Orbital lifetime predictions

An assessment of model-based ballistic coefficient estimations and adjustment for temporal drag coefficient variations

M.R. Haneveer

MSc Thesis Aerospace Engineering

TU Delft

Orbital lifetime predictions

An assessment of model-based ballistic coefficient estimations and adjustment for temporal drag coefficient variations

by

M.R. Haneveer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday June 1, 2017 at 14:00 PM.

Student number:	4077334	
Project duration:	September 1, 2016 – June 1, 2017	
Thesis committee:	Dr. ir. E. N. Doornbos,	TU Delft, supervisor
	Dr. ir. E. J. O. Schrama,	TU Delft
	ir. K. J. Cowan MBA	TU Delft

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

Summary

Objects in Low Earth Orbit (LEO) experience low levels of drag due to the interaction with the outer layers of Earth's atmosphere. The atmospheric drag reduces the velocity of the object, resulting in a gradual decrease in altitude. With each decayed kilometer the object enters denser portions of the atmosphere accelerating the orbit decay until eventually the object cannot sustain a stable orbit anymore and either crashes onto Earth's surface or burns up in its atmosphere. The capability of predicting the time an object stays in orbit, whether that object is space junk or a satellite, allows for an estimate of its orbital lifetime - an estimate satellite operators work with to schedule science missions and commercial services, as well as use to prove compliance with international agreements stating no passively controlled object is to orbit in LEO longer than 25 years.

An orbital lifetime prediction rests primarily on three pillars - numerical orbit propagation, thermospheric density prediction, and the ballistic coefficient of the spacecraft. The latter is an indication of the interaction the spacecraft has with local atmosphere and how heavily it is influenced by the atmospheric drag. Linearly dependent on the spacecraft's mass and reference area, as well as the local drag coefficient, the ballistic coefficient is commonly well-known for larger spacecraft. For micro- and pico satellites an accurate ballistic coefficient is often not known, as well as for most space debris. This thesis discusses a method capable of estimating an arbitrary 1U CubeSat's ballistic coefficient by analyzing both its historical Two-Line Element data and the density it experienced along its trajectory according to the empirical thermospheric density model NRLMSISE-00. In parallel the same is done for a spherical satellite which already has a preliminary well-known ballistic coefficient, given the two objects are experiencing near-similar atmospheric conditions. Any density error associated with NRLMSISE-00 along that trajectory can be largely mitigated, resulting in a model-dependent estimated ballistic coefficient β_E. This procedure can be iterated upon with a batch of CubeSats, leading to a single method capable of estimating the ballistic coefficient of a large group of target satellites. In this research, around 60 CubeSats have been subjected to this method.

Literature commonly addresses the drag coefficient used within the ballistic coefficient to be constant at $C_D = 2.2$ for both CubeSats and spherical satellites. This would imply that upon computing β_E its value would remain constant throughout its lifetime given no mass is expelled from the spacecraft. However, an object's drag coefficient is dependent on the ambient gas composition, as well as its geometry and freestream velocity and can thus not be considered constant. This thesis additionally implements a method capable of adjusting an epoch-associated β_E to account for changing ambient gas composition over time and altitude, for which the output of NRLMSISE-00 is used. Within the analytical C_D computation, based on Sentman's rarefied aerodynamic equations, the energy accommodation coefficient α plays an important role, and yet again in most literature this is considered constant at values ranging between 0.8 and 1.0. Langmuir's adsorption model however considers α variable and a function of the presence of local atomic oxygen. Second to β_E being adjusted for the epoch-associated C_D, this thesis also investigates the impact a constant $\alpha = 0.8$ and the effect that incorporation of Langmuir's isotherm has on adjusting β_E to $\beta_{C_{D,\alpha=0.8}}$ and $\beta_{C_D Langmuir}$ respectively.

A distinction has been made between CubeSats at altitude higher than 500 km (in-orbit group) and CubeSats which already re-entered (re-entry group). These objects have had their three discussed β computed and evaluated with a reference and a verification orbit based on their historical TLE. The re-entry group demonstrated improvements of 35.65% for its β_E values compared to a standard β_0 of 0.03125, and have shown Langmuir's adsorption model becomes unsuited for orbits lower than 300 km. Two out of three in-orbit groups showed an improvement of 5% with β_E, yet this rose to 10% with $\beta_{C_D Langmuir}$, thus showing for higher orbits Langmuir's adsorption model does perform better instead of constant α. Furthermore, the groups highlighted the sensitivity of the iteration process and the criteria of when two objects experience near-identical atmospheric conditions. Overall, this thesis demonstrated the possibility of a global ballistic coefficient estimation model which can be extended with other satellites and/or space debris. The adjustment of the epoch-associated β_E to account for the varying C_D significantly improved the decay prediction for higher orbits, though for objects close to re-entry a deeper investigation into the computation of α would have to be considered.

Abstract

The ballistic coefficient β of a satellite indicates the influence atmospheric drag has on the orbital decay of said satellite. The method developed in this study estimates the ballistic coefficients of 1U CubeSats by analyzing both its historical TLE data and the density it experienced along its trajectory according to the empirical density model NRLMSISE-00. In parallel the same is done for a spherical satellite with an already known ballistic coefficient, given the object is experiencing near-similar atmospheric conditions as that of the CubeSat. Any density error associated with NRLMSISE-00 along that trajectory is consecutively largely mitigated, resulting in an improved model-dependent estimated ballistic coefficient for the CubeSat. This procedure is iterated upon with a batch of 60+ CubeSats, leading to a single procedure capable of estimating the ballistic coefficient of a large group of objects.

The estimated β are consecutively adjusted for the epoch- and altitude-dependent drag coefficient, performed analytically as a function of ambient gas-composition and spacecraft geometry. Additionally, an investigation in the computation of the energy accommodation coefficient α, an essential variable for the C_D computation, is performed, assessing whether a constant $\alpha = 0.8$ can be assumed or if Langmuir's adsorption model based on the ambient presence of atomic oxygen would not be more suited.

When subjecting the estimated and adjusted β values to a propagator build in TUDAT, based on Runge-Kutta-Fehlberg 7(8), and comparing them to their actual orbits, prediction improvements of up to 40% are observed compared to regular non-adjusted β values for CubeSats re-entering the atmosphere. For CubeSats orbiting Earth above 500 km, improvements between 5% and 15% are observed. The research furthermore highlighted the sensitivity of the iterative process and the orbital element selection criteria through which satellite pairs are assumed to have near-similar atmospheric conditions – the tighter the selection criteria the better the quality of the β estimations were, though at the cost of fewer β estimations to occur.

Overall, it can be concluded that the β estimation method can be applied on a scale covering multiple satellites, potentially growing to a global ballistic coefficient estimation model which could also include space debris. Improvements to orbital lifetime predictions are primarily seen in the lower parts of LEO. Furthermore, the adjustment for the variable drag coefficient proved most useful for the CubeSats above 500 km, given Langmuir's adsorption model was used. For the lower orbits, Langmuir's adsorption model lost its accuracy and an adaptation of Langmuir's model which considers an object's historical orbit data is suggested.

Preface

Concluding this thesis is the pinnacle of 7 years of my education at Delft, University of Technology. Starting in September 2010 with the Bachelor Aerospace Engineering, I quickly found my enthusiasm towards anything that had to do with rockets and outer space. From almost day one I immediately joined the Delft Aerospace Rocket Engineering (DARE) student team, a voluntarily student organization working on in-house developed rockets. A couple of years of building, launching, and yes, also crashing rockets was an unforgettable experience. This experience in combination with my Bachelor thesis work revolving around a Phase-A study of an Earth Observation satellite made me decided to continue my academic career within the Spaceflight Master Track at the Aerospace Faculty.

Initially following the Space Engineering specialization, I remember my most favored course being an elective course from the Space Exploration specialization - Mission Geometry and Orbit Design taught by Ir. R. Noomen to be precise. Though still enthusiastic about space engineering, I was more attracted to conclude my Master with a thesis subject from the Space Exploration department. After numerous talks with R. Noomen on potential graduation topics, the idea of working on orbital lifetime predictions was the one that excited me the most and made me decide to formally switch specialization, and thus graduate from the Space Exploration department under the supervision of assistant professor Dr. Ir. Eelco Doornbos, the faculty's expert in thermospheric research.

Straight from the get-go I was made aware of the complexity of orbital lifetime predictions, and the huge amount of uncertainties associated with it. The two month literature study taught me a great deal, and it made me appreciate my topic choice more and more. In the end I decided to contribute to the body of knowledge by researching ballistic coefficient estimations and its adjustments for the temporal drag coefficient, an aspect I knew little about barely a year ago but of which I can now confidently say I have a thorough understanding of. Second to the gained knowledge, the thesis research enabled me to become more disciplined and structured when conducting an academic experiment - an asset I shall carry with me indefinitely.

During the entire graduation process, from literature study to this final thesis report, Eelco's guidance ensured that the research was always focused towards the final goal and that I wouldn't stray too far from that path whenever I wanted to investigate seemingly interesting yet unrelated things. Without his insights and continuous availability throughout the project this research would not have been where it is now - for this I'd like to thank him. Furthermore, I would like to thank Dr. Ir. Dominic Dirkx for his tremendous help with the TUDAT library. Whenever a TUDAT-related programming issue surfaced, Dominic managed to understandably break-down the problem, enabling me to use my time more efficiently.

Last but not least I'm thanking my parents, for having at all times supported me on my journey in becoming an aerospace engineer. It wasn't always an easy road, but no matter what I was facing they were there for me. For that I'm forever grateful.

M.R. Haneveer
Delft, June 2017

Contents

1

Introduction

This document presents an investigation into a ballistic coefficient (β) estimation procedure, as well as the influence a variable drag coefficient has on β component. The research is directly related to orbital lifetime predictions, and the thereby associated topics of atmospheric densities and orbit propagation. The selection of this topic came to be when discussing potential thesis topics with a few professors from the Space Exploration department, where the prospect of graduating on orbital lifetimes caught my enthusiasm. The work presented in this document covers a total period of nine months, two of which have been dedicated to a literature review and the remaining seven on the actual research. With these nine months being concluded, a thesis defense and engineering exam are held on the 01-06-2017 to assess the academic quality of the conducted research, after which the author shall in all likelihood receive his Master of Science title.

To start things of, a brief introduction is given in the next two pages highlighting the importance of accurate lifetime predictions, on what specific branches within the body of knowledge this research has its focus on, and how this research is conducted. In chapter 2 and chapter 3 a deeper technical briefing is given on the theoretical background necessary for this research - though the rough lines and extend of this project should become apparent with solely this chapter. Concluding this section, the research question and accompanying sub-questions are stated, as well as a directive on to accomplish them.

1.1. Orbital lifetime predictions in the thermosphere

Though the space around Earth is huge, and there are many different orbits at various altitudes possible, there is always the small chance an object will collide with another. Such a collision, whether between space debris or an active satellite, can result in an escalating effect by generating more space debris and thereby increasing the risk of future collisions. This risk is especially present in the thermospheric region of Earth atmosphere, spanning an altitude range between 200 km and 2,000 km. In this Low Earth Orbit (LEO) region around 500[7] commercial, Earth Observation, military, and educational satellites orbit Earth are present, as well as mankind's current only space habitat; the International Space Station. Knowing accurately how long an object remains in orbit around Earth, whether an active satellite or space junk, results in a better insight on how much future space objects have to be removed to contain or de-escalate the risk on in-orbit collisions.

International agreements[36] have been put in place to push satellite operators to remove their satellite from crowded and commonly used orbits either by de-orbiting the satellite or by putting it in a graveyard orbit - an orbit generally a few hundred kilometer higher. Moreover, during the mission design phase, the customer might request for example 12 years minimum operational lifetime. As the designer, the worst-case scenario could ensure these 12 years in orbit, yet during nominal or optimal Solar conditions this lifetime could suddenly jump up to 15+ years. Obviously this is not a negative thing for the satellite operator - they can now use the satellite longer than intended (solely considering orbital lifetime, not functionality). However, if a satellite's lifetime suddenly exceeds the 25 years threshold, the mission design might have to be adjusted to include disposal of the satellite. A longer lifetime furthermore requires a new estimate on operational costs for that particular mission, as that would also increase if over time the satellite's lifetime proves to be longer than calculated. The opposite has an equal impact on the operators business model as well, as a shorter mission duration impacts the amount of science that can be conducted or the amount of commercial services that can be provided.

For mainly these reasons it is important to aim for an accurate orbital lifetime prediction - both for a better understanding towards space debris mitigation policies and for commercial purposes. However, as this chapter will clarify, orbital lifetime predictions are no straightforward analytical calculations. With the main contributor to orbital decay being atmospheric drag, and that drag being a function of atmospheric behavior and satellite characteristics there are many fluctuating parameters and uncertainties that need to be considered. Whether it is through reducing uncertainties in predicting future Solar activity, or estimating a better ballistic coefficient of the satellite as opposed to the β calculated initially by the satellite operator, efforts can and have to be made to decrease the gap between worst-case and optimal orbital lifetime prediction.

1.2. Research framework and objective

To address the stated problem, a literature review has been performed prior to this thesis research. The sole purpose of this review was to clarify the current state-of-the-art science on orbital lifetime predictions, and to identify in which areas a contribution could still be made towards this body of knowledge. Concluding this literature review, it was decided upon to extend the research performed by J.T. Emmert, P. Mehta, and B. Bowman, where the orbital decay of spherical objects and target satellites are analyzed to derive model-dependent ballistic coefficients for these target satellites. In short, a spherical 'parent' satellite and a target 'child' satellite are identified based on whether they are experiencing near-similar atmospheric conditions. Consecutively the matched satellites have their Two-Line-Element (TLE) data analyzed to, in corporation with a thermospheric density model, estimate a better β value for the child satellite. With this method the error coming from the empirical thermospheric density model, one of the main uncertainties in orbital lifetime predictions[10], can be largely mitigated, and by incorporating an iterative process the β values for a large group of satellites can be estimated simultaneously.

Additionally, the epoch-dependent estimated β values are being adjusted to account for temporal drag coefficient (C_D) variations - a variable upon which β is linearly related. Generally C_D is assumed constant with a value of 2.2 in orbital lifetime predictions, though according to the theory it would be better estimated as a variable. The computation of these drag coefficients are performed analytically as a function of satellite geometry and ambient gas composition, based on the work of M. Pilinski[41] and Sentman's aerodynamic equations. Within the theory of analytically computing a drag coefficient, a choice can be made to either keep the energy accommodation coefficient α constant or to let it be dependent on the presence of atomic oxygen in the ambient gas composition - the latter being referred to as Langmuir's adsorption model. Within this research, both the impact of a constant $\alpha = 0.8$ and Langmuir's isotherm are investigated.

This would lead to a total of three different β values per object, one being the estimated β_E and the other two the adjusted for the varying C_D, or $\beta_{C_D,\alpha=0.8}$ and $\beta_{C_D,Langmuir}$. The performance of these ballistic coefficients are assessed by subjecting them to a special perturbation propagator and compare them to the object's verification orbits build-up from their historical TLE data. To keep the research from growing too complex, it has been decided to solely consider 1U CubeSats in the target satellite database, primarily due to their uniform shape and large presence in LEO and as a follow-up research from D. Oltrogge[35] who performed an evaluation of CubeSat orbital decay back in 2011. Around a hundred 1U CubeSats have been found that would be suited for this research, together with close to thirty spherical satellites for the spherical satellite database. With the need and framework for this research having been discussed, the research question can thus be formulated as follows:

How do estimated ballistic coefficients adjusted for atmospheric density errors and the inclusion of an ambient gas-composition dependent C_D affect orbital lifetime predictions?

Researching this statement can provide insight in how historical data sets can be used to obtain accurate lifetime predictions for satellite operators, as well as potentially debunk the commonly used assumption that a constant C_D of 2.2 is a valid constant throughout a complete orbital lifetime analysis.

To support the stated hypothesis, this thesis shall also investigate and answer the following sub-questions:

- How can variations in C_D as function of gas composition be accurately modeled for calibration spheres and tumbling CubeSats?
- Is Langmuir's isotherm model suited for thermospheric energy accommodation coefficients?
- In what regions do this method perform best, and how sensitive are the results to the initial sample size?
- Can this method be extended for satellites with more complex geometries?

To fully be able to answer these question, the methods described by Emmert, Bowman, Vallado, and Pilinski are to be incorporated into a computer program, include a function for the C_D variation. Subsequently verification of the software by reproducing their datasets shall confirm the righteous implementation of the theory and models. Ultimately, the propagation results should provide insight into whether the inclusion of the C_D variation does indeed result in more accurate lifetime predictions, as well as the β estimation procedure is valid for large groups of satellites. Further details on the experimental set-up can be found in chapter 6.

A set of objectives have been set up to assist in the development of the complete experiment. These objectives shall form the backbone for the programming outline and project planning, and make use of the methodologies described in section 4.1 and section 4.2.

- **Objective 1**: Demonstrate capability of batch orbit propagations.
- **Objective 2**: Introduce an adequate TLE filtering algorithm.
- **Objective 3**: Develop the ability to estimate β_E values according to J.T. Emmert's theory[13].
- **Objective 4**: Demonstrate capability of computing C_D as function of ambient gas composition
- **Objective 5**: Incorporate Langmuir's adsorption model into objective 4.
- **Objective 6**: Propagate objects for the different β values and analyze the propagation results.

The described objectives are in a logical development order, where first an emphasis is set on being able to properly propagate a TLE to predict its orbital lifetime, and from there continue building the program until the research questions can be answered.

1.3. Thesis outline

The structure of this document starts with a discussion on the state-of-the-art technology in orbital lifetime predictions and thermospheric modeling (chapter 2 and chapter 3), as well as a discussion on the β estimation procedure in section 4.1 and the computation of an gas composition dependent drag coefficient in section 4.2. The implementation of this theory is discussed in the methodology chapter, after which the software implementations are verified separately at the end of that chapter. Having established a verified software package, the experimental set-up as well as the object databases are discussed in chapter 6, where the various scenarios and assessment merits are explained as well.

Logically, the scenarios and objects discussed earlier are subjected to the verified software package, from which propagation results are the product. This data is presented in chapter 7 for the different groups and scenarios, though a discussion on these results are held later in chapter 8. The discussion will form the foundation upon which the hypothesis and sub-questions are answered, which is done in the same chapter. The document concludes with an overview of the findings, complimented by a recommendation section on what could or should be done in the future when one is considering to extend this research.

2

The Basics of Orbital Lifetime Predictions

This chapter will provide the reader with an overview of the essentials in orbital lifetime predictions. In the introduction an indicative in the three main pillars have been presented, which shall be put in context in the next few pages. The relation between the atmospheric drag, thermospheric density models, and numerical propagators in the determination of the orbital lifetime is presented here. In subsequent chapters the discussed topics will be reviewed more closely, and a deeper insight into the current state of technology will be discussed. Later in the report, techniques that could potentially improve the accuracy of OLP are discussed, simulated, verified, and their results analyzed and presented.

2.1. Reference Frames

A brief introduction to the reference frames used in this thesis is presented in this chapter. The coordinate system most regularly used is the cartesian coordinate system, a system that specifies points within a three dimensional space using a set of three numerical coordinates, commonly denoted **x**, **y**, and **z**, which are signed distances from a fixed centre point measured in an equal unit of length. The set of coordinates can be extended to include position related information, such as the position derived velocity and acceleration. Within this report, the Cartesian coordinate system is used to express an object's position (x,y,z) and velocity (X,Y,Z) .

Cartesian coordinates can be translated to a spherical coordinate system, which is useful when the points of interest describe a sphere, such as is the case when geographical (position on Earth) or celestial (orbital plane) representations are expected. The Cartesian coordinates **x**, **y**, **z** aretranslated to the spherical coordinates **radial distance** $\boldsymbol{r}$, **azimuthal angle** θ, and **polar angle** ϕ using a simple set of three equations not further discussed in this report.

The spherical coordinate system uses radians instead of degrees and measures the azimuthal angle counter-clockwise from the x-axis to the y-axis. The polar angle is zero at the horizon and positive towards the North pole. The spherical coordinate system can subsequently be changed towards a geographical coordinate system where the positions are measured in latitude, longitude, and altitude, convenient for when the Earth is taken as reference center and the coordinates are to be related to specific geographic positions on or above Earth. Figure 2.1 gives a visual presentation the Cartesian and spherical coordinate systems.

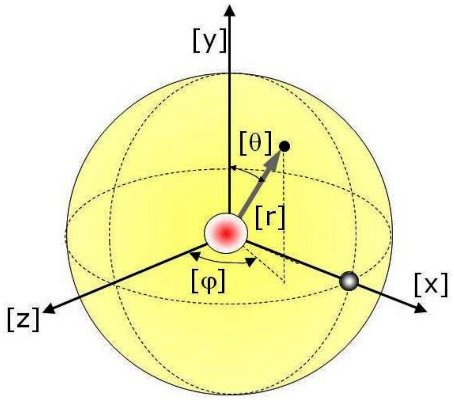


Figure 2.1: Cartesian and spherical coordinate system

2.1.1. Earth-Centered Inertial (ECI) and Earth-Centered Earth-Fixed (ECEF)

Earth-Centered Intertial (ECI) is the inertial reference frame and often used frame for this thesis. The frame is non-rotating and therefore common when considering space objects. The x-y plane coincides with the Earth's equatorial plane. The x_{eci} axis is permanently fixed in a direction relative to the celestial sphere (which does not rotate like the Earth does). The z_{eci} axis lies at a 90 angle to the equatorial plane and extends through the North Pole. Due to forces exerted from the sun and moon, the Earth's equatorial plane moves with respect to the celestial sphere. The Earth rotates, the ECI Coordinate system does not. The origin of the system lies in Earth's center of mass.

The effects of nutation, precession, and the acceleration of the Earth around the Sun results in the system being a quasi-inertial reference frame, and as mentioned result in the equoatorial plane moving with respect to the celestial sphere. Varying according to date, particular epochs have been specified to ensure the scientific community use the same reference frame notations. The two ECI-frames used in this thesis are the J2000 and TEME reference frame - J2000 in the Runge-Kutta 7(8) propagation in TUDAT and TEME as output from the SGP4 propagator.

J2000 - One commonly used ECI frame is defined with the Earth's Mean Equator and Equinox at 12:00 Terrestrial Time on 1 January 2000. It can be referred to as J2000 or EME2000. The x-axis is aligned with the mean equinox. The z-axis is aligned with the Earth's spin axis or celestial North Pole. The y-axis is rotated by 90° East about the celestial equator.

TEME - The ECI frame used for the NORAD two-line elements is called True Equator, Mean Equinox (TEME). An exact operational definition of TEME is very difficult to find in the literature, but conceptually its primary direction is related to the "uniform equinox"[50]. This cited work provides the reader with the coordinate frames definition used for the US space object catalogs.

To clarify, if Earth would be modeled in the ECI frame, it would be seen rotating over time whilst with the ECEF frame the Earth would be seen as being stationary (thus the ECEF frame itself is rotating). For orbital motions and the accompanying equations of motion it is recommended to perform these calculation in the ECI frame. Also, the ECI frame is more useful when specifying the direction towards celestial objects such as the Sun, Moon, and Mars, all three of them used in the equations of motion described in chapter 3.

Earth Centered Earth Fixed, also an inertial reference frame in Cartesian coordinates, is used as well in this thesis and denoted with the subscript *ecef.* with the point (0,0,0) located in the centre of mass of the Earth. Its axis are aligned with the international reference pole (IRP) and international reference meridian (IRM), fixed with respect to Earth's surface. The z-axis is aligned with the true North, not to be confused with Earth's rotational axis. The ECEF frame co-rotates with the Earth's around the Z_i axis, with the subscript i standing for the ECI z-axis. This frame is commonly used to project positions onto Earth's surface in the form of latitude and longitude. The International Terrestrial Reference Frame (ITRF) is an ECEF frame maintained

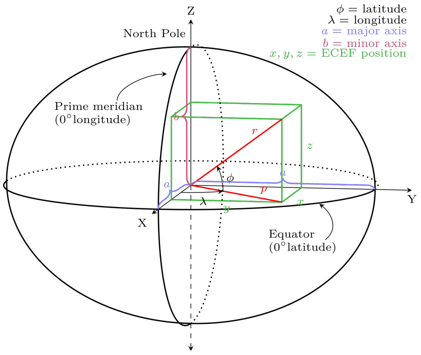


Figure 2.2: Earth Centered Earth Fixed (ECEF) reference frame

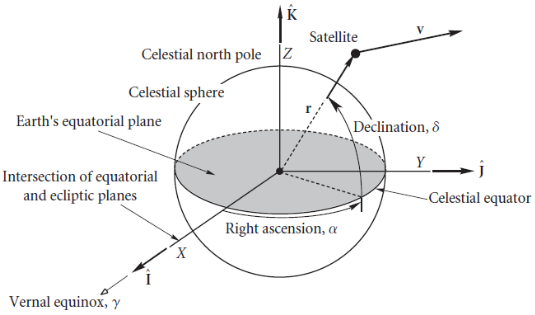


Figure 2.3: Earth Centered Intertial (ECI) reference frame

by the International Earth Rotation Service (IERS) and are subjected to small variations such as tectonic and tidal deformations.

