, obtained from and celestrack.org with the initial mean inclination, eccentricity and motion shown.

Case	Satellite	i [deg]	e [-]	n_0 [rev/day]	No. of iterations	Correct result
1	ABS-6	0.0595	0.0002493	1.00272645	40	yes
2	O3BFM8	0.0516	0.0002598	5.00115107	4	yes
3	Eutelsat-5	0.0487	0.0004933	1.00270865	18	yes
4	O3BFM3	0.0486	0.0001171	5.00115267	4	yes
5	NSS-6	0.0460	0.0002804	1.00270734	26	yes
6	ASTRA-1KR	0.0454	0.0005450	1.00274958	7	yes
7	COSMOS	0.0451	0.0002884	1.00269413	22	yes
8	COMSATBW-2	0.0445	0.0002305	1.00274982	72	no
9	ASTRA-4A	0.0440	0.0002983	1.00272242	DNC	no
10	EHOSTAR-7	0.0434	0.0002152	1.00270551	DNC	no
11	HISPASAT-1D	0.0360	0.0004202	1.00272913	29	yes
12	BSAT-3A	0.0323	0.0005075	1.00273821	9	yes

7.3.1. Implementation of the Satellite Catalogue

To obtain a better understanding of which values for i and e convergence is reached, the complete TLE catalogue from space-track.org is used as input of the software. This catalogue, which consists of all current tracked objects in Earth orbit, contains 17542 objects at October 24, 2018. In order to use the entire satellite catalogue, the designed software is rewritten in a way that it is capable to read multiple TLE's from one input file. The SGP4 algorithm is run at initial epoch only and thus the output elements should be equal to the input mean elements. The maximum number of iterations is limited to 50 since it was found that if a high number of iterations was required, the converged result was often not equal to the input mean elements (see Table 7.1). The output file is changed too. For every TLE, the satellite number, eccentricity, inclination, number of iterations, and a 1 for convergence and a 0 for non-convergence is printed to the output file. This process for the entire satellite catalogue took 248 milliseconds using a 2018 generation laptop.

It was found that out of the 17542 objects, only 96 failed to converge, which converts to a success rate of 99.5%. For each spacecraft, the eccentricity versus the inclination was plotted in a scatter plot. The results of the verification run is shown in Figure 7.5a. The color indicates the number of iterations required to converge.

Figure 7.5a is not conclusive for which values the solution does not converge. It does prove however that there are no cases of non-convergence at high values for inclination and/or eccentricity, although the number of iterations increase at values near 1 for eccentricity. Prognoz-6 has with 0.9252 the highest value for eccentricity in the satellite catalogue and it required 16 iterations to obtain the correct mean elements.

Zooming in on Figure 7.5a near zero for inclination and eccentricity results in Figure 7.5b and it can be clearly seen that at (very) low values for inclination and eccentricity the algorithm does not converge. When using the spacecraft catalogue for geosynchronous satellites from space-track.org, which includes all geostationary satellites with small inclinations, the convergence rate is clearly lower. Out of the 853 cases, 775 converged and 78 failed. This results in a convergence rate of 90.9%.

From these results it was found that the combination of (very) low inclination and eccentricity can result in a non-convergence case. Using an inclination angle lower than 0.05 degrees is not advised when the eccentricity is lower than $6 \cdot 10^{-4}$. If the inclination angle is higher, a lower value of eccentricity is accepted.

Modified equinoctial elements are undefined when $i = \pi$. This singularity is avoided by calculating the modified equinoctial elements directly from the Cartesian coordinates.

To exclude the other orbital elements as cause of non-convergence, they are plotted as well. The mean motion is plotted against the mean anomaly in Figure 7.6. The mean anomaly should not have any effect on

the convergence requirements, but it is simply used here to obtain an even spread of the mean motion for all spacecraft. It can be seen in Figure 7.6a that substantial part of the satellite catalogue is situated in LEO orbits, with the mean motion ranging from 12-16 revolutions per day. Another band is located at a mean motion of two revolutions per day, corresponding to an altitude where a.o. Global Positioning System (GPS) satellites reside. At one revolution per day, at geostationary altitude, the required iterations suddenly increases and convergence is not always reached for these geostationary satellites. A zoomed-in view of this region is shown in Figure 7.6b.

It is however wrong to conclude that the mean motion has an influence on the convergence of the Modified Walter Method. In Figure 7.7, the inclination is plotted versus the mean motion. Subfigure 7.7a shows all spacecraft and for high values of inclination and a mean motion of one revolution per day, no failures are found. Zooming-in for small values of inclination, there are indeed non-convergence cases. It can also be clearly seen that there are no spacecraft in other orbits with very small inclinations than GEO orbits, except 15 spacecraft with a mean motion of 5 revolutions per day, but these converge without issue.

The eccentricity is plotted versus the mean motion in Figure 7.8. A limit for mean motion can be observed. For increasing eccentricity, the mean motion decreases. This can be explained as the limit of eccentricity an orbit can have for a certain semi-major axis. The altitude at perigee is then below zero.

The non-convergence cases are found again at low values for eccentricity, at an mean motion of one revolution per day. It can thus be concluded that the mean motion has no influence on the convergence of the Modified Walter Method. The non-convergence cases only occur for combinations of small values of inclination and eccentricity, which only occur in geostationary orbits.

7.3.2. Conclusions

Due to the use of the modified equinoctial elements, the performance of this Modified Walter's Algorithm is significantly increased compared to the original algorithm published by (Walter, 1967), where it was concluded that convergence is met when:

$$0.1^\circ \leq i \leq 90.0^\circ \quad (7.8)$$

and eccentricity varying between

$$0.01 \leq e \leq 0.30 \quad (7.9)$$

One can ask why non-convergent cases still occur: the modified equinoctial elements were applied to avoid the singularities of inclination and eccentricity.

Inclination and eccentricity are still required by SGP4 and are thus converted from equinoctial elements again at the end of each iteration. At very small values for either inclination or eccentricity, floating point errors occur during this conversion. This results in wrong results for inclination or eccentricity and eventually leading in a non-convergence case. Convergence can still be possible but is then mainly due to luck. It is possible that due to a floating point error, the iteration actually takes a step in the correct direction.

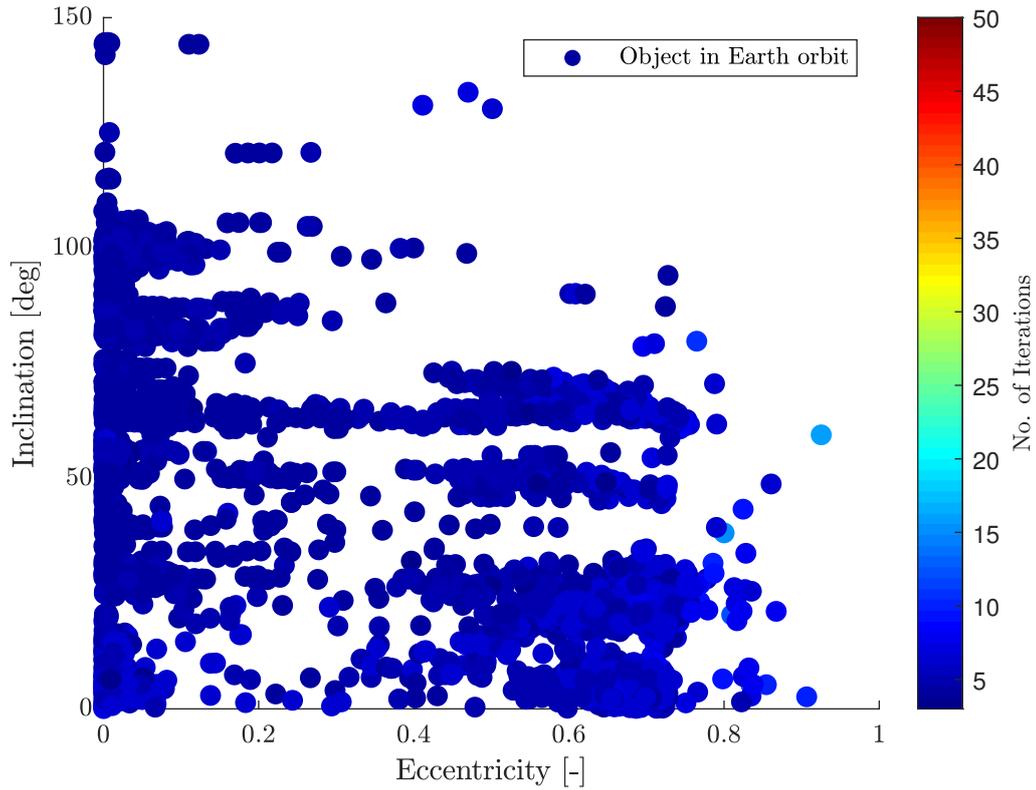
To ensure convergence, the new limits for inclination and eccentricity are set as:

$$0.05^\circ \leq i \leq 180^\circ \quad (7.10)$$

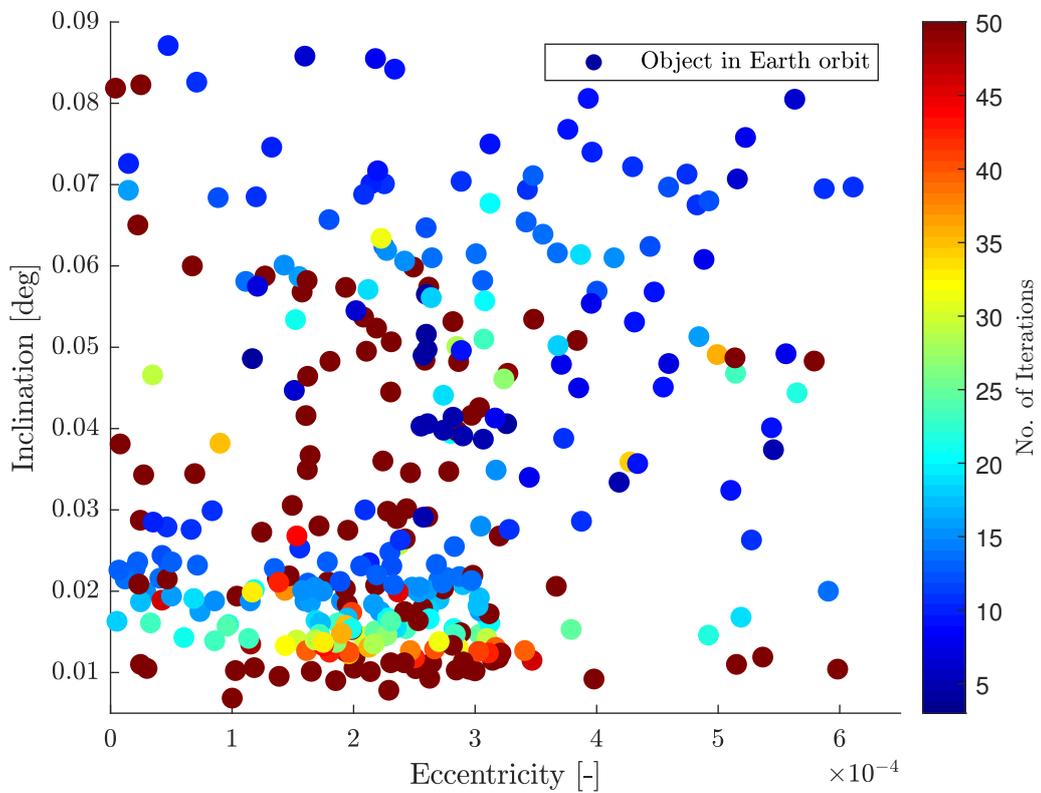
for

$$4e-6 \leq e < 1 \quad (7.11)$$

For lower values of eccentricity, an inclination angle of at least 0.085° is advised.

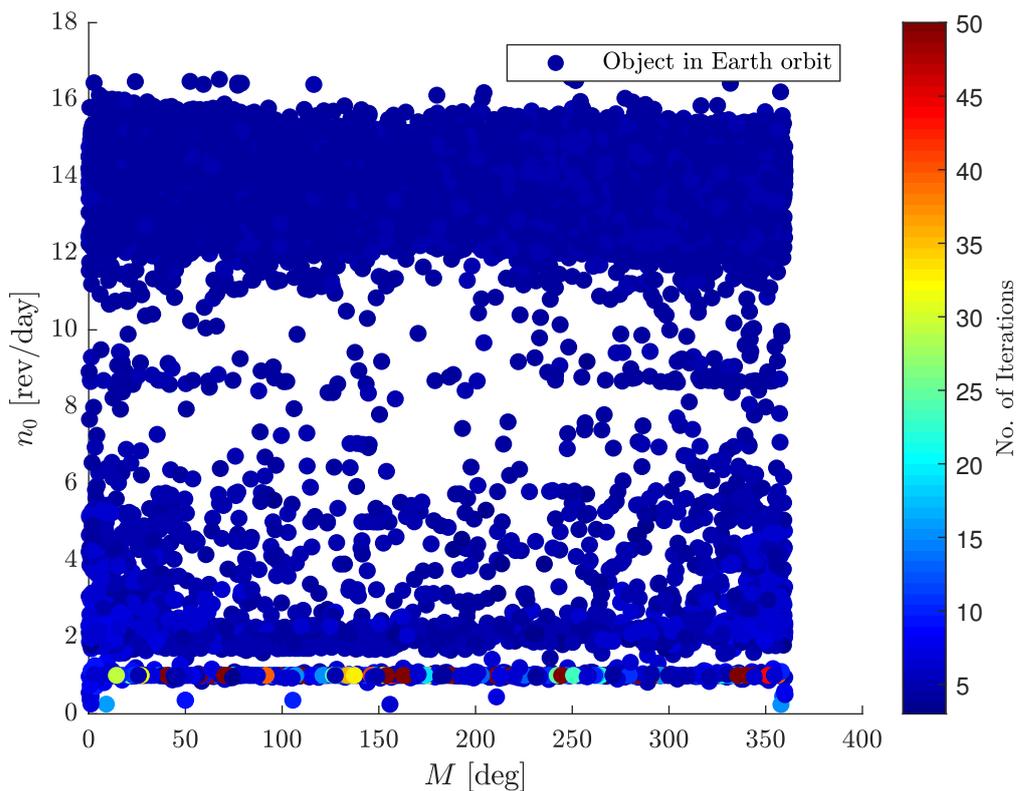


(a) Results of verification run of the entire spacecraft catalog with eccentricity versus inclination.

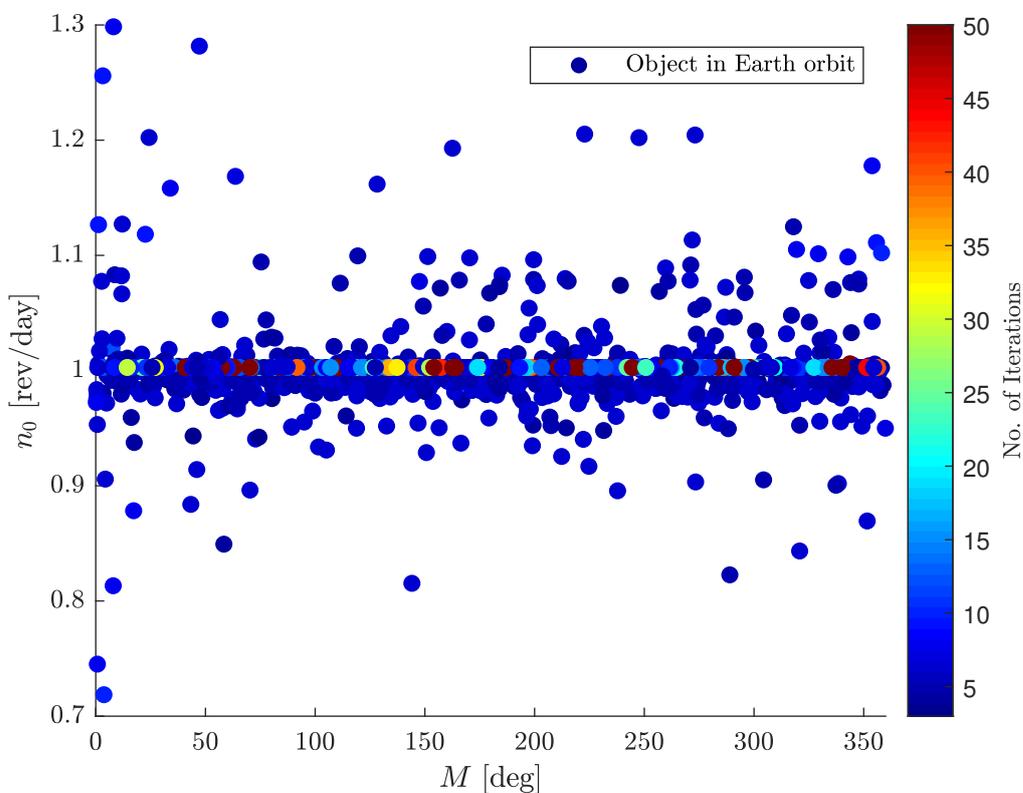


(b) Zoomed-in results of verification run of the spacecraft catalog for small values of inclination and eccentricity.

Figure 7.5: Results of verification run of the spacecraft catalog obtained from space-track.org. The eccentricity is plotted against the inclination for each space object. The color indicates the number of iterations.

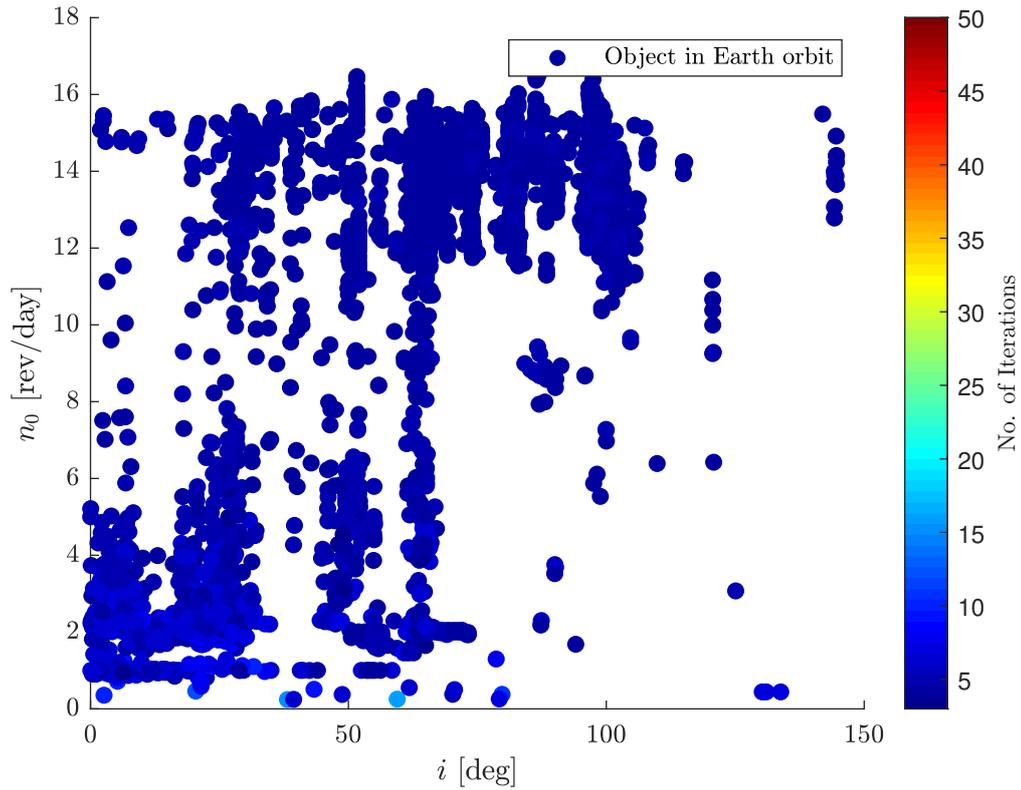


(a) Results of verification run of the entire spacecraft catalog.

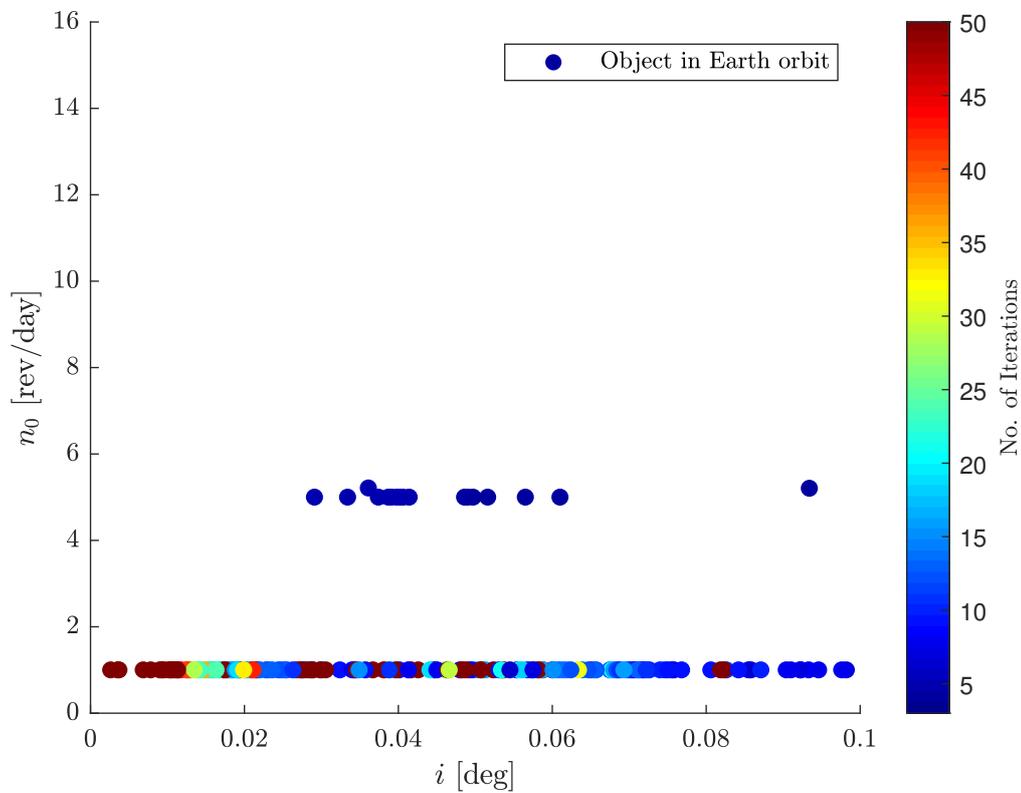


(b) Zoomed-in results of verification run of the spacecraft catalog for small values of inclination and eccentricity.

Figure 7.6: Results of verification run of the spacecraft catalog obtained from space-track.org. The eccentricity is plotted against the inclination for each space object. The color indicates the number of iterations.

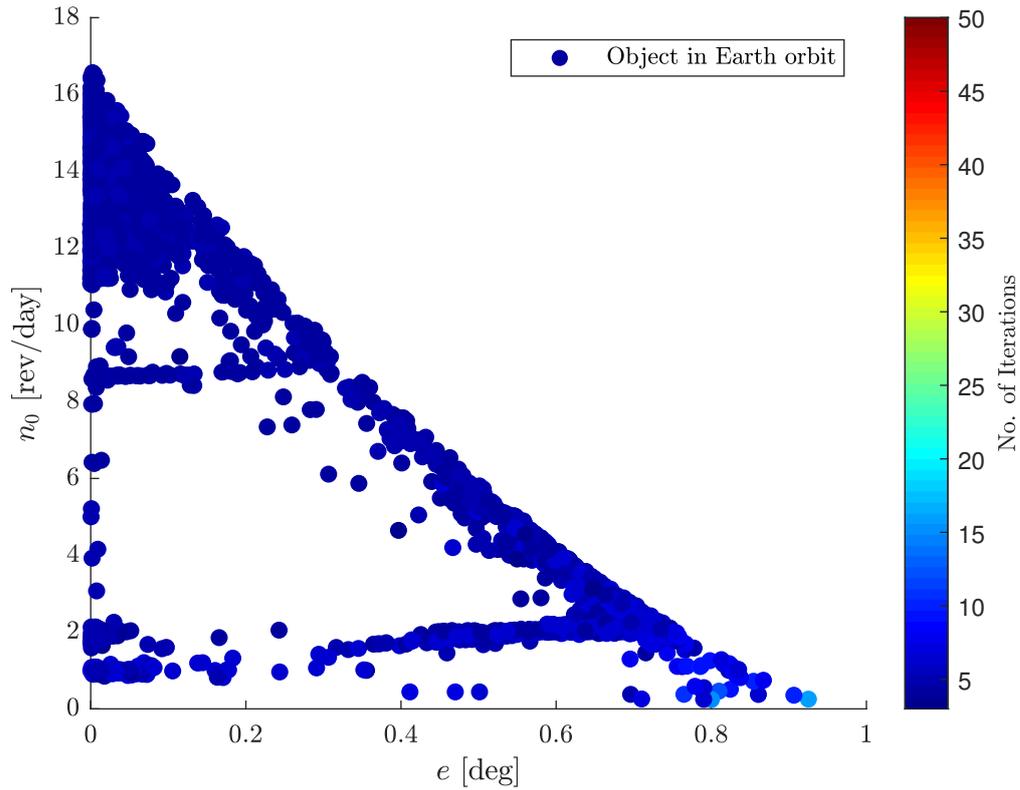


(a) Results of verification run of the entire spacecraft catalog.

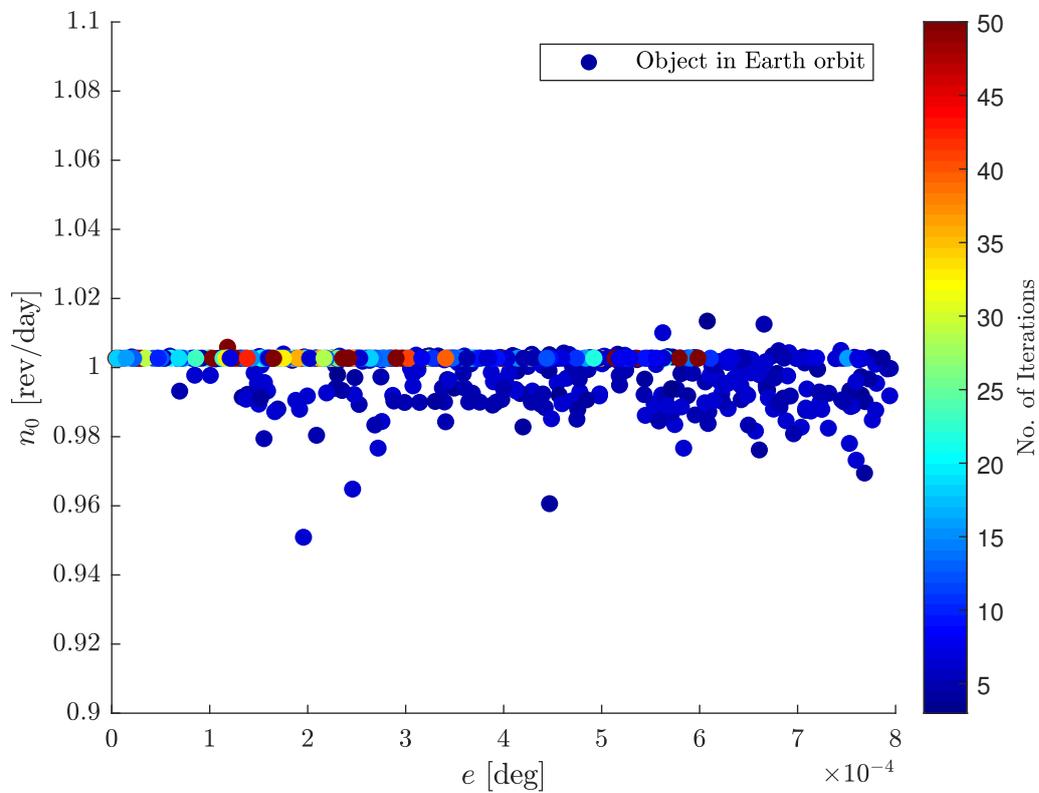


(b) Zoomed-in results of verification run of the spacecraft catalog for small values of inclination and eccentricity.

Figure 7.7: Results of verification run of the spacecraft catalog obtained from space-track.org. The eccentricity is plotted against the inclination for each space object. The color indicates the number of iterations.



(a) Results of verification run of the entire spacecraft catalog.



(b) Zoomed-in results of verification run of the spacecraft catalog for small values of inclination and eccentricity.

Figure 7.8: Results of verification run of the spacecraft catalog obtained from space-track.org. The eccentricity is plotted against the inclination for each space object. The color indicates the number of iterations.

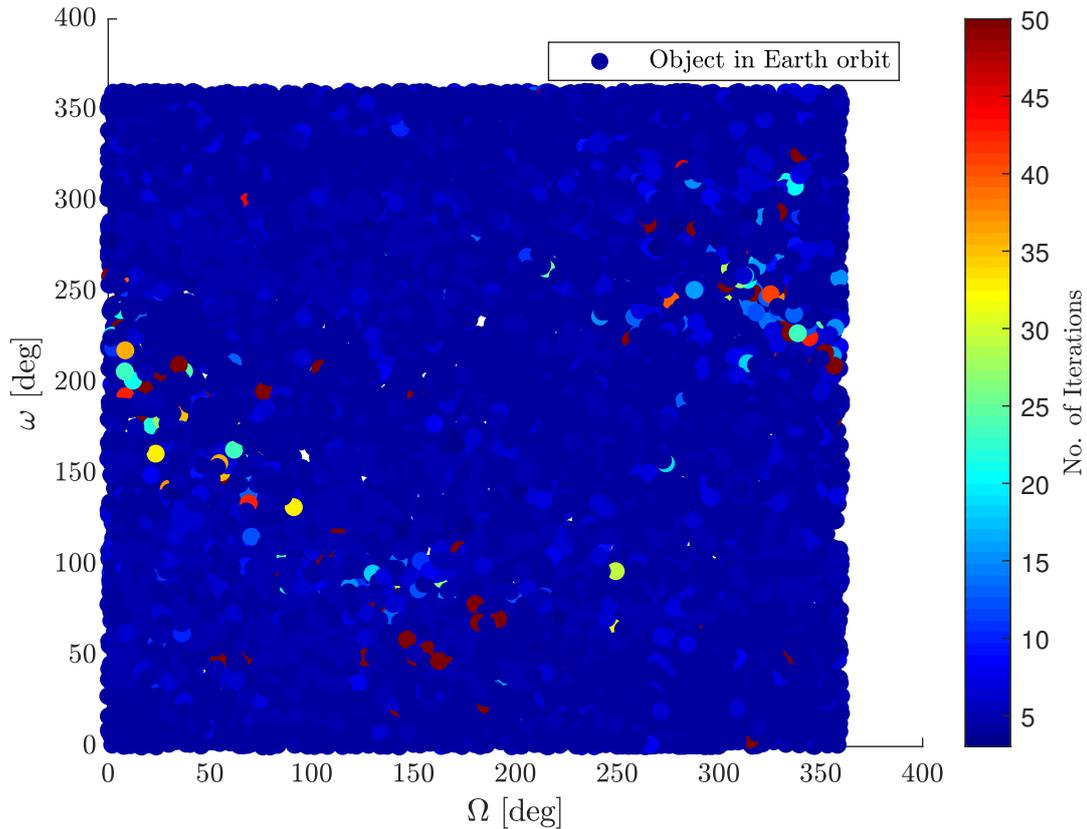


Figure 7.9: Results of verification run of the entire spacecraft catalogue with ω versus Ω .

7.4. Iterative SGP4 behaviour

The Modified Walter Method is used to propagate further in time and the results are compared with the original SGP4 algorithm. The test cases used in Chapter 5 are used again.

In the first example, the spacecraft are propagated for 10 days and the differences between the iterative SGP4 and the original SGP4 are expressed using the Radial, In-track, Cross-track (RIC) coordinate reference system, with the origin in the orbit created by the original SGP4 model. The transformations required to obtain the RIC coordinate system can be found in Appendix A. It is observed that for each case the along-track difference is largest. Furthermore, large differences occur between the different cases which cannot be ascribed to the different treatments in SGP4 for near-Earth or deep-space satellites. Although convergence was reached, the along-track differences can go up to 60 km after 10 days for PROGRESS-M17, where the smallest along-track difference, around 2 meters after 10 days, was found for SPOT-1. The along-track difference is the dominant difference, the radial and cross-track differences are in general a factor 100 smaller.

In the previous section it was concluded that the modified Walter method is correct, and therefore these observed differences do not occur as a result of an error in the conversion to mean elements, but are a result of the behaviour of the original SGP4 algorithm, which has, according to (Hoots et al., 2004b) an error in the along-track direction in the kilometre range after a day, compared to actual (GPS) data. Furthermore, the results obtained by (Kelso, 2007) show similarity between the RIC differences for the original SGP4 versus GPS data and the iterative SGP4 versus the original SGP4. It is therefore decided to compare the results of the iterative SGP4 with actual data. (Kelso, 2007) used ephemerides data of GPS satellites for the comparison. The GPS constellation is continuously monitored by a global network of sensors and its high-precision ephemerides are readily available to the public by the US National Geospatial-Intelligence Agency (NGA). To be able to use this data, which is presented in ECEF WGS-84 coordinate system, a conversion is required to the correct coordinate and reference system, since SGP4 uses TEME WGS-72. This could be done using

Systems Tool Kit (STK), a software-package used to perform analyses of a.o. space applications, or extensive coordinate transformations. This is beyond the scope of this thesis research and therefore not considered further.

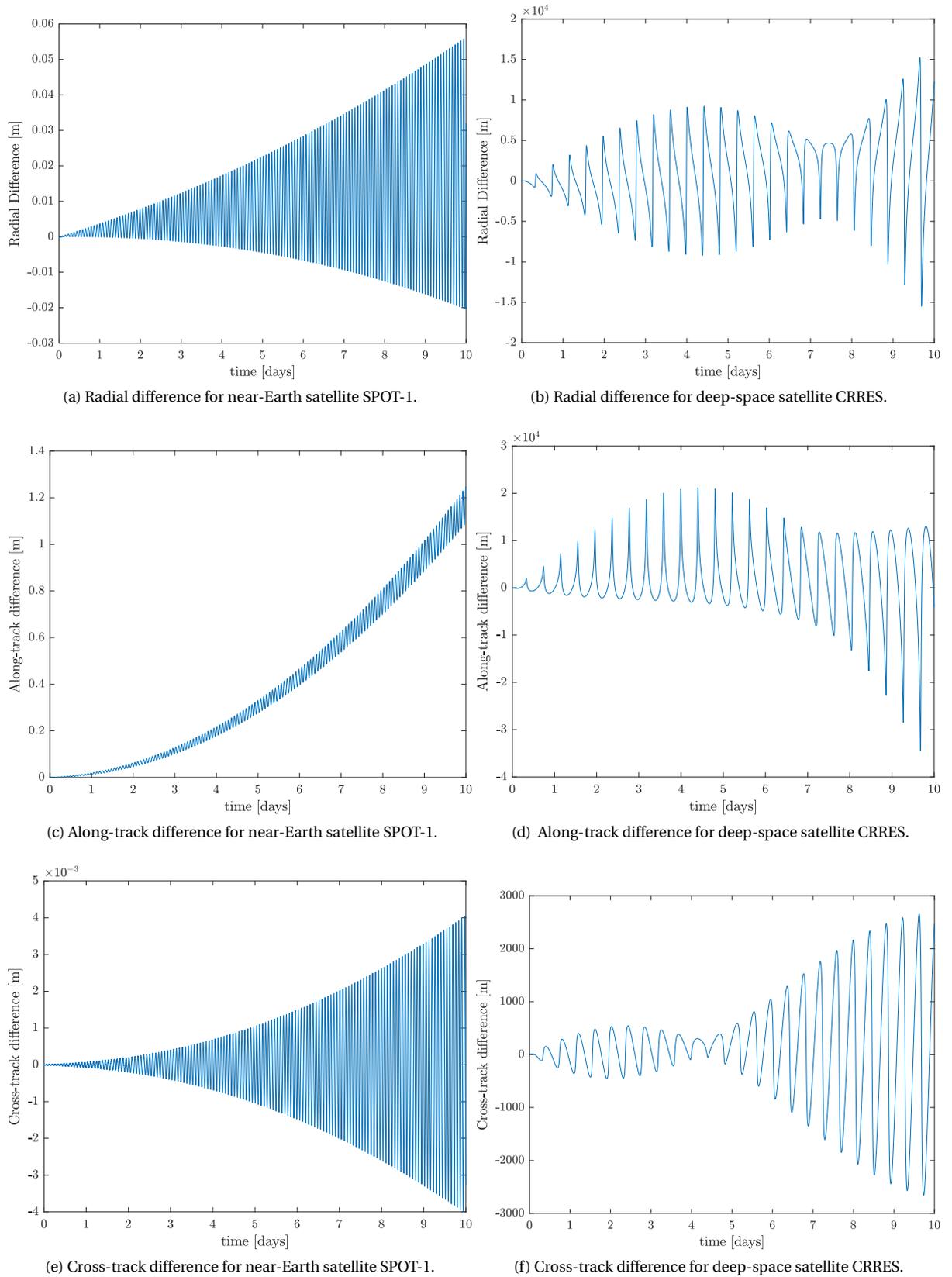


Figure 7.10: Differences between propagation of the iterative SGP4 and the original SGP4 for radial, along-track and cross-track directions, for respectively a near-Earth and a deep-space satellite.

7.5. SGP4-LT: comparison to analytical solution

With all parts of the SGP4-LT tool verified, the complete tool can be verified and tested. In this section, the results of SGP4-LT are compared with the results of the pure Edelbaum solution, which was validated in Section 7.1. The numerical Edelbaum solution obtains the orbital elements in time whereas the analytical solution only obtains the semi-major axis and inclination in time and assumes all other elements constant. Using the numerical solution it was found that this assumption holds up since the changes in eccentricity, argument of perigee and right ascension of ascending node are relatively small. Due to the perturbations added in SGP4-LT it is expected that these orbital elements will change and therefore it is decided to use the numerical Edelbaum solution for comparison.

The two test cases used in Section 7.1 are used again. For both test cases the right ascension of the ascending node is set to 180 degrees and the argument of perigee to zero degrees. This value of 180 degrees is chosen since it was found in Section 7.1 that the Right Ascension of the Ascending Node (RAAN) is not constant. Although these changes are small, it results in a continuously shifting between 0 and 360 degrees. To avoid this shifting as much as possible, a value of 180 degrees was chosen.

The resulting orbital elements and yaw angle versus time for this first case can be found in Figure 7.11. From the plots for semi-major axis and inclination there is hardly any difference visible for the semi-major axis and inclination. The transfer trajectory calculated by the numerical Edelbaum solution requires a flight time of 190.25 days, the transfer by SGP4-LT takes 190.75 days which is a half day difference. The eccentricity calculated by SGP4-LT hardly changes compared to the numerical solution, although both solutions only change on a small scale. This lower change by SGP4-LT is only due to the SGP4 model, since the analytical Edelbaum solution in SGP4-LT always assumes the eccentricity equal to zero and requires no eccentricity value. Thus the eccentricity does change due to perturbations when near Earth, as was shown in Chapter 4 but the low-thrust module assumes zero at every instance. At 67.5 days, when the altitude is high enough to initialize the deep-space module, the eccentricity becomes constant.

The RAAN changes significantly compared by the RAAN obtained by the numerical solution. Due to the perturbations and in particular due to the J_2 effect, this angle decreases rapidly when near Earth but at higher altitudes this effect decreases and the slope flattens out and is near-constant in GEO, with a decrease of 0.0141 degrees per day.