In this research, every object (or TLE) is initially considered a point mass in the ECI frame. When using the SGP4 propagator to retrieve the orbital elements from the TLE data, the used reference from is the TEME (True Equator Mean Equinox), an ECI-frame which incorporates the mean motion, precession, and nutation for Earth's rotation. To include thse effects, additional input factors for the transformation from TLE to TEME are required, namely the polar motion coefficients as a function of Julian date.

The motion of the rotation axis of the Earth relative to the crust, known as polar motion, is represented in the Earth Orientation Parameter x_p and y_p, or EOP. In the terrestrial ECEF frame, the equatorial coordinate of the pole are x and -y. It has two major components: (i) a free oscillation with period about 435 days (Chandler wobble) and (ii) an annual oscillation forced by the seasonal displacement of air and water masses. The EOP x_p and y_p are denoted in degrees and needed for the transformation from TEME to the terrestrial ECEF frame. The resulting absolute difference is visualized in Figure 2.4 for the years 2000-2011. The term polhody refers to a chart depicting the motion of the terrestrial pole as a function of time.

ITRF: The International Terrestrial Reference Frame is a reference frame suitable for use with measurements on or near and specified by Cartesian ECEF coordinates. The ITRF solutions are in a **geocentric coordinate system** and thus do not directly use an ellipsoid. If needed the Cartesian coordinates can be transformed to geographical coordinates when referred to an ellipsoid. Updated yearly, the ITRF is commonly use for accurate geographical coordinate transformations. The WGS-84, discussed below, has been derived from an ITRF referred an an ellipsoid and is identical at one meter level[58].

WGS-84: As Earth is not a perfect sphere and hence local altitude and gravitational accelerations are not uniform over the sphere, the geographical coordinate system is extended to a **geodetic coordinate system**. The geodetic system represents Earth as a reference ellipsoid derived from raw altitude data. The most commonly used geodetic reference system is the **World Geodetic System (WGS)**. This comprises a standard Earth coordinate system with a standard spheroidal reference surface and gravitational equipotential. The latest and most accurate version is WGS84 is the one selected for this research due to its accuracy and availability in MatLab and TUDAT.

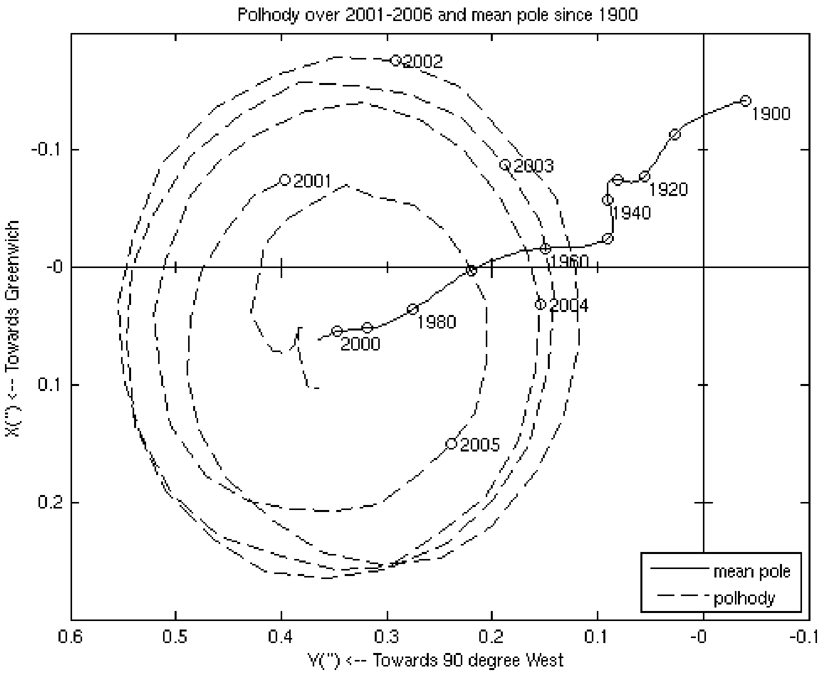


Figure 2.4: Polar motion, 2001-2006.Solid line : mean pole displacement, 1900-2006[44].

2.1.2. Conversion between TEME, ECEF, and J2000

Celestrak.com provides proper documentation on the process of converting TLE data to the TEME, ECEF, and J2000 reference frames, based on the work of Vallado[55]. When intending to convert a TEME ephemeris to another ECI ephemeris, they recommend not doing this rotation directly as there are many variables unknown. Instead, going to ECEF first is better and simpler. From there the ECEF data can be converted to J2000. These transformations are described below, together with the MatLab files provided by *Celestrak.com*.

- Use SGP4 to transform a TLE to TEME Cartesian coordinates.
- Rotate from the TEME to ECEF coordinate frames (teme2ecef.m).
- Use the ECEF vectors to find latitude and longitude (ijk2ll.m).
- Convert from ECEF to J2000 (ecef2eci.m).

In subsection 5.1.3 these Matlab files are used for the coordinate frame transformations. Due to the widespread use of Vallado's theory and software, the mathematical details behind the transformations shall not be elaborated on but assumed correctly incorporated by Vallado in these functions. No tempering with these original MatLab files shall be done throughout the thesis work.

2.1.3. UT1, UTC, and TAI

Within this thesis, three time notations are used in which the aforementioned reference frames can be expressed - UT1, UTC, and TAI. The International Atomic Time, or TAI, is an atomic time standard using the combined output of about 400 atomic clocks spread around the globe. Though highly accurate - only deviating 1 second in up to 100 millions years -, it does not take into account Earth's slowing rotation. Universal Time, or UT1, is a measure of time related to sidereal time, or more specifically the rotation of Earth relative to a fixed celestial background, and thus does include the variations induced by Earth's rotation. A constant comparison between the two is made to ensure the difference between the two time scales does not exceed 0.9 seconds. If this difference does however occur, a leap second is added to the Coordinated Universal Time

(UTC), the third time scale considered the world's standard and onto which all clocks and timings are regulated.

Daily values for the difference between UT1 and UTC are provided by the International Earth Rotation and Reference Systems Service (IERS), as well as regular updates of the current Earth Orientation Parameters (EOP) x_p and y_p. For transformation between either timescales or reference frames, the latest values from IERS are to be consistently used. To give an example relevant to this thesis, the output of the orbital propagator SGP4 are coordinates in the TEME reference frame and UTC time scale. If these coordinates are to be used in for instance the atmospheric density model NRLMSISE-00, these coordinates would first have to be transformed to the ECEF-frame and adjusted to the UT1 timescale, as these are the expected input reference frame and timing.

2.2. Aerodynamic Drag

Beginning by the basics, the *aerodynamic drag* is the most influential non-gravitational force acting on objects orbiting Earth in orbits below 2,000 km. Rarefied gas particles interact with an object such as a satellite, resulting in energy dissipation and thus in a reduction in orbital period and altitude. An accurate determination in the aerodynamic force is thus key in deriving the lifetime of an object. Let's first start with a small introduction into the fundamentals behind aerodynamic drag force.

Even in an orbit around the Earth, in the near-vacuum of space, a satellite still experiences an aerodynamic drag due to Earth's atmosphere. This drag slows the spacecraft down (energy dissipation) and results in the degradation of the semi-major axis. Over time, the orbit will have degraded so much it starts to re-enter the atmosphere and most likely burn up. The drag an object experiences can be best quantified by determining its drag coefficient. The drag coefficient C_D is a normalized representation of the drag force F_D experienced by the object:

$$C_D = \frac{F_D}{A_{ref}\frac{1}{2}\rho v_r^2} = \frac{ma_D}{A_{ref}\frac{1}{2}\rho v_r^2} \tag{2.1}$$

in which m is the mass of the object, a_D the drag acceleration, A_{ref} a reference area, ρ the local mass density and v_r the relative velocity of the atmospheric particles interacting with the satellite. The dimensionless drag coefficient is often considered to be constant and therefore a convenient way to quantify performance compared to other objects. The drag acceleration itself, $a_D = F_D/m$, can generally be derived from accelerometers such as on GRACE, CHAMP, and GOCE[14] and Two-Line Elements (TLE) from satellite tracking observations[39]. For this research, solely TLE data shall be used as this data is excessive and openly accessible.

Using this approach leads to an *estimated drag coefficient*. This estimated drag coefficient will come close to the value of the true drag coefficient, but due to its empirical nature the estimated drag coefficient will always have a small relative error. Moreover, the true mass density is also not a parameter readily available due to the high complexity of the atmosphere and the lack of global recurring in-situ observations. Generally, ρ is predicted through a density model, resulting in a *model-based estimated drag coefficient*. Many thermospheric density models are currently in use, each with a different approach on the mass density computations. Obviously this leads to discrepancies amongst the models, and each associated model-based estimated drag coefficient. E. Doornbos, S. Bruinsma, M. Pilinski, and B. Bowman, discuss the implications this has on current mass density derivation techniques and urge for a global standard in satellite drag computation[12].

$$a_D = \frac{F_D}{m} = C_D \frac{A_{ref}}{m} \frac{1}{2} \rho v_r^2 \tag{2.2}$$

Using this estimated drag coefficients, the drag acceleration can be computed using Equation 2.2, which can subsequently be used for the derivation of the decay rate and the associated orbital lifetime. However, the reference area and/or mass of an object is not always known. Satellites can be tumbling, resulting in a fluctuating reference area, or due to out-gassing have lost a portion of their mass. Or consider space debris, where the reference area and mass are even harder to accurately determine. For these objects it makes more sense to estimate the (inverse) ballistic coefficient according to Equation 2.3. This term is especially useful when the right-hand side is considered constant and is largely determined by the shape and attitude of the satellite as compared to variation of temperature and gas composition at the specific altitude.

$$B = \frac{1}{\beta} = C_D \frac{A_{ref}}{m} \tag{2.3}$$

This thesis takes it one step further though to retrieve accurate β estimations through analysis of historical TLE data and combine this in orbit propagation with the inclusion of the variable drag coefficient as function of ambient gas composition. This discussion will be continued in chapter 2, but the importance of the drag and ballistic coefficient to the decay rate and associated orbital lifetime has become clear.

2.3. Orbit Determination

The contribution of atmospheric drag to the orbital decay rate has now been established. However, at lower levels of the thermosphere the mass density increases and thus results in a larger decay rate, or, the decay rate is not a constant but is a function of the local mass density. To estimate an orbital lifetime one has to first establish an initial unperturbed state and current position of the object. During the design phase an estimate is made on where the satellite is injected into orbit, and this will be considered the unperturbed initial state, or orbit injection state. However, when observing orbiting objects in space it becomes important to acquire an accurate estimation of its current position. Orbit determination (OD), or the tracking of objects, is amongst other done by NORAD and NASA, who have a publicly available database on the (historical) orbital elements of over 16,000 objects presented in a standard called Two-Line Elements (TLEs).

The TLEs are based on a number of observations of the object, after which the simplified perturbation SGP4 model derives the orbital elements. This is done by applying the perturbation model to the observations and find fitting orbital elements, resulting in the final TLE not being the precise orbital elements but an average over a certain time span. Same as with the drag coefficients, these are *estimated orbital elements* and thus include a magnitude of error in them. The SGP4 perturbation model does include the main perturbing forces (gravitational, radiation, drag, third-bodies, see section 3.2), but its analytical nature does impose limitations. During a single day, the semi-major axis error could accumulate to 1 km per day, an error which can significantly propagate when the TLE is not updated frequently. There are other ways to more accurately determine the orbital elements, for instance by enhancement of GPS or Satellite Laser Ranging (SLR) observations or by using special perturbation models, see subsection 3.3.1. However, the NORAD database accessible through CelesTrak.com provides such an abundance of TLE information that in most cases these orbital elements are used for orbital lifetime predictions and thermospheric mass density computations[39][4][11][13].

The Two-Line Elements provide temporal orbital element sets, determined at an epoch inconsistent throughout the day. Using these orbital elements, the decay rate can be deduced by observing the change in semi-major axis, see Equation 4.2. The historical TLE data can furthermore be investigated to increase the accuracy of the orbital lifetime predictions. For instance, in their paper, C. Levit and W. Marshall[27] demonstrated a major improvement in reducing the error drift by applying a Least-Square Batch Approach.

The importance of the atmospheric drag and unperturbed orbit determination for the orbital lifetime prediction have been discussed. Subsequently an introduction has been made on the use of TLE data and the sense of using the (inverse) ballistic coefficient, as well as the distinction between true and (model-based) estimate drag coefficient has been discussed. The orbital prediction is still far away, as the influence of the solar activity and its effect on the thermospheric density and gas composition have not been reviewed yet. Thermospheric density prediction models are essential when considering orbital lifetime, and how exactly this fits in the bigger picture is presented in the next section.

2.4. Solar Influence and Thermospheric Density Prediction

The energy input of the Sun into the atmosphere is not constant, but varies with time. Mostly the changes can be attributed to established solar cycles, but instantaneous events on the Sun can also have an influence in the space weather around Earth, though within a smaller order of magnitude. Electromagnetic radiation and charged particles in the form of Solar wind interacts with both Earth's magnetic field and the composition of the atmosphere, most extremely in the Magnetosphere, Ionosphere, and Thermosphere. The fluctuating energy input results in expansion and shrinking of the atmosphere, similar to when a balloon filled with air is heated or cooled. Expansion of these layers of the atmosphere in which an object is orbiting Earth can significantly increase the local mass density, sometimes even up to 10x the nominal value during high solar activity[11]. Thus when considering orbital lifetime one should also include a prediction model on the thermospheric mass density as the effects of solar activity can result in extremes of predicted lifetimes of 2 years being shortened or extended by tens of percentages, see Figure 2.5.

Standards on the solar activity and the geomagnetic activity have been established that can be used as input to the thermospheric density models. These indices, such as the F10.7 solar radio flux proxy or the K_p global geomagnetic activity, serve as indications on the future solar weather. In subsection 3.4.1 the details on the solar activity and its relation to the fluctuations in local mass density is discussed, as well as the origin and usefulness of these indices.

At the moment there are a couple of commonly used empirical thermospheric density models, see Figure 3.6, which use parameters such as the solar and geomagnetic indices together with the location of interest to estimate the mass density set in the future. The model families have their own underlying principles and are based on past observations (hence empirical) such as radar and accelerometer data, or a combination of the two. Though all applicable to the thermospheric regime, the mass densities might differ in certain periods of a solar cycle or at the boundaries of the thermosphere. Furthermore, not all models are made publicly available, forcing scientist to make due with the most suited model available. However, the models can be calibrated with additional accelerometer data if need be. This thesis shall however not consider that option. Amongst the output variables is the local mass density, which can subsequently be used for the computation of the atmospheric drag as explained in previous sections.

2.5. Computing Orbital Decay

At this point, all the input variables for the computation of the orbital decay are ready. The initial unperturbed position of the object has been discussed, along with the expected local mass density values and the estimated drag coefficient. From here it is the question how to model the trajectory of the object for the next weeks, months, or even years. As will be mentioned in section 3.3, a perturbation model will have to compute the position of the object for the upcoming time span and with the SGP4 perturbation model an error of 1 km per day could arise. Imagine if this error is extrapolated over a time span of years the impact of this error could be enormous. Figure 2.5 provides an example of how errors in lifetime predictions could creep on towards large final lifetime estimation errors, the *sensitivity of orbit decay prediction.*

The example used is not of a real spacecraft, but depicts a clear image on how errors propagate in time. The perturbation model will have to take into account the main perturbing forces, but for obvious reasons the errors are induced by neglecting the smaller perturbations or by taking too large step sizes. A perturbation model including all known perturbations and taking step sizes in the order of seconds will drastically reduce the error, but at a severe cost of computational time. The main distinction for perturbation models and propagators are whether an (semi-)analytical or numerical approach is used. An analytical approach will provide an absolute result and will receive its from the sophistication of the functions describing the forces acting on the body, whilst the numerical approach will provide an highly accurate approximation, seeing its accuracy primarily dependent on step size and truncation errors. section 3.3 presents the commonly used general perturbations and special perturbations method.

The initial unperturbed state, whether theoretical or observed (be aware of potential bias error), is subsequently projected into the future with an orbit propagator. Primarily due to the atmospheric drag at the lower altitudes the orbit of a satellite will slowly decay, or more strictly experience a reduction in its semi-major axis until it will eventually re-enter Earth's atmosphere. The propagation of the orbit can be done in various ways, with the distinctions being made between analytical and numerical, see section 3.3. Though orbital mechanics is subjected to high-order terms, and is therefore tedious if not impossible to solve accurately analytically. Therefore orbit propagation is often performed numerically, which can provide much higher accuracies. After running the propagator to the new time of interest, the user would want to receive new/updated state

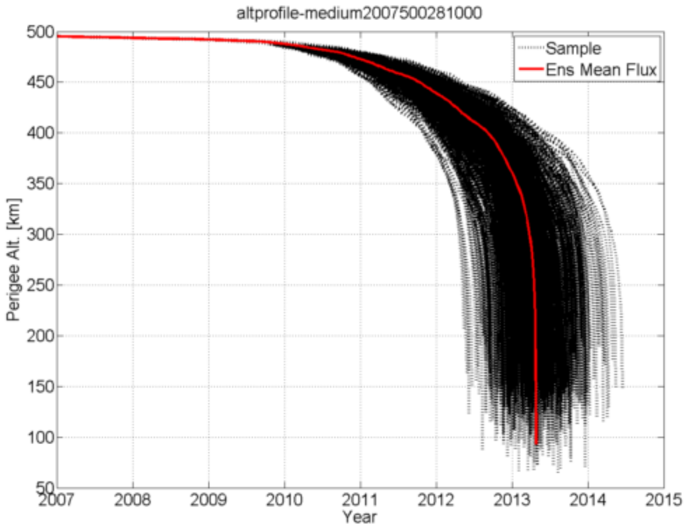


Figure 2.5: Example error sensitivity for 28°inclination 500 km altitude orbit, with a medium peak flux for cycle 25[2].

vectors or orbital elements from which the orbital lifetime can be deduced.

At one point, the orbit propagator will state that the satellite is at a certain altitude at which the mass density of the atmosphere is simply too high and will cause the satellite to burn up. At this point, the satellite will cease to exist and will have come to the end of his lifetime. The importance of prediction this lifetime is tied to both mission planning and the engineering process. Stakeholders are expecting the satellite to perform a certain amount of time, his mission lifetime. Engineers will have to ensure the satellite will be able to perform this long, not only mechanically but also taking into account the decay of its orbit and the corresponding impact it has on satellite performance. Furthermore, if it might become apparent during the design phase that the satellite could potentially orbit Earth longer than 25 years, considerations regarding a re-entry propulsion system might have to be made.

Due to the many inaccuracies in the aspects leading up to the orbit lifetime prediction, the predicted values often deviate from the actual lifetime, sometimes in the order of only days but potentially also in the order of weeks to months. To reduce this error, various techniques to reduce the errors emerging from uncertainties in atmospheric behaviour, solar activity, and satellite in-orbit properties, have been thought of. Two of those, focused on the ballistic coefficient estimation and the inclusion of a varying drag coefficient, shall be the focus of this research, and are further discussed in section 4.1 and section 4.2 respectively.

3

Numerical Orbit Propagation

To this point, the need for orbital lifetime predictions, scope of this thesis, reference frames, and research objects have been discussed. The chapter shall provide the framework for the transformation of specific initial orbital positions to positions in the future, and thereby provide the possibility to predict the orbital lifetime of an object.

Defining the orbital lifetime requires knowledge on the determination of unperturbed orbit, the addition of perturbing forces, and how to propagate these forces numerically. As atmospheric drag is the prime contributor to orbit decay, special attention shall be given to the estimation of the local atmospheric densities and the thermospheric models used in this thesis.

3.1. Determination of the Unperturbed Orbit

When determining an initial (or unperturbed) state multiple approaches are possible, such as satellite laser ranging (SLR), radar tracking, optical sightings, and GPS observations. The US Space Surveillance Network (SSN) provides radar and optical tracking data from which orbital elements are derived. CelesTrak.com uses this data, together with publicly available GPS data, to generate Two-Line Elements (TLEs), a data set consisting of orbital elements of an object. These TLEs are elaborated on in more detail below.

3.1.1. Two-Line Elements (TLE)

Two-Line Elements are standard data sets used by NORAD and NASA to represent observed orbits of radar and optical observations. The elements in the TLEs are mean elements calculated to fit the set of observations using the SGP4 orbital model, and are in the Earth-centred inertial reference frame (ECI) and delivered in UTC. Published SGP4 TLEs are Kozai mean elements modified to express the mean motion in its original Brouwer form[39]. Kozai elements are from the Kozai theory-based Simplified General Perturbations (SGP) astrodynamics model, and Brouwer elements thus underlying formalism in the SGP4 propagator. The SGP4 propagator first converts the published TLEs from the Kozai mean elements to the Brouwer values, before computing a state vector. As the TLEs are fitted on a set of observations, there will be a slight error in the observed orbital elements and the actual orbital elements.

TLEs are consistently used in orbit determination, and there set format will therefore be shortly discussed in this section. A extensive data base consisting of TLEs of around 16,000 objects, defined at a reference epoch and orbiting Earth, can be publicly accessed on CelesTrak.com. Data for each TLE object consists of three lines in the following format below. Line 0 is a twenty-four character name consistent with the name length in the NORAD satellite catalogue (SATCAT). Line 1 and 2 are the standard Two-Line Orbital Element Set Format, an example is presented in Figure 3.1. Archived historical TLE data is publicly available through CelesTrak.

A small note on the $\beta^\star$ value appearing in the TLE format, the $\beta^\star$ is a drag-related parameter and could

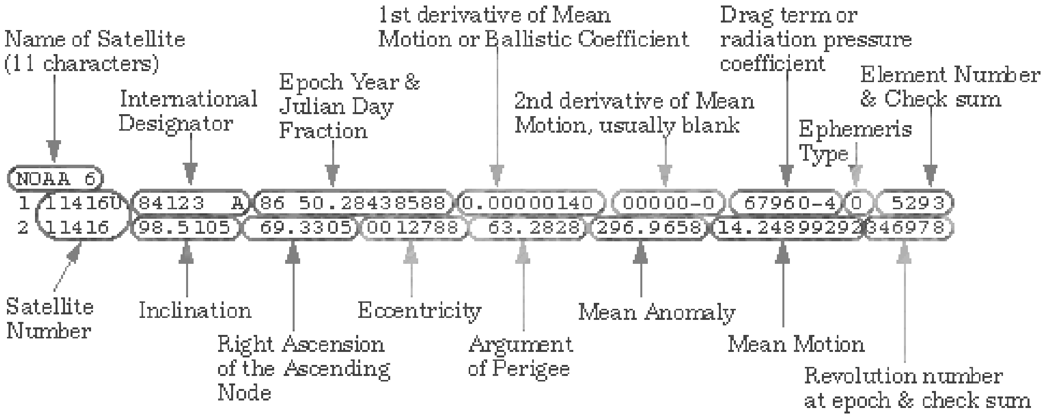


Figure 3.1: Two-Line Elements Break-Down [Credit: NASA HSF]

be converted to the ballistic coefficient β according to the following formula:

$$\beta = \frac{1}{12.74162\beta^{\star}} kg.m^{-2} \tag{3.1}$$

The $\beta^{\star}$ drag term is used in the SGP4 propagator and estimates the effects of atmospheric drag on the satellite's motion. This drag coefficient incorporates not only atmospheric drag, but serves as a fit parameter in the orbit determination/update process. Therefore,this term may also include effects due to sensor bias, non-modeled force terms (e.g. in solar radiation pressure and atmospheric perturbations), variation in the drag coefficient C_D, etcetera. It is therefore not advised to use the $\beta^{\star}$ drag coefficient in any mass density or ballistic coefficient derivation, $\beta^{\star}$ shall thus not be used in this research.

3.1.2. Equations of Motion

Without taking into account perturbations, the equations of motion of an object orbiting Earth can be derived from Kepler's law for a two-body astrodynamics problem. In this unperturbed situation, all orbits are solutions of this two-body problem in an inertial Earth centred (ECI) reference frame considering only the gravitational attraction by the assumed spherically symmetric Earth:

$$\ddot{r} = -\frac{\mu}{r^2}\hat{r} \tag{3.2}$$

where r is the inertial position vector of the satellite and μ equals the Earth's gravitational parameter, equal to the product of the gravitational constant and the mass of Earth. In this equation, the mass of the satellite is neglected compared to the magnitude of Earth's mass. Manipulating this equation according to the laws of astrodynamics presented in many astrodynamics textbooks such as that of Wakker[59] demonstrates that the orbit can be described by the semi-major axis a and eccentricity e:

$$r = \frac{a(1-e^2)}{1+ecos(\theta)} \tag{3.3}$$

With r and θ being the polar coordinates. Figure 3.2 represents a visualization of this 2D orbit of an elliptical satellite orbit. Kepler's third law can be derived after further manipulation of Equation 3.2, relating the semi-major axis with the orbital period P, see Equation 3.4. Furthermore, the mean motion expressing the number of orbital revolutions per unit of time is given is Equation 3.5.

$$P = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{3.4}$$

$$n = \frac{2\pi}{P} = \sqrt{\frac{\mu}{a^3}} \tag{3.5}$$

Especially the mean motion is useful as it can be directly retrieved from TLE data, see subsection 3.1.1. These basics of orbital mechanics do only take into account the gravitational attraction of a spherically symmetric Earth (with a homogeneous density distribution) and no other forces acting on the satellite. In real life though this is not the case, and the satellite will be subjected to other accelerations due to deviations in Earth's shape and mass distribution, as well as contributing factors from solar radiation, atmospheric drag, and third-body accelerations from celestial bodies such as the Moon and Sun; these will be described further in section 3.2.

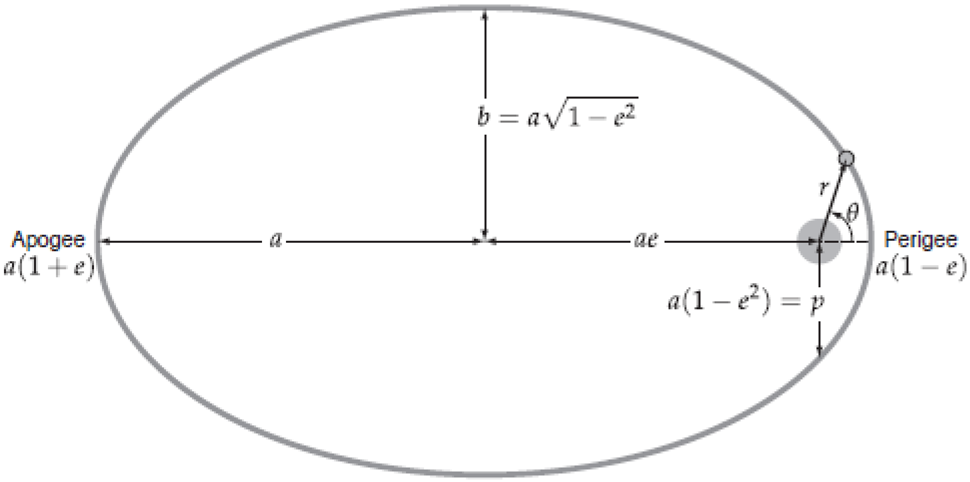


Figure 3.2: Geometrical parameters of an elliptical satellite orbit[11]

For orbital lifetime predictions an initial unperturbed state is chosen after which a perturbing force model assesses the change in semi-major and consequently the decay rate and lifetime of the satellite's orbit. This unperturbed state can be determined theoretically during the design phase for orbit lifetime predictions before the start of the satellites mission (often the orbit injection state), or an observation in for instance the TLE format can be used as the initial unperturbed state when the satellite is already orbiting Earth. Though orbit lifetime predictions before the mission commences are getting more and more accurate, corrections will still need to be made to this prediction once the satellite is actually in orbit. Before heading into these perturbation forces and models, a little elaboration on this TLE format is given in the next section. An important observation on using TLEs as starting state, in their paper C. Levit and W. Marshall[27] noted that using the latest TLE as starting state might not improve the orbit propagation when comparing a SP approach to a GP approach. The key is to finding a better initial starting state derived from multiple historic TLE data. TLE orbits are most accurate when evaluated 1-3 days before the TLE epoch[11].

3.2. Perturbing forces

The acquisition of an unperturbed state of the object has been discussed, and it is time to introduce it to the many perturbing forces acting on bodies in LEO. To keep within the scope of this theis only four perturbing forces will be discussed, see list below. These perturbing forces are selected on their order of magnitude and their presence (in the form of parametric equations) in the most wide-spread perturbation models. An example on the impact of the individual perturbing forces on the ISS in LEO at 380 x 390 km and an inclination of 51.6°is presented in Figure 3.4[60].

- **Non-spherical Gravity**
- **Solar Radiation Pressure**
- **Atmospheric Drag**
- **Third Body Perturbations**

The following sections will provide an introduction to these perturbations and their influence on the orbital elements. How these are consecutively incorporated in perturbation models and which models are used for this research will be discussed in section 3.3.

Non-spherical Gravity

Earth is not a perfect sphere, and neither does it have a homogeneous mass distribution. The shape of the Earth can be approximated by an ellipsoid with an equatorial radius about 20 km larger than the polar radius. Together with the non-homogeneous mass distribution the gravitational force varies with the latitude and longitude of an object, resulting in an Earth potential V that can be expressed by a spherical harmonic expansion:

$$V = \frac{\mu}{r}[1 - \sum_{n=2}^{\infty}(\frac{a_e}{r}^{n} J_n P_n(sin(\phi)) + \sum_{m=1}^{\infty}\sum_{n=2}^{\infty}(\frac{a_e}{r}^{n}[C_{nm}cos(m\lambda) + S_{nm}sin(m\lambda)]P_{nm}(sin(\phi))] \tag{3.6}$$

with a_e being the semi-major axis, r the geocentric distance of the satellite, and ϕ and λ being the latitude and longitude respectively. J_n, C_{nm}, and S_{nm} represent zonal (m=0) and tesseral (m≠0) coefficients of the harmonic development determined through a known Earth model. Finally, P_n and P_{nm} represent the Legendre polynomials and the corresponding Legendre functions. Variations in these groups occur with increasing subscript, for this research the spherical harmonics shall be to a degree of 5.

Primary secular variations of orbital parameters are produced by even-degree zonal coefficients, and the odd-degree zonal coefficients produce long-period perturbations. The tesseral coefficients on the other hand produce short-periodic perturbations. The largest order of magnitude is induced by the J_2 term, the harmonic coefficient modeling the ellipsoidal shape of the Earth considering a homogeneous density. For a satellite in a GPS orbit the perturbation due to the J_2 effect is roughly $5 * 10^{-5} \frac{m}{s^2}$. Though an order 10^{-4} lower than the central gravitational force of Earth it is still an order magnitude of 2 higher than the remaining harmonics. Nodal regression is one of the main changes in orbital elements due to the non-spherical shape and non-homogeneous gravity field of Earth. Figure 3.3 demonstrates an exaggerated effect of spherical harmonics, and then more specifically the J2-perturbation, on the precession of the right ascension of the ascending node of an arbitrary elliptical orbit.

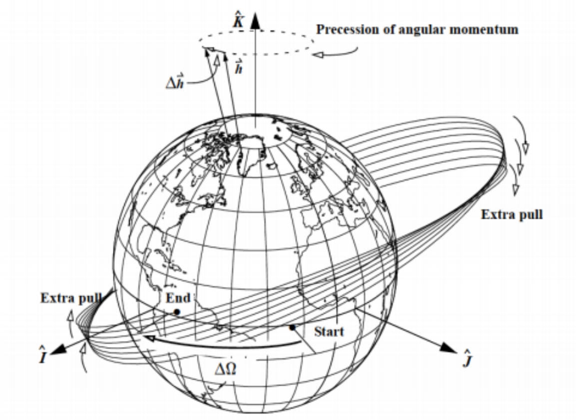


Figure 3.3: Nodal Regression due to J_2[60]

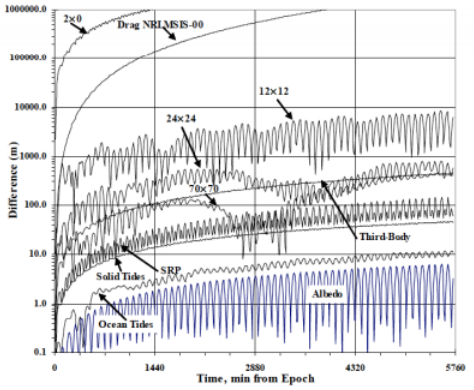


Figure 3.4: Impact of individual perturbing forces on the ISS in LEO[60]

Solar Radiation Pressure

Expelled by the Sun, photons are traveling at high speed through space away from the Sun and upon impact with an object result in a minute exchange in energy and momentum. This exchange in energy perturbs the orbit of the object, the severity of which depending on factors such as the cross-sectional area exposed to the Sun, the solar activity, and the mass and reflectivity of the satellite. The order of magnitude is roughly in the $1 * 10^{-7} \frac{m}{s^2}$ range and around 700 km can match the perturbation experienced by atmospheric drag. Solar radiation is more predictable and consistent though, and does not require sophisticated numerical models.

Atmospheric Drag

Similar to Solar radiation pressure, small particles from Earth's atmosphere interact with an object resulting

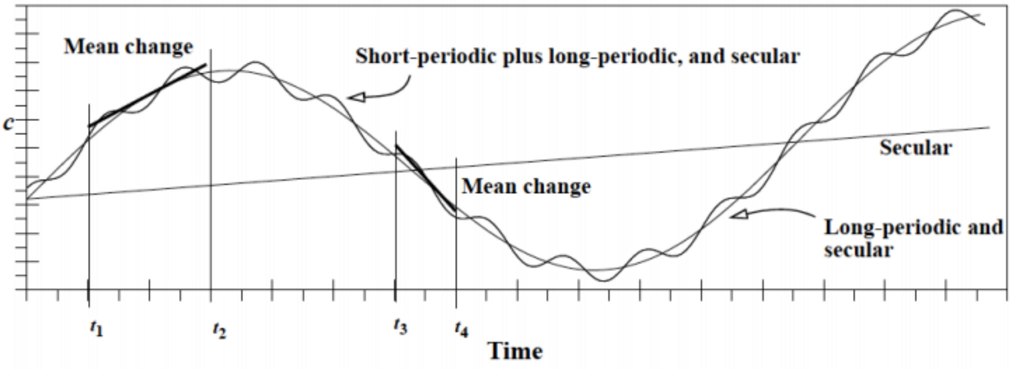


Figure 3.5: Effects of Perturbation Forces on Orbital Elements

in energy dissipation and the decline of the objects orbit (or more accurate, the reduction of the semi-major axis and eccentricity). Geometrical factors such as reference surface area as well as the mass of the object are key in the influence the particles exert on the body, as well as the ambient mass density and gas composition. Being the most important perturbing force in LEO when considering orbital lifetime predictions and being the main topic for this research, additional information is presented in chapter 4.

Third Body Perturbations
The presence of other celestial bodies such as the Sun and the Moon allow for a perturbing force due to their respective gravitational field, with the Moon producing the largest perturbation force. Though the Sun is much more massive than the Moon, the gravitational effect is proportional to the inverse of the square of the distance and therefore the effect of the Moon is larger than the Sun's. What is more, these forces induce tidal effects deforming Earth's shape and affecting its gravitational potential. These tidal effects produce accelerations in the order of $1 * 10^{-9} \frac{m}{s^2}$ on GPS satellites and are still three magnitudes lower than that of the gravitational forces of the Moon and Sun. For most propagation models the tidal effects are neglected.

The effect of these perturbations on the orbital elements is presented in Figure 3.5, with the change in the orbital element c being attributed to the secular effects (straight line), long-periodic effects (large oscillating line), and the short-period effects (small oscillating line)[60]. The presented example is exaggerated but representative of orbit degradation and fluctuation.

3.3. Numerical Orbit Integration

In most orbital dynamic computations, higher-accuracy orbits are not determined analytically but require effective algorithms in a numerical orbit integration approach due to their complex nature. Orbit-integration problems range in complexity following from the time span, number of orbiting bodies, and inclusion of perturbing forces. An integrator can be assessed in the accuracy of the integrated values, the memory usage, computational time. In this section a brief introduction on the various integration methods is presented, followed by the integrators commonly used in orbital mechanics. Basically, a division into three groups have been made; a *numerical table look-up*, *semi-analytical*, and *high-precision numerical* approach. Within this thesis and the respective required accuracy no use shall be made of the numerical look-up tables.

In a semi-analytical method the mean orbital elements will be subjected to the larger order orbit perturbations such as J_2 and J_3, as well as estimated drag-induced perturbations, expressed in analytical expressions. This method is not a direct 3-dimensional numerical integration, and though it lacks in the accuracy of a high-precision numerical integrator the inclusion of the larger perturbations significantly increase the accuracy as compared to the numerical look-up table. An altitude-dependent ballistic coefficient can be incorporated, but it is generally advisable to estimate an average ballistic coefficient[52]. This is the approach that is taken by the *General Perturbations (GP)* method, where general differential equations of the motion or change in orbital elements are solved analytically. An analytical approach to a problem solves the orbit determination directly by a set of equations. The higher and more accurate the number of equations the higher the accuracy of the solution will be, though also at the cost of computation time. Due to its analytical nature, the complexity of the equations of motion is limited by the computation time and in orbital mechanics is less preferable as compared to a numerical approach if high-accuracy is a requirement, see subsection 3.3.1.

3.3.1. General and Special Perturbations

Orbit integration using GP takes advantage of the fact that orbits are decaying at a very slow rate under the influence of their perturbations. A two-body orbit approximation with only a few of the higher order perturbations is already sufficient, as the GP approach characterizes an orbit analytically in terms of an ellipse, and describes a low-order analytic solution to Newton's second law with a realistic gravitational potential and atmospheric environment. To be more precise, the GP processor only includes the zonal harmonics through J_5 and has a few enhancements for resonant tesseral terms[61], leaving the element sets incapable of reproducing the satellites motion perturbed by short periodic terms as not enough accuracy in the tesseral terms is acquired.

The GP approach using the SGP4 perturbation model, see next subsection, has been used to derive the TLEs from observed changes in the mean motion and semi-major axis by the Naval Space Command and NORAD, resulting in the earlier mentioned TLE set of six orbital elements with an average error drift of 1.0-1.5 km/day. However, the increase in available processing power over the years have enable the possibility to achieve a higher accuracy when using high-precision numerical integration.

The Simplified General Perturbations (SGP) integrators are commonly used to calculate orbital state vectors of satellites and space debris, relative to ECI. There are five mathematical models (SGP, SGP4, SDP4, SGP8, and SDP8), of which SGP4 is most often used for the generation of TLEs by NORAD and NASA. SDP stands for Simplified Deep Space Perturbations and is applied to trajectories exceeding orbital periods of 225 minutes. For any trajectory with shorter orbital periods the SGP is the proper model to select. These models predict the perturbations due to the Earth's shape, drag, radiation, and third body gravitational effects. Though often used in orbital lifetime predictions, these models often have a ~1.0 km altitude error at epoch which can grow to ~1.0-1.5 km a day. The Naval Space Command use the Simplified General Perturbations-4 (SGP4) propagator[39] to characterize the orbits of the objects it tracks until recently, when it decided to start switching switching to an SP database.

A note when using state vectors as input for the SGP4 propagator. The TLE use the mean motion in terms of the Kozai's theory (1959), though the SGP4 model is based on Brouwer's theory (1959). This has no implication when using SGP4 to convert TLEs to initial conditions, though take care when considering converting a state vector from an ephemeris generated from another model. In that case, the semi-major axis needs to be converted first to the suitable Brouwer's mean motion before using SGP4. In his paper of 2008, David A. Vallado[55] provides the motivations and technicalities on how to do this. This shall not be discussed further here as the TLEs are directly inserted into the SGP4 propagator within this project.

Though an analytical approach provides an absolute solution to the problem, in astrodynamics and especially orbital mechanics solving an analytical solution involves too many equations and is not efficient time wise. Problems in orbital mechanics are more often solved numerically, which basically uses a set of differential equations describing the behavior of the object over time and then determining the change in the initial variables over a period of time. This approach will not provide an absolute solution, but can provide high accuracy approximations. The most accurate method is using a numerical integrator which includes an accurate high-order perturbation model. In methods of *Special Perturbations (SP)*, datasets consisting position, velocity, and acceleration are subjected to numerical integration of the equations of motion. Basically, the position and velocity is perturbed directly as opposed to using the observed changes in orbital elements. The possibility of using ever-increasing processing power enables orbit determination to be done more accurate with the use of the SP method as compared to a GP approach.

The SP approach performs a direct numerical integration of Newton's differential equation with detailed representations of the forces experience by the object[14]. The accuracy achieved using an SP propagator is significant,with 5-10 times smaller error growth as compared to the GP propagator. In their paper, C. Levit and W. Marshall[27] concluded that with using an SP propagator the error growth was only 0.1 km/day against the 1.0-1.5 km/day resulting from the GP method. An effort is made to also have the incoming observations

Table 3.1: Comparison between Cowell's and Encke's RK 7(8) propagation in TUDAT with similar initial conditions. Three satellites propagated for 100 days starting 21.04.2016 and their final semi-major axes compared to their respective historical TLE data. Values shown are the difference between propagation and verification orbit.

NORAD	Cowell	Encke
33497	0.305 km	0.382 km
35932	0.221 km	0.381 km
35933	0.131 km	0.313 km

into the SP format, together with the current GP derived TLE elements, according to the procedure described by Wilkins[61]. This will provide users with both TLE data presented in GP orbital elements and SP state vectors. In the future, the maintenance of the GP catalog will most likely be canceled as the GP elements can also be derived from the SP state vectors, mitigating the need for catalog machinery dedicated for the GP catalog.

When a user is deciding for a numerical SP approach, it is possible to develop their own propagator based on **Cowell's** or **Encke's** method combined. Cowell's method is the simplest SP method which basically sums the Newtonian forces on body i from other bodies j in x, y, z components and integrate these numerically to form new velocity and position vectors. Encke's method includes the osculation of orbits as described in Figure 3.5 and can compute more accurate orbits. The added complexity and the need for the occasionally required updating of the osculating orbit (known as rectification) in Encke's method are (recently) incorporated in the TUDAT libraries. Though the initial preference for this thesis was to use Encke's method, it has been decided to use Cowell's method instead due to faster computation times and in practice better results as compared to Encke's method using the TUDAT libraries.

During the selection process, a comparison has been made to justify the choice for Encke's method over Cowell's method, three satellites have been propagated for a period of 100 days with similar initial settings. The outcome semi-major axes have in turn been compared to their historical TLE dataset of the same timespan. The results can be seen in Table 3.1, which unexpectedly show a decrease in accuracy when using Encke's method. Moreover, the observed computation time was between 2x and 4x longer as compared to Cowell's method. Potentially this unexpected result could be due to parts of Encke's script in TUDAT not being verified properly yet, but irregardless of the error source the choice has been made to switch back to using Cowell's method.

3.3.2. Runge-Kutta Integration Methods

Numerical integration requires an integration method for solving the differential equation set with a balance between accuracy and computation time. In orbital mechanics, (variations on) the Runge-Kutta method is often applied. Developed around 1900 by the German mathematicians C. Runge and M. Kutta, these methods are often used in approximation of solutions of ordinary differential equations (ODE) such as for orbit problems in celestial mechanics[57]. A difference between implicit and explicit integration methods can be made. Without going into too much mathematical detail, implicit integration methods find a solution by solving an equation involving both the current state and the later state of a system, while an explicit method calculates the state of a system at a later time from the state of the current time.

Mathematically the state $Y(t)$ of the current system and $Y(t+\Delta t)$ the state of the later time, would be solved with the implicit method by:

$$G(Y(t), Y(t+\Delta t)) = 0 \tag{3.7}$$

and for the explicit method:

$$Y(t+\Delta t) = F(Y(t)) \tag{3.8}$$

The implicit method requires an extra computation to solve for $Y(t+\Delta t)$ and is therefore harder to implement. However, orbital mechanics problems are often stiff, or in other words, are comprised of differential equations for which the numerical methods could become numerically unstable unless step sizes are taken at extremely small sizes. Therefore, using an explicit method often requires an impractical small step size Δt to keep the error within bound. In their paper, J. Aristoff and A. Poore[1] demonstrate the superiority of implicit methods over explicit methods in orbital propagation for objects in LEO, GEO, and highly elliptic orbits.

Computation time and (truncation) error control are the two important considerations when selecting a integration method, but apart from only making a selection between an implicit or explicit method, one could also decide to select between a fixed or adaptive method. The difference between these two lies in the step size, with the first having a *fixed step size* and the latter a *varying step size*. An adaptive integration method redefines its step size at the beginning of every integration step by calculating all the derivatives of the ODE and lower the step size if the derivatives grow above a predefined value or will increase the step size if the derivatives go below this value. In his paper, Oleg Golberg[15] presents a quick and comprehensive overview of the advantages of an adaptive step size numerical method by implementing it into a standard Euler integration. He demonstrates the computational effectiveness of an adaptive integration method for ODE. The *Runge-Kutta-Fehlberg (RKF78)*, *Runge-Kutta-Nyström (RKN1210)*, and *Dormand-Prince (RK87)* method also incorporate adaptive step sizes and are used more commonly in practice. Some of these integrators such as the RK 7(8) and SGP4 are available in TUDAT, a function library from Delft University of Technology, and in MatLab code packages respectively.

3.4. Thermospheric density models

The presence of rarefied gas in the upper atmosphere is the main contributor to orbital decay in LEO. Particles interacting with the surface of the satellite slow down the object into atmospheric layers with higher density, thereby decreasing the semi-major axis. In the previous chapter the effect of Solar activity on the atmosphere has been described, and how atmospheric layers expand and shrink. Continuing from this, a review on the existing thermospheric models will be presented in this chapter. What are the advantages and limitations of the most up-to-date models, what are the major uncertainties and how are these induced, how are the models calibrated, and which model is most suited for thermospheric mass density computations?

In their comparison of density models, Montenbruck and Gill [31] conclude that the “models have statistical inaccuracies of about 15%, and there has been no significant improvement in density models over the past two decades.” Though since the publication of their book in 2001 there have been improvements in the density models, it is apparent that inaccuracies still dominantly persist in these models, especially during geomagnetic storms or with large variations in solar activity. Therefore most models are evaluated and compared to each other on different global regions and time spans to determine which model performs best under what conditions and how future models can be calibrated to include these improvements. The models discussed in this chapter are only empirical models.

3.4.1. Solar and geomagnetic indices

To retrieve an atmospheric gas composition and local air density estimation from an atmospheric model, a couple of necessary inputs are required. **Epoch date** and **position** of the position of interest are obviously required, but so is the influence the Solar and geomagnetic activity has on the atmosphere in that period of time. To describe the solar activity in a quantitative sense, various indices have been defined over time. The first such index is the relative sunspot number R. This index stood for 100 years, until the discovery of solar radio radiation the 10.7 cm radio flux index F10.7, or Covington index CI was introduced in 1946. Corresponding to the mean spectral energy flux density of the solar disk at a frequency of 2800 MHz, the F10.7 index is measured in ”solar flux units” (s.f.u.) and widely used for research related to solar weather predictions[34]. The F10.7 index has been recorded since its introduction in 1946 and tracks well with observed sensitivities in the Ionosphere and upper atmosphere imposed by solar extreme ultraviolet (EUV) radiation. It also relates strongly to its predecessor index, the number of sunspots R, which was later discovered is a meaningful indicator of future solar activity[17]. Unlike many solar indices, the F10.7 radio flux can easily be measured reliably on a day-to-day basis from the Earth’s surface, in all types of weather[2]. This data is publicly available through the on-line data service of the Space Weather Prediction Center (SWPC) of the National Oceanic and Atmospheric Administration (NOAA). To give an example, when the solar activity is quiet, or below nominal, the corresponding Covington index CI = 66, while in extreme cases values of CI = 300 can be reached.