In Figure 7.11e the argument of perigee shows a large increase until around 200 degrees and starts to decrease until around 150 degrees after 70 days. At that point, there is a sudden change from a decreasing trend to an increasing trend before stabilizing at around 190 degrees. This sudden change in gradient is due to transition from near-Earth to deep-space in the SGP4 algorithm. In the deep-space module other perturbations are taken into account, resulting in a different behaviour of the argument of perigee. The argument of perigee is however poorly defined for (near-) circular orbits and thus no conclusions should be drawn.

Finally, the yaw angle of the SGP4-LT changes slightly slower than the numerical solution, following the pattern of the semi-major axis and inclination.

Figure 7.12 shows the orbital elements and yaw angle for the second test case. Again, the two methods hardly show any difference for the semi-major axis and inclination: the transfer trajectory of the SGP4-LT tool requires 280.19 days, the numerical Edelbaum solution requires 280.07 days, which equals a difference of 172.8 minutes. At 66.95 days, the deep-space module is initialized for the first time, which can be clearly seen in plots for eccentricity and argument of perigee, Figures 7.12b and 7.12e, respectively.

The variations of Ω in time for the initial orbit and final orbit are plotted together with the Ω of the transfer orbit. The result can be found in Figure 7.13. As expected, the slope of the transfer Ω at t_0 and t_f is equal to the slope of the RAAN of the initial and final orbit, respectively, showing that the RAAN is not altered by the low-thrust module.

The argument of perigee shows a large change compared to the numerical Edelbaum result and it is clear when the deep space phase is entered: there is a sudden change from a decreasing trend to an increasing trend. One should however realize that the orbit is near-circular and thus the argument of perigee is not well defined.

Using SGP4-LT, the trajectory from LEO to GEO with an inclination change of 28.5 degrees requires a computation time of 5 seconds. For the purely numerically Edelbaum solution, this required already 7 minutes. This is a large difference in computation time but it should be noted that the numerical Edelbaum computation is performed in MATLAB, whereas SGP4-LT is written in C++. Nonetheless, this is a large difference in

calculation time.

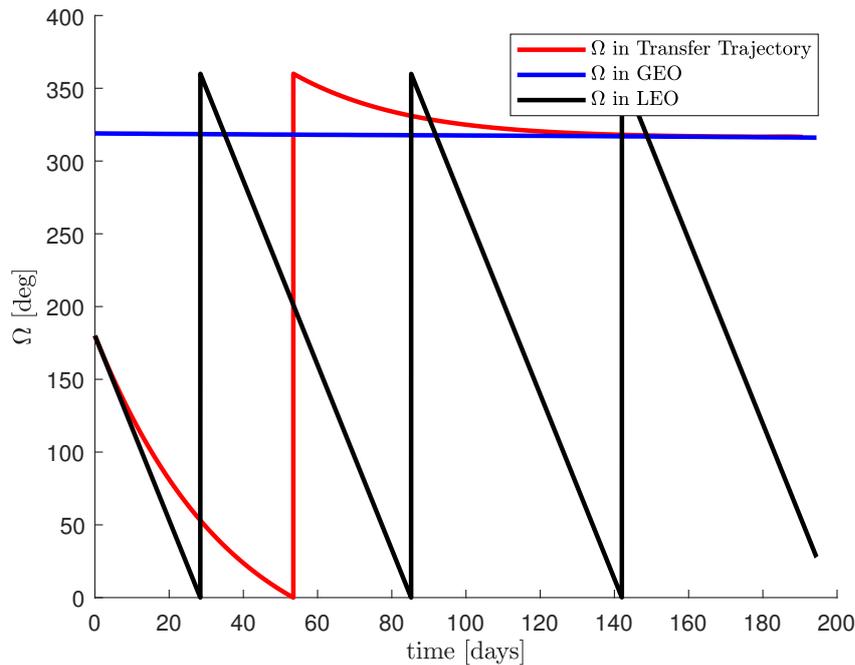


Figure 7.13: Evolution of Ω for the transfer trajectory, from LEO to GEO with an inclination change of 28.5 degrees, and the evolution of Ω in LEO and GEO

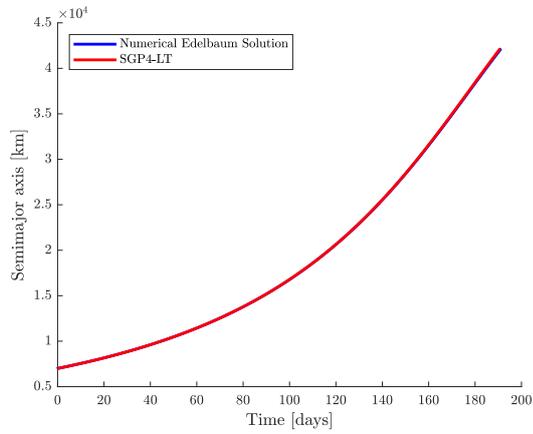
7.6. SGP4-LT: Comparison to numerical results

The SGP4-LT tool is compared with the numerical results obtained by (Herman and Spencer, 2002). In this paper, optimal low-thrust orbits are designed using higher-order collocation methods. Several example cases were analysed where a spacecraft is transferred from LEO to GEO, Medium Earth Orbit (MEO) and Highly Elliptical Orbit (HEO) orbits. This research is a good candidate for result comparison: its initial orbit and final orbits, LEO and GEO, are circular, which is a requirement for the SGP4-LT tool. The HEO orbit is not considered in this comparison since it has an eccentricity of 0.73.

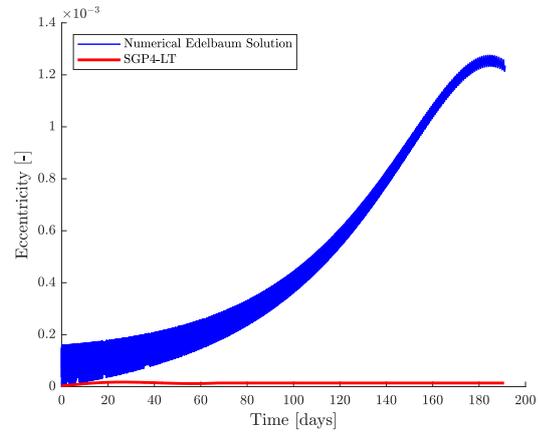
Furthermore, the direction of the thrust vector T is defined by the angles α and β whereas the SGP4-LT tool only uses β and assumes $\alpha = 0$. According to the results in (Herman and Spencer, 2002), the eccentricity during the transfer remains relatively small and the values increase in case of a higher thrust acceleration. Lastly, the SGP4-LT tool uses constant thrust, where the model of (Herman and Spencer, 2002) is capable of multiple burns and coasting in between. Still, it can be concluded that these test cases can be used to validate the SGP4-LT tool, although not the same results in terms of ΔV and time of flight can be expected. Table 7.2 shows the inputs for the orbital elements as used by (Herman and Spencer, 2002). Table 7.3 shows the results obtained by (Herman and Spencer, 2002) and SGP4-LT for different thrust accelerations. The transfer trajectories can be found in Figures 7.14 and 7.15. The percentage differences are calculated by using:

$$PD = \frac{|x_1 - x_2|}{(x_1 + x_2)/2} * 100\% \quad (7.12)$$

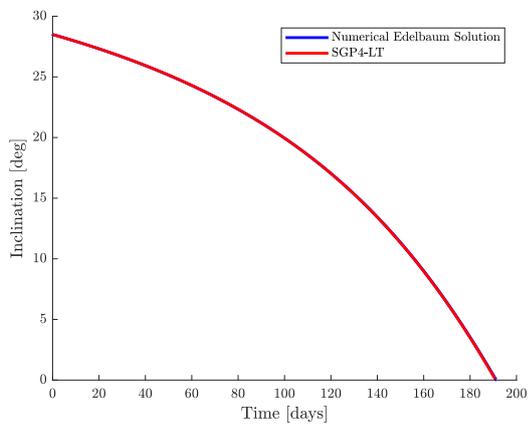
The ΔV and transfer time for the values of $1e-2$ and $1e-3$ km/s^2 are far off compared by the results obtained by (Herman and Spencer, 2002). These values are comparable to the accelerations of a high-thrust spacecraft. Looking at the trajectories with these thrust accelerations in Figures 7.14a - 7.14b and 7.15a - 7.15b, the trajectories have the looks of a high-thrust trajectory as well. The calculated values by SGP4-LT are completely off and a transfer time of 15 minutes from LEO to GEO is unrealistic. For the lower values of acceleration, the transfer trajectory represents a low-thrust trajectory. The compared values have smaller differences. These differences are logical to occur since the results from (Herman and Spencer, 2002) are obtained using an optimized numerical algorithm where thrust is not constant, coasting is allowed and there is no constraint on thrust angle and eccentricity. These constraints are required for Edelbaum's solution and it was shown in the



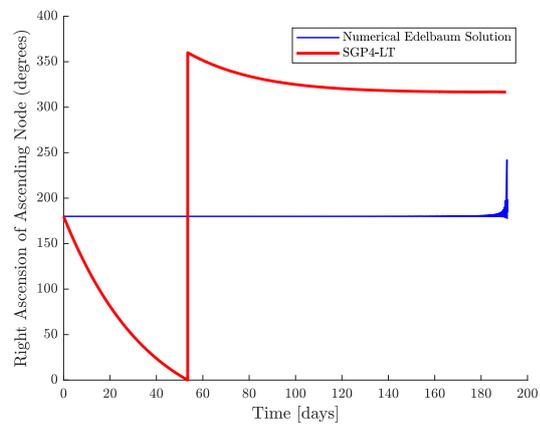
(a) Semi-major axis versus flight time



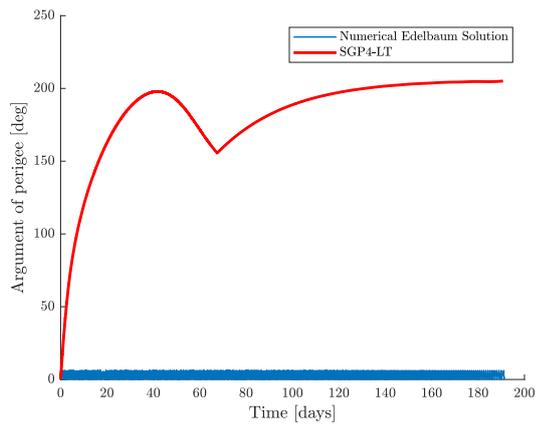
(b) Eccentricity versus flight time



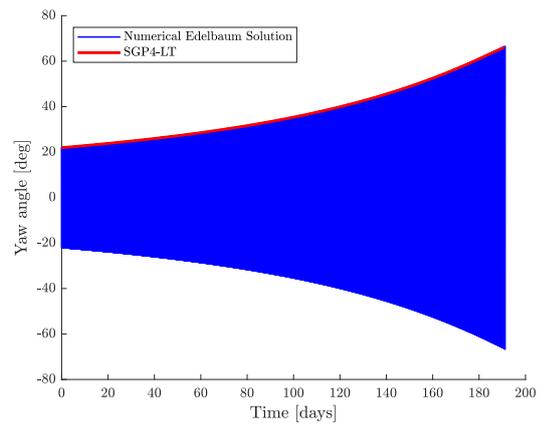
(c) Inclination versus flight time.



(d) Right ascension of the ascending node versus flight time.



(e) Argument of perigee versus flight time



(f) Yaw angle versus time.

Figure 7.11: Orbital elements and yaw angle versus flight time as for a LEO to MEO transfer calculated with SGP4-LT and the numerical Edelbaum solution.

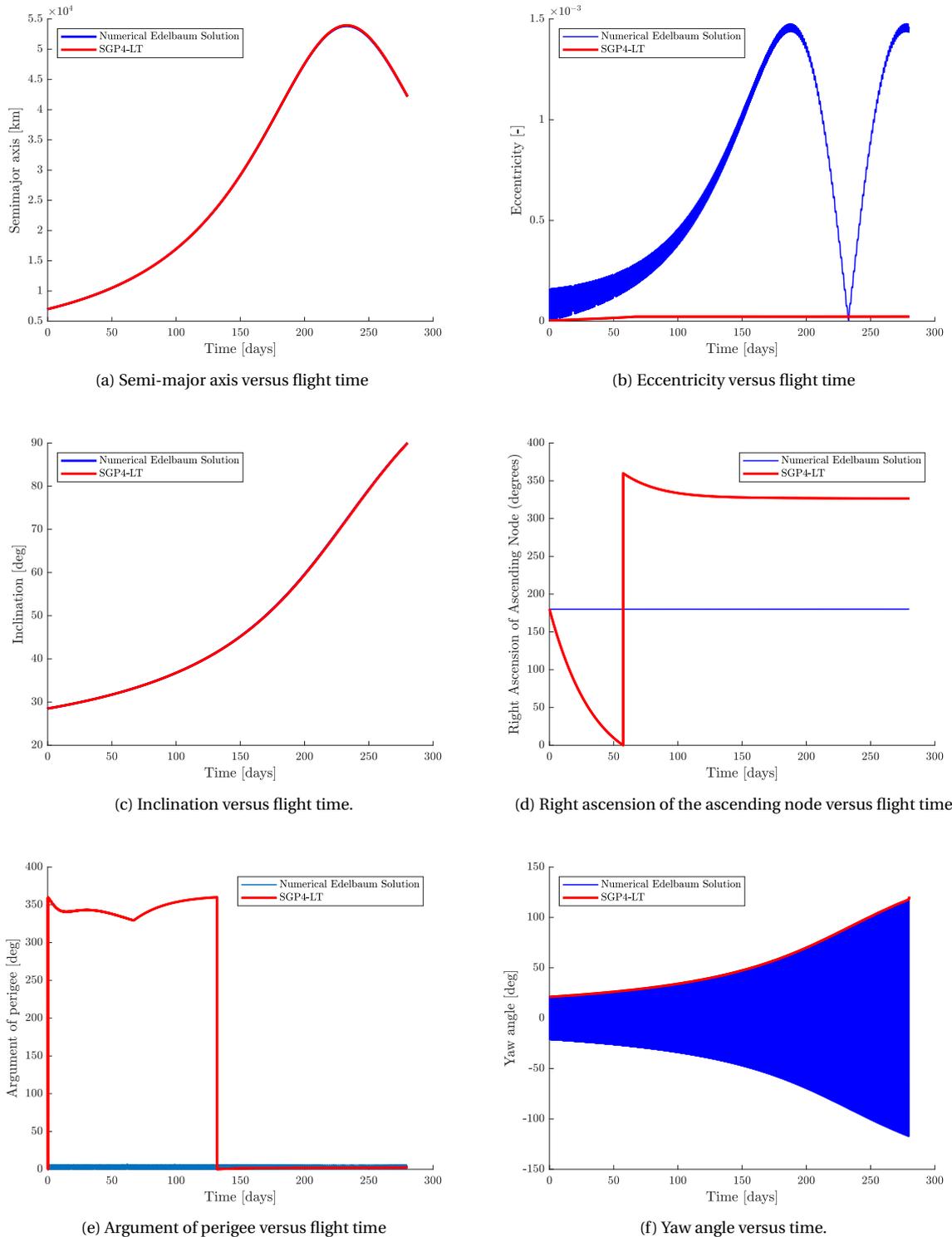


Figure 7.12: Orbital elements and yaw angle versus flight time as for a LEO to MEO transfer calculated with SGP4-LT and the numerical Edelbaum solution.

previous section that SGP4-LT functions as expected compared to solely Edelbaum's solution. It can be concluded that the decreasing differences for lower accelerations show that SGP4-LT obtains fairly good results considering its constraints. The results obtained by SGP4-LT are a direct result of the solution suggested by Edelbaum.

Table 7.2: Kepler elements input used by (Herman and Spencer, 2002).

Orbital Elements	LEO	GEO	MEO
a	7003	42287	26560
e	0	0	0
i	28.5	0	54.7
Ω	0	0	0
ω	0	0	0
M	Free	Free	Free

Table 7.3: Results of (Herman and Spencer, 2002) and SGP4-LT for a transfer trajectory from LEO to GEO.

Thrust Acceleration [km/s ²]	Herman's Results		SGP4-LT			
	Total ΔV [km/s]	Transfer Time [hours]	Total ΔV [km/s]	Transfer Time [hours]	ΔV Difference [%]	Time Difference [%]
1e-2	4.13	5.40	9.00	0.25	74.18	182.30
1e-3	4.31	6.06	6.08	1.69	34.07	112.77
1e-4	5.17	18.32	5.78	16.05	11.14	13.02
1e-5	5.70	149.59	5.77	160.43	1.22	6.99

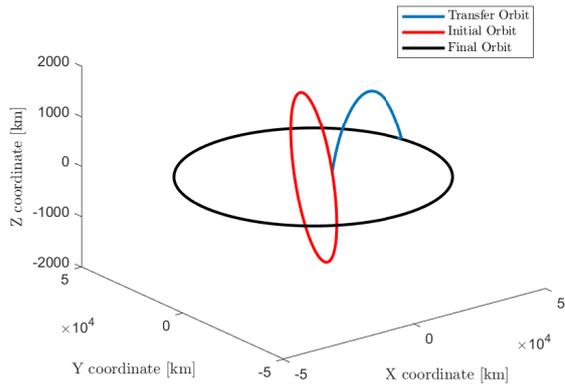
Table 7.4: Results of (Herman and Spencer, 2002) and SGP4-LT for a transfer trajectory from LEO to MEO.

Thrust Acceleration [km/s ²]	Herman's Results		SGP4-LT			
	Total ΔV [km/s]	Transfer Time [hours]	Total ΔV [km/s]	Transfer Time [hours]	ΔV Difference [%]	Time Difference [%]
1e-2	3.86	3.06	15.17	0.42	118.87	151.37
1e-3	3.97	3.60	5.29	1.47	28.51	84.02
1e-4	4.73	14.56	5.28	14.68	10.98	0.82
1e-5	5.12	135.23	5.28	146.78	3.08	8.19

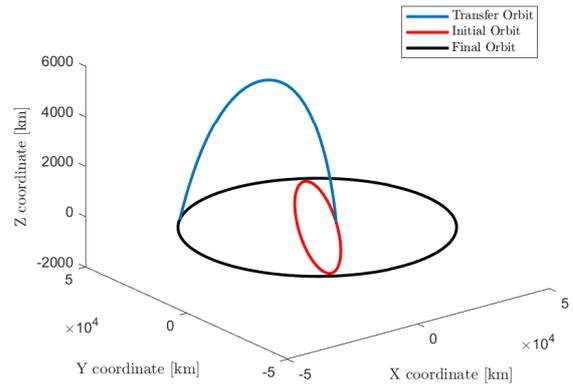
7.7. Conclusions

In this chapter the separate parts of SGP4-LT were tested and verified. It was found that Edelbaum's solution functions as expected and was verified using an RK78 integrator for inclination change and pure orbit raising. The SGP4 algorithm was verified using the cases published in (Vallado et al., 2006) and works correctly. The Modified Walter Method was tested for the entire satellite catalogue which consists of 17452 objects and a convergence rate of 99.5% was reached. For correct results, it is advised to have inclinations between 0.05 and 180 degrees and eccentricities between $4 \cdot 10^{-6}$ and 1.

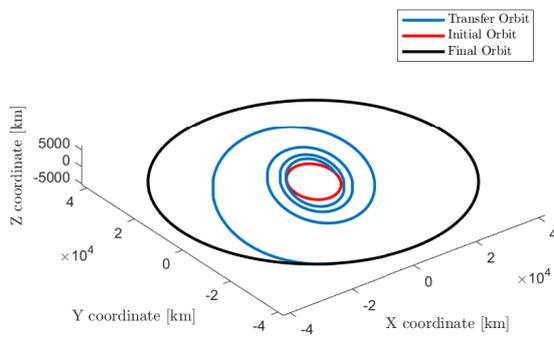
With use of the Modified Walter's Method, the SGP4 algorithm was rewritten into an iterative algorithm and its results were compared in the radial, along-track, and cross-track directions with the original SGP4 algorithm. The obtained differences vary per spacecraft but its differences are comparable and for some cases even better than obtained by (Lee, 2002). These differences which increase by time are due to the nature of the general SGP4 algorithm where its accuracy quickly diverges by a kilometre per day (Hoots and Roehrich, 1988).



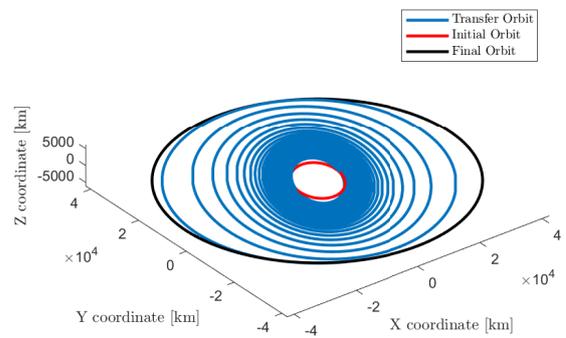
(a) LEO to GEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-2} \text{ km/s}^2$, with z-axis not to scale.



(b) LEO to GEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-3} \text{ km/s}^2$, with z-axis not to scale.

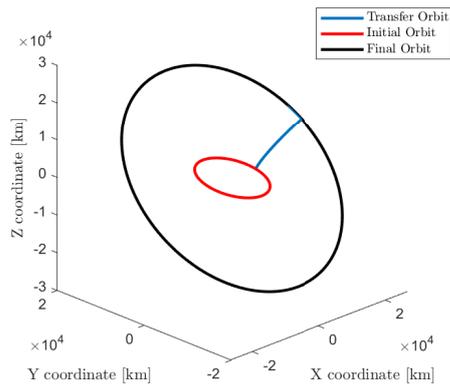


(c) LEO to GEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-4} \text{ km/s}^2$.

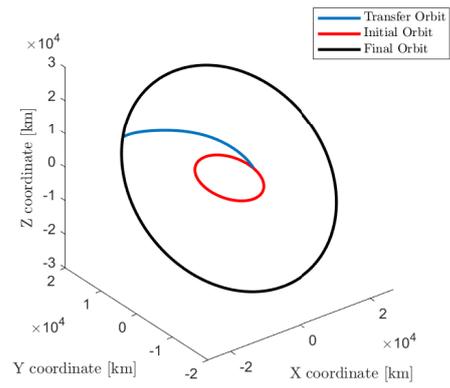


(d) LEO to GEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-5} \text{ km/s}^2$.

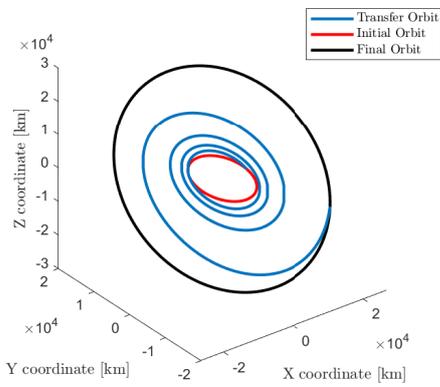
Figure 7.14: LEO to GEO transfer trajectories for different amounts of thrust acceleration.



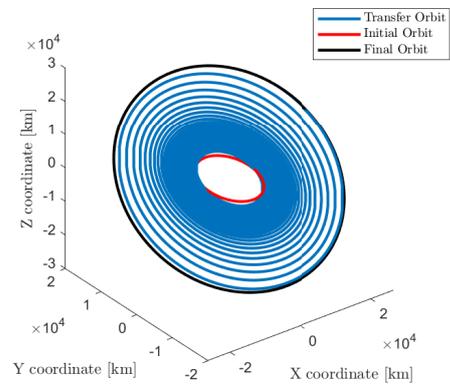
(a) LEO to MEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-2} \text{ km/s}^2$



(b) LEO to MEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-3} \text{ km/s}^2$.



(c) LEO to MEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-4} \text{ km/s}^2$.



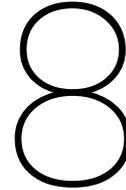
(d) LEO to MEO transfer trajectory for $f_{acc} = 1 \cdot 10^{-5} \text{ km/s}^2$.

Figure 7.15: LEO to MEO transfer trajectories for different amounts of thrust acceleration.

With all components of SGP4-LT validated, SGP4-LT was compared to the numerical Edelbaum's solution and it was found that the semi-major axis and inclination behave as expected due to the low-thrust force by Edelbaum. The other elements behave as expected due to the perturbations of SGP4. The argument of perigee acts unexpected, but this is due to the fact that ω is undefined for (near-) circular orbits. The computation time of SGP4-LT is impressive. It obtains an optimized low-thrust transfer trajectory after five seconds. The numerical Edelbaum solution requires seven minutes to obtain almost the same results without perturbations.

Finally, the transfer orbits obtained from SGP4-LT were compared with the transfer orbits from (Herman and Spencer, 2002). These orbits were designed using a numerical optimization and had less constraints than SGP4-LT. There was no constraint on eccentricity, thrust angles, and coasting was allowed. It was found that for high values of thrust acceleration, SGP4-LT obtains incorrect results, but with decreasing and thus more reasonable thrust accelerations the results actually become closer to the results obtained by (Herman and Spencer, 2002).

SGP4-LT is validated and it is concluded that with its constraints due to Edelbaum's solution and with the accuracy of SGP4 in mind, it offers a good and fast first approximation for a low-thrust transfer orbit around Earth.



Practical Applications

"The universe is big. It is vast and complicated and ridiculous. And sometimes, very rarely, impossible things just happen and we call them miracles"

—The 11th Doctor (Doctor Who)

In the previous chapters, the SGP4-LT tool was developed and tested to design near-circular low-thrust transfer trajectories using analytical methods. In this chapter, the tool is used to design transfer trajectories of interesting cases to show the purpose and multi-functionality of SGP4-LT.

The following TLE is used throughout this chapter unless indicated otherwise. All used TLE's are shown in Appendix D.

```
0 XXXXX
1 XXXXU XXXXA 18221.28032416 .00001255 00000-0 00000-0 0 9993
2 XXXX YYY.YYYY 114.0000 0000000 40.0000 000.0000 YY.YYYYYYYY000000
```

The XXX represent values such as satellite name and international designator and have no influence on a trajectory calculated by SGP4-LT. The YYY indicates specific values for inclination and mean motion and these values differ per test case. The numbers shown in the above TLE indicate the epoch year with the epoch day and its fraction, a set right ascension of ascending node of 114 degrees and an argument of perigee of 40 degrees. The spacecraft has an ion engine with a specific impulse of 3000 seconds and has an assumed dry mass of 900 kg. A low-thrust acceleration of $3.5e-7$ km/s² is used.

Three different cases are considered. First, orbit raising cases are evaluated using the SGP4-LT tool and the outcome is compared with the high-thrust Hohmann transfer, which was briefly discussed in Section 2.7. This is followed by Section 8.2 which discusses different cases for non-coplanar orbit raising from LEO to GEO or MEO orbits.

The high-thrust transfer is performed by a conventional bi-propellant thruster with a specific impulse of 380 seconds (McCormack, 2007). The low-thrust transfers are performed by an ion thruster with a specific impulse of 3000 sec (Herman et al., 2007). The dry mass of the considered spacecraft is 900 kg. In Section 8.3 the SGP4-LT tool is tested for continuous drag resistance, which is a completely different application. This is based on the GOCE mission.

8.1. Orbit Raising

In this section, orbit raising is considered. The results of the SGP4-LT tool are compared with a high-thrust Hohmann transfer. The initial parking orbit has an altitude of 180 km and the final orbits have a radius of 42164 or 27320 km. There is no inclination change. During the generation of the results it became clear that the inclination change cannot be equal to zero: the initial and final inclination cannot be exactly the same. This would result in a division by zero in Equation 5.21 since $\beta_0 = 0$. Therefore the difference between the initial and final inclination is set to $1 \cdot 10^{-10}$ degrees. Furthermore, the inclination and eccentricity cannot be

exactly 0 and are increased to 0.1 degrees and $4 \cdot 10^{-6}$ respectively in case these are set to zero, as was found in Chapter 7.

Four test cases are selected. All four test cases have an initial parking orbit with an altitude of 180 km. The final orbit has either a radius of 42164 or 26578 km. The orbit raising of test cases 1 and 3 occurs at 0 degrees inclination, whereas test cases 2 and 4 have an inclination of 28.5 degrees. A difference in the results due to the inclination is not expected. The test cases are summarized in Table 8.1.

The transfer trajectories are calculated in three different ways. The energy-efficient Hohmann method for high-thrust propulsion is calculated using the equations presented in Section 2.7. The purely Edelbaum solution is followed and finally the SGP4-LT tool is used to design the transfer trajectories. The results can be found in Table 8.2.

The ΔV 's obtained using the low-thrust methods are higher than the ΔV obtained by the Hohmann transfer. This does not mean that these low-thrust transfers are less efficient than a Hohmann transfer. The propellant mass is calculated using Tsiolkovsky's rocket equation (Chapter 3) to show the advantage of low-thrust propulsion.

Table 8.1: Test cases with initial and final conditions

Test Case	R_0 [km]	R_f [km]	i_0 [deg]	i_f [deg]
1	6558	42164	0	0
2	6558	42164	28.5	28.5
3	6558	26578	0	0
4	6558	26578	28.5	28.5

Table 8.2: Results for orbit raising using Hohmann transfer, Edelbaums's solution and SGP4-LT.

Test Case	Hohmann			Edelbaum			SGP4-LT		
	t_f [min]	ΔV_{tot} [km/s]	M_{prop} [kg]	t_f [min]	ΔV_{tot} [km/s]	M_{prop} [kg]	t_f [min]	ΔV_{tot} [km/s]	M_{prop} [kg]
1	3.15e2	3.94	1689.75	2.22e5	4.66	154.51	2.25e5	4.72	156.63
2	3.15e2	3.94	1689.75	2.22e5	4.66	154.51	2.25e5	4.72	156.63
3	1.28e2	3.12	1175.60	1.57e5	3.29	106.63	1.55e5	3.35	108.64
4	1.28e2	3.12	1175.60	1.57e5	3.29	106.63	1.60e5	3.35	108.64

The spacecraft propelled by high-thrust requires a ten times larger amount of propellant compared to the low-thrust propelled spacecraft, which is caused by the difference between the specific impulses. This clearly shows the efficiency of a low-thrust engine over a high-thrust engine. The difference in ΔV , and thus propellant mass, for the pure Edelbaum method and SGP4-LT results is due to the longer flight time, which is caused by the perturbations in SGP4 which are absent in Edelbaum's method.

8.2. Transfer between two non-coplanar orbits

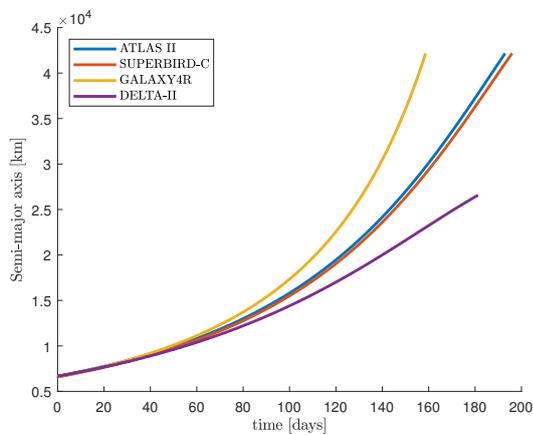
Spacecraft with a geostationary orbit as target should be launched from a launch site near the equator in order to avoid the expensive plane change manoeuvres (Wakker, 2010). Today, Kourou in French Guyana is the main launch site used for the launch of geostationary satellites. Cape Canaveral in the United States of America is used as well to launch satellites intended for geostationary orbits. Four real life examples are used in this Section. All four examples were placed in a geostationary transfer orbit, which is a high elliptical orbit with a large apogee. Then, using high-thrust engines, this apogee is lowered together with the inclination to the final altitude and zero inclination of a geostationary satellite. These examples are compared with the SGP4-LT tool in terms of ΔV and transfer time. Since the SGP4-LT trajectory design tool is only designed to work with near-circular orbits, the perigee altitude is used as radius of the parking orbit. Furthermore, these four examples are put in this geostationary transfer orbit by the third and final burn of the launch vehicle, which results in a lower ΔV used by the spacecraft to reach GEO.

The initial data is filled in a TLE format, with arbitrary values chosen for ω and Ω . The mean anomaly is

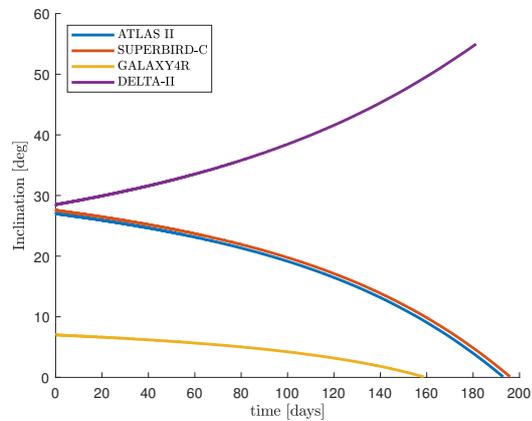
set to zero. The results are shown in Table 8.3. The semi-major axis, inclination and right ascension of the ascending node are plotted versus the transfer time and these plots are shown in Figure 8.1.

Table 8.3: Spacecraft with their launch site, initial and final orbit properties. (K) is Kourou, (KSC) stands for Kennedy Space Center. The calculated time of flight and ΔV are shown in columns 7 and 8. The total CPU time required is shown in the final column.

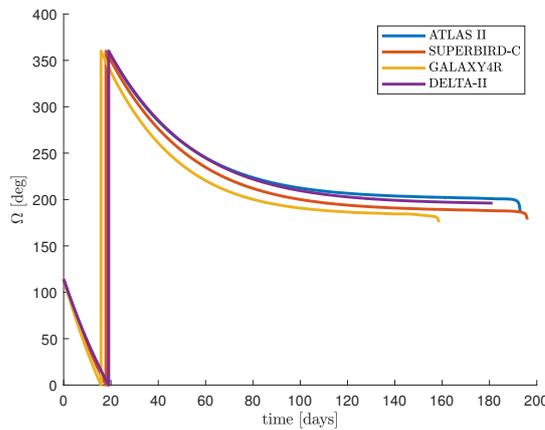
Spacecraft	Initial Radius [km]	Initial Inclination [deg]	Used Eccentricities [-]	Final Radius [km]	Final Inclination [deg]	t_f [days]	ΔV_{tot} [km/s]	CPU Time [sec]
Atlas II (KSC)	6663	27	4e-6	42164	0.10	192.93	5.83	4.65
Superbird-C (KSC)	6563	27.55	4e-6	42614	0.10	195.94	5.94	5.92
Galaxy 4R (K)	6578	6.998	4e-6	42614	0.00	158.66	4.80	4.04
Delta II (KSC)	6659	28.5	4e-6	26578	55.00	181.30	5.48	4.19



(a) Semi-major axis versus transfer time.



(b) Inclination versus transfer time.



(c) Right Ascension of the ascending node versus transfer time.

Figure 8.1: Results for the semi-major axis, inclination and right ascension of the ascending node of the transfer trajectories for ATLAS II, SUPERBIRD-C, GALAXY4R and DELTA-II.

Due to the large increase in semi-major axis, the inclination change is performed during the transfer with the steepest decrease near the final altitude. From GALAXY4R it can be observed that the inclination change consumes a significant part of the transfer time. An inclination change of only 7 degrees instead of around 28 degrees results in a decrease of 40 days of travel time. By increasing the semi-major axis to the final 26578 km

radius of DELTA-II, which is only half of the 421644 km semi-major axis, the decrease of travel time is 15 days. The right ascension of the ascending node shows the behaviour as expected and is verified in Section 7.5.

The total CPU time required to obtain these transfer trajectories is shown in the final column of Table 8.2. An average CPU time of a mere 5 seconds is required for a low-thrust transfer trajectory calculation.

8.3. Station-keeping in Low Earth Orbit

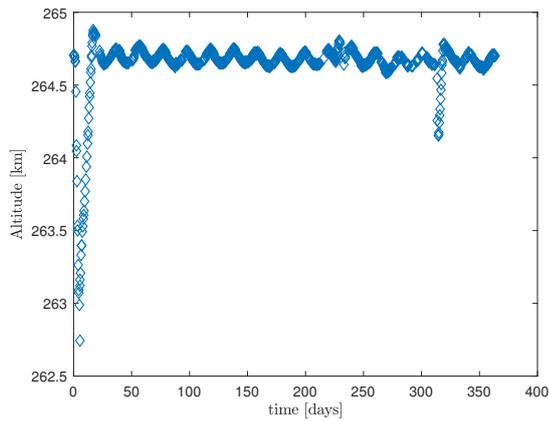
The versatility of the SGP4-LT tool is shown by an example of station keeping to compensate the loss of orbital energy due to atmospheric drag. As an example, the GOCE satellite is selected. This satellite was in a very low Earth Orbit, with an altitude of around 260 km (Wallace et al., 2011). The GOCE spacecraft was one of the first spacecraft to use low-thrust ion propulsion to counteract the atmospheric drag. In Chapter 3 more information on the GOCE satellite is provided. Historical TLE data for GOCE was obtained from space-track.org for its entire mission time. The spacecraft had a constant altitude in 2012, in 2013 its altitude was lowered twice and eventually GOCE disintegrated in the atmosphere.

The first available TLE from 2012 for GOCE is used and the satellite is propagated using SGP4-LT with thrust turned off. All TLE from 2012 are loaded in SGP4-LT as well and are run at initial epoch only to obtain one state vector for each TLE. Then, all state vectors are converted to orbital elements. The altitude, derived from the semi-major axis, the orbital element altered most by atmospheric drag, is plotted in Figure 8.2a. Using that first available TLE, GOCE is propagated for one year using SGP4-LT and the resulting altitude is plotted in Figure 8.2b. The propagated SGP4 altitude varies about 20 km, which is standard SGP4 behaviour as was shown in Chapter 5. Without any counter effects, the spacecraft's altitude is descending rapidly: in less than 150 days, the spacecraft has re-entered Earth's atmosphere. Looking at the historical TLE data, the altitude is essentially constant due to the ion propulsion system. To obtain a better visual understanding, both are depicted in Figure 8.2a.

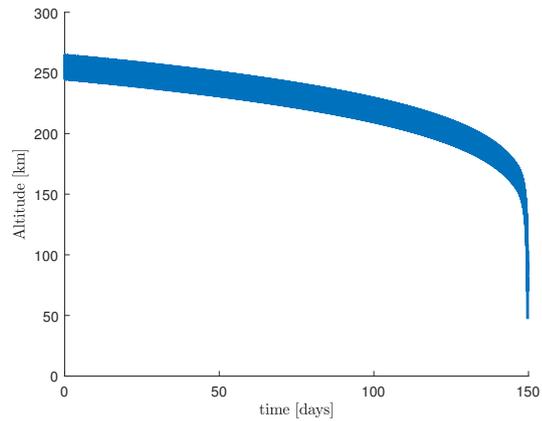
In Section 8.1 it was found that the inclination change cannot be exactly equal to zero. This was solved by adding a margin at the initialization of SGP4-LT. When the final inclination and final semi-major axis are reached, the algorithm stops and the results are printed. During station-keeping, SGP4-LT only stops after the time given as user input is reached. During this time, the inclination and semi-major axis are constantly varying around the desired values. Since there is no stopping requirement other than the flight time, the rare case that the inclination at t_n is exactly equal to the desired inclination, occurred. This was due to the fact that inside the low-thrust module there was no case for an inclination difference of exactly zero. This was either solved at initialization or the algorithm was terminated before a difference of exact zero was reached. Initially, two if-cases were possible inside the low-thrust module; either the current inclination is larger than the final inclination or the current inclination is smaller. If the difference was exactly zero, the Edelbaum solution was not initialized and the value for the semi-major axis was only initialized as a zero variable, which eventually resulted in an infinite loop. This was solved by temporally adding the set margin to the value of the final inclination.