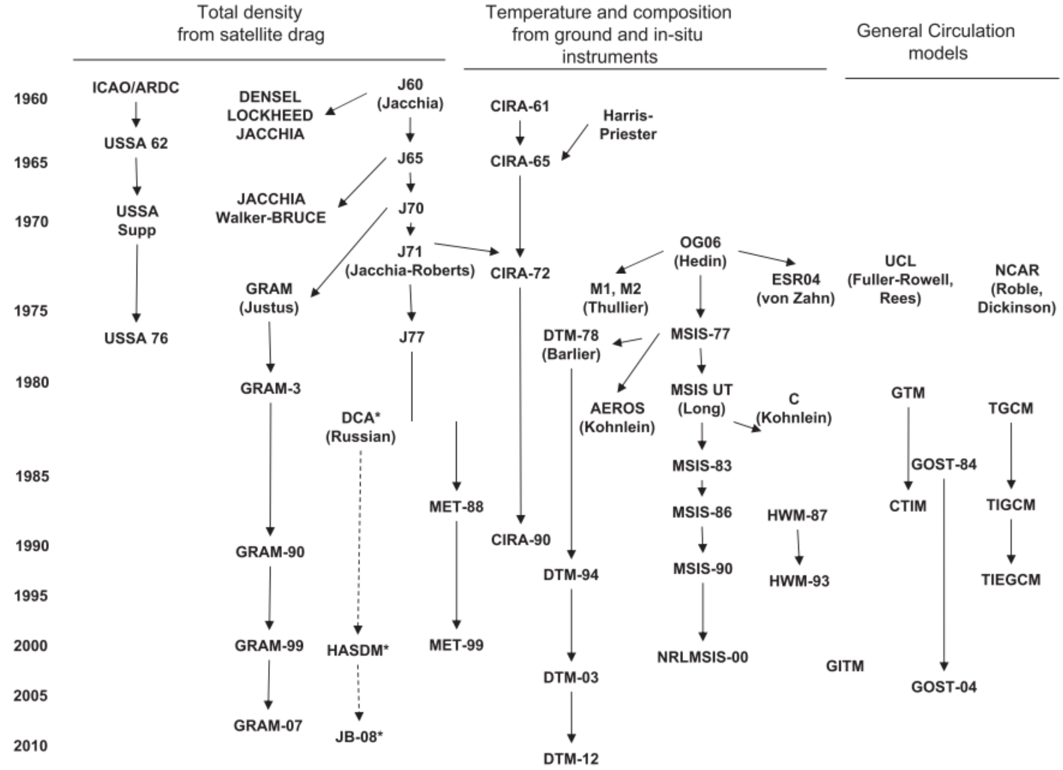
Geomagnetic indices are used to characterize the variability of Earth's magnetic field. These indices are represented in single numbers and can be divided into two main categories, the K-index (non-linear, 0-9 scale) and A-index (linear, 0-400 scale). In these categories, subscripts denote further specialization of each index. The main geomagnetic indices (K_p and a_p) are subsequently being used most often as input for atmospheric models are presented below. The definitions of the indices in this report are deduced from information by NOAA and the International Service of Geomagnetic Indices (ISGI).

- K_p - A non-linear index used to characterize the intensity of geomagnetic activity on a planetary scale (however because of the historical context at the time of its creation, the Kp network is heavily weighted towards Europe and Northern America).
- a_p - To linearize the K_p index, an average of eight 3-hour K_p values is converted to a_p values, providing a daily average level for geomagnetic activity.
- a_a - To measure the amplitude of global geomagnetic activity during 3-hour intervals normalized to geomagnetic latitude around 50 degrees. aa was introduced to monitor geomagnetic activity over the longest possible time period.
- a_m - To provide a characterization of global geomagnetic activity using a large set of stations representing all longitudes and possible hemispheric discrepancies. am = (an + as) / 2.
- a_n - The geomagnetic index to characterize geomagnetic activity on the Northern hemisphere.
- a_s - he geomagnetic index to characterize geomagnetic activity on the Southern hemisphere.
- Dst - To monitor the axis-symmetric magnetic signature of magnetosphere currents, including mainly the ring current, the tail currents and also the magnetopause Chapman-Ferraro current.
- AE - The Auroral Electro-jet (AE) index measures the global electro-jet activity in the auroral zone.

Depending on the atmospheric model, a combination of these indices are inserted in the model to retrieve the local air density. Within the computations performed for this thesis, these indices are automatically downloaded and stored. Take note though that when performing an orbital lifetime prediction, predicted values of F10.7 and the APH values are provided by NOAA yet they increase in inaccuracy the further away the prediction goes into the future. The predicted values are not wild guesses though, as Solar activity analysis from data going back decades[19][37][43] has already provided insights in the most dominant Solar cycles upon which future predictions can be extrapolated from.

3.4.2. Empirical density models

Empirical models are based on fitting parametric equations to a set of observations, the accuracy of these models depending greatly on the quality of the observations and the complexity of the equations, as well as the ability to interpolate and extrapolate the observation data between time steps. The earliest empirical models go back as early as the Harris and Priester model presented in 1962[16], and with the availability of more and more data these models have frequently been updated with increasing accuracies. In Figure 3.6 an overview of the development of these atmospheric models is presented. There are three model families that have been used most widely within thermospheric research; *Jacchia*, *MSIS*, and *DTM*. Development within these families dates back all the way to the early 1960's and 1970's[21], several of these will be discussed further in this chapter.



* Corrections to existing models

Figure 3.6: Development of atmospheric model [56].

This research uses solely open-source software, and therefore does not consider models such as HASDM. The NRLMSISE-00 mentioned in this report has calibration capabilities, though no elaboration shall be given as this calibration shall not be performed.

4

Ballistic Coefficient and Drag Coefficient

Numerical propagating an orbit is not the novelty on which this thesis research is focused on, instead the research has its emphasis on two important parameters needed for orbital lifetime prediction - namely the **ballistic coefficient** β and **drag coefficient** C_D. The ballistic coefficient is essentially an indication of the influence the atmosphere has on the decay of an object; simply stated a small heavy object orbits Earth longer than a large light object with the same initial conditions and under similar atmospheric conditions. Though computing the ballistic coefficient is simple and straight-forward (see Equation 2.3), being able to provide accurate values for the mass, drag coefficient, and reference area prove to be a difficulty.

Most man-made space objects are well-documented, and the mass and reference area can be found in literature quite easily. However, some (small) satellite operators tend to present rounded values for their mass, e.g. 1 kg instead of the actual 800 gr mass, or the satellite is freely tumbling and does not have a properly defined reference area for the incoming aerodynamic flow. Furthermore, the drag coefficient is subject to variation as well throughout the object's lifetime - though in most orbital lifetime prediction tools this drag coefficient is considered constant with a value around $C_D = 2.2$[49][28]. The theory described in this chapter shall elaborate on ways to get a better estimate of the ballistic coefficient of an object by analyzing its historical TLE data and comparing it with decay data of objects experiencing the same atmospheric conditions, and the implementation of a varying drag coefficient as opposed to a constant one.

To summarize the identified problem, there are two categories of uncertainty when determining the ballistic coefficient of an object - first the uncertainty in an object's mass and reference area and second the variability and uncertainty of the drag coefficient. Based on the literature study performed prior to this research, methods that could mitigate these uncertainties have been found and are to be incorporated in this research. Afterwards, an assessment shall be performed to determine the actual contribution of these new methods as explained in chapter 6. The two main research topics for this thesis are thus:

- **Ballistic coefficient estimation** through analysis of historical TLE and density data based on the work of J.T. Emmert[13].
- **Drag coefficient determination** as a function of ambient gas composition and object geometry based on the work of M. Pilinski[41], L. Sentman[51], and K. Moe[30].

In this chapter solely the theory is described, the implementation and verification shall be discussed in section 5.2 and section 5.3. The experimental set up, including the CubeSat subjected to the various experimental scenarios, hypotheses, and sub-questions are discussed in chapter 6.

4.1. Ballistic Coefficient Estimation

The importance of estimating the ballistic coefficient lies in the uncertainties in an object's initial estimation, the dependency orbital decay calculations have on this parameter, and the variability it can have throughout its lifetime. As seen in Equation 2.3, β is dependent on mass m, reference area A_{ref}, and drag-coefficient C_D. Assuming the objects of interest do not expel mass during their mission, it is still possible to have a varying A_{ref} when an object exhibits a tumbling behavior or has its attitude actively controlled. Furthermore, C_D is a function of ambient gas composition and geometry, and thus has a temporal variation as the experienced ambient gas composition varies with altitude and solar activity. Though at the start of a mission the β can be assumed, it could prove that the actual β^T deviates significantly from this initial value.

This chapter will provide the reader with the ballistic coefficient estimation method, the backbone of this thesis. The theory shall be discussed first, followed by how this is implemented into the software. Apart from purely the software implementation, the focus shall also be on any assumptions made, database set-up, and intermediate procedure steps. This chapter shall conclude with the verification of the software in accordance with the original theory. An estimated β using this procedure is expected to result in more accurate β values, which is expected to be seen back in the assessment of this theory in chapter 7.

In theory, the β estimation procedure is based on one single aspect - the cancellation of the thermospheric model error between two objects experiencing similar local atmospheric conditions. How similar exactly one wants to have these local conditions depend on a set of criteria based on epoch and orbital element range; more on this in subsection 5.2.2. Assumed is that the only contributing factor to an objects orbital decay is the local thermospheric mass density ρ, a value that can be obtained in two ways:

- **Model-estimated mass density** ρ^M - by inserting the object's positioning and epoch in the NRLMSISE-00 model and retrieve its total mass density as output.

- **Orbit-derived mass density** ρ^T **-** by quantifying the change in observed time-integrated value of the drag-induced acceleration $\dot{v}$, which can be derived from observing the changing mean semi-major axis seen in historical TLE data.

When both ρ^M and ρ^T are computed over the same time period the difference can be attributed to errors in the NRLMSISE00 model - errors that are temporal and spatial and therefore are not consistent. Now consider this procedure being applied to an object with an accurately known β. When a second object is within the same atmospheric region, and thus experiencing the same atmospheric conditions, the atmospheric error can be canceled out and a more accurate β is estimated for the second object.

4.1.1. Orbit-derived mass density

The most straight-forward way to derive thermospheric mass density from observation is by assessing the drag-induced acceleration experienced by satellites orbiting in the thermosphere. The drag-induced acceleration is given by:

$$a_d = \dot{v} = -\frac{1}{2}\rho\beta|v - v_a|^2 e_{v-v_a} \tag{4.1}$$

where $\dot{v}$, or a_d, represents the drag-induced acceleration, ρ the local atmospheric mass density, v the velocity of the satellite, v_a the local velocity of the atmosphere, e_{v-v_a} the directional unit vector of relative velocity, and the inverse ballistic coefficient is represented by β (see Equation 2.3). From this equation, density can be directly determined when β, $\dot{v}$, and v_a are known. The acceleration can be measured by examining orbital changes through for instance TLE data or accelerometers on-board the satellite, resulting in orbit-derived density and accelerometer-derived density (see Equation 2.2) respectively. The orbit-derived mass density is discussed in the next subsection. The determination of β is discussed in subsection 4.1.2.

Drag acceleration changes the trajectories of orbiting bodies, and when this change is observed time-integrated values of $\dot{v}$, and thus ρ can be derived. The primary effect of drag acceleration is the change in semi-major axis which decreases over time. The perturbation theory gives:

$$\frac{da}{dt} = -\frac{a^2 \rho \beta |v - v_a|(v - v_a) * v}{GM} \tag{4.2}$$

where a represents the semi-major axis, G the gravitational constant and M the central body mass (in this case, that of Earth). In their paper, J. Picone, J. Emmert, and J. Lean[39] estimated the wind-related error in the drag force to be largest for an equatorial orbit and for altitude below 400 km; the peak average error is estimated to be around 3%. The changes in the acceleration are not measured instantaneous, but are observations with non-uniform time-spacing. Therefore, the density derived from this approach are average along the trajectory. The temporal resolution is generally in hours to days, depending on the observation instruments and the priority the object has. The ISS orbiting Earth in LEO will have more frequently updated TLEs as compared to space debris in GEO.

Furthermore, additional perturbing forces have to be taken into account. The forces associated with the aspherical gravitational field of Earth are typical larger than the drag, but are periodic for altitude and thus average out[14]. Moreover, the radiation pressure perturbation will have to be subtracted from the observation data before any drag accelerations data can be extracted. How these perturbations are computed and the effect this has on the TLE computations will be discussed further in section 3.2.

4.1.2. Estimation of model-dependent β and ρ values

As mentioned before, the estimation of the ballistic coefficient β and thermospheric mass density ρ is still difficult due to lack of true β and ρ values and simplifications in the estimated β and ρ values. Model-based estimated β and ρ values are however computed more often by analyzing TLE data. In 2013, J. Sang, J. Bennett, and C. Smith[47] presented a method to estimate the ballistic coefficient of LEO debris objects from historical TLE data, using the drag perturbation equation. Studies on β are mainly related to the estimation of the drag coefficient and attitude of the satellite, though mass might also be a factor of uncertainty. For for instance spherical objects with known surface properties, C_D can be theoretically computed depending on the atmospheric gas composition and mass of the satellite, as the reference area will remain the same. B. Bowman and K. Moe, 2005[4] have concluded that the C_D value computed with these theoretical methods have an uncertainty of only 3% and have used this method to calibrate the Jacchia-70 thermospheric model.

In 2009, J.T. Emmert[13] used TLE data to estimate thermospheric mass densities and estimated his accuracy to be 2%, and for the long-term accuracy around 5 - 10%. The long-term accuracy is limited due to geometry uncertainties in his used reference spherical object, the Starshine 1. Furthermore, variations of the drag coefficient with ambient composition haven't been considered in his method, which he recommends to be done in the future. However, an important aspect is in how he estimated the ballistic coefficient. In his paper published in 2005[39], a method to estimate the inverse ballistic coefficient for an irregularly shaped object ("A"') from a near-spherical reference object ("S") is described:

$$B_A^{E\star} = B_A^M \frac{B_S^E}{B_S^M} \tag{4.3}$$

where the M stands for the model derived ballistic coefficient by considering mass densities computed by NRLMSISE-00, the E stands for the estimated ballistic coefficient, and $E\star$ for the model-based estimated ballistic coefficient. This method is only possible if the reference object is in the vicinity of object A, otherwise the computed mass densities would deviate too much from each other. The more accurately B_S^E and the ambient composition and density are known, the higher the accuracy of $B_A^{E\star}$ can be derived. Here the usefulness of a calibration sphere is again highlighted thanks to the consistent attitude-independent reference area.

The model-based estimation of the ballistic coefficient $B_A^{E\star}$ can be done at a single point in time and position in the orbit, or taken over a period of time. If a non-spherical object is tumbling, the estimated ballistic coefficient derived at a single point in time might be an outlier as opposed to the average ballistic coefficient.

Table 4.1: Objects groups for the beta estimation algorithm, with the spherical parent satellites having pre-determined β values.

Subscript	**Description**
$\{S\}$	Parent object - satellite with B^E_S. Initially from spherical satellite database
$\{A\}$	Child object - Satellite for $B^{E\star}_A$. Consisting solely of 1U CubeSats.

Table 4.2: Time spans for minimum TLE pair julian date spread, β^M evaluation period, and integration time steps

Condition	**Time span**
TLE pair t_{ij}	≥ 3 days
B^M evaluation period t_e	$\simeq 30$ days
Integration time step in Equation 4.4	1 minute

Ambient composition will change the ballistic coefficient over time as well, furthermore reducing the efficiency of determining $B^{E\star}_A$ at a single point in time. It is however also possible to evaluate $B^{E\star}_A$ at multiple instances in a given time period and derive an averaged value for the object, or to retrieve the change in drag coefficient from it for research in how satellite bodies interact with the thermospheric gas composition. For the β estimation procedure discussed in this chapter, a single $B^{E\star}_A$ is computed with the TLE pair covering at least a period $t_{ij} > 3$ days to reduce the effect observation errors have on the TLE pair. This is repeated with new >3 day TLE pairs within a total evaluation period of $t_e = 30$ days, thus resulting in thirty different $B^{E\star}_A$ values, which are subsequently averaged out resulting in the estimated β of object A. Table 4.2 summarizes the used evaluation periods and integration time steps.

In Equation 4.3, the value B^E_S is already known and assumed constant over the evaluation period. $B^M_{\Delta t}(t_{ij})$ is left to be determined, representing the ballistic coefficient $C_D A_{ref}/m$ of the objects, and basically the relation between the observed orbit-derived mass density and model-estimated mass density. These values are calculated using the following equation:

$$B^M_{\Delta t}(t_{ij}) = \frac{\frac{2}{3}\mu^{2/3}[n_M(t_{ij})]^{-1/3}\Delta_{ij} n_M}{\int_{t_j}^{t_i} \rho^M v^3 F dt} \tag{4.4}$$

where $\mu = GM$ is the gravitational constant, n_M the Kozai mean motion of the orbit, i and j the indices of the pair of TLE's, ρ_M the mass density from the NRLMSISE-00 model, v the orbital velocity, and F the dimensionless windfactor, see Equation 5.3. The propagator SGP4 is used for the orbital trajectory computation of each TLE [18].

With object A having a newly estimated $B^{E\star}_A$, this object can subsequently be inserted in the S database with its β becoming a new B^E_S for the next iteration. With this iterative approach, new objects in the A database can be matched and have its β estimated. Within this report, all spherical satellite database have the subscript S and act in the first iteration as the parent for all matched child CubeSats with subscript A. The next iteration, the objects with estimated $B^{E\star}_A$ values become the parents with subscript S to interact with the remaining child CubeSats, and so forth.

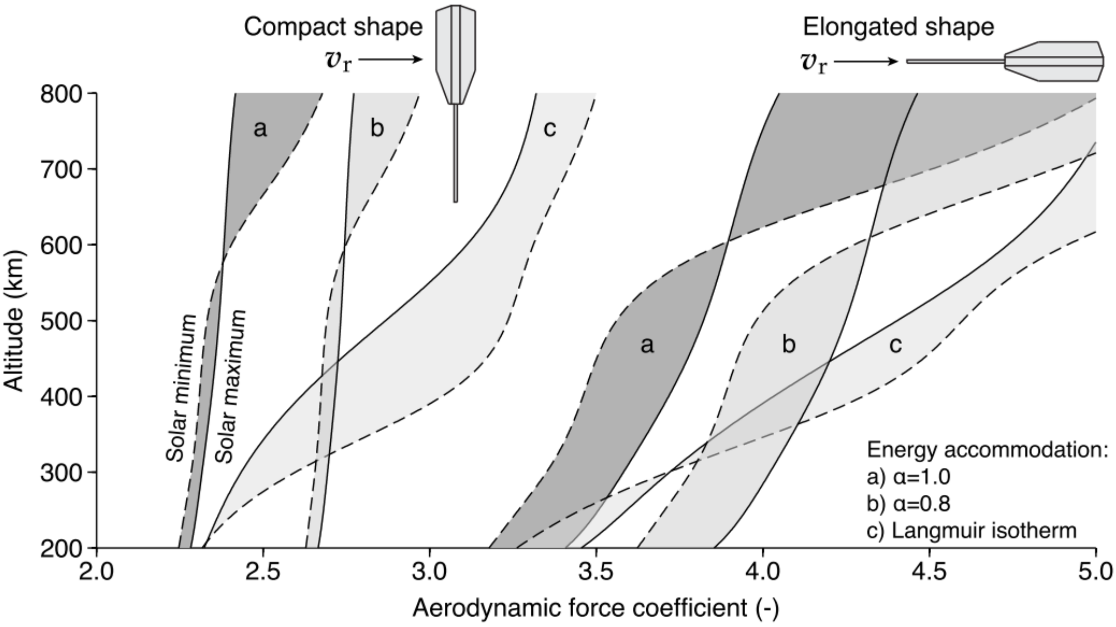


Figure 4.1: Drag coefficient variation of geometry and altitude[11]

4.2. Variable Drag Coefficient

The ballistic coefficient estimated in section 4.1 is from an associated epoch, an epoch that has its own specific atmospheric conditions. When time and position progress, these conditions start deviating from the original conditions, either due to the thermosphere expanding or contracting due to solar activity, or simply because the satellite is undergoing decay and thus starting to orbit through lower altitudes, consequently having the satellite experience a varying C_D instead of a constant value. The implication this can have on β is not insignificant, as β is directly related to the drag coefficient C_D, see Equation 2.3.

This chapter shall focus on the determination of a satellite's drag coefficient as a function of gas composition and geometry, starting with the underlying theory, the current approach on computing C_D, and how this approach can be included in the orbit propagation software such that β estimates start varying over time as well.

Theory: the C_D varies over time and location due to changing atmospheric conditions such as temperature and ambient ga composition and the impact these conditions have on the drag coefficient.

Problem: in most propagation methods a constant around $C_D = 2.2$[49][28] is assumed, as well as constant energy accommodation coefficients α

Need: a tool that determines the drag coefficient of a cube and sphere as a function of ambient gas composition and have this incorporated in the propagation software.

- **Drag coefficient for a sphere** shall be used to provide better initial of the ballistic coefficients of the spheres from the spherical satellite database as their mass and diameter are already well-documented and known.
- **Drag coefficient for a cube** shall be used to determine the change in estimated ballistic coefficient β_E from section 4.1 between first the epoch of the β_E estimation and start of propagation period and second over the complete propagation period.

The drag experienced by a satellite in orbit is solely due to the interaction its geometry has with the ambient gas particles. These particles hit the satellite under a certain angle, depending on the satellite's geometry

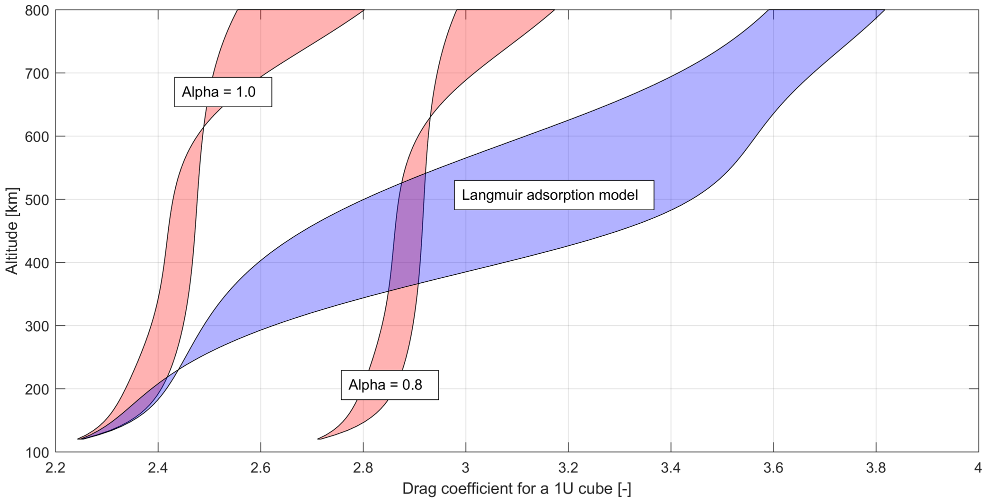


Figure 4.2: C_D algorithm output for a 1U CubeSat for different energy accommodation coefficients as a function of altitude.

and attitude, and thereby decelerate the satellite. The drag coefficient C_D is a dimensionless parameter used to quantify the object's drag or resistance behavior in a fluid environment - in the case of the near-vacuum of Earth's atmosphere the 'fluid environment' is considered the sparse presence of rarefied gas. Having the capability to analytically determine C_D and eventually the change of C_D over the propagation period is an important facet of this research. This chapter shall focus on two ways of computing the drag coefficient, one being applied to spherical shapes and the other for cubical shapes.