Next, the low-thrust module is turned on with the final semi-major axis and inclination equal to the initial values. To make SGP4-LT suitable for altitude maintenance, a transfer option was added. This option can be turned on or off by the user. To turn this function on, a "1" is inserted which results in the termination of the tool when the final conditions are met. For altitude maintenance, the transfer function is turned off ("0"). This results in SGP4-LT to stop when the total time, as set by the user, is reached instead of at final conditions. The results of SGP4-LT with a thrust acceleration of $3.5 \cdot 10^{-7} \text{ km/s}^2$ can be found in Figure 8.2d.

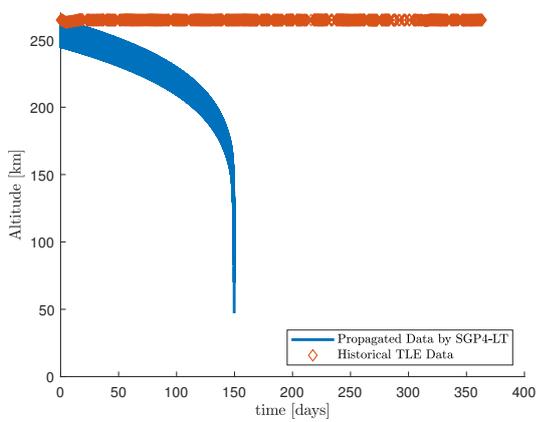
With the thrust turned on, the average semi-major axis remains constant. However one should consider the fact that the thrust is always on with its constant acceleration; even when the spacecraft is very close to its desired altitude, the same amount of acceleration is used to reach its desired altitude and thus thrust is wasted. Comparing Figures 8.2d and 8.3a, it can be observed that the same altitude with the same deviation is maintained while the thrust accelerations differ one order of magnitude. In both situations, the change in location due to the perturbations is equal. The only difference is that a higher thrust acceleration results in a faster return to the satellites desired altitude. However, due to the deviation in SGP4 this cannot be observed. Therefore no difference between the two plots can be found.



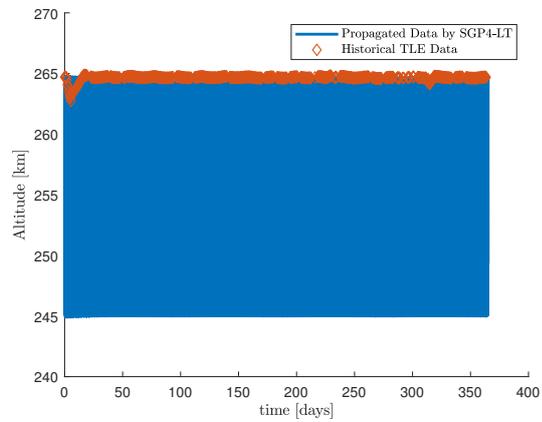
(a) Altitude versus time obtained from historical TLE data of GOCE of the year 2012.



(b) Altitude versus time obtained by propagation without thrust of GOCE TLE data from January 1, 2012.



(c) The altitude of GOCE plotted versus the time, starting at January 1, 2012, obtained by propagating the first TLE available on January 1, 2012 without thrust, and the altitude obtained from historical TLE data of GOCE during 2012.



(d) Propagated using SGP4-LT with a thrust acceleration of $3.5 \cdot 10^{-7}$ km/s², together with the historical data for the year 2012

Figure 8.2: The semi-major axis versus time, obtained from historical TLE data, and propagated from January 1, 2012 onwards.

The thrust acceleration was decreased multiple times with one order of magnitude. As concluded previously, the altitude was kept constant using a thrust acceleration of $3.5 \cdot 10^{-7} \text{ km/s}^2$. However, decreasing the thrust acceleration to $1 \cdot 10^{-9} \text{ km/s}^2$ results in a re-entry after 240 days. This thrust acceleration can thus not overcome the drag perturbation. The associated plot of altitude versus time is shown in Figure 8.3b. Next, the thrust acceleration was lowered with smaller increments and it was found that $1.8 \cdot 10^{-9} \text{ km/s}^2$ was the minimum thrust acceleration possible to maintain the altitude, as can be seen in Figure 8.3c.

To obtain the propellant mass required for one year, it is assumed that the total mass of the spacecraft is 1000 kg at January 1, 2012. The ion thruster is a Qinetiq T5 thruster with Xenon propellant as used on GOCE. Assuming constant total mass and with an acceleration of $1.8 \cdot 10^{-9} \text{ km/s}^2$, the thrust equals 1.8 mN, which follows from:

$$T = f_{acc} M_{sc} \quad (8.1)$$

This calculated thrust force is used in Figure 8.4 to obtain the specific impulse I_{sp} . The plot in this figure is from (Wallace et al., 2011) and represents the specific impulse as a function of thrust force. A specific impulse of 800 seconds is selected. The required ΔV is found by:

$$\Delta V = f_{acc} t_f \quad (8.2)$$

With t_f the total time, which is a year. This results in a required ΔV of 56.803 m/s. Using Tsiolkovsky's Equation (Chapter 3)

$$\Delta V = I_{sp} g_0 \ln \left(\frac{M_0}{M_f} \right) \quad (8.3)$$

resulting in a total propellant mass of 7.21 kg of xenon required to maintain the correct altitude for a year. In reality, the thrust acceleration will not be constant and not turned on at all times during the length of the mission. Furthermore there is only out of plane thrust available due to the use of Edelbaum's solution, which has negligible effect on the orbital elements other than semi-major axis and inclination.

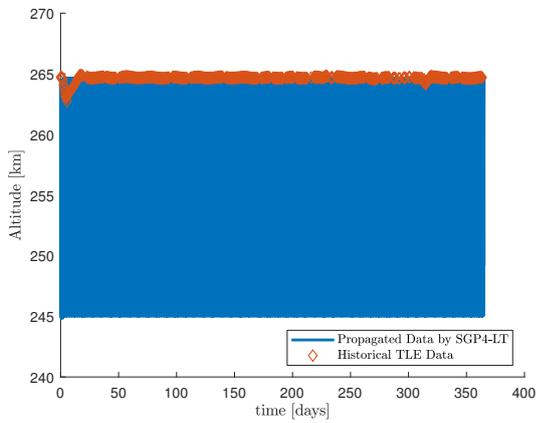
The GOCE spacecraft had 40 kg of Xenon propellant on board, which lasted 4.5 years (Wallace et al., 2011). All GOCE TLE is used to obtain the course of the altitude during the mission time and this is depicted in Figure 8.5. During 1176 days, the altitude was kept constant. It is assumed that when there is a decrease in altitude, the low-thrust engine is turned off and no xenon is used. Furthermore, after day 1400, near the end of its mission life, it became more difficult to maintain the correct altitude due to lower pressure in the xenon tank (Wallace et al., 2011). To approximate this end-of-life period which was not taken into account in the calculation, an additional 200 days is added to the 1176 days. In this first guess approximation, it is thus assumed that the ion engine was turned on for a total of 1376 days, or 3.77 years. This would result in a total required xenon propellant mass of 27.18 kg, which is only 67.95% of the real amount of propellant used by GOCE. This would thus not be enough propellant for the GOCE mission. Since SGP4-LT is only used for first guess approximations, it might be a better idea to use a higher value for thrust acceleration or include a safety margin. However, it can be concluded that the values obtained by SGP4-LT are reasonable since the thrust values required to maintain the orbit are all in the range of the thrust of GOCE as shown in Figure 8.4. This was a fast first approximation including multiple assumptions, but it again shows the versatility of the SGP4-LT tool.

The following assumptions were made to calculate the propellant mass.

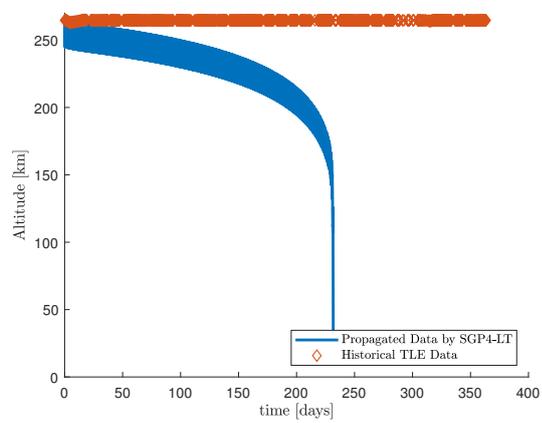
- When GOCE decreased its altitude it was assumed that the thrust was turned off.
- The thrust force and thus specific impulse is assumed constant during the entire mission due to a constant mass.
- When using SGP4-LT, the thrust is always turned on.
- The altitude is the only element maintained.

8.4. Conclusion

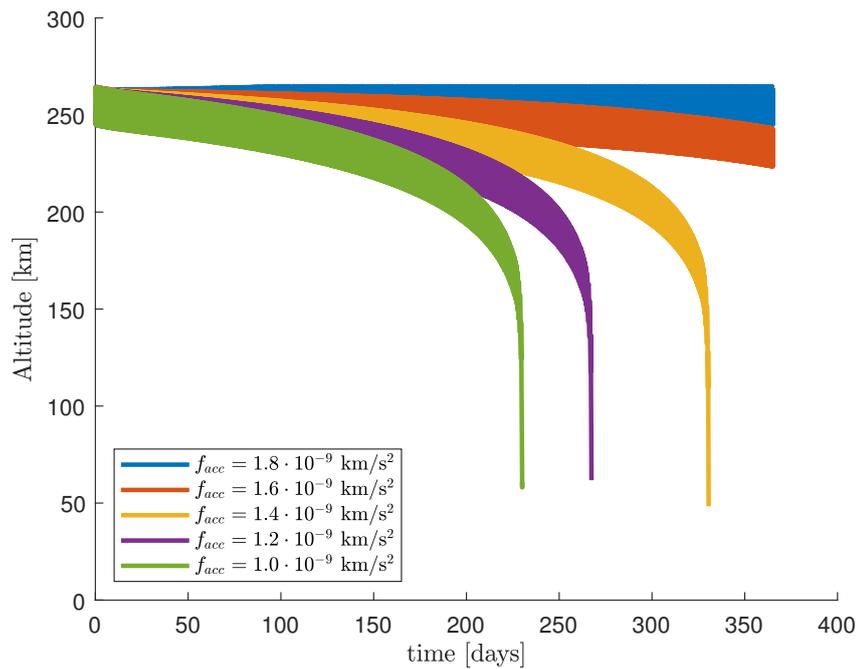
This chapter discussed three different cases for which the SGP4-LT tool can be used. In the case of orbit raising, logical results are obtained, but it was found that the SGP4-LT tool cannot handle an inclination difference of exactly zero. An exception is therefore added to the tool to incorporate orbital raising. In case of



(a) Altitude versus time with a thrust acceleration of $1.0 \cdot 10^{-8}$ km/s².



(b) Altitude versus time with a thrust acceleration of $1.0 \cdot 10^{-9}$ km/s².



(c) The propagated altitude of GOCE using different values for thrust acceleration for one year.

Figure 8.3: The semi-major axis versus time, obtained from historical TLE data, and propagated from January 1, 2012 onwards.

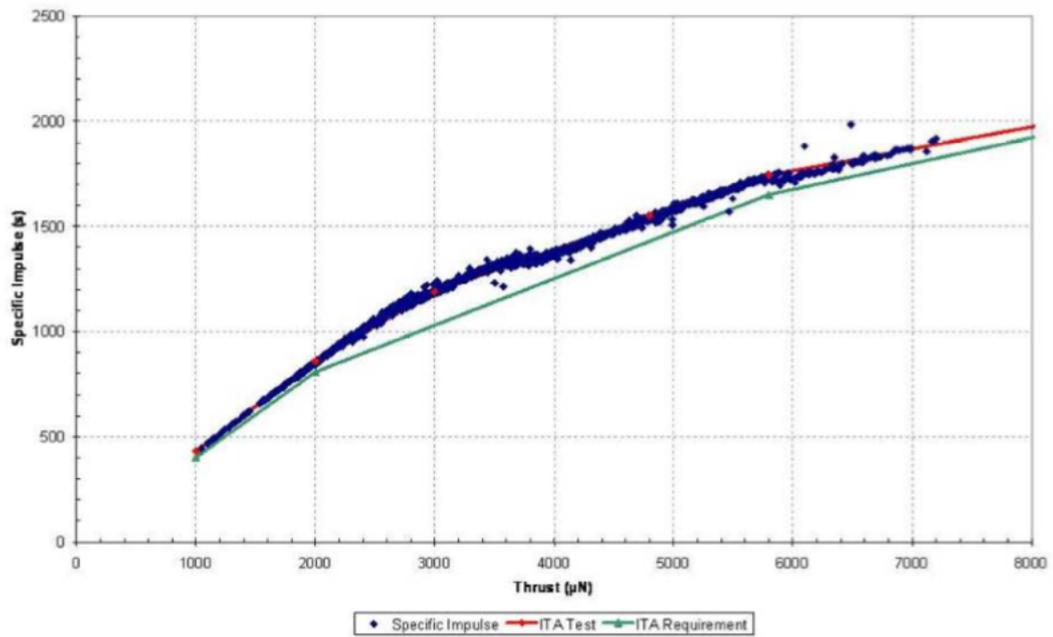


Figure 8.4: Directly calculated specific impulse of GOCE, plot obtained from (Wallace et al., 2011).

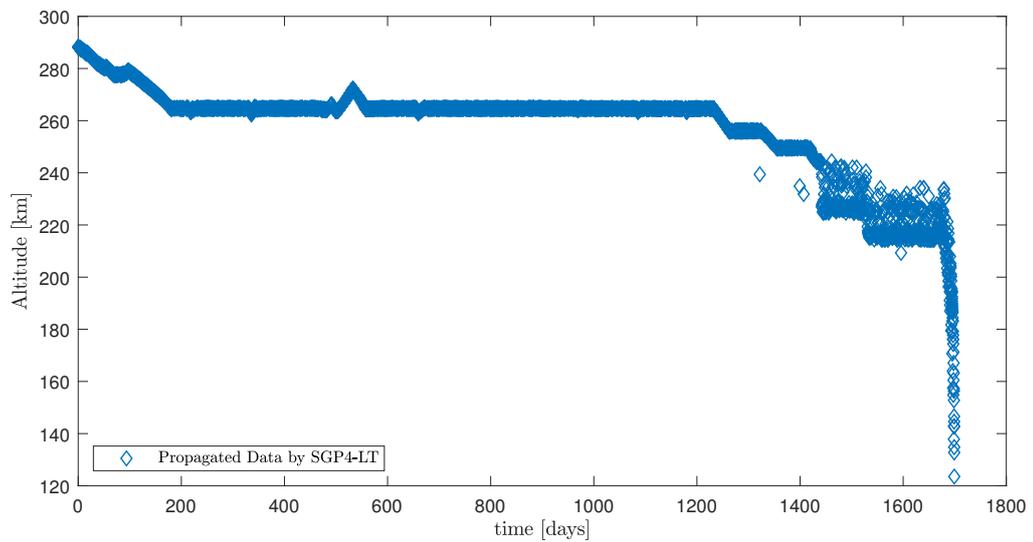


Figure 8.5: The altitude of GOCE during its entire mission time, obtained by using historical TLE data for GOCE.

zero inclination difference, the difference is set to $1e-9$ degrees. The mass savings compared to high thrust propulsion were also provided.

The second application considered a transfer between two non-coplanar orbits, and real-life examples were used. These examples required an alteration in initial orbit for SGP4-LT to be functional, but for all 4 examples realistic results were found.

The SGP4-LT tool was finally used to maintain the altitude of a satellite in a low orbit around Earth. Assumptions were required to calculate a propellant mass. SGP4-LT obtained a propellant mass of 27.18 kg for maintaining the altitude at 265 km for a period of 3.77 years. This is only 67.95 % of the propellant mass GOCE had available, but it shows that the thrust accelerations required are in the range of the used ion thruster. Using SGP4-LT is a novel approach of the design of low-thrust orbits whilst taking perturbations into account. The SGP4-LT tool is fast: within five seconds a transfer trajectory of 180 days is computed. The most time consuming part is the storing of data in an output file. The fast nature of this tool makes it a good educational tool. Plots can be easily generated to provide a better understanding of low-thrust orbits and, due to the optimized Edelbaum's solution, it can be clearly observed that inclination change is less expensive at higher altitudes.

9

Conclusions and Recommendations

"He knew that all the hazards and perils were now drawing together to a point: the next day would be a day of doom, the day of final effort or disaster, the last gasp."

—J.R.R. Tolkien, *The Return of the King*

In this chapter, the most relevant finding of this thesis research and recommendations are provided. In Section 9.1 the main conclusions are given to answer the research question and its subquestions. Section 9.2 provides the recommendations given for further research and development of the SGP4-LT tool.

9.1. Conclusions

In this section, the research question introduced in Chapter 1 is answered. The research question and its subquestions are repeated here:

To what extent would the SGP4 model adapted for low-thrust propulsion perform, compared to numerically low-thrust Earth-orbit transfer trajectory design methods?

Its subquestions are:

1. How does SGP4 work?
 - (a) How is the SGP4 algorithm structured?
 - (b) How do orbital perturbations affect the motion of a spacecraft modelled in SGP4?
 - (c) What are the main sources of propagation uncertainty using SGP4?

These subquestions are answered in Subsection 9.1.1.

2. How can SGP4 be used to design low-thrust trajectories?
 - (a) What kind of low-thrust trajectory design method is required to work in cohesion with SGP4?
 - (b) Which modifications have to be made to the original SGP4 algorithm in order to be able to implement a low-thrust module?

These subquestions are answered in Subsections 9.1.2 and 9.1.3.

3. Are the results obtained by SGP4-LT representative for a first approximation trajectory design tool?
 - (a) How do small changes in (the direction of) the low-thrust force affect the performance of the optimization tool?
 - (b) What are the limitations of SGP4-LT?
 - (c) What is the accuracy of SGP4-LT compared to other methods?
 - (d) For which purposes can SGP4-LT be used?

These subquestions are answered in Subsections 9.1.4-9.1.6.

9.1.1. Simplified General Perturbations Model

The SGP4-LT tool combines an analytical low-thrust method with the Simplified General Perturbations Model 4, SGP4. This model is widely used today by the scientific community to propagate spacecraft in orbit around Earth, by using two line elements TLE, which is an ASCII file containing the mean orbital elements of a spacecraft at a certain epoch.

The SGP4 model as we know it today is the result of an evolution of an extensive analytical method to describe the effect of perturbations on a spacecraft in orbit around Earth with its foundations in the 1960's. The SGP4 model uses mean orbital elements as input which are orbital elements with the periodic variations removed and are therefore not equal to osculating orbital elements. The output is a state vector with Cartesian coordinates and velocity components. The accuracy of the SGP4 algorithm decreases fast after a few days, with an error of a magnitude in kilometers. The structure was extensively discussed in Chapter 4.

9.1.2. Analytical Low-Thrust Solution

An analytical low-thrust solution suggested by Edelbaum was extended into a minimum flight problem by (Kechichian, 1997). This solution is used in the SGP4-LT tool due to its equations for semi-major axis and inclination as functions of the initial and final conditions, and time. To implement a low thrust in SGP4, it was required that this low-thrust solution was capable of returning orbital elements as a function of time. The use of other analytical methods, e.g. shape-based methods, was not suitable to be used in SGP4 since these solutions require a numerical optimization. Therefore Edelbaum's solution was selected. Edelbaum's solution has however certain constraints that decrease the usability of the SGP4-LT tool. The initial and final orbits must be (near) circular and this also hold for the transfer trajectory. Furthermore, the thrust is constant and always on during the transfer. The thrust angle is only out-of-plane and assumed constant during a revolution. A numerical comparison using these constraints obtained similar results, but this solution is still not completely optimized.