The theory shall be primarily based on Sentman's rarified aerodynamic equations, explained in detail in the upcoming sections. Within these equations, the energy accommodation coefficient α plays an important role as well, a coefficient that is often considered constant with a value ranging between 1.0 and 0.8, though it is also possible to vary this coefficient as a function of the presence of ambient atomic oxygen. The impact a variable drag coefficient has on the propagation results shall be assessed for these two conditions, one where α is kept constant and one where it is varied in accordance with Langmuir's adsorption model. This shall be discussed further in section 4.2.

4.2.1. Sentman's rarified aerodynamic equations

A 1U CubeSat can be seen as a block of six panels, each of the same dimensions of 0.1 x 0.1 meter. These panels are interacting with gas particles found in the thermosphere - though not present in a large quantity these particles are enough to eventually decelerate the spacecraft until re-entry. To assess determine the drag coefficient of this 1U cubesat as a function of local gas composition, Sentman's rarified aerodynamic equations are used. These equations have also been used in the work of E. Doornbos[11], which includes data results which shall be used for verification of the implementation of these equations in subsection 5.4.3.

First let's examine a single flat panel. Depending on the orientation (or θ angle) the drag, lift, and normal vectors can be determined according to Equation 4.5, Equation 4.6, and Equation 4.7, see visualized in Figure 4.3. Though the lift coefficient C_L shall not be used in this thesis the complete formula shall still be verified in subsection 5.4.3 for the verification of correct formula implementation.

$$\hat{u}_D = \frac{v_r}{||v_r||} \tag{4.5}$$

$$\hat{u}_{L,i} = \frac{(\hat{u}_D x \hat{n}_i) x \hat{u}_D}{||(\hat{u}_D x \hat{n}_i) x \hat{u}_D||} \tag{4.6}$$

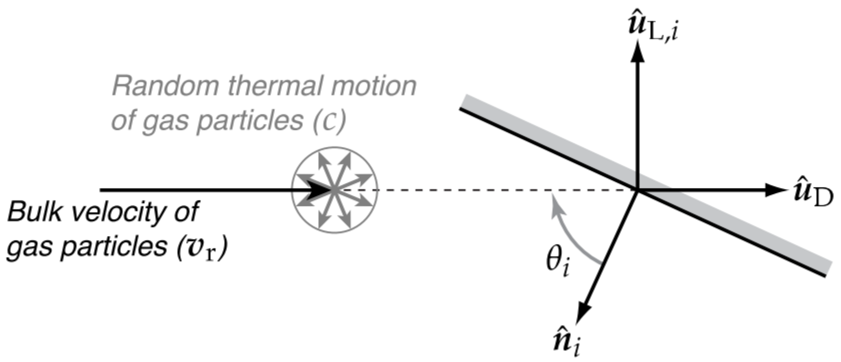


Figure 4.3: Definition of the velocity components and unit vectors used in the aerodynamic calculations for a flat panel[11].

$$\gamma_i = cos(\theta_i) = -\hat{U}_D \cdot \hat{n}_i, \qquad l_i = -\hat{u}_L \cdot \hat{n}_i \tag{4.7}$$

Having defined the angles and normal vectors, the impact various gas composition have on each panel can be assessed. It should be noted that even when γ = 90 degrees, there shall still be a portion of drag coefficient present due to the random thermal motion of the gas flowing over the panel. It is for this reason that long slender satellites experience a large drag coefficient as can be seen in Figure 4.1. The formulas for the drag and lift coefficient, used in the same manner as [53] and [30], are presented in Equation 4.8 and Equation 4.9.

$$C_{D,i,j} = \left[\frac{P_{i,j}}{\sqrt{\pi}} + \gamma_i Q_j Z_{i,j} + \frac{\gamma_i}{2}\frac{v_{re}}{v_{inc}}(\gamma_i \sqrt{\pi} Z_{i,j} + P_{i,j})\right]\frac{A_i}{A_{ref}} \tag{4.8}$$

$$C_{L,i,j} = \left[l_i Q_j Z_{i,j} + \frac{l_i}{2}\frac{v_{re}}{v_{inc}}(\gamma_i \sqrt{\pi} Z_{i,j} + P_{i,j})\right]\frac{A_i}{A_{ref}} \tag{4.9}$$

where

$$G_j = \frac{1}{2S_j^2}, \qquad P_{i,j} = \frac{1}{S_j}\exp(-\gamma^2 S_j^{\gamma}), \qquad Q_j = 1 + G_j, \qquad Z_{i,j} = 1 + \mathrm{erf}(\gamma_i S_j) \tag{4.10}$$

With the error function erf being defined as:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-y^2)dy \tag{4.11}$$

The relation a single particle has to the drag coefficient is defined in the speed ratio S of the bulk velocity to the most probable thermal velocity, see Equation 4.12. This most probable thermal velocity c_{mp} of molecules and atoms is a function of the particle's molecular mass m_j and presented in the gas kinetic theory[3]. The subscript *j* represents the different molecular masses present in the ambient gas composition. For this thesis, and as output from the NRLMSISE00 thermospheric model: $[j = He, O, N_2, O_2, Ar, H, N$,and atomic O]. Furthermore, k_b is the Boltzmann constant and T the local temperature (second temperature output from NRLMSISE00).

$$c_{mp,j} = \sqrt{2\frac{k_b}{m_j}T}, \qquad S_j = \frac{v_r}{c_{mp,j}} \tag{4.12}$$

Both Equation 4.8 and Equation 4.9 rely on the ratio of the velocity of the re-emitted particles (v_{re}) to that of the incoming particles ($v_{inc} = v_r$). This ratio can be described as a function of the energy coefficient α and the wall temperature T_W as presented in Equation 4.13. This function has been proposed by [23] after analysing the velocity ratio derivation presented by [30].

$$\frac{v_{re}}{v_{inc}} = \sqrt{\frac{1}{2}\left[1+\alpha\left(\frac{4RT_W}{v_{inc}^2}-1\right)\right]} \tag{4.13}$$

R represents the gas constant, which is Avogado's constant Na multiplied by the Boltzman Constant k_b. Throughout the research a value of R = 8.314459 was used.

Using the above equations, the drag and lift coefficient of a single panel can be derived for an arbitrary particle with a known molecular mass as a function of the attitude related angle θ. Theoretically one could simulate an arbitrary body under different attitude conditions as long as this body has been build up by multiple panel representations. To get the total aerodynamic force on the body the lift and drag coefficients for each panel will have to be added and multiplied by the particle's density ratio to the total density of the flow, see Equation 4.14. The subscript *j* relates to the different molecular masses and the subscript *i* to each panel used to construct the satellite's body.

$$C_a = \sum_i \sum_j \frac{\rho_j}{\rho}(C_{D,i,j}\,\hat{u}_{D,i} + C_{L,i,j}\hat{U}_{L,i}) \tag{4.14}$$

For this thesis, this method is solely used to determine the drag coefficient of a non-rotating 1U CubeSat. Eventually this particular method could be extended to include a rotating body, though that is left to be out of the scope of this research. The verification of the implementation of subsequent equations is performed in subsection 5.4.3, and the final discussion on the effect the energy accommodation coefficient has on the propagation accuracy can be found in subsection 8.1.4.

4.2.2. Drag Coefficient for a sphere

A different method is used for the determination of the drag coefficient of a sphere. Where for a single cube only two orientation angles θ had to be considered (0 and 90) and simply added for one frontal plate and four side panels, the method for a sphere does not use the same panel principle. Instead, Pilinski[41] provides a method to determine the drag coefficient of a sphere as function of ambient gas composition by assessing the incoming kinetic temperature with the re-emitted kinetic temperature of each gas particle in the freestream flow.

$$T_{k,i} = \frac{mv_i^2}{3k)b} \tag{4.15}$$

$$T_{k,r} = \frac{m}{3k_b}v_i^2(1-\alpha) + \alpha T_w \tag{4.16}$$

where $T_{k,i}$ is the kinetic temperature carried to the surface by an incoming molecule. Furthermore, $T_{k,r}$ represents the kinetic temperature of the reflected molecule and T_W the kinetic temperature the molecule would have if it was re-emitted at the wall temperature, or simply wall temperature[41].

$$s = v_i\beta, \qquad \beta = \sqrt{\frac{m_j}{2k_bT_{k,i}}} \tag{4.17}$$

The speed, or velocity, ratio is denoted in Equation 4.17 where again the subscript *j* relates to the molecular mass of the incoming particle and k_b the Boltzmann constant. Together with $T_{k,i}$ and $T_{k,r}$, these are the only parameters needed to compute the drag coefficient $C_{D,j}$ as a function of an incoming particle, see Equation 4.18. The total drag coefficient $C_{D,sphere}$ is computed in a similar fashion to Equation 4.14 except without the summation of the *i* term, solely the addition of the *j* subscripts multiplied with the respective partial mass density ratio to the total mass density.

$$C_{D,j} = \frac{2s^2+1}{\sqrt{\pi}s^3} exp(-s^2) + \frac{4s^4+4s^2-1}{2s^4} erf(s) + \frac{2\sqrt{\pi}}{3s} \sqrt{T_{k,r}/T_{k,i}} \tag{4.18}$$

The advantage of this method compared to a full CFD-simulation is that a simple straight-forward analytical implementation can be performed, thereby not straining the computational time of the propagation program.

4.2.3. Langmuir's Energy Accommodation

In both the drag coefficient computations for the cube and sphere geometries the parameter α is present, representing the energy accommodation coefficient. α characterizes the behavior of gas particles in their collisions with a body surface and depends most dominantly on the composition and pressure of the gas mixture and surface material of the body. As can be seen in Figure 5.15, the energy accommodation coefficient does have a clear effect on the drag coefficient, thereby demonstrating the importance of choosing a value for α in the drag computations.

Often constant or a table look-up values of $\alpha = 1.0$ and $\alpha = 0.8$ are considered when computing C_D, as is the case with for instance the research of B. Bowman[4] and P.M. Mehta[29]. However, the presence of atomic oxygen in the ambient gas mixture affect this energy accommodation coefficient as it attaches itself to the surface layer of the body and softens the impact other particles have on the body. Pilinski [41] described it as a 'softening' of the outer plate and thereby absorbing more momentum energy from the incoming particles. With low quantities of atomic oxygen this can have a large effect on the energy accommodation, where α can theoretically start to assume values well below $\alpha = 0.8$, see Figure 5.16.

One of the cases investigated in this thesis is the addition of Langmuir's energy accommodation in the determination of the drag coefficient C_D. The formulation for this term as described by Pilinski is presented in eq. (4.19).

$$\alpha = \frac{K \cdot P}{1 + K \cdot P} \tag{4.19}$$

where there partial pressure P of atomic oxygen is defined as the product of the number density and temperature $P_O = n_O T$, both outputs of the NRLMSISE-00 model. The constant K has been given a value of 7.5 $\cdot 10^{-17}$ after fitting energy accommodation data to the curve of Figure 5.16.

Note however that this constant K is selected as an optimum fitting to certain solar activity, and that no fitting points have been present at altitudes above 500 km, apart from a single data point at 650 km. Though the inclusion of the Langmuir isotherm does provide a variable and gas composition dependent α, it might still not optimally represent the energy accommodation coefficient at different solar activity or higher altitudes. As can be seen in Figure 5.16, α quickly drops, based on the fitted data - yet in real life this curve might not be the best representation of α as a function of atomic oxygen. To investigate this a little further, chapter 7 also includes results where the energy accommodation is being kept constant at 0.8 and compared to Langmuir's term for higher altitudes.

5

Methodology

The previous chapters provided the fundamentals of numerical propagation orbit and thermospheric density models, as well as demonstrate the influence that β and C_D have on the decay of the satellite. Having described the theory, this chapter shall elaborate on the theory implementation and methodology used for this specific research. This chapter is divided in three separate section, focused on **orbit propagation**, **β estimation**, and **C_D determination** respectively. For each of these sections, the underlying theory is explained, followed by the software implementation and verification:

- **Orbit Propagation** - section 5.1 elaborates on the models and software used for the orbit propagations and the specifics of the thermospheric density model settings.
- **Ballistic coefficient estimation** - section 5.2 presents the theoretical background on deriving a ballistic coefficient from the historical TLE datasets of two objects experiencing similar atmospheric conditions. The section ends with the software implementation being discussed and verified.
- **Drag coefficient determination** - section 5.3 emphasizes on an analytical method determining C_D values for spheres and cubically shaped objects as a function of ambient gas composition. Similarly this section ends with a discussion on the software implementation and verification.

At the end of this chapter, the reader should have a clear understanding of the three pillars that make up this research. In chapter 6 the experimental set-up following from the software packages and the data formats are discussed, upon which consecutively the results are presented in chapter 7.

5.1. Orbit propagation software

Previously the necessities for adequate orbit propagation have been discussed, such as the potential candidates for any mathematical integration, the possible empirical thermospheric density models, and the perturbing forces that are all to be put together into an orbit propagation model. However, the accuracy requirements for the propagation model change depending on the scenario - for the creation of the actual orbit propagation advanced SP perturbation models are required whilst for the ballistic coefficient estimation procedure a less accurate GP perturbation model can be used, hence the earlier described decision to script the models in two different coding languages, MatLab and C++.

This section elaborates on the selection procedure for the atmospheric density model, the integrators, and the perturbation models, as well as how these could be recreated by any researcher wishing to verify this work. At the end of this section an overview of the final propagation model, perturbing forces, and integrator selection is presented.

5.1.1. Thermospheric density model

As reviewed in section 3.4, there are three candidate models for this thesis research - **JB2008**, **NRLMSISE-00**, and **DTM2013**. As a reminder, these three candidates have been selected based on open-source availability, accuracy, and ease of integration. Since assessing the accuracy for these models in detail is not the essential part of this thesis, and as the accuracies do not deviate much from each other to begin with, the decision was made to incorporate NRLMSISE-00 based on two main criteria:

- NRLMSISE-00 is directly available and incorporated in the MatLab and TUDAT libraries.
- NRLMSISE-00 has been used in research of J.T. Emmert[13] and J.M. Picone[39] which form the fundamental of this thesis.

NRLMSISE-00, which stands for *Naval Research Laboratory* and *Mass Spectrometer and Incoherent Scatter Radar*, is the latest model in the NRLMSIS family. As input, this model relies on F10.7 data (81 day average and daily), geomagnetic indices (daily), and aspects such as date, time, and location in geodetic altitude, geodetic latitude, and longitude. The model and the corresponding NRLMSIS database include (1) total mass density from satellite accelerometers and from previous OD data of Jacchia and Barlier, (2) temperature from incoherent scatter radar, and (3) molecular oxygen number density $[O_2]$[38]. The latter allows for oxygen contributions to the total mass density at high altitudes, commonly starting from an altitude of 500 km. It bases its density computations on a history of a_p values for 57 hours prior to the time of interest[5]

The NRLMSISE-00 model incorporates the advantages of the other two models Jacchia-70 and MSISE-90, which are the improved predictions of the MSIS-class models in winter conditions and higher accuracy of the Jacchia-70 model under a combination of summer, low solar activity, high latitudes, and high altitudes. In their paper, Picone, Hedin, and Drob[38], a detailed comparison between the achievements of NRLMSISE-00, its predecessor MSISE-90, and the model Jacchia-70 is presented. Here it is demonstrated that the NRLMSISE-00 model considers both regimes together and achieves improved predictions in comparison to MSISE-90 and Jacchia-70. The DTM2013 has not been selected since when selection the best suited model there was only one evaluation of the DTM2013 model present[6], which was an evaluation by an employee of the institute responsible for the development of DTM2013 - therefore it was decided to not consider DTM2013 until additional evaluations would have been performed.

For NRLMSISE-00 to be used, the variables depicted in Table C.1 are to be inserted. These variables are related to position in the geodetic WGS-84 ECEF reference frame, epoch, F10.7 and APH values, and an array of flags for the 23 setting parameters, see Table C.3. The F10.7 and APH values are automatically downloaded from *Celestrak.com* and stored in a database for use with each respective epoch. As output, the model provides the local temperatures and gas density compositions, see Table C.2.

In all computations done with NRLMSISE, the flags are by default set to 1 except for Flag(9) = -1. This setting uses the entire matrix APH rather than just APH(:,1). Furthermore, the OTYPE input is set to include Oxygen for the total mass density output. The sixth column in the density output matrix is used for the total mass densities as opposed to summation of the individual gas density components.

NRLMSISE-00 requires the latitude, longitude, and altitude in the ECEF coordinate system, yet the output of the SGP4 propagator is in TEME coordinates. In subsection 5.1.3 the procedure to perform these transformations are discussed. However, for this transformation additional inputs are required, namely the polar motion variables x_p and y_p and the UT1-UTC timing difference, see section 2.1. An EOP database consisting of the EOP data from 1962 to now has been downloaded from IERS, and used in the transformation from TEME to ECEF and consequently to the J2000 frames.

5.1.2. Propagation models

Having established the empirical thermospheric density model, the propagation methods can be discussed. As the reader might has seen in previous sections, there shall be two orbital propagation methods used in this research - one for conversion of TLE data and low-accuracy propagation and another for a high-accuracy propagation:

- **General Perturbation** propagation with SGP4 in the MatLab 2016a environment. Used for converting TLE data to Cartesian coordinates and orbital elements, and for the ballistic coefficient estimation procedure.
- **Special Perturbation** propagation using the Runge-Kutta Fehlberg 7(8) integrator performed with Cowell's method. Due to its computationally intensiveness this propagation method is performed in compiled C++ in the TUDAT environment.

The software of both models, as well as instructions on how to integrate these in one's own propagation model, is all open-source. Details on both propagators, why they have been selected, and their usage are presented in the next two subsections. The special perturbation model allows for more accurate propagation results, though as mentioned in computationally intensive. Since SGP4 is also a necessity for TLE coordinate conversion, the decision was made to also perform the ballistic coefficient estimation procedure with this propagator, as for this procedure a high accuracy is not required.

5.1.3. SGP4 propagator

SGP4 has been selected as a necessity for the conversion from TLE data to TEME coordinates. This propagator shall also be used for establishing TLE-derived verification orbits. A general perturbation propagator, SGP4 is unsuited for long-term orbit propagation without a significant loss of accuracy - hence a second, special perturbation propagator is selected for the actual orbit lifetime predictions. SGP4 has an open-source software package based on Vallado's research readily available use with MatLab. This section elaborates on the origin of the script, the in- and outputs, and the integration and usage in MatLab. The SGP4 propagator is directly downloaded from *Celestrak.com*. The script is open-source, fully compatible with MatLab, and programmed by David Vallado based on the principles described in his book Fundamentals of Astrodynamics and Applications[9]. The SGP4 propagator is used to fulfill three functions:

- **1)** TLE conversion to Cartesian coordinates in the TEME frame and the subsequent conversion to orbital elements.
- **2)** Propagation of two orbits in a 30-day interval used for the estimation of an updated ballistic coefficient, see section 4.1.
- **3)** Construction of verification orbits with a set of historical TLE data, to be used for the performance assessment of the TUDAT propagated orbit outputs.

In order to properly execute these three functions, the SGP4 software package will need to be set up and used as intended. David Vallado has described this procedure in detail, and is listed in , Table 5.1 - note that the first function of TLE conversion relates to the first procedure. The second procedure, involves the creation of the latitude and longitude ephemeris, which are needed as inputs for the NRLMSISE-00 atmospheric density program, and the last procedure transforms the TEME coordinates to J2000 for input in the TUDAT model.

The three procedures discussed in Table 5.1 are the backbone for determining the initial conditions for both the SGP4 and TUDAT propagation, the use of NRLMSISE-00, and the creating of the verification orbits. When the initial conditions are retrieved from the TLE data, they are converted to J2000 coordinates and written to a .text file which is consecutively read by TUDAT. The program has been set up such that an array of objects can have their initial conditions determined simultaneously, as well have the propagation in MatLab and TUDAT run parallel to each other.

Table 5.1: Vallado's SGP4 software architecture

	Creating Cartesian position and velocity ephemeris from a TLE
1	Read in the data file containing the TLE information
2	Convert the TLE information (twoline2rv.m)
3	Initialize SGP4 (sgp4.m with 0.0 time)
4	Loop through the desired time period:
	-Call SGP4 and obtain the position and velocity vectors in TEME (sgp4.m)
	-Write out the relevant parameters: time since epoch, TEME position and velocity
5	End loop

	Create a latitude and longitude ephemeris from a TLE
1	Read in the data file containing the TLE information
2	Convert the TLE information (twoline2rv.m)
3	Initialize SGP4 (sgp4.m with 0.0 time)
4	Loop through the desired time period:
	-Call SGP4 and obtain the position and velocity vectors in TEME (sgp4.m)
	-Convert the position and velocity vectors to latitude and longitude:
	-Rotate from the TEME to ECEF coordinate frame (teme2ecef.m)
	-Use the ECEF vectors to find latitude and longitude (ijk2ll.m)
	Write out the relevant parameters: time since epoch, latitude, longitude
5	End loop

	Converting a TEME ephemeris to a J2000 ephemeris
1	Convert the position and velocity vectors from TEME to ECF (teme2ecef.m)
2	Convert from ECEF to J2000 (ecef2eci.m)
3	Write out the relevant parameters: time since epoch, ECI position and velocity

Regarding the main MatLab file, *sgp4.m*, two internal operation modes are used. The first operation mode is **the manual 'm' mode**, where a single TLE is converted to Cartesian coordinates at solely its epoch period. This mode is used as a preparation for the verification orbit. Multiple TLE can be converted at once using the manual mode, resulting in a set of epochs related to each individual TLE. With these epochs defined, middle points between the epochs can be determined which can in turn be related to a specific number of minutes before and after an epoch. So let's say for a verification orbit consisting of an arbitrary number of TLE the TLE data files will have to be modified to include the time to and from the two closest middle points of their neighboring epochs, the procedure to do so would look as described above is visualized in Figure 5.2.

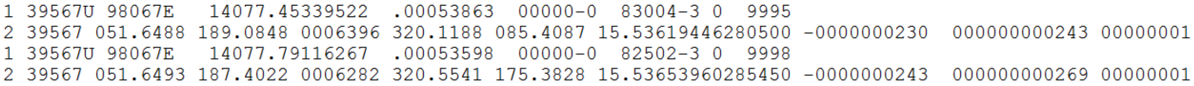
- 1. Request start and end epoch of the verification orbit
- 2. Retrieve all TLE data within this epoch range
- 3. Extract the epochs from each TLE
- 4. Find the middle point between each consecutive TLE
- 5. Determine minutes between original epoch and middle point of the two neighboring TLEs.
- 6. Store these values back into the original TLE in the exact format represented in Figure 5.1.

The resulting format, displayed in Figure 5.1 and Table 5.2, are exactly the input needed for the second internal operation mode of the SGP4 propagator - **the verification 'v' orbit mode**. In this mode, a TLE set is read with for each TLE the start and end time in minutes before and after the respective epoch. With the verification mode a verification orbit can be set up based on the historical TLE data of an object, and the initial conditions of the first TLE are stored and send to TUDAT for the actual propagation.

Table 5.2: TLE format for verification orbit input of the SGP4 propagator.

TLE Line	Position	Description
Line 2	Character 71 to 81	Start epoch of propagation, in minutes before TLE epoch.
Line 2	Character 84 to 95	End epoch of propagation, in minutes after TLE epoch.
Line 2	Character 97 to 104	Step size of propagation, in minutes.

Output of the SGP4 propagation are coordinates in the TEME reference frame, see section 2.1. Using the *teme2ecef.m* function these coordinates are transformation to ECEF coordinates - a necessity when one intends to use the NRLMSISE-00 atmospheric density model. The open-source SGP4 software is well-documented and is not difficult to use, though special care has to be taken to follow the right steps and to properly modify the TLE .txt files when creating a verification orbit.



```
1 39567U 98067E   14077.45339522  .00053863  00000-0  83004-3 0  9995
2 39567 051.6488 189.0848 0006396 320.1188 085.4087 15.53619446280500 -0000000230  000000000243 00000001
1 39567U 98067E   14077.79116267  .00053598  00000-0  82502-3 0  9998
2 39567 051.6493 187.4022 0006282 320.5541 175.3828 15.53653960285450 -0000000243  000000000269 00000001
```

Figure 5.1: Visualisation of the TLE format for verification orbit input of the SGP4 propagator

5.1.4. Runge-Kutta Fehlberg 7(8)

SGP4 is certainly beneficial for the functions described earlier, yet it lacks in accuracy in long term orbit propagation - that is, when looking at more than a couple of days. In section 3.2 the advantages of a special perturbation propagator have been discussed, along with Runge-Kutta integration methods. The need for a SP propagator is high, as without it any potential advances made with the estimation of the ballistic coefficient and incorporation of a variable drag coefficient might not be retrievable from the data as the errors induced by the SGP4 propagator simply are in a higher order. Therefore, an accurate numerical integrator is required - which is provided by the Runge-Kutta Fehlberg 7(8) integrator.

The Runge-Kutta Fehlberg 7(8) integrator shall be the backbone of the SP propagator, together with the NRLMSISE-00 density model. To reduce the computation time, a decision has been made to incorporate the SP propagator in C++, and more specifically using the TUDAT environment, see subsection 5.1.5. The incorporation of the Runge-Kutta Fehlberg 7(8) integrator in TUDAT can be done with a simple line of code, which due to the way the TUDAT library is set up can be done in a simple 'plug-and-play' manner - the script behind the integrator is already verified and present in the TUDAT library.

Table 5.3: Input settings Runge-Kutta-Fehlberg 7(8) and accompanying evaluation periods.

Description	Value
Absolute Error Tolerance	1.0E-12
Safety Factor for next stepsize	0.8
Maximum Factor Increase for next stepsize	4.0
Minimum Factor Decrease for next stepsize	0.1
TLE pair t_{ij}	≥ 3 days
B^M evaluation period t_e	≃ 30 days
Integration time step in Equation 4.4	1 minute

The additional input settings for the selected Runga-Kutta Fehlberg 7(8) are described in Table 5.3. In this table, the timing periods used for the propagation are presented as well.

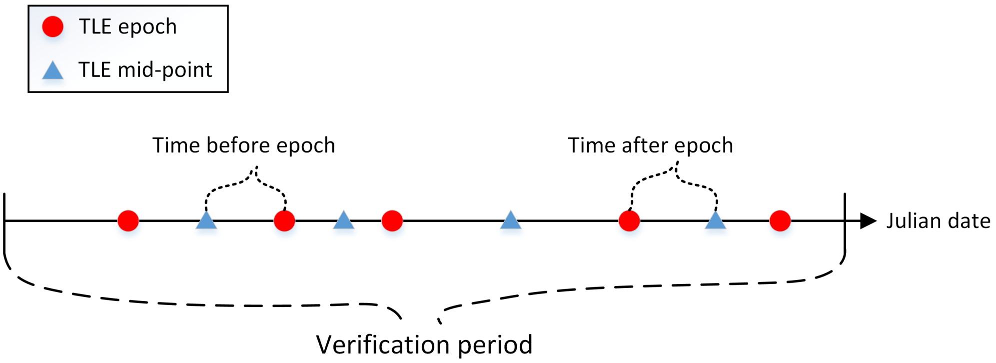


Figure 5.2: Visualization of modification TLE for verification

5.1.5. TU Delft Astrodynamics Toolbox - TUDAT

The software is written in two languages - MatLab and C++. MatLab is a widely used programming environment and shall therefore, by itself, not be elaborated on. TUDAT on the other hand is an environment most probable not encountered often outside of astrodynamic computations and shall therefore be discussed briefly in this section.

The TU Delft Astrodynamics Toolbox (Tudat) library is a set of open-source C++ software libraries, developed and maintained by staff and students in the Astrodynamics and Space Missions research group from Delft, University of Technology. This toolbox provides a user with a robust and large set of astrodynamic applications, such as atmospheric models, mathematical integrators, and acceleration models. These C++ libraries can be implemented into a custom astrodynamics application, as has been done for this research. The choice for TUDAT was made due to its (open-source) availability, the computationally fast C++ programming language (up to 50x faster compared to MatLab), and the customization possibilities. Here, an explanation regarding the various libraries used for this research, as well as a road map of the software structure.

Setting up TUDAT libraries

A detailed installation guide on how to use the TUDAT libraries can be found on tudat.tudelft.nl, however a quick recap shall be presented here. Via GitHub a user can download the whole TUDAT library, including the Eigen, CSpice, and Boost packages. The data on GitHub is actively and regularly updated by the Astrodynamics and Space Missions research group, so be sure to frequently check if you have the latest version. A handy tool for this is SmartGit, a program which can search for any updates in the libraries on your computer and the files available through GitHub. Any new updates can be incorporated automatically on your local machine.

To be able to compile the files, a C++ compiler and a software development program should be present on the local machine. TU Delft advises using CMake as the C++ compiler and QT Creator as software developer. Installation of these programs are straight forward, after which the user can start compiling the libraries. Compilation of the complete TUDAT library can take up to an hour, but after all functionality is present. Before heading into making a custom application, it is advised to inspect the overall structure and components of TUDAT, including the functionality of Boost, CSpice, and Eigen. Furthermore, there are frequently updated example applications, as well as an easy to use template application. The example applications provide a solid basis on how to set up a new application using the pre-made template. The functions of the propagation model created in TUDAT are presented in Table 5.4.

With these functions present in the propagation software, it is possible for initial conditions in the J2000 frame to be propagated into time using the aforementioned NRLMSISE-00 model and integration method. The initial conditions are created in MatLab from transformed TLE data to TEME coordinates, a process described further in subsection 5.1.3, and propagated according to the scenarios described in chapter 6. The objects subject to the propagation are stated in section 6.1. Depending on the scenario, the option for the inclusion of the C_D calculation is turned on or off. All output is stored as .dat files and analyzed in MatLab.

Table 5.4: Functionality of TUDAT propagation script

Function	Description
Input and Output	Loading the initial conditions set up in MatLab and the WGS84 coefficients. Data output stored automatically to .dat files.
Body Definition	Defining the celestial bodies Sun, Earth, Mars, Moon, and Venus.
Body Objects	Creates body properties such as the spherical harmonic gravity field and atmospheric settings for Earth.
Frame Orientation	Setting all frames to the J2000 ECEF frame.
S/C Definition	Defining the S/C using input values for the mass, reference area, and initial drag coefficient.
C_D Calculation	Option to continuously update the drag coefficient as function of ambient gas composition and S/C geometry, see section 4.2.
S/C Objects	Creates S/C properties such as aerodynamic coefficient settings and satellite radiation pressure settings.
Set acceleration	Links the bodies with the S/C with central gravity accelerations for all celestial bodies, cannon ball radiation pressure from the Sun, and the aerodynamic acceleration from Earth's atmosphere. All accelerations are stored in a acceleration model map.
Set propagation	Defines the distance and velocity vectors (initial conditions) used in the propagation as well as defines the mathematical integrator model (RK78) and the selection of Cowell's method.
Propagate Orbit	Actual propagation of the orbit based on input values for start and end ephemeris using the defined bodies and the acceleration and propagation settings. Output is stored and saved in .dat files.

To summarize, Table 5.5 presents an overview of the used models. All apart from the SGP4 propagator are in the TUDAT environment, MatLab only uses the SGP4 propagator and NRLMSISE-00 thermospheric density model.

Table 5.5: The models and their specifics used within this research.

Propagation models	
General Perturbation	SGP4 (MatLab)
Special Perturbation	TUDAT (C++), Cowell's method

Perturbing forces	
Non-spherical Gravity	WGS-84, Spherical Harmonics up to degree 5
Solar Radiation Pressure	Cannon ball radiation pressure, A_{ref} assumed constant 1U
Atmospheric Drag	Thermospheric density from NRLMSISE-00
Third-Body Perturbations	Earth, Moon, Sun, Mars, and Venus

Integrator	
Variable step-size, Explicit	Runge-Kutta Fehlberg 7(8) absoluteErrorTolerance = 1.0E-12 safetyFactorForNextStepSize = 0.8 maximumFactorIncreaseForNextStepSize = 4.0 minimumFactorDecreaseForNextStepSize=0.1

5.2. Ballistic Coefficient Estimation

The propagation model has been discussed, and its implementation shall be verified later in this chapter. One of the two main aspects of this research focuses on the estimation of an object's ballistic coefficient - that method is discussed in more depth here. in short, the CubeSat and spherical satellite databases are analyzed to detect whether any set of sphere (parent) and CubeSat (child) can be found that experience near-similar atmospheric conditions. To establish whether a match occurs, a proximity algorithm has been set up. When a match, or a group of matches, have been found, the model-dependent ballistic coefficient of both objects are determined by examining the mean motion difference retrieved from the TLE data and combining that with the expected density output of NRLMSISE-00 for that specific trajectory. The resulting two β^M values have atmospheric density errors induced by NRLMSISE-00 incorporated in them, though due to having experienced near-similar atmospheric conditions the ratio between the two β^M values and the true β^T values are the same. Hence, with β^T known for the parent, an estimated β_E for the child can be established.

Important details to this method include a TLE filtering algorithm, the development of the proximity algorithm, and eventually the establishment of an iterative β_E estimation procedure. These aspects are discussed further in this section.

5.2.1. Two-Line Element Filtering

Every propagation performed in this research has its initial conditions and object characteristics derived from Two-Line Element data. The proper transformation from TLE data to initial conditions in the J2000 frame have been discussed in subsection 5.1.3. Though the TLE are not solely used for derivation of initial conditions - they are also used to assess an object's mean motion difference, as well as used to create verification orbits. The first, the assessment of the average mean motion and mean motion difference, is key to this β estimation method, and if possible corrupt or bad TLE data are preferably removed from the TLE database before usage. Such a TLE filtering algorithm has been set up performing a set of three filtering procedures in accordance with J.T. Emmert's paper[13].

J.T. Emmert argues that the TLEs are preferably not to be taken directly after each other, but that a separation of preferably three days is required when selecting an optimal TLE set. This is to ensure a better reading of the change of mean motion - with the difference and absolute mean motion being from here on associated to an epoch in the middle of the two TLE sets. Referred to as the *3-day monotonicity*, this filter has been incorporated in this thesis's algorithms.

Apart from solely a selection based on the julian dates of the epochs (3 days difference), a filtering based on the mean motion (element 8 on Line 2 in the TLE) can ensure TLEs containing a rise in orbit due to for instance actively controlled thrusters are discarded as well. Logically, when considering passive satellites solely experiencing drag, the mean motion is expected to rise. For a 3-day separated TLE set, a negative mean motion difference is expected to be seen, any positive difference implies either a maneuver to have taken place or noise in the TLE time series is present - these TLE sets are obviously filtered out.

A third and final filter is incorporated, one that dictates that no match between parent and child is to be made within 30 days of an object's re-entry with the atmosphere as the rapid orbital changes experienced in this phase are deemed to increase the noise of the observation data. The filter is easily applied by determining the epoch of the match and compare that with the re-entry epoch, if that exists for either the parent or child historical TLE set.

In his paper J.T. Emmert incorporates two additional filters, one to filter out highly elliptical orbits and deep space objects, and another determining the Drag-to-Noise ratio of an object to be able to exclude objects with a too low mass or too low reference area which could result in inaccurate atmospheric drag readings. As the initial CubeSat and spherical satellite databases consist solely of (near-)circular LEO spacecrafts, it has been decided to not incorporate these two filters as it would not result in any filtering from those databases.

5.2.2. Proximity algorithm

The atmospheric error can only be canceled out if the atmospheric conditions between the matched parent and child objects are considered near-similar. Looking at Figure 5.3 and Figure 5.4 it can be seen that thermospheric densities varies primarily over inclination, right ascension of the ascending node, epoch, and altitude. Figure 5.3 clearly demonstrates a variation at the two different epochs, as sthe gas composition can be seen to change significantly between the year 2000 and 2006 for a same altitude - indicating the importance epoch and altitude have when trying to establish the aforementioned similar atmospheric conditions. This document has elaborated on that in section 4.2 where the contribution a varying gas composition has on the C_D of an object and the impact this has on the ballistic coefficient is discussed.

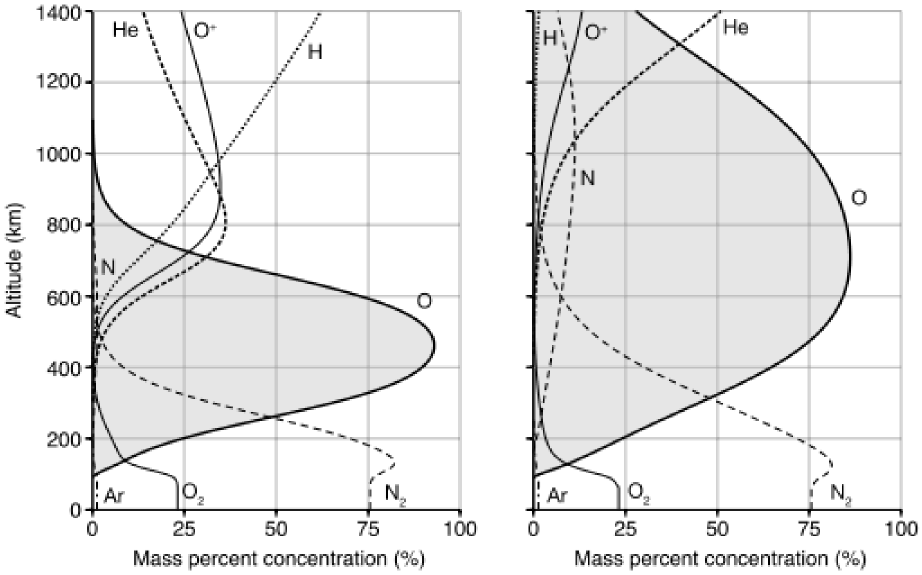


Figure 5.3: Altitude profiles of atmospheric mass percent concentrations for low (*left*) and high(*right*) solar activity levels, according to the NRLMSISE-00 model, evaluated over Delft, at 18:00 on July 15, 2000 and 2006[11].

Having established **epoch** and **altitude** as prime criteria, Figure 5.4 demonstrates how determining an inclination and RAAN range are important as well. Though this figure only represents a single epoch, it can clearly be seen that there is a density variability spread over the Earth, at the altitude of 400 kilometer in this particular case. Same as goes with the altitude and gas composition profile, these contour maps change considerably over altitude and epoch - an object with an inclination of 5° and RAAN of 30° will most likely encounter vastly different mass densities as opposed to an object with an inclination of 55° and RAAN of 10° orbiting Earth at the same altitude. For this reason, **inclination** and **RAAN** have been selected to form the other two criteria for the proximity algorithm. Note that in this particular case the wording 'proximity' is not directly related to absolute position but whether the orbital parameters of the two objects are within predefined criteria ranges.

Within the proximity algorithm, these criteria for these four orbital elements have to be set up to optimize the number of potential matches whilst still ensuring only small thermospheric density deviations. Ideally, one would aim to keep the range on epoch within a mere few days and the altitude in the tens of kilometer range as these two are most defining for the density variations, however, regarding the perfect ranges for these criteria nothing much can be said at the moment.

In chapter 6 it shall be mentioned that multiple groups are established for this research, all with different criteria sets. The specific values for these sets are clearly stated there, and are selected to later in chapter 7 discuss the influence certain ranges might have on the quality of the parent-child matches. To summarize, the four orbital element selection criteria are presented below order to their assumed priority in minimizing the range for that particular criteria:

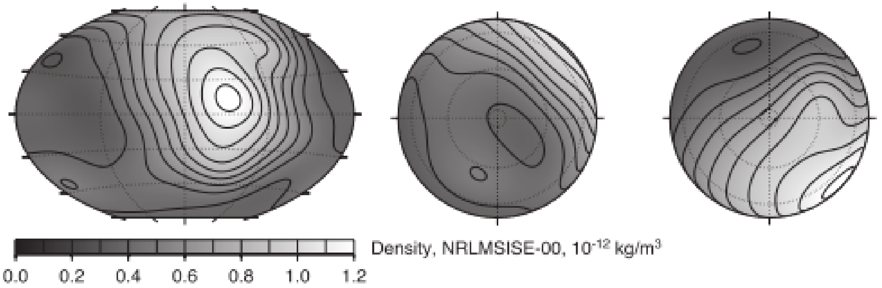


Figure 5.4: Maps of modelled density at 400 km altitude,on July 15, 2006 at 18:00:00 UTC (F10.7=70.2, ap=3). The map projections are, from left to right, a Winkel Tripel projection, and two orthographic hemispherical projections centered on the geographic South Pole and North Pole[11].

- 1)*Epoch*
- 2) *Altitude*
- 3) *Right Ascension of the Ascending Node*
- 4) *Inclination*

The criteria for determining whether a parent, being it a spherical satellite or a CubeSat with its β_E already estimated, can be paired with a child (always CubeSats) have now been established - the exact values follow in chapter 6. The proximity algorithm initiates with converting all TLEs in the spheres and CubeSat databases to their orbital elements using the SGP4 propagator. This output is stored, together with their respective row number for look-up purposes. A series of for-loops iterates over these four criteria and establishes when all criteria are met, consecutively storing the corresponding row numbers of the paired TLEs. It is possible multiple instances occur when objects meet each other - in that case the algorithm selects the TLEs corresponding to the largest spread between first encounter and last encounter. For example, if object 1077 meets object 3022 on two occasions, once covering only twenty matching TLEs and the other covering 430 matching TLEs, the latter is chosen as the best match between these objects.

There is one more step necessary before the ballistic coefficient for the CubeSats can be determined, and that is to select the Julian dates and corresponding TLEs that cover the aforementioned evaluation period t_e of 30 days. Taking the example mentioned above, the middle of the 430 TLEs is selected as the centre point and the TLEs spanning 15 days before and after this point are stored as the TLE set to be used for the β_E estimation. In the end, the large initial sphere and CubeSat databases have been reduced to solely comprise of matched objects with for at least 30 days of overlapping TLE data.

5.2.3. Ballistic coefficient estimation

Estimating the ballistic coefficient of the objects in the CubeSat database, $B_A^{E\star}$, requires the computation of two other ballistic coefficients first - namely B_A^M and B_S^M. The subscripts A refers to the child, or CubeSat, ahd S to the parent. This is done in accordance with Equation 4.4, though this does require some attention. First of all, lets focus on the numerator, which essentially is the portion related to the *orbit-derived mass density*:

$$\frac{2}{3}\mu^{2/3}[n_M(t_{ij})]^{-1/3}\Delta_{ij}n_M \tag{5.1}$$

As mentioned, t_{ij} is defined as a TLE pair at least covering 3 days, and evaluated at its mid-point. Over this set, the mean and difference between the Kozai mean motions are determined. In order to later divide this by the denominator, the amount of data points will have to be the same. Therefore, for every mid-point position of a TLE pair data points for the mean and difference in n_M are determined, resulting in a set of epochs in Julian dates with the two corresponding $\Delta_{ij}n_M$ and $n_M(t_{ij})$. The epochs are rounded to their nearest minute, after which the $\Delta_{ij} n_M$ and $n_M(t_{ij})$ are interpolated with an integration time step t_{step} of 1 minute. Hence, a m x 3 matrix is created with the first column being the Julian data, the second the interpolated average of the mean motion, and the third the interpolated difference of the mean motion.

The denominator represent the portion related to the *model-estimated mass density*, and consists of the density ρ^M, its orbital velocity v and the wind factor F:

$$\int_{t_j}^{t_i} \rho^M v^3 F dt \tag{5.2}$$

For each TLE pair in t_{ij} there is a start and end epoch. The algorithm takes the starting TLE and, with the SGP4 propagator, integrates to the end epoch of the TLE pair with a integration time of again 1 minute. The output are the position and velocity vectors in the TEME-frame, see section 2.1. The velocity vector, containing v_x, v_y and v_z are used to compute the v parameter through $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$. With the same data in the TEME-frame, the wind factor is computed in accordance with Equation 5.3. This wind factor is an approximation which King-Hele[22] argues to be a suitable one compared to for instance more realistic wind models such as HWM-07.

$$F \simeq (1 - \frac{rw}{v} cos(i))^2 \tag{5.3}$$

where r is the distance of the object from the centre of the Earth, w the angular velocity of the Earth's rotation, and i is the angle of inclination of the orbit. The plot at the third row, second column position in Figure 5.9 shows the approximation in comparison to the HWM-93 model.

The final parameter from Equation 5.2 is the model-estimated mass density, which in the software requires an additional few steps before this value is computed. The procedure below describes the acquisition process of these mass density values. Keep in mind that the same vector length as for the velocity v and wind-factor F is used - that is, the total integration period with time-steps of one minute.

- **Step 1:** Determine epoch dates for integration period.
- **Step 2:** Determine x_p and y_p values to account for polar motion, see section 2.1.
- **Step 3:** Convert position and velocity vectors from TEME to ECEF in the WGS-84 system.
- **Step 4:** Acquire APH values corresponding to the integration period.
- **Step 5:** Initiate NRLMSISE-00 model with ECEF geodetic λ, ϕ, altitude, date, and APH values. For additional settings, see subsection 5.1.1.
- **Step 6:** Store total mass density output from NRLMSISE-00.

Equation 5.2 is at this point evaluated over the integration period and subsequently used to divide Equation 5.1 resulting in the $B^M_{\Delta t}(t_{ij})$. Having performed this for both B^M_A and B^M_S (the model-dependent ballistic coefficients for parent and child), and with the value of B^E_S pre-determined (the 'true' ballistic coefficient of the parent), $B^{E\star}_A$ from Equation 4.3 is estimated and assigned to the respective child object.

To summarize, for each object, its ballistic coefficient is computed using its historical TLE observation data and the thermospheric density model NRLMSISE-00. The ballistic coefficients, B^M_S and B^M_A, both have an error introduced to it based on the errors in the density model, see section 3.4. However, this error is mitigated using Equation 4.3 through assessing the ballistic coefficient of the parent, here B^M_S experiencing similar atmospheric conditions, and whose ballistic coefficient, B^E_S, is predetermined and more accurately known. In this manner, a ballistic coefficient $B^{E\star}_A$ relieved of most of its atmospheric error is computed. Figure 5.5 shows the output ballistic coefficients on the left-hand side of Equation 4.3 of the matched spherical parent Westpac and the child CubeSat DTUSAT. As expected, the ballistic coefficients of both objects varies quite significantly over time, dominantly through atmospheric errors, and potentially also due to the object tumbling or TLE observation errors, though the general trend for both objects are reasonably similar. When evaluating Equation 4.3, the mean of both B_S^M and B_A^M are taken to compute $B_A^{E\star}$.

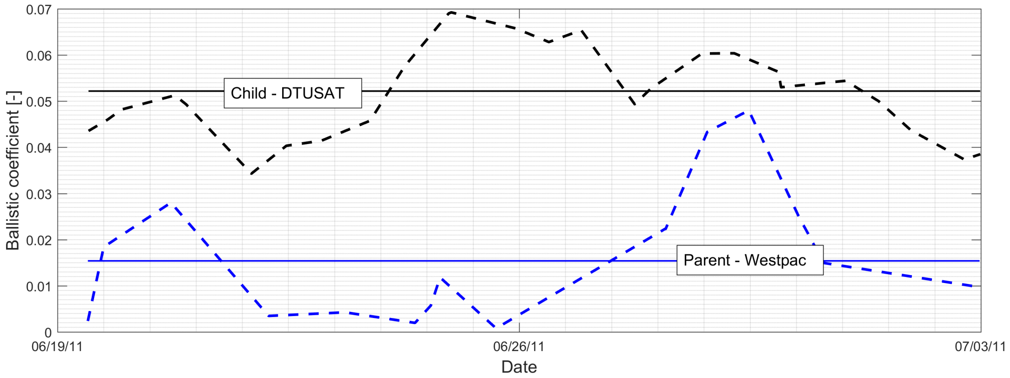


Figure 5.5: Example of left-hand side of Equation 4.3 for the matched spherical parent Westpac (blue) and the child CubeSat DTUSAT (black). The averaged lines for both satellites are considered to be B_S^M and B_A^M respectively.