9.1.3. Iterative SGP4

The original SGP4 algorithm is initialized at its starting time and uses stored constants during the propagation. The orbit of the spacecraft changes remarkably when a low-thrust force is applied. This results in that the stored values at the starting time are not valid later in time. It was therefore required to rewrite the SGP4 algorithm to initialize all values at every time step. As a result, the output is required as input in order to propagate to the next time step. An issue arose here, since the Cartesian state vector could not easily be converted into mean orbital elements that are used as input. Various solutions were found in literature ((Walter, 1967), (Andersen, 1994), (Ely, 2013), (Lee, 2002), (Der and Danchick, 1996)) but these lack large convergence rates or the mean elements are solved by brute numerical calculations.

A solution was found by using the solution suggested by (Walter, 1967) and (Andersen, 1994). This solution, the Modified Walter Method, was corrected and extended by using modified equinoctial elements. It is an elegant, iterative solution and has a convergence rate of 99.5% percent, when tested the entire satellite catalogue, which consist of 17542 objects in Earth orbit. Convergence was not reached in the case of combinations for very small values for inclination and eccentricity. The cases that failed to converge had a combination of very small values for inclination and eccentricity, which only occurred in GEO orbits according to the satellite catalogue. It was found that convergence is reached when:

$$0.05^\circ \leq i \leq 180.0^\circ \quad (9.1)$$

for

$$4e-6 \leq e < 1.0 \quad (9.2)$$

This is a large improvement compared with the solution by (Walter, 1967), which had only convergence reached for $0.1^\circ \leq i \leq 90.0^\circ$ and $0.01 \leq e \leq 0.30$.

It is expected that the non-convergence is caused by floating-point errors when converging the modified equinoctial elements back into osculating orbital elements.

9.1.4. SGP4-LT

Various changes in the handler code as published by (Vallado et al., 2006) were made to obtain a fully functional SGP4 handler algorithm to provide the proper input and output required for SGP4-LT. In this handler file, different modes of operation for SGP4-LT can be selected.

Three modes of operation can be identified. The low-thrust force can be turned on or off, and its magnitude

can be changed. If the thrust is turned off, the SGP4-LT will act as the normal (iterative) SGP4 algorithm. If the thrust is turned on, two modes are available. The first mode is the design of a transfer to an orbit as specified by the user. The third mode of operation is the maintenance of altitude. If this case is selected, the final orbit conditions must be equal to the initial orbit conditions. A stop time is set whereas for a transfer trajectory SGP4-LT would continue until the target orbit is reached. Furthermore, the step size can be set. In general, a step size of one minute was used in this thesis research. Finally, the names of the output files can be provided.

Additionally, some settings as provided by (Vallado et al., 2006) can be changed, such as the type of world geodetic system, but is recommended to leave these as they are for general use.

It followed from the Modified Walter Method that combinations of very small values for inclination and eccentricity must be avoided in SGP4-LT. Therefore, the input values are checked and if they are below the values required for convergence, they are automatically adjusted to exactly these values, 0.05° and $4e-6$ for inclination and eccentricity, respectively.

9.1.5. Verification

SGP4-LT was verified by calculating transfer trajectories which were verified for Edelbaum's solution. It was found that the SGP4-LT tool combines the perturbations and low-thrust acceleration correctly. Due to the perturbations, the transfer time will increase slightly. The evaluation of semi-major axis and inclination showed, besides the effect due to a slightly longer flight time, the same behaviour for a pure Edelbaum solution. When looking at the orientation of the orbit, the right ascension of the ascending node changed with a rate as expected by SGP4 at its corresponding altitude. Furthermore, a pure analytical Edelbaum solution cannot give a graphical interpretation of the transfer trajectory. To obtain a plot of the trajectory, or orbital elements other than the semi-major axis or inclination, a numerical integration is required. This integration was performed in Matlab and needed a computation time up to seven minutes, whereas it took SGP4-LT only a mere 5 seconds to produce the data required to plot the trajectory.

It is therefore concluded that the SGP4-LT works correct according to SGP4 and Edelbaum's solution and is extremely fast.

The SGP4-LT tool was tested with a numerical optimization tool for circular-to-circular orbits. In the first instance, this might be a comparison of apples and oranges: SGP4-LT uses continuously constant out-of-plane thrust and the transfer trajectory should have a (near) zero eccentricity. The transfer trajectory of numerical solution by (Herman and Spencer, 2002) is not constrained by eccentricity, coasting is allowed, the thrust angle can vary in more directions, and no perturbations are taken into account. However, it was found that the eccentricity calculated by (Herman and Spencer, 2002) remains near zero, and that made it a good candidate to compare the limits of Edelbaum's solution with an optimized numerical solution. For high values of thrust acceleration, the results obtained by SGP4-LT were wrong. However, by lowering the thrust acceleration, the trajectory started to act more like a low-thrust trajectory and the time of flight and ΔV for both solution became comparable, although the results obtained by (Herman and Spencer, 2002) are still better optimized.

9.1.6. Practical Application

The SGP4-LT tool was tested with different test cases. For orbit raising, it was found that SGP4-LT cannot handle inclination changes exactly equal to zero. This was solved by applying a margin of $1 \cdot 10^{-9}$ degrees to the final inclination in case an inclination change of zero degrees is given as input.

A comparison was made to a high-thrust Hohmann transfer and the mass savings due to low-thrust propulsion were shown. The low-thrust propellant mass was about a tenth of the high-thrust propellant mass.

The next application was an non-coplaner orbit change for just-launched satellites. Four launched satellites were used as an example, with their initial and final conditions given. It was shown how expensive inclination change is compared to increasing the semi-major axis of an orbit. Again, these orbits were calculated with SGP4-LT in 5 seconds.

A third, completely different use of SGP4-LT was found in maintaining the altitude of a satellite in a low Earth orbit. Using GOCE as an example, it was shown that SGP4-LT can be used as a first guess for the propellant use required to maintain the altitude of an orbit.

9.1.7. Research Objective

The research objective was stated as:

Extend the existing SGP4 model with a low-thrust propulsion module and develop a fast, reliable and robust optimization tool which uses this extended model.

This was succeeded by the development of SGP4-LT, which includes a low-thrust optimization solution to obtain optimized transfer trajectories. Furthermore, it can be used for altitude maintaining calculations.

9.2. Recommendations and Future Research

A practical and fast analytical transfer trajectory design method was developed for low-thrust spacecraft in orbits around Earth, while undergoing perturbations. The research question has been answered and the research objective has been met, but recommendations can still be made for future research.

First, Edelbaum's solution is used in SGP4-LT. This analytical low-thrust method has strict constraints for initial and final conditions. Improvements to Edelbaum's solution exist, such as the addition of shadow arcs. Implementing these improvements would result in more realistic results.

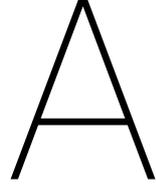
Inclination change near the anti-nodes is highly efficient, but Edelbaum's solution requires constant thrust. The iterative nature of SGP4-LT can be used in its advantage to reduce the waste of propellant near the anti-nodes of the orbit. The thrust module could either be completely turned off when near the anti-nodes, or the transfer problem is temporarily changed in a pure orbit-raising problem.

The relevance of SGP4-LT would increase if a different solution, which does not have the constraints set by Edelbaum's solution, could be inserted. One should take into account that the use of numerical optimization methods is not suitable. Due to the perturbations, the low-thrust method should be initialized again at every time step.

Further research is required for the Modified Walter's Method to solve the non-convergence cases for very small values of inclination and eccentricity, which occur due to singularity in the conversion of modified equinoctial elements to orbital elements.

The iterative SGP4 algorithm was verified by comparing its results to the results of the original SGP4 algorithm. It was found that the results differ per spacecraft, and slightly better results were found when compared with (Lee, 2002). To get a complete understanding of the accuracy of the iterative SGP4 algorithm, it is suggested to compare the iterative SGP4 with the accurate data from GPS satellites.

SGP4-LT is written in C++ in order to make it compatible with TU Delft Astrodynamics Toolbox (TUDAT), but is not yet implemented into TUDAT. Once implemented, its results can be easier compared to the results of numerical propagations obtained by TUDAT and is easier accessible for students.



Transformations

A.1. Frame Transformations

Transformation between Earth-centered inertial frame and Satellite-based orbit Frame RSW

In an ECI coordinate system, the unit vectors of the three axes of the RSW coordinate system are:

$$\hat{\mathbf{R}} = \frac{\mathbf{r}}{|\mathbf{r}|}, \quad \hat{\mathbf{W}} = \frac{\mathbf{r} \times \mathbf{v}}{|\mathbf{r} \times \mathbf{v}|}, \quad \hat{\mathbf{S}} = \hat{\mathbf{W}} \times \hat{\mathbf{R}} \quad (\text{A.1})$$

The transfer matrices between the RSW coordinate system and the ECI coordinate system are:

$$M_{RSW \rightarrow ECI} = [\hat{\mathbf{R}} \quad \hat{\mathbf{S}} \quad \hat{\mathbf{W}}] \quad (\text{A.2})$$

$$M_{ECI \rightarrow RSW} = [\hat{\mathbf{R}} \quad \hat{\mathbf{S}} \quad \hat{\mathbf{W}}]^T \quad (\text{A.3})$$

A.2. Coordinate Transformations

Kepler Elements to Cartesian Coordinates

The transformation from Kepler elements, $[a \ e \ i \ \omega \ \Omega \ \theta]$, to Cartesian elements $[x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]$ is derived in (Wakker, 2010) and can be done using *spherical geometry* or *rotation matrices*. The method using rotation matrices is explained here. The velocity components $\dot{x}, \dot{y}, \dot{z}$ are obtained using the spherical geometry method.

A non-rotating reference frame with axes ξ, η, ζ with its origin at the main body center is defined. The $\xi\eta$ -plane is the orbital plane, $+\xi$ points towards pericenter and $+\zeta$ points towards the angular momentum vector.

Three rotations of the reference frame are required. First, the $\xi\eta\zeta$ -frame is rotated around the ζ -axis by an angle $-\omega$. Next, a rotation around the line of nodes by an angle of $-i$. Finally, a rotation around the Z -axis over an angle of $-\Omega$ is performed. Combining these rotations and creating one rotation matrix leads to:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \quad (\text{A.4})$$

with:

$$\begin{aligned}
l_1 &= \cos \omega \cos \Omega - \sin \omega \sin \Omega \cos i \\
m_1 &= \cos \omega \sin \Omega + \sin \omega \cos \Omega \cos i \\
n_1 &= \sin \omega + \sin \omega \cos \Omega \cos i \\
l_2 &= -\sin \omega \cos \Omega - \cos \omega \sin \Omega \cos i \\
m_2 &= -\sin \omega \sin \Omega + \cos \omega \cos \Omega \cos i \\
n_2 &= \cos \omega \sin i \\
l_3 &= \sin \Omega \\
m_3 &= -\cos \Omega \sin i \\
n_3 &= \cos i
\end{aligned} \tag{A.5}$$

From the spherical geometry method (Wakker, 2010) one obtains:

$$\begin{aligned}
\dot{x} &= \frac{\mu}{H} [-l_1 \sin \theta + l_2 (e + \cos \theta)] \\
\dot{y} &= \frac{\mu}{H} [-m_1 \sin \theta + m_2 (e + \cos \theta)] \\
\dot{z} &= \frac{\mu}{H} [-n_1 \sin \theta + n_2 (e + \cos \theta)]
\end{aligned} \tag{A.6}$$

Cartesian Coordinates to Kepler Elements

The derivations and equations used are again from (Wakker, 2010).

The radius r and velocity V are determined with:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{A.7}$$

$$V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{A.8}$$

The angular momentum \mathbf{H} can be written as:

$$\mathbf{H} = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} y\dot{z} - z\dot{y} \\ x\dot{z} - z\dot{x} \\ x\dot{y} - y\dot{x} \end{bmatrix} \tag{A.9}$$

with magnitude:

$$H = \sqrt{H_x^2 + H_y^2 + H_z^2} \tag{A.10}$$

this leads to:

$$a = \frac{r}{2 - rV^2/\mu} \tag{A.11}$$

$$i = \arccos\left(\frac{H_z}{H}\right) \tag{A.12}$$

$$\sin \Omega = \frac{H_x}{H \sin i} \quad ; \quad \cos \Omega = \frac{H_y}{H \sin i} \tag{A.13}$$

E and e follow from:

$$e \sin E = \sqrt{\frac{1}{\mu a}} (x\dot{x} + y\dot{y} + z\dot{z}) \tag{A.14}$$

and

$$e \cos E = 1 - \frac{r}{a} \tag{A.15}$$

θ follows from:

$$\sin(\theta + \omega) = \frac{x}{r} \cos \Omega + \frac{y}{r} + \sin \Omega \quad ; \quad \sin(\omega + \theta) = \frac{z}{r \sin i} \tag{A.16}$$

Classical Orbital Elements to Modified Equinoctial Elements

The transformation follows directly from the definition of the modified equinoctial elements.

$$p = a(1 - e^2) \quad (\text{A.17})$$

$$f_{mee} = e \cos(\omega + \Omega) \quad (\text{A.18})$$

$$g_{mee} = e \sin(\omega + \Omega) \quad (\text{A.19})$$

$$h_{mee} = \tan(i/2) \cos \Omega \quad (\text{A.20})$$

$$k_{mee} = \tan(i/2) \sin \Omega \quad (\text{A.21})$$

$$L_{mee} = \Omega + \omega + \theta \quad (\text{A.22})$$

Modified Equinoctial Elements to Classical Orbital Elements

$$a = \frac{p}{1 - f_{mee}^2 - g_{mee}^2} \quad (\text{A.23})$$

$$e = \sqrt{f_{mee}^2 + g_{mee}^2} \quad (\text{A.24})$$

$$i = \arctan 2 \left(2\sqrt{h_{mee}^2 + k_{mee}^2}, 1 - h_{mee}^2 - k_{mee}^2 \right) \quad (\text{A.25})$$

$$\omega = \arctan 2 (g_{mee} h_{mee} - f_{mee} k_{mee}, f_{mee} h_{mee} + g_{mee} k_{mee}) \quad (\text{A.26})$$

$$\Omega = \arctan 2 (k_{mee}, h_{mee}) \quad (\text{A.27})$$

$$\theta = L_{mee} - (\Omega + \omega) \quad (\text{A.28})$$

Then the eccentric anomaly and mean anomaly follow from:

$$E = 2 \arctan \left(\tan \left(\frac{\theta}{2} \right) \sqrt{\frac{1-e}{1+e}} \right) \quad (\text{A.29})$$

$$M = E - e \sin E \quad (\text{A.30})$$

Cartesian Coordinates to Modified Equinoctial Elements

Again, the angular momentum vector \mathbf{H} is obtained:

The radius r and velocity V are determined with:

$$r = \sqrt{x^2 + y^2 + z^2} = |\mathbf{r}| \quad (\text{A.31})$$

$$V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} = |\mathbf{v}| \quad (\text{A.32})$$

and

$$\hat{\mathbf{u}} = \mathbf{r} / r \quad (\text{A.33})$$

$$\hat{\mathbf{v}} = \mathbf{v} / v \quad (\text{A.34})$$

The angular momentum \mathbf{H} can be written as:

$$\mathbf{H} = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} y\dot{z} - z\dot{y} \\ x\dot{z} - z\dot{x} \\ x\dot{y} - y\dot{x} \end{bmatrix} \quad (\text{A.35})$$

with magnitude:

$$H = \sqrt{H_x^2 + H_y^2 + H_z^2} \quad (\text{A.36})$$

Then the semi-latus rectum p is obtained by:

$$p = \frac{H^2}{\mu} \quad (\text{A.37})$$

The unit angular momentum vector:

$$\hat{\mathbf{h}} = \frac{\mathbf{H}}{H} \quad (\text{A.38})$$

h_{mee} and k_{mee} can then be computed:

$$h_{mee} = \frac{\hat{h}_x}{1 + \hat{h}_z} \quad (\text{A.39})$$

$$k_{mee} = \frac{-\hat{h}_y}{1 + \hat{h}_z} \quad (\text{A.40})$$

The unit vectors in the equinoctial frame are determined by:

$$\mathbf{f} = \begin{bmatrix} 1 - k_{mee}^2 + h_{mee}^2 \\ 2h_{mee}k_{mee} \\ -2k_{mee} \end{bmatrix} \quad (\text{A.41})$$

$$\mathbf{g} = \begin{bmatrix} f_y \\ 1 + k_{mee}^2 - h_{mee}^2 \\ -2h_{mee} \end{bmatrix} \quad (\text{A.42})$$

Equation A.41 and A.42 are normalized using

$$\hat{\mathbf{f}} = \frac{\mathbf{f}}{1 + k_{mee}^2 + h_{mee}^2} \quad (\text{A.43})$$

$$\hat{\mathbf{g}} = \frac{\mathbf{g}}{1 + k_{mee}^2 + h_{mee}^2} \quad (\text{A.44})$$

Finally, f_{mee} and g_{mee} are obtained by:

$$f_{mee} = \mathbf{e} \cdot \hat{\mathbf{f}} \quad (\text{A.45})$$

$$g_{mee} = \mathbf{e} \cdot \hat{\mathbf{g}} \quad (\text{A.46})$$

with \mathbf{e} :

$$\mathbf{e} = \frac{\mathbf{v} \times \mathbf{H}}{\mu} - \hat{\mathbf{u}} \quad (\text{A.47})$$

The true longitude is calculated using:

$$L_{mee} = \arctan 2(\hat{\mathbf{u}}_x + \hat{\mathbf{v}}_y, \hat{\mathbf{u}}_y - \hat{\mathbf{v}}_x) \quad (\text{A.48})$$

B

Gauss' Form of Lagrange Planetary Equations for near-circular orbits

Gauss' form of Lagrange Planetary Equations are shown in Equations B.1 - B.6.

$$\frac{da}{dt} = 2 \frac{a^2}{\sqrt{\mu p}} \left[f_S e \sin \theta + f_N \frac{p}{r} \right] \quad (\text{B.1})$$

$$\frac{de}{dt} = \sqrt{\frac{p}{\mu}} \left[f_S \sin \theta + f_N (\cos E + \cos \theta) \right] \quad (\text{B.2})$$

$$\frac{di}{dt} = f_W \frac{r}{\sqrt{\mu p}} \cos u \quad (\text{B.3})$$

$$\frac{d\omega}{dt} = -\sqrt{\frac{p}{\mu}} \left[f_W \frac{r}{p} \cot i \sin u + \frac{1}{e} \{ f_S \cos \theta - f_N \left(1 + \frac{r}{p} \right) \sin \theta \} \right] \quad (\text{B.4})$$

$$\frac{d\Omega}{dt} = f_W \frac{r}{\sqrt{\mu p} \sin i} \sin u \quad (\text{B.5})$$

$$\frac{dM}{dt} = n - f_S \left[\frac{2r}{\sqrt{\mu a}} - \frac{1-e^2}{e} \sqrt{\frac{a}{\mu}} \cos \theta \right] - f_N \frac{1-e^2}{e} \sqrt{\frac{a}{\mu}} \left(1 + \frac{r}{p} \right) \sin \theta \quad (\text{B.6})$$

Furthermore, realizing that:

$$p = a(1 - e^2) \quad (\text{B.7})$$

$$H = \sqrt{\mu p} \quad (\text{B.8})$$

$$\dot{r} = V \sin \gamma = \frac{\mu}{H} e \sin \theta \quad (\text{B.9})$$

$$r \cos \theta = a \cos E - ae \quad (\text{B.10})$$

$$u = \omega + \theta \quad (\text{B.11})$$

The velocity in a circular orbit is calculated using:

$$V = \sqrt{\frac{\mu}{a}} \quad (\text{B.12})$$

Variation of a

Combining Equations B.7 - B.12 with Equation B.1 it follows:

$$\frac{da}{dt} = \frac{2a}{V} \left[f_S \sin \gamma + f_N \cos \gamma \right] \quad (\text{B.13})$$

However, for circular orbits $\gamma = 0$, resulting in:

$$\frac{da}{dt} = \frac{2af_N}{V} \quad (\text{B.14})$$

Variation of e

Knowing that $e = 0$ it follows from Equation B.10 that $E = \theta$. This results in:

$$\frac{de}{dt} = \frac{f_S \sin \theta + 2f_N \cos \theta}{V} \quad (\text{B.15})$$

Variation of i

Realizing that $a/\sqrt{\mu a} = 1/V$ leads to:

$$\frac{di}{dt} = \frac{f_W \cos(\omega + \theta)}{V} \quad (\text{B.16})$$

Variation of Ω

$r/\sqrt{\mu p} = 1/V$. For small inclinations, one can assume: $\sin i \approx i$. Equation B.5 reduces then to:

$$\frac{d\Omega}{dt} = \frac{f_W \sin(\omega + \theta)}{Vi} \quad (\text{B.17})$$

Variation of ω

Using the geometrical relations:

$$\cot i = \frac{1}{\tan i} = \frac{\cos i}{\sin i} \quad (\text{B.18})$$

and again using the small angle approximation:

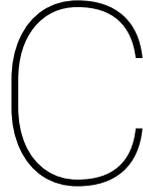
$$\cot i \approx \frac{1}{i} \quad (\text{B.19})$$

This results in:

$$\frac{d\omega}{dt} = -\frac{1}{V} \left[\frac{f_W \sin(\omega + \theta)}{i} + \frac{1}{e} (f_S \cos \theta - 2f_N \sin \theta) \right] \quad (\text{B.20})$$

which can be written as

$$\frac{d\omega}{dt} = -\frac{d\Omega}{dt} + \frac{1}{eV} (f_N \sin \theta - f_S \cos \theta) \quad (\text{B.21})$$



SGP4-LT Version 1

In this appendix the initial version of the SGP4 algorithm with low-thrust module is briefly explained and its results are discussed.