5.2.4. Iteration procedure

The β estimation procedure above describes the steps necessary to first determine whether a child object is experiencing the same atmospheric conditions as a parent object and how consecutively a model-based $B_A^{E\star}$ was estimated. With the newly acquired $B_A^{E\star}$, other objects within $\{A\}$ (the child object dataset)could potentially have its ballistic coefficient estimated as well - after all, child of the first iteration might come within range of other objects in CubeSats that the original spheres didn't come even remotely close to. This can have an escalating effect, where for instance in the first iteration a single parent object meets three child objects, which in turn find an additional few child objects each, and so forth. This iteration procedure is initiated relatively simple. At the end of each iteration round, all child objects with a $B_A^{E\star}$ are removed from the child database and used to create a new parent data set.

Once indeed the algorithm detects no new matches being found, the algorithm terminates. Depending on the initial epoch and orbital element criteria set at the start, the group of CubeSats will have their ballistic coefficients estimated. A small note, a child object could potentially have been interacting with more than one parent object and have thereby its $B_A^{E\star}$ estimated multiple times as well. In that case, mean for the multiple ballistic coefficients have been taken for that specific NORAD identifier.

Since the estimation of a new $B_A^{E\star}$ takes roughly six minutes computation time on the set-up used for this thesis, intermediate data saves have been incorporated in the software. After each $B_A^{E\star}$ estimation, the value is stored both externally in an Excel file as in a .mat matrix. The β estimation software is thus only required to run once and is not required every time a new orbital lifetime is predicted. However, as a future recommendation, it could be beneficial to translate the code to C++ and have the computational time reduced.

5.3. Varying Drag Coefficient

Theoretically the drag coefficient is dependent on the ambient gas composition, the satellite's geometry, and its freestream velocity. As shown in Figure 4.1, a range of drag coefficients for cubical and spherical shapes can be observed throughout the lower (<1000km) regions of the atmosphere. Furthermore, in section 4.2 the energy accommodation coefficient α modeled by Pilinski[41] using Langmuir's isotherm, was also discussed demonstrating the effect anomalous oxygen has on the total drag coefficient. Though the aforementioned drag coefficient range was between $C_D = 2.0$ and $C_D = 3.5$, still a constant $C_D = 2.2$ is generally considered when computing the ballistic coefficient of an object orbiting Earth in LEO - an assumption that introduces a potential error source during orbit propagation. This section shall describe the process on how the theoretical background is incorporated in the software algorithms, that is in the ballistic coefficient estimation software as well as in the orbit propagation software.

Figure 5.6: Visualization of ballistic coefficient estimation and the corresponding drag coefficients used in Equation 5.4

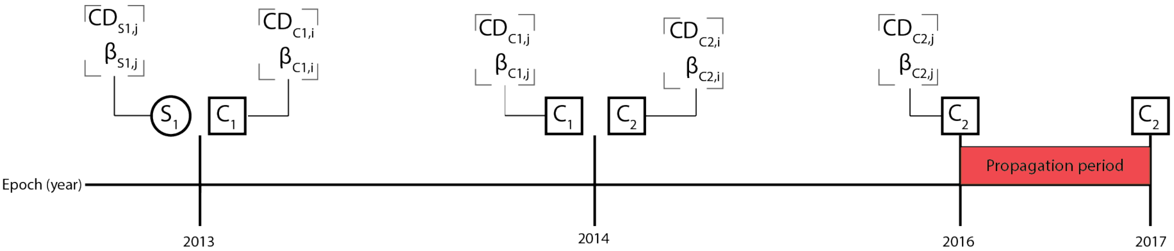


Figure 5.6 visualizes the process of estimating β_{C2} and consecutively propagating Cube C2 - a process similar to what shall eventually be performed to all β_E estimations for the scenarios described in chapter 6. In this particular case, sphere S_1 matched with Cube C1 and had its $\beta^{\star}_{cube_{C1}}$ estimated. After a time interval the orbital parameters of Cube C1 approaches those of Cube C2 and in a similar fashion $\beta^{\star}_{cube_{C2}}$, or β_{C2} is estimated at that particular epoch. Eventually Cube C2 is propagated with a modified β_{C2} value to compensate for the changed C_D.

One should keep in mind that at each β estimation epoch a C_D is associated. Apart from documenting the $\beta^{\star}_M$, it is also key to document the associated C_D value. If at another epoch the $\beta^{\star}_M$ is required again, whether it is for a new β estimation or for propagating purposes, its value will have to modified in accordance to Equation 5.4 to accommodate for the drag coefficient at that particular epoch. The assumption for this to be valid is that the mass and frontal area of the target object remains constant over time and thus the only varying parameter in the β computation is that of the drag coefficient.

$$\beta_j = \beta_i \frac{C_{D,j}}{C_{D,i}} \tag{5.4}$$

where the subscript i corresponds to the epoch at which that particular β was estimated and j the epoch at which β is again called upon. During orbit propagation the initial ballistic coefficient β_j would thus have to predetermined after which the drag coefficient is to be constantly computed and updated throughout the propagation period. The formulation for the β estimation as described in eq. (4.3) hence becomes:

$$B_A^{E\star} = B_A^M \frac{B_S^E}{B_S^M} \frac{C_{D,S,j}}{C_{D,S,i}} \tag{5.5}$$

For example, the ballistic coefficient of a spherical satellite with known diameter and mass is initially determined by assuming a $C_{D,i} = 2.2$, yet at the epoch at which Equation 5.5 the actual $C_{D,j}$ is computed and turns out to be 2.3 Hence the B_S^E is adjusted in accordance to the ratio described in Equation 5.4. The resulting $B_{A,j}^{E\star}$ is saved, though **with** its own respective epoch-associated drag coefficient according to its geometry. If B_A is also a sphere, the same value of $C_D = 2.3$ can be stored. However, as for this research B_A is a CubeSat, the $C_{D,j}$ for this geometry is computed simultaneously and stored along with $B_A^{E\star}$. For any future computation with $B_A^{E\star}$, the associated C_D is used as the $C_{D,i}$ value to be inserted in Equation 5.4.

5.3.1. Algorithm implementation

In order for the varying drag coefficient to be computed both during the propagation and for adjustment of the β_E estimations, three separate algorithms have been written. These three algorithms relate directly to the theory from section 4.2, namely the calculations for the C_D of a sphere and a cube, and the calculations using Langmuir's isotherm to model the energy accommodation coefficient:

- C_D **cube** - considers a non-rotating stabilized 1U CubeSat and derives the total drag coefficient by analyzing the impact the gas mixture has on each panel.
- C_D **sphere** - applicable for a spherical object and derived the total drag coefficient as a function of the kinetic temperature of the incoming particles $T_{k,i}$ and the kinetic temperature of the re-emitted particles $T_{k,r}$.
- **Langmuir's isotherm** - calculates the energy accommodation coefficient α as a function of presence of atomic oxygen in the ambient gas mixture. α is a required parameter for the two above scripts.

The scripts have been written in both MatLab and C++, such that it can also be used in the TUDAT propagation program during the propagation period. The software architecture is similar for the two programs, and is displayed in Figure 5.7. The MatLab scripts are stand-alone and are just to adjusts a set of β_E estimations given the epoch and altitude of the matched objects are also provided. These two are needed to allocate the parent's and child's C_D to that specific epoch.

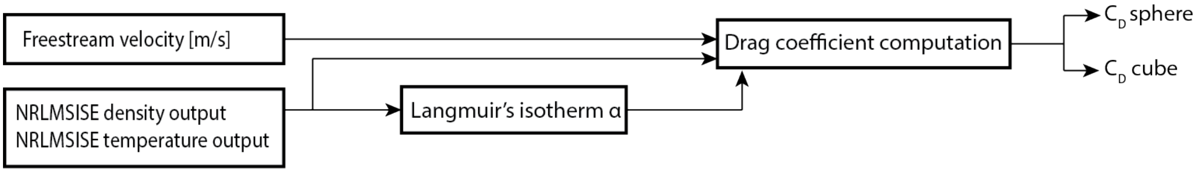


Figure 5.7: Inputs and outputs for the drag coefficient computation for an object at an arbitrary position. The input NRLMSISE-00 data is respective to that position.

An important aspect of having the capability to assess an object's drag coefficient is that this can now be combined with the output of the initially estimated ballistic coefficients $B_A^{E\star}$. For the objects which have had their β estimated, it is possible to adjust their β values without re-running the whole β estimation procedure. By using Equation 5.6 the ratio between $C_{D,j}$ and $C_{D,i}$ together with the ratio of $\beta_{C1_{old}}$ and $\beta_{C1,i}$ are used to update the ballistic coefficients.

$$B_A^{E\star} = B_A^M \frac{B_S^E}{B_S^M} \frac{C_{D,S,j}}{C_{D,S,i}} \tag{5.6}$$

In order to do so, all that is required is a list of corresponding C_D values for each object at epoch at which they were matched with their parent satellite - for the first iteration these values would of course need to be computed with the C_D sphere algorithm, all the others are computed with the C_D cube algorithm. An important note though for this computation - the average density and temperature output from NRLMSISE-00 evaluated over one revolution starting at the epoch is used as input for these functions. This is to retrieve an accurate representation of the average felt C_D instead of a value associated with specific position, which as can be seen in Figure 5.4 can vary over latitude and longitude.

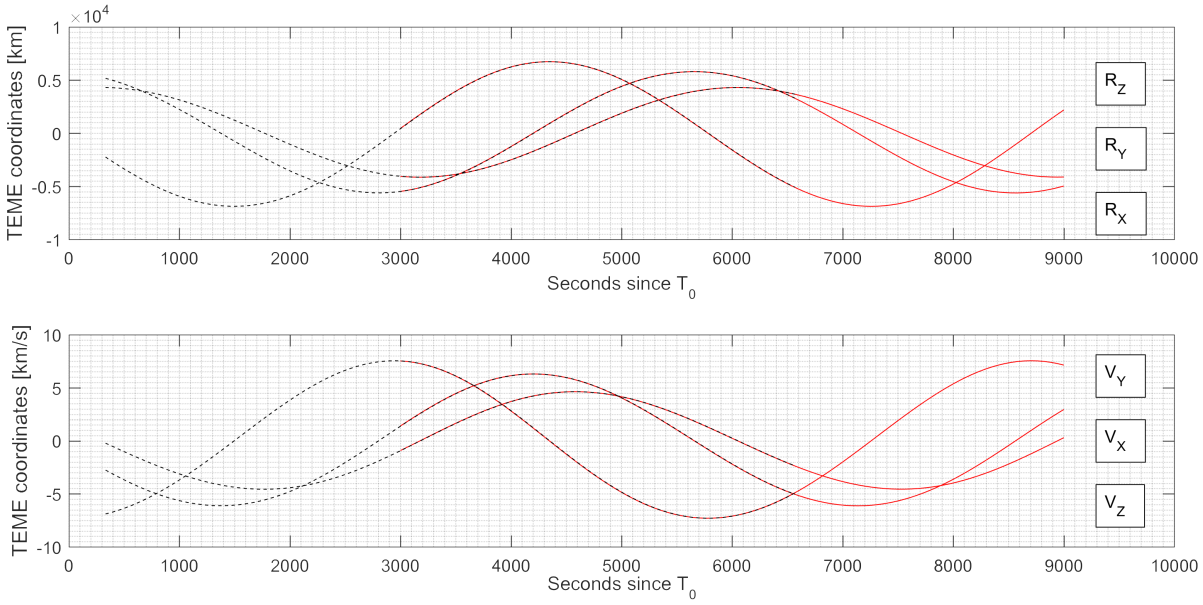
5.4. Verification

Having discussed the theory behind the methods described in this chapter and elaborated on the software implementation of the theory, this final section focuses on verifying the correct implementation of the software. Literature essential for this research, such as the work of J.M. Picone[39], M.D. Pilinski[41], and E. Doornbos[11], provided (graphical) data with which the software implementations can be verified. Output position and velocity coordinates from the SGP4 and TUDAT propagator are also investigated to ensure proper usage of the reference frames and proper transformations between those two.

5.4.1. SGP4 and TUDAT

An arbitrary satellite was chosen to verify the propagation outputs from the SGP4 propagator and the TUDAT propagation model. The selected object was 37851, in which in Figure 5.8 the position and velocity coordinates in the TEME reference frame are depicted. The red line represents the output data from the SGP4 propagator and the black dotted line that of the TUDAT propagator. The verification has been performed by starting at the same time, T_0, though for demonstration purposes the SGP4 propagation results, or the red-line, are appearing in the graph after roughly 3000 seconds to ensure the black dotted line is still visible and not continuously overlapped by the red line.

Figure 5.8: Software verification of object 37851 demonstrating correct reference frame usage for SGP4 and TUDAT propagation models.



As can be seen, the data overlaps each other - which was to be expected, especially close with the data being so close to the T_0. If one were to either zoom in or extend the propagation for a couple of weeks, deviations between the SGP4 verification orbit and TUDAT propagation would logically start to appear. However, the sole purpose of Figure 5.8 was to demonstrate similar initial conditions are used for both propagators, initial conditions derived from Two-Line Elements to the TEME frame, converted to the J2000 ECI frame for propagation with TUDAT, and when converted back to the TEME ECI frame it proved to still be overlapping each other. Concluding, it can be stated that the reference frame transformations are performed properly and in accordance with the theory described in section 2.1.

5.4.2. Ballistic coefficient estimation

This section discusses the verification process for the β estimation procedure. The theory behind this procedure is largely based on work of J.T. Emmert[13] and J.M. Picone [13][39]. In their papers, they provide intermediate figures and datasets of their steps, related mostly to Equation 4.4. These figures have been reproduced using the details provided in their papers.

The focus of the verification process for the β estimation software lies in three components - **TLE element conversion**, **NRLMSISE-00 usage** and **atmospheric error computation**. Demonstrating the software is capable of reproducing the data mentioned before and performing these three components accordingly shall be considered prove of verification.

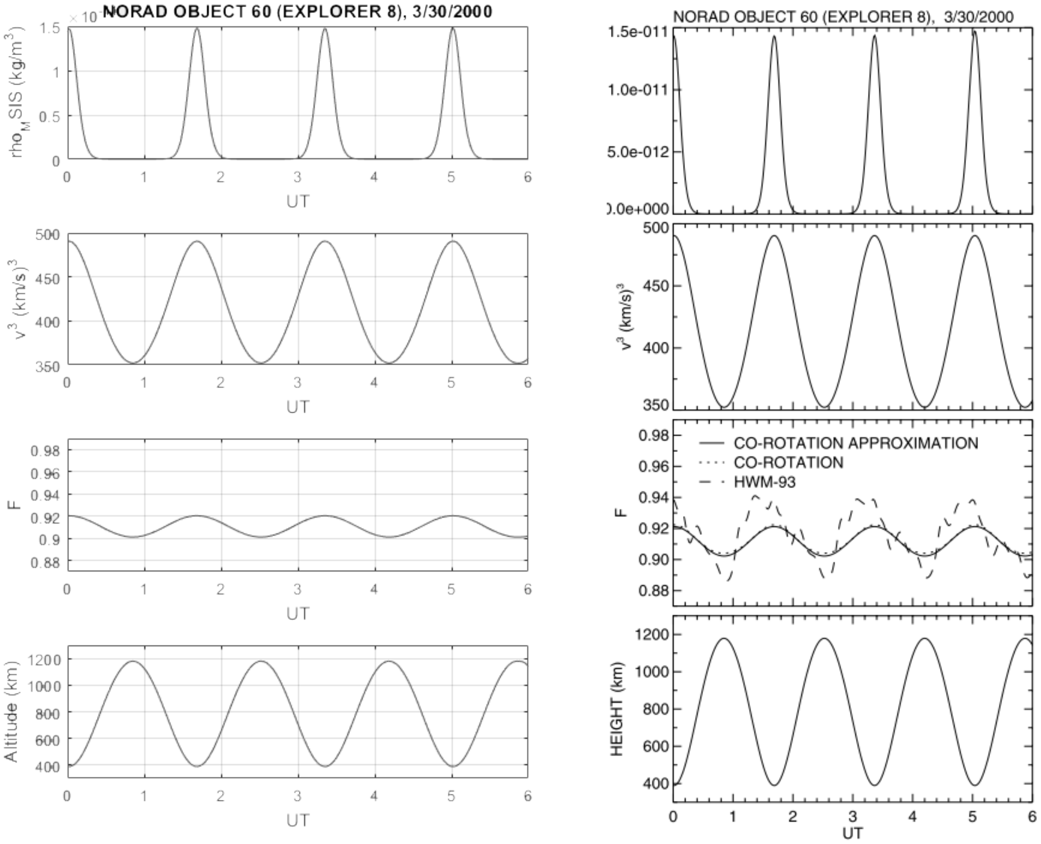


Figure 5.9: Software verification for computation of mass densities, velocities, velocity factors [F], and altitude of Explorer 6 (NORAD 60). Left side is the developed software and on the right side the graphs from Picone's work[39].

Starting with Figure 5.9, the right hand side showcases the original figures of J.M. Picone, with on the left hand the figures resulting from the thesis software. Present are the density, v^3, windfactor, and altitude of NORAD object 60 (Explorer 8) on the 30th of March 2000 from 00:00 UT to 06:00 UT. The data match each other perfectly, verifying the capability of downloading an arbitrary TLE dataset, convert the data to TEME coordinates using SGP4 and consecutively transform this to the ECEF frame. Furthermore, it verifies correct NRLMSISE-00 usage and the computation of the windfactor approximation.

The verification of the above five points form the backbone for the $B_A^{E\star}$ estimation, which shall be discussed now. As mentioned, the $B_A^{E\star}$ is basically the orbit-derived mass density from the TLE data divided by the model-estimated mass density from the NRLMSISE-00 model. In his paper, Picone[39] provides the ratio of observed effective density to the NRLMSISE-00 effective mass density for again the Explorer-8, yet now computed over the full year 2000. This figure, shown in Figure 5.10, showcases the atmospheric error the NRLMSISE-00 model has over that year compared to the TLE data. With the β estimation software, this figure has been reproduced, see Figure 5.11. The capability of the software to compute $B_A^{E\star}$ is henceforth verified.

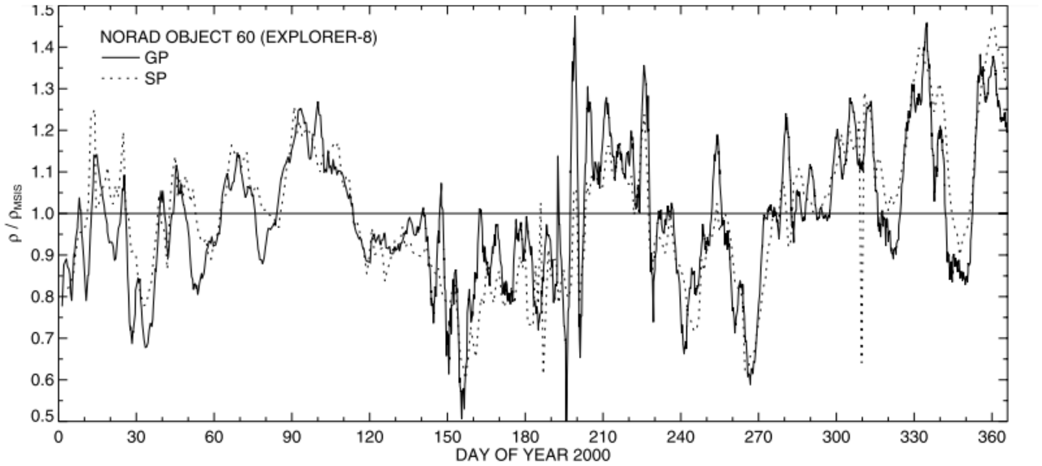


Figure 5.10: The ratio of the observed effective density to the NRLMSISE-00 effective model density, computed from TLEs (SGP4) and from SP state vectors/B-values for Explorer-8 during 2000. The nominal integration time interval for the TLE-based density ratio, equation (22), was 3 days, as was the SP fit span[39]

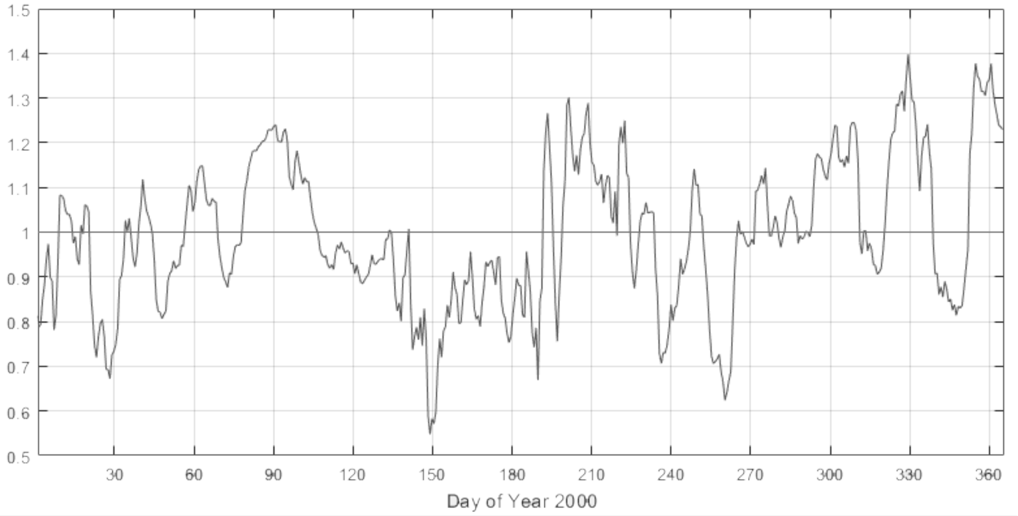


Figure 5.11: Reproduced Figure 5.10, demonstrating the capability of reproducing the ratio of observed effective density to the NRLMSISE-00 model and the effective density retrieved from analysis of historical TLE data .

To summarize, a verified software tool has been created that is capable of estimation ballistic coefficients of objects orbiting Earth in near-similar atmospheric conditions as that of the initial dataset consisting of spherical satellites. The next section shall elaborate on the incorporation of a variable drag coefficient, dependent on the local gas composition and temperature. Similar to this chapter, the theory shall be elaborated on first, followed by the implementation and eventually the verification.

5.4.3. Variable drag coefficient

Verifying the software is key before use can be made during the propagation. This verification process is performed in two phases - first, a Matlab script is made and verified with the literature of Doornbos and Pilinski. Second, the script is translated to C++ and incorporated in the TUDAT library, where it is again verified with data from Doornbos. The reasoning behind this is that Matlab allowed for rapid prototyping and debugging, saving quite some time before implementing the code in C++. Moreover, now the same scripts can be called upon in both Matlab and TUDAT, instead of solely in TUDAT.

Starting with Seltman's rarified aerodynamics equations, the related equations are used to provide drag and lift coefficients as a function of angle θ and molecular mass. Figure 5.12 demonstrates the results Eelco Doornbos has from his work[11] at the top half and the values retrieved from the MatLab script on the bottom half. The molecular mass associated to these plots is that of the oxygen atom, thus $m_j = 16$, assuming a velocity of $7600 m/s$ and an energy accommodation coefficient $\alpha = 1.0$.