The initial version of SGP4-LT uses the osculating elements, converted from the output state vector, as input for the low-thrust module. It should be noted that this version is not the iterative method as is been used in the final version. It was known beforehand that osculating elements are not equal to mean elements, but it was nonetheless investigated if good results could be found.

An issue arises when using Edelbaum in the middle of the SGP4 algorithm. During the SGP4 algorithm the Brouwer convention of the elements is used which leads to an incompatible semi-major axis a :

To obtain the semi-major axis, the mean motion in the "Kozai" convention, n_0 in revolutions per day, is needed. At the initialization of SGP4, the Kozai mean motion is found in this form in the TLE data and converted to the "Brouwer" convention using the following equations (Hoots et al., 2004b):

$$a_1 = \left(\frac{k_e}{n_0} \right)^{2/3} \quad (C.1)$$

$$\delta_1 = \frac{3}{2} \frac{k_2}{a_1^2} \frac{3 \cos^2 i_0 - 1}{(1 - e_0^2)^{3/2}} \quad (C.2)$$

$$a_2 = a_1 \left(1 - \frac{1}{3} \delta_1 - \delta_1^2 - \frac{134}{81} \delta_1^3 \right) \quad (C.3)$$

$$\delta_0 = \frac{3}{2} \frac{k_2}{a_2^2} \frac{3 \cos^2 i_0 - 1}{(1 - e_0^2)^{3/2}} \quad (C.4)$$

$$n_0'' = \frac{n_0}{1 + \delta_0} \quad (C.5)$$

$$a_0'' = \left(\frac{k_e}{n_0''} \right)^{2/3} \quad (C.6)$$

using

$$k_2 = \frac{1}{2} J_2 a_E^2 \quad [\text{units of (Earth radii)}^2]$$

$$J_2 = 1.082616e - 3$$

$$k_e = \sqrt{GM} = 0.0743669161 \quad [\text{units of (Earth radii)}^{1.5}/\text{minute}]$$

G = universal gravitational constant

M = mass of the Earth

a_E = equatorial radius of the Earth

These equations lead to a value for the semi-major axis in the Brouwer convention (a_0'') which is used throughout the SGP4 algorithm. It is only at the end of SGP4 that a conversion is made to an state vector containing

x , y and z . As explained above, the updated semi-major axis is required for the addition of the low-thrust disturbance, before the long period periodics are applied. This requires an semi-major axis in the classical Kepler convention. To obtain this semi-major axis, a , first the mean motion in Kozai convention (n_0) must be recovered which is carried out by solving Equation C.7 for n_0 :

$$Y \equiv \left(\frac{k_e}{n_0''(n_0)} \right)^{2/3} - a_0'' = 0 \quad (\text{C.7})$$

where n_0'' is a function of n_0 . For a range of n_0'' values the resulting curve is plotted in Figure C.1. This curve represents the data of the EutelSat I-F1 at epoch June 25, with the known semi-major axis in the Brouwer's convention of 6.7. Finding the root of this equation results in the corresponding mean motion n_0'' . From the TLE data of this satellite, the corresponding mean motion is 0.98 rev/day.

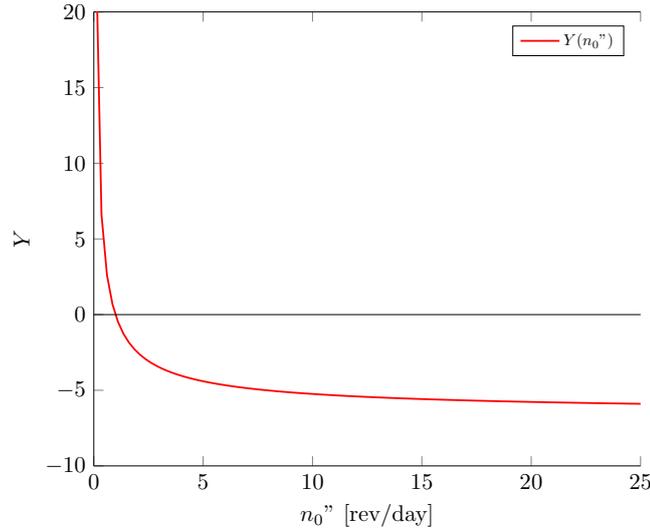


Figure C.1: Curve Y which requires solving at $Y = 0$, obtained by using SGP4 for satellite EutelSat I-F1 at epoch June 25, 2006

This equation cannot be solved analytically and is therefore solved numerically. When solved for n_0 , the semi-major axis can be retrieved using Equation C.8:

$$a = \left[\mu \left(\frac{43200}{\pi n_0} \right)^2 \right]^{1/3} \quad (\text{C.8})$$

SGP4 gives at this point in the algorithm updated values for the elements after time step h for t_{n+1} . Let's call these elements $a_{SGP4(n+1)}$ and $i_{SGP4(n+1)}$, the only two elements that change in Edelbaum's analytical method. The Edelbaum algorithm runs now for the values a_n , i_n , and calculates the corresponding $a_{EB(n+1)}$ and $i_{EB(n+1)}$. This implies that Edelbaum completely renews for every time step taken: it takes a_n and i_n as it were the orbit parameters of the initial orbit and calculates a new transfer orbit at each time step.

Using this approach results at t_{n+1} in two different values, an SGP4 value and an Edelbaum value, for a and i . The new, updated values for a_{t+1} and i_{t+1} then become:

$$a_{n+1} = a_{SGP4(n+1)} + (a_{EB(n+1)} - a_n) \quad (\text{C.9})$$

$$i_{n+1} = i_{SGP4(n+1)} + (i_{EB(n+1)} - i_n) \quad (\text{C.10})$$

The updated semi-major axis a_{n+1} is converted back to the Brouwer's convention, by calculating the mean motion in [rev/day] and using Equations C.1-C.6. The updated elements are fed to the remaining part of the SGP4 algorithm and are stored so that they can be used again at the next epoch.

The results obtained using this version showed interesting behaviour: at time steps smaller than 10 minutes,

the spacecraft would crash into Earth, since the mean motion, which was obtained as osculating element, was too large compared to its correct mean value and per time epoch this mean motion became larger until it corresponded to a semi-major axis equal to the radius of the Earth.

Using larger time steps, the change due to the low-thrust module was large enough to counteract this increase in mean motion and eventually the target orbit was reached. This was however only possible using a ten-times larger amount of thrust compared to the thrust required when using Edelbaum's method solely. It was therefore decided that using this approach will not result in correct results. Mean elements have to be used as input for SGP4.



Used Two Line Elements Sets

TLE used in Chapter 7

0 INTELSAT 4A-F1

1 8330U 75091A 18296.31435785 -.00000009 +00000-0 +00000-0 0 9998
2 8330 013.3105 330.4179 0003125 221.8469 134.8761 00.99938176093542

0 SL-16 R/B

1 23088U 94023B 18296.54081114 +.00000138 +00000-0 +96541-4 0 9990
2 23088 071.0008 326.5745 0001410 034.1053 326.0160 14.14674283265048

0 CASE1

1 88888U 80275.98708465 .00073094 13844-3 66816-4 0 8
2 88888 72.8435 115.9689 0086731 52.6988 110.5714 16.05824518 105

0 CASE1b

1 88888U 80275.98708465 .00073094 13844-3 66816-4 0 8
2 88888 72.8435 115.9689 206580 160.6988 1.5714 11.05824518 105

0 CASE2

1 11801U 80230.29629788 .01431103 00000-0 14311-1
2 11801 46.7916 230.4354 7318036 47.4722 10.4117 2.28537848

0 CASE2b

1 11801U 80230.29629788 .01431103 00000-0 14311-1
2 11801 46.7916 230.4354 57356 29.4722 30.4117 5.28537848

0 TDRS4-1994

1 19883U 89021 B 94002.42255033 -.00000242 +00000-0 +00000-0 0 9997
2 19883 000.0414 169.8057 0000818 191.2596 311.8029 01.00267832087807

0 CRRES

1 20712U 90065A 18287.62108234 -.00000177 00000-0 20695-3 0 9995
2 20712 18.0992 66.8910 7102958 217.1459 61.7437 2.46160481240381

0 SPOT1

1 16613U 86019A 86220.84554216 .00000101 00000-0 51859-4 0 2036
2 16613 98.7227 294.5877 0001355 138.2330 221.8534 14.20028330 24082

0 MIR

1 16609U 86017A 92191.82959504 .00010233 00000-0 14927-3 0 4145

2 16609 51.5978 133.4246 0016682 265.2458 94.6703 15.56108601365801

0 KVANT1

1 17845U 87030A 87132.70786077 .00009332 00000-0 72819-4 0 759

2 17845 51.6303 339.4435 0016920 342.4341 17.6028 15.71553631 6732

0 EUVE

1 21987U 92031A 94152.24458228 .00000959 00000-0 40400-4 0 3166

2 21987 28.4320 160.7323 0010443 162.5892 197.5064 15.18546497109973

0 PROGRESSM17

1 22588U 93019A 94001.63076894 .00153542 20172-4 30443-3 0 4217

2 22588 51.6202 291.8750 0084012 88.0951 273.0021 15.93214724 43191

0 HST

1 20580U 90037B 18289.13114583 .00000536 00000-0 21861-4 0 9994

2 20580 28.4705 35.5559 0002798 6.7531 16.9647 15.09112650363750

TLE used in Chapter 8

0 ATLASII

1 25544U 98067A 18221.28032416 .00001255 00000-0 00000-0 0 9993

2 25544 27.0000 114.1026 0000004 41.9073 000.0000 15.96239457400234

0 SUPERBIRD-C

1 25544U 98067A 18221.28032416 .00001255 00000-0 00000-0 0 9993

2 25544 27.5500 114.1026 0000004 41.9073 000.0000 16.27278683673710

0 GALAXY4R

1 25544U 98067A 18221.28032416 .00001255 00000-0 00000-0 0 9993

2 25544 06.9980 114.1026 0000004 41.9073 000.0000 16.27278768388088

0 DeltaII

1 25544U 98067A 18221.28032416 .00001255 00000-0 00000-0 0 9993

2 25544 28.5000 114.1026 0000004 41.9073 000.0000 15.97677942889040

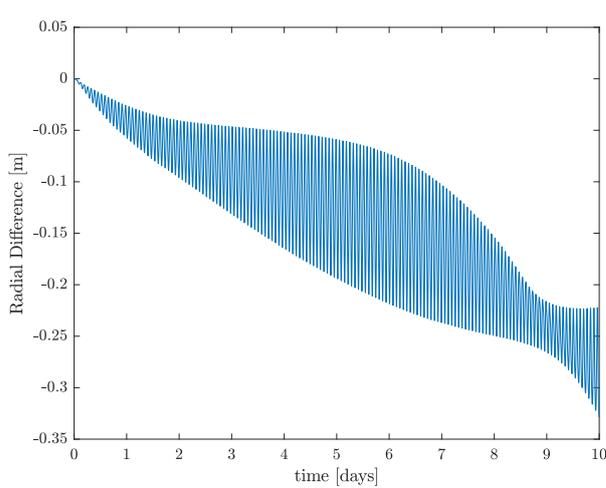
0 GOCE

1 34602U 09013A 12002.47825061 .00043121 27153-5 60428-4 0 9994

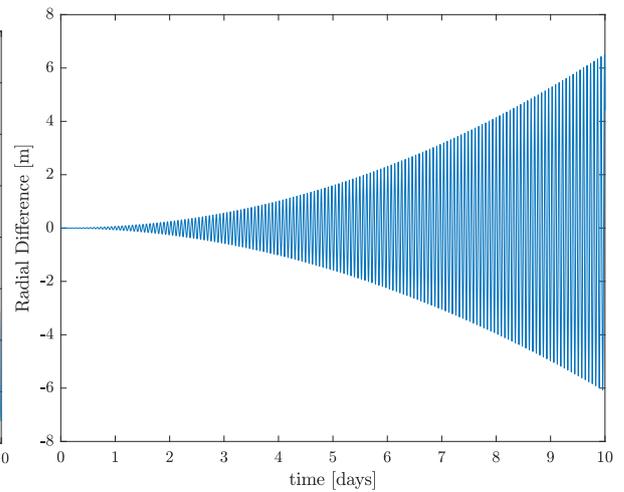
2 34602 096.6138 025.9009 0006012 122.9895 237.3028 16.06005362163745

E

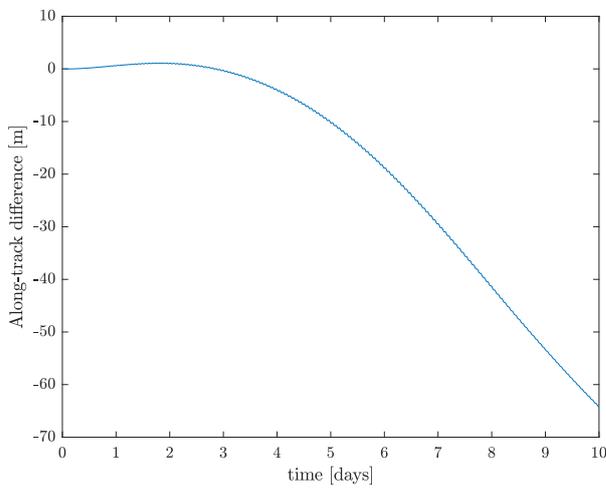
Results of validation of iterative SGP4



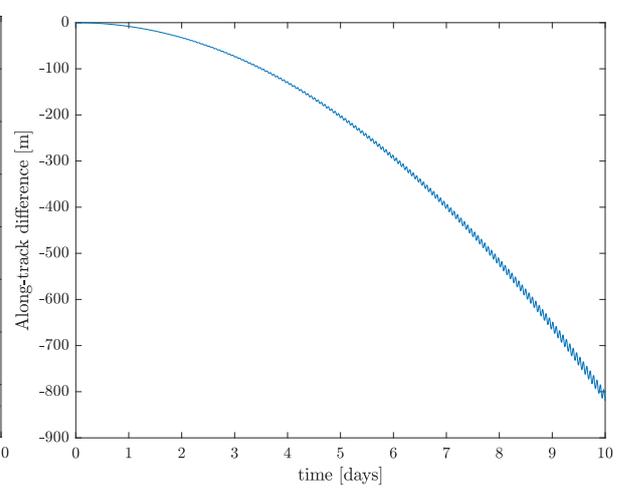
(a) Radial difference for near-Earth satellite EUVE.



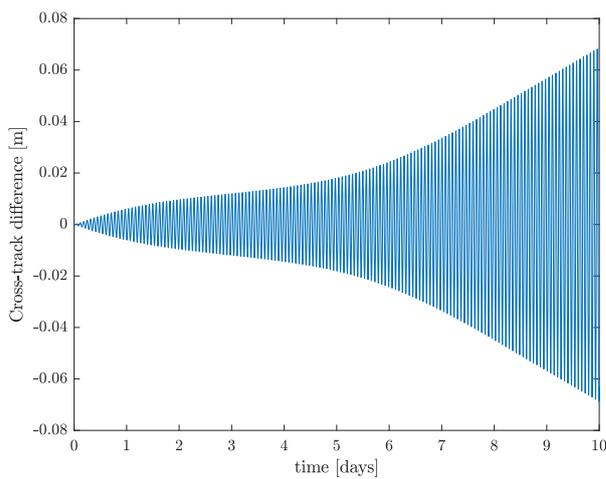
(b) Radial difference for near Earth satellite MIR.



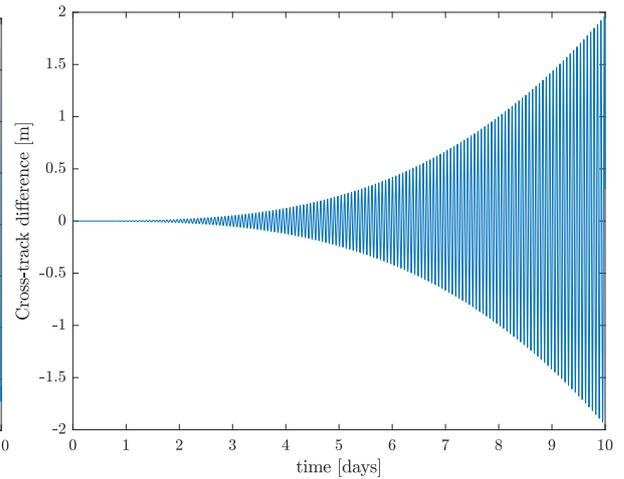
(c) Along-track difference for near-Earth satellite EUVE.



(d) Along-track difference for near-Earth satellite MIR.

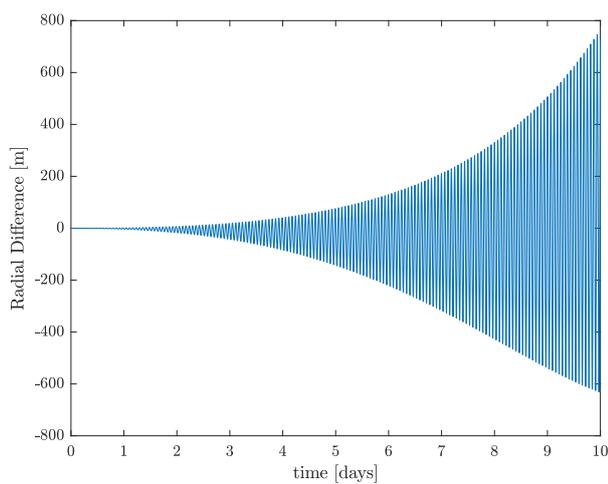


(e) Cross-track difference for near-Earth satellite EUVE.

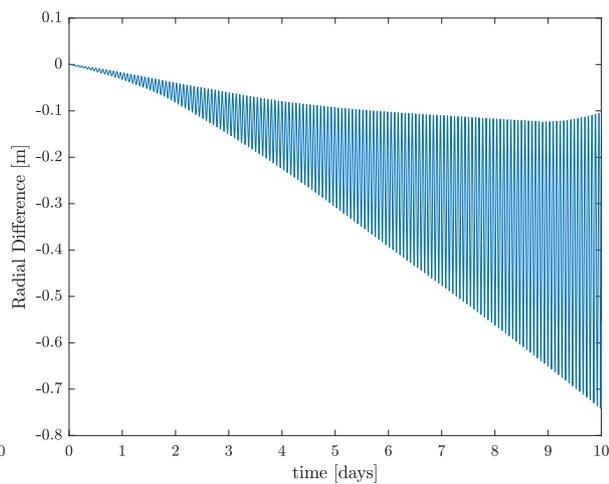


(f) Cross-track difference for near-Earth satellite MIR.

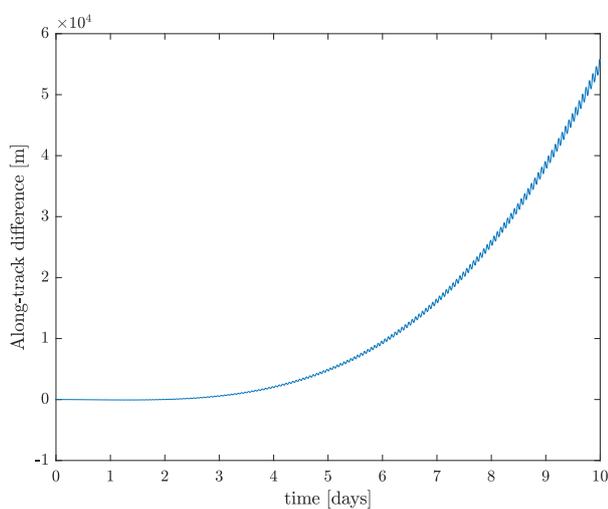
Figure E.1: Differences between propagation of the iterative SGP4 and the original SGP4 for radial, along-track and cross-track directions, for respectively a near-Earth and a deep-space satellite.



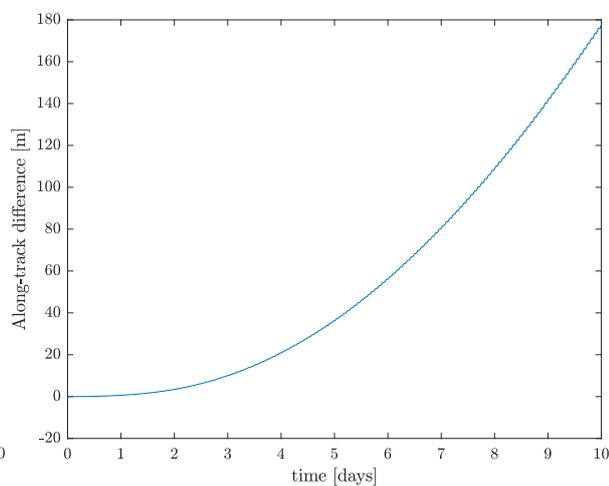
(a) Radial difference for near-Earth satellite PROGRESS-M17.



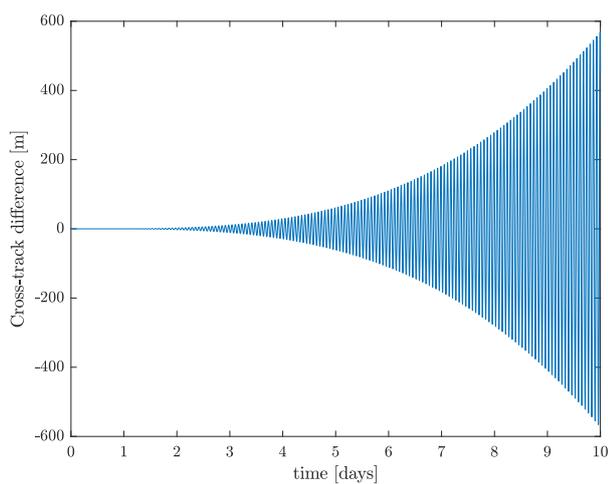
(b) Radial difference for near Earth satellite HST.



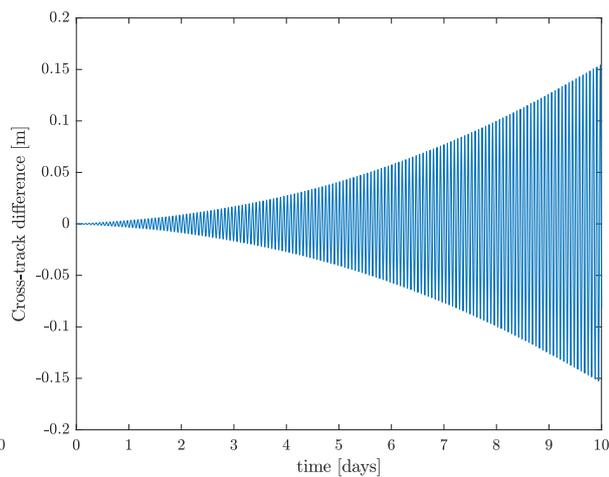
(c) Along-track difference for near-Earth satellite PROGRESS-M17.



(d) Along-track difference for near-Earth satellite HST.



(e) Cross-track difference for near-Earth satellite PROGRESS-M17.



(f) Cross-track difference for near-Earth satellite HST.

Figure E.2: Differences between propagation of the iterative SGP4 and the original SGP4 for radial, along-track and cross-track directions, for respectively a near-Earth and a deep-space satellite.

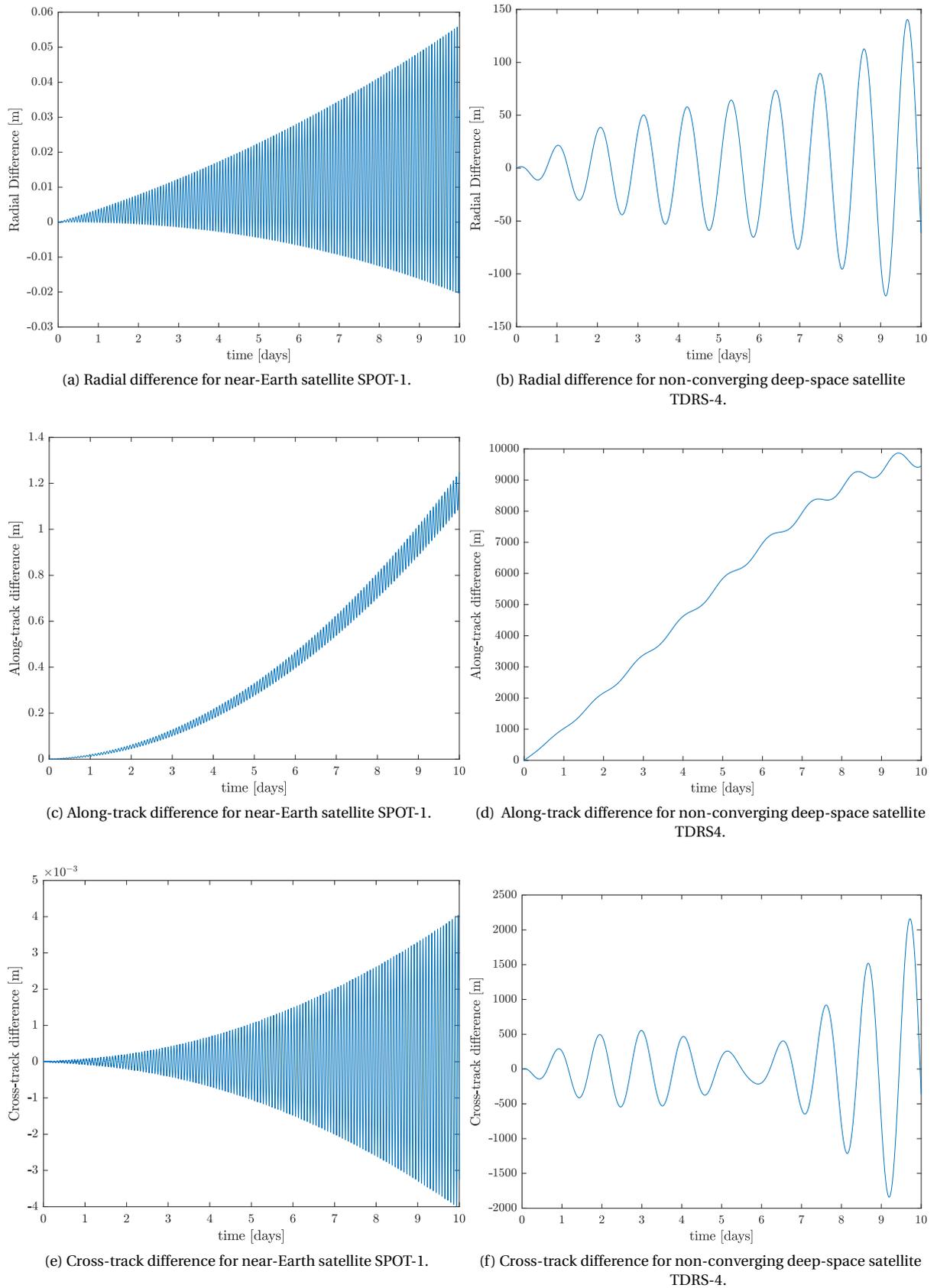


Figure E.3: Differences between propagation of the iterative SGP4 and the original SGP4 for radial, along-track and cross-track directions, for respectively a near-Earth and a deep-space satellite.

Bibliography

- Dwight E. Andersen. Computing NORAD Mean Orbital Elements from a State Vector. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, December 1994.
- M. Ansele. All-electric Propulsion Satellites. Website, viewed on 17-11-2018, June 2017. URL <https://www.airbus.com/newsroom/news/en/2017/06/all-electric-propulsion-satellites.html>.
- J. Benkhoff et al. BepiColombo - Comprehensive Exploration of Mercury: Mission Overview and Science Goals. *Planetary and Space Science*, 58:2–20, 2010.
- R. A. Broucke and P.J. Cefola. On The Equinoctial Orbit Elements. *Celestial Mechanics*, (5):303–310, 1972.
- J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley and Sons, Ltd, 2008.
- D. Conte and D. Spencer. Targeting the Martian moons via Direct Insertion into Mars' Orbit. *Astrodynamic Specialist Conference, Vail, United States of America*, 2015.
- J. Cornelisse, H. Schöyer, and K. Wakker. *Rocket Propulsion and Spaceflight Dynamics*. Pitman, 1979.
- G. Der and R. Danchick. Conversion of Osculating Orbital Elements. *Flight Mechanics Estimation Theory Symposium, Greenbelt, USA*, 3333:317–332, 1996.
- Todd Ely. Transforming Mean and Osculating Elements Using Numerical Methods. *AAS Astrodynamics Specialist Conference, Pittsburgh, Pennsylvania, August*, 2013.
- D. Estublier, G. Saccoccia, and J. Gonzalez del Almo. Electric Propulsion on SMART-1. ESA Bulletin, Directorate of Technical and Quality Management ESTEC, Noordwijk The Netherlands, February 2007.
- B. Foing et al. SMART-1 Mission Overview from Launch, Lunar Orbit to Impact. *Lunar and Planetary Science*, 38:1915–1916, 2007.
- S. Franzen. Thermal radiation. Lecture Notes CH437, NC State University, viewed on 26-06-2017, 2016. URL http://www4.ncsu.edu/~franzen/public_html/CH437/lec1/pdf/earth_T.pdf.
- P. Gjerlov. NEUCC. *IFIP WG 9.7 Conference, Copenhagen, Denmark, August 2014*, 4:109–115, 2015.
- D. Goebel, J. Polk, I. Sandler, I. Mikellides, J. Brophy, W. Tighe, and K. Chien. Evaluation of 25-cm XIPS Thruster Life for Deep Space Mission Applications. *31st International Electric Propulsion Conference, Ann Arbor, United States of America*, 2009.
- D. Gondelach. A Hodographic-Shaping Method for Low-Thrust Trajectory Design. Master thesis, Delft University of Technology, 2012.
- A. L. Herman and D. B. Spencer. Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods. *Journal of Guidance, Control, and Dynamics*, 25(1):40–47, January 2002.
- D. A. Herman, G. C. Soulas, and M. J. Patterson. Performance Evaluation of the Prototype Model NEXT Ion Thruster. *43rd Joint Propulsion Conference, Cincinnati, United States, July 8-11*, 2007.
- F. Hoots, P. Schumacher Jr., and R. Glover. History of Analytical Orbit Modeling in the U.S. Space Surveillance System. *Journal of Guidance, Control, and Dynamics*, 27(2):174–185, 2004a.
- FR. Hoots and R.L. Roehrich. Spacetrack Report No. 3. Technical report, NORAD, 1988. URL <https://www.celestrak.com/NORAD/documentation/spacetrk.pdf>.
- FR. Hoots, Jr P. W. Schumacher, and R. A. Glover. A History of Analytical Orbit Modelling in the United States Space Surveillance System. *Journal of Guidance, Control, and Dynamics*, 27(2):174–185, 2004b.

- R. Jahn and E. Choueiri. *Encyclopedia of Physical Science and Technology*, volume 5. Academic Press, third edition, 2001.
- R. G. Jahn. *Physics of Electric Propulsion*. Dover Publications, INC, 2006.
- J.G.P. de Jong. Analytical Low-Thrust Trajectory Design. Literature Study, Delft Univeristy of Technology, October 2017.
- J. A. Kechichian. Reformulation of Edelbaum's Low-Thrust Transfer Problem Using Optimal Control Theory. *Journal of Guidance, Control, and Dynamics*, 20(5):988–994, 1997.
- Dr. T.S. Kelso. Celestrak. Website, viewed on 13/11/2018, September 2018a. URL <https://www.celestrak.com/>.
- Dr. T.S. Kelso. Space surveillance. Website, viewed on 28/10/2018, June 2018b. URL <https://www.celestrak.com/columns/v04n01/>.
- T.S. Kelso. Validation of SGP4 and IS-GPS-200D against GPS Precision Ephemerides. *17th AAS/AIAA Space Flight Mechanics Conference, Sedona, Arizona, January 28- February 1, 2007*.
- B. Kennedy, S. Bhaskaran, J. Riedel, and M. Wang. Deep Space 1 Navigation: Extended Missions. Jet Propulsion Laboratory California Institute of Technology, Pasadena California, September 2003.
- V. Khayms, K. Kannenberg, B. Meyer, and L. Werthman. Overview of Hall Current Thruster Integration Activities at Lockheed Martin Space Systems Company. *Electric Rocket Propulsion Society*, 2001.
- C. A. Kleuver. Using Edelbaum's Method to Compute Low-Thrust Transfers with Earth-Shadow Eclipses. *Journal of Guidance, Control, and Dynamics*, 34(1):300–303, 2011.
- Chen L., Bai XZ, Liang YG., and Li KB. *Orbital Prediction Error Propagation of Space Objects*. Springer, Singapore, 2017.
- B.S Lee. NORAD TLE Conversion from Osculating Orbital Element. *Journal of Astronomy and Space Sciences*, 19(4):395–402, November 2002.
- J. Leloux. Filtering Techniques for Orbital Debris Conjunction Analysis. Master's thesis, Delft University of Technology, 2012.
- Wilfried Ley, Klaus Wittmann, and Willi Hallman, editors. *Handbook of Space Technology*. Wiley, 2008.
- R. Litchford. High Power Flex-Propellant Arcjet Performance. *American Institute of Aeronautics and Astronautics*, 2012.
- Mr. McCormack. Space Handbook: Astronautics and its Applications. website, viewed on 15/10/2018, 2007. URL <https://history.nasa.gov/conghand/propelnt.htm>.
- D. Milligan, D. Gestal, O. Camino, P. Pardo-Voss, et al. SMART-1 Electric Propulsion Operational Experience. In *International Electric Propulsion Conference, Princeton, United States of America*, 2005.
- O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2001.
- NASA. Magnetoplasmadynamic thrusters. website, viewed on 20/07/2017, 2010. URL <https://www.nasa.gov/centers/glenn/about/fs22grc.html>.
- NASA. NASA's Juno Spacecraft Breaks Solar Power Distance Record. Website, viewed on 20-07-2017, 2016. URL <https://www.jpl.nasa.gov/news/news.php?feature=4818>.
- B. Neta and D. Vallado. On Satellite Ubra/Penumbra Entry and Exit Positions. *Journal of the Astronautical Sciences*, 46(1):91–104, 1998.
- R. Noomen. AE4878 Space Mission Design: Integrators. Lecture Notes, Delft University of Technology, December 2016.

- W. L. Oberkampf, T. G. Trucano, and C. Hirsch. Verification, validation, and predictive capability in computational engineering and physics. Foundations for Verification and validation in the 21st Century Workshop, Laurel, Maryland, October 2002.
- M. Poole and M. Ho. Boeing Low-Thrust Geosynchronous Transfer Mission Experience. *20th International Symposium on Space Flight Dynamics, Annapolis, United States of America, 2007.*
- W. Press. *Numerical Recipes: The Art of Scientific Computing*. Cambridge, 2007.
- M. Rayman. Results from the Deep Space 1 Technology Validation Mission. *50th International Astronautical Congress, 1999, Amsterdam, The Netherlands, October 1999.*
- M. Rayman and S. Williams. Design of the First Interplanetary Solar Electric Propulsion Mission. *Journal of Spacecraft and Rockets*, 39(4):589–595, 2002.
- M. Rayman, P. Chadbourne, J. Culwell, and S. Williams. MISSION Design for Deep Space 1: A Low-Thrust Technology Validation Mission. *Acta Astronautica*, 45(4):381–388, 1999.
- John Redden. *Advanced Algebra v.1.0*. Creative Commons, 2012.
- B. Reid. *The Influence of Neutral Flow Rate in the Operation of Hall Thrusters*. PhD thesis, University of Michigan, 2009.
- A. Rocchi and R. Jehn. Low-thrust navigation tools at esoc mission analysis section. *6th International Conference on Astrodynamics Tools and Techniques, 2016, Darmstadt, 2016.*
- G. Schmidt, D. Manzella, et al. Radioisotope electric propulsion (REP): A near-term approach to nuclear propulsion. *Acta Astronautica*, 66:501–507, 2010.
- D. Shen, B. Jia, G. Chen, K. Pham, and E. Blasch. Space based sensor management strategies based on informational uncertainty pursuit-evasion games. *Aerospace and Electronics Conference, 2015, Dayton, United States of America, 2015.*
- C. Steiger, E. Montagon, and A. Accomazzo. Flight Operations Preparation for the BepiColombo Mission to Mercury: Concepts and Challenges. *14th International Conference on Space Operations, Daejeon, Korea, 2016.*
- J. Szabo and Y. Azziz. Characterization of a High Specific Impulse Xenon Hall Effect Thruster. *Electric Propulsion Conference, Princeton University, October 2005.*
- D. Vallado, P. Crawford, R. Hujsak, and T. Kelso. Revisiting Spacetrack Report 3: Rev2. *AIAA*, 2006.
- K.F. Wakker. *AE4874-I Astrodynamics*. Lecture Notes, Delft University of Technology, 2010.
- M. Walker, B. Ireland, and J. Owens. A set of Modified Equinoctial Orbit Elements. *Celestial Mechanics*, 36: 409–419, August 1985.
- N. Wallace, P. Jameson, and C. Saunders et al. The GOCE Ion Propulsion Assembly - Lessons Learnt from the First 22 Months of Flight Operations. *The 32nd International Electric Propulsion Conference, Wiesbaden, Germany, September 11-15, 2011.*
- H. G. Walter. Conversion of Osculating Orbital Elements into Mean Elements. *The Astronomical Journal*, 72 (8):994–997, 1967.
- D. Yarnoz, R. Jehn, and M. Croon. Interplanetary Navigation Along the Low-Thrust Trajectory of Bepi-Colombo. *Acta Astronautica*, 59:284–293, 2006.