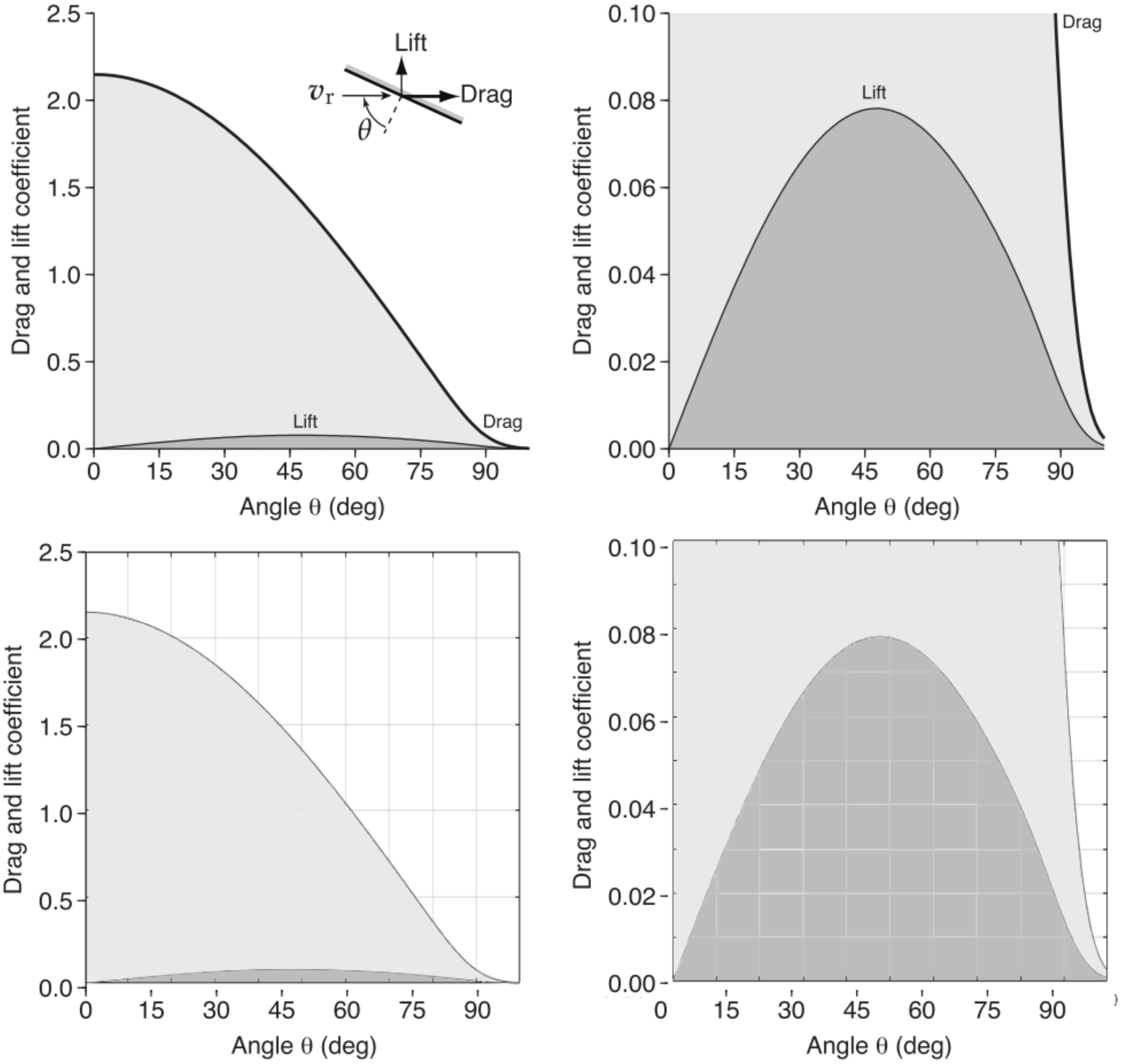


Figure 5.12: Verification of drag coefficient computation for a flat plate. Top side is data from E. Doornbos[11] and the bottom part are the recreated values from the Matlab script.

Clearly, the two profiles correspond, and therefore verify the script's capability of deducing a C_D and C_L value from an arbitrary plate under arbitrary gas composition outputs from NRLMSISE-00. A note on the 90 degree angle and non-zero drag coefficient - the random thermal motion of the gas particles also impact the plate when it is parallel to the velocity vector, which creates a small yet notable drag coefficient contribution. It is due to this phenomena that slender satellites experience higher C_D values than their shorter counterparts.

Following a similar fashion, the drag and lift coefficient as a function of energy accommodation coefficient are verified by recreating Figure 5.13, where the top is from Eelco Doornbos[11] and the bottom from the Matlab script. The same velocity and molecular mass is used (7600 m/s and 16 kg/mol), yet the plate's orientation is set as perpendicular to the flow field. Finally, the energy accommodation coefficient is verified with data from Pilinski and Doornbos in Figure 5.15. Here, the constant K as described in Equation 4.19 has been set to 7.50 x 10$^-$17.

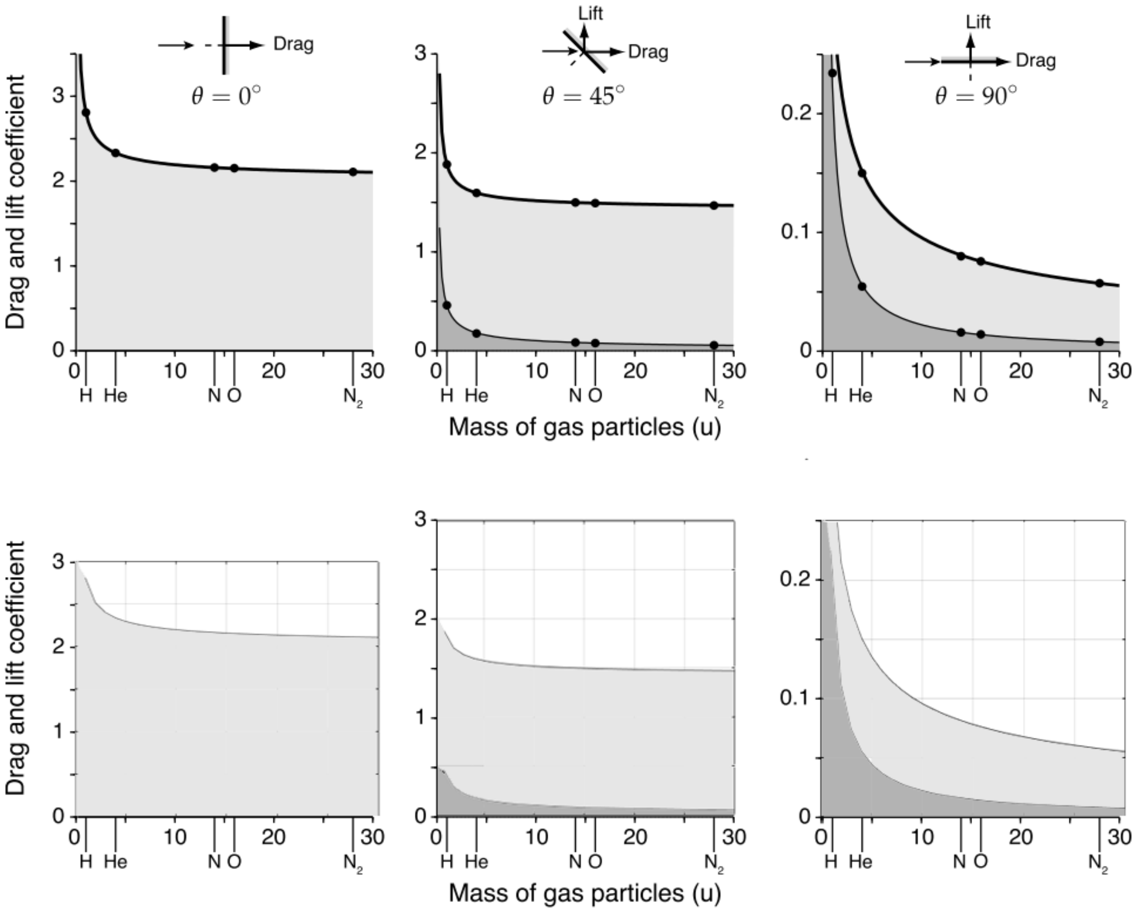


Figure 5.13: Verification of C_D and C_L computation for a panel as a function of gas particle mass. Top side is data from E. Doornbos[11] and the bottom part are the recreated values from the Matlab script.

From verifying Figure 5.12, Figure 5.13, Figure 5.15, and Figure 5.16 the following can be said about the implementation of the drag coefficient determination software based on the interaction between flat panels and the ambient gas mixture:

1) C_D and C_L for a plate under arbitrary θ can be calculated.
2) The incorporation of α is done correctly.
3) Langmuir's isotherm follows the procedure described by Pilinski[41]

In a similar fashion as done with Figure 4.2, Figure 5.14 presents the drag coefficient algorithm output of an arbitrary sphere under solar maximum and minimum conditions for two constant energy accommodation coefficients (0.8 and 1.0), as well as computed with Langmuir's adsorption model. The C_D ranges from 2.0 to 2.3, a much lower spread compared to the CubeSats, though this was expected through the discussion held in section 4.2.

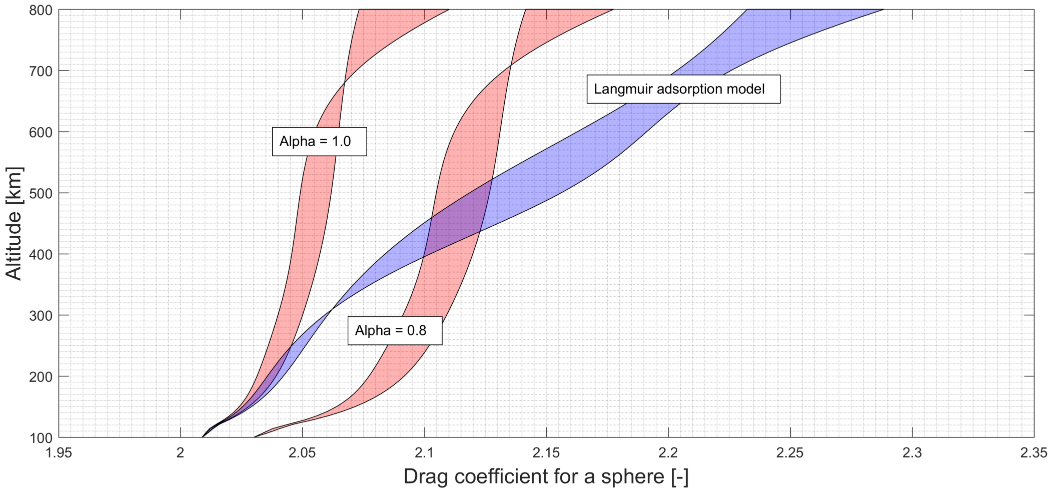


Figure 5.14: C_D algorithm output for an arbitrary spherical satellite for different energy accommodation coefficients as a function of altitude.

Having described the methodology for estimating β_E values and adjusting them for the varying drag coefficient, both for a constant $\alpha = 0.8$ and according to Langmuir's adsorption model, the software implementations have consecutively been verified and deemed ready for usage. The next chapter elaborates on the experimental set up and which spherical satellites and CubeSats are having their ballistic coefficients updated.

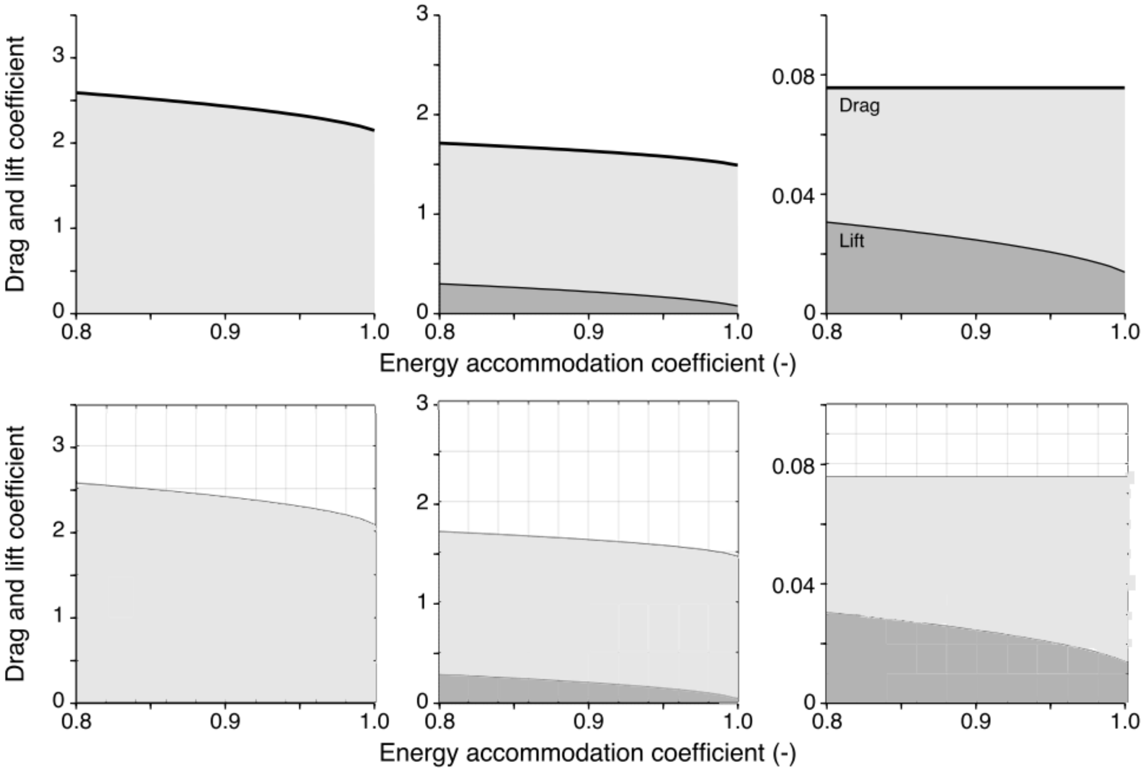


Figure 5.15: Verification of the $C_{D,cube}$ software (bottom) with Doornbos's work[11](top). From left to right, the θ angles are 90, 45, and 0 degrees respectively.

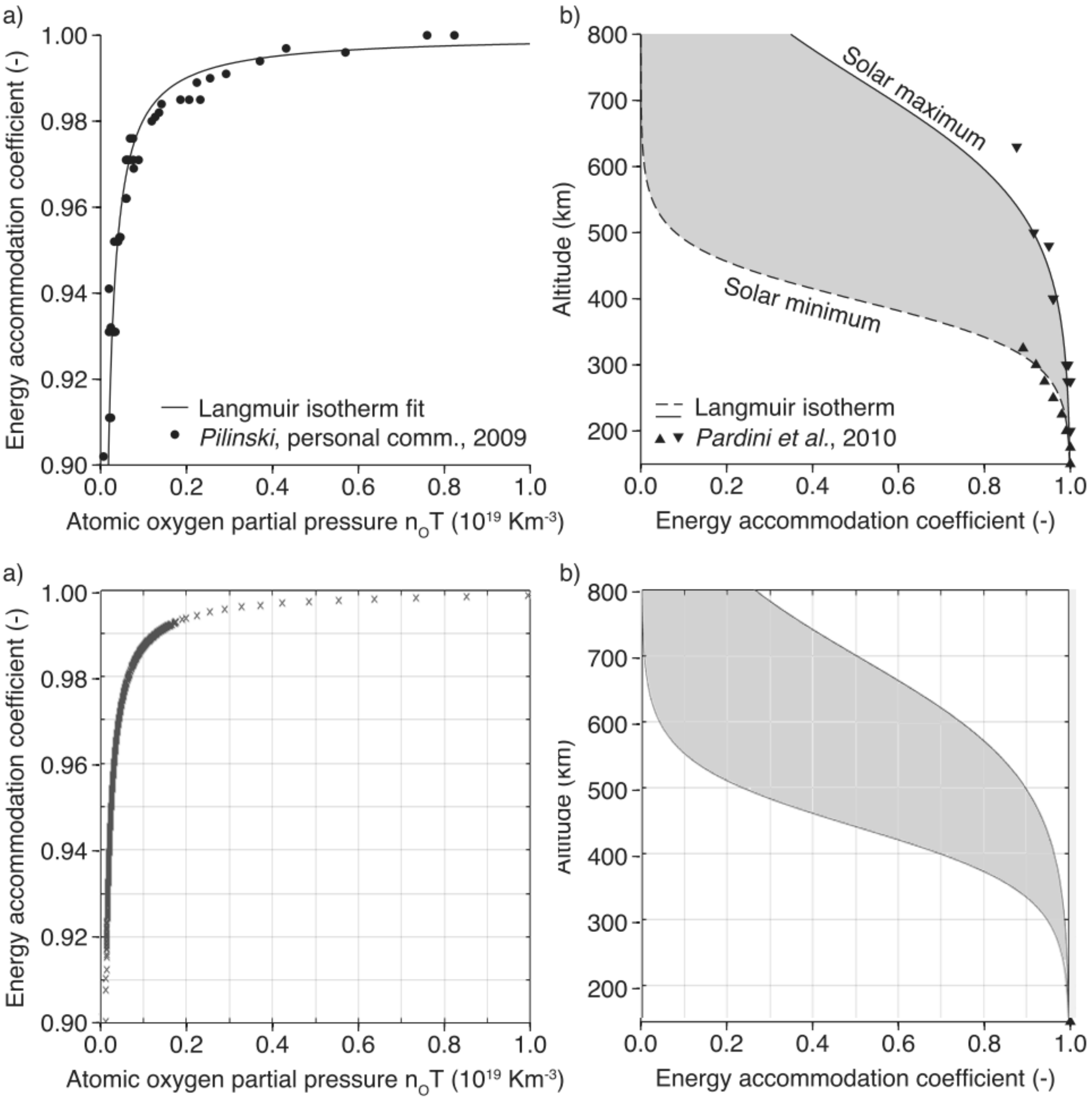


Figure 5.16: Verification of Langmuir's isotherm and the energy accommodation coefficient as a function of altitude and solar activity inputs. Top side is data from Pilinski[41] and the bottom part are the recreated values from MatLab.

6

Experimental Set up

The objects that are to be subjected to the methodology described in chapter 5 are described in this chapter - both the initial parent dataset consisting of spherical satellites as the child dataset consisting of 1U CubeSats. For all satellites, a distinction is made between objects still orbiting Earth and those that already have re-entered the atmosphere. Within these two categories, the in-orbit or re-entry category, the proximity algorithm is used to establish three satellite groups for the in-orbit category and two for the re-entry category. The satellite composition of these groups are discussed in section 6.1.

These satellite groups are all subjected to four propagation scenarios that have been established to properly assess the accuracy of the estimated ballistic coefficients, the adjusted β_E values for the varying drag coefficient, and another where the energy coefficient follows Langmuir's adsorption model. These scenarios and the assessment merits through which the performance is quantified is discussed in section 6.2.

6.1. Satellite Test Group

This thesis focuses on orbital lifetime prediction updates for the CubeSat community, and more specifically for a method for estimating ballistic coefficient for CubeSats in the 1U class. Selecting this class is based on a simple need - to have a large set of test objects with comparable geometric properties. CubeSats are increasingly present in Low Earth Orbits (LEO) due to recent advancements in satellite miniaturization and launch cost reductions, resulting in hundreds of potential candidate objects over the last ten years. Moreover, the dimension standardization for CubeSats allows the candidate objects to be easily categorized according to their geometries.

Apart from the CubeSat database of test objects, a database containing spherical satellites have also been set up, which as discussed in section 4.1 function as parent objects in the first iteration. Due to their geometry the observed reference area of the satellite is independent of their attitude, which allows for the ballistic coefficient estimation of the CubeSat test objects when both a sphere and Cube are experiencing the same atmospheric conditions, more on this in subsection 5.2.2.

6.1.1. CubeSat database

A CubeSat is a miniaturized satellite with as primary mission providing affordable access to space for small payloads - whether for commercial, educational, scientific, or technology demonstration purposes. Started in 1999, the purpose of the CubeSat Program was to provide a standard for design of picosatellites to reduce cost and development time, increase accessibility to space, and sustain frequent launches[54]. A CubeSat is build up in Units, with a single Unit (1U) being a 10 x 10 x 10 cm cube with a mass of up to 1.33 kg. If a mission desires, these cubes can be grouped together to form 2U, 3U, or sometimes even 6U and 12U satellites.

Due to their sizing limitations, functionality of a single CubeSat does not compare to that of a classical larger satellite. Yet advancements in miniaturization, in-orbit inter-satellite cooperation, and the cheap development and launch cost have seen the volume of CubeSats present in LEO grow exponentially over time - in mid-February 2017, a PSLV has delivered 104 different satellites into orbit in a single launch, of which 80+ were 3U CubeSats. Such volume growths and complex launch operations are unprecedented in the space business and is one of many signs that industry is starting to embrace this sector and nurture its growth.

The dimension standardization is however not a guarantee that all 1U CubeSats are exactly the same - there are still numerous design possibilities enabling 1U CubeSats to exceed the initial 1U volume once injected into their orbit.Deployable solar arrays, antennae, or baffles, show that not all CubeSats will have the same reference area. Figure 6.1, Figure 6.2, and Figure 6.3 show three different 1U CubeSats, each with their own design and outer dimensions, to give the reader a feel for the variability in deployed outer dimensions. As a final note, there are missions that have active stabilization while others might just be tumbling through space. This has an impact on the computation of experienced drag acceleration, as tumbling 1U CubeSats will have a changing and generally larger reference area as opposed to an actively stabilized CubeSat. Though, as discussed in section 4.2, during the drag computations all CubeSats are considered to be active attitude-controlled with only one panel directly facing the freestream velocity to avoid additional complexity and computation time.

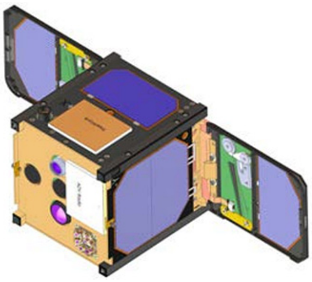


Figure 6.1: Aerocube 4 (NORAD 38769)

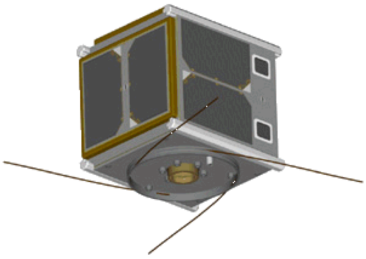


Figure 6.2: AAU CubeSat (NORAD 27846)

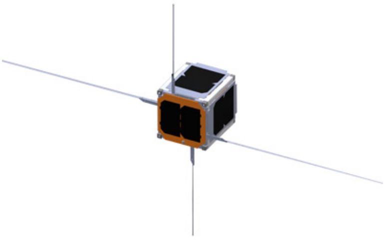


Figure 6.3: Duchifat-1 (NORAD 40021)

A database has been set up consisting of 96 1U CubeSats which have been placed in LEO between January 2009 and December 2016, resulting in two sub-categories:

- **Re-entry group** - consisting of roughly 25 CubeSats deployed from the ISS or at the same altitude as the ISS. These ISS deployed CubeSats have almost all already re-entered. This sub-category shall be used when evaluating the effectiveness of the proposed orbital lifetime estimation in the lower part of the thermosphere and closer to the end of life of an object. The assessment merit shall be the decay epoch, see section 6.3.
- **In-orbit group** - the majority of the CubeSats have been injected at orbits above 500 kilometer and are still orbiting Earth. This sub-category is used to evaluate how estimating β_E and incorporation a varying C_D affects the prediction of altitude decay. The assessment merit shall be the mean altitude of the last 10 orbits of the object compared to a verification orbit, see section 6.3.

The database solely consists of the satellite's name and NORAD designator - see Table 6.1. As the ballistic coefficients of these objects shall be estimated according to the procedure discussed in section 4.1, no data on the mass and reference area is stored. An attempt has been made to get an accurate initial mass and reference area, yet often the information provided in the satellite's data sheet proved to be too inaccurate and inconsistent to be used. Instead, the following three values for the mass, reference area and drag coefficient are assumed for all 1U CubeSats:

The reference area is larger than the standardized 10 cm x 10 cm dimensions for a 1U CubeSat, and has been intentionally increased to 12.5 cm x 12.5 cm (hence the $0.01562\ m^2$) as it is assumed that most of the CubeSats are not actively controlled and thus experiencing a tumbling behavior. Secondly, a drag coefficient of 2.8 has been selected as initial estimate when computing the β_E based on Figure 4.1.

Table 6.1: 1U CubeSat database - total of 96 objects present

Name	NORAD ID
DTUSAT	27842
CUTE-1	27844
AAU CUBESAT	27846
CANX-1	27847
CUBESAT XI 4	27848
NCUBE-2	28897
UWE-1	28892
CUBESAT XI 5	28895
EGYPTSAT 1	31117
CSTB 1	31122
LIBERTAD 1	31128
CP3	31129
CAPE 1	31130
CP4	31132
AEROCUBE 2	31133
COMPASS 1	32787
AAUSAT CUBESAT 2	32788
SEEDS	32791
SDS-1	33497
CP6	35003
HawkSat I	35004
AEROCUBE 3	35005
SWISSCUBE	35932
BEESAT	35933
UWE-2	35934
ITUPSAT 1	35935
HAYATO	36573
Waseda-SAT2	36574
Negai	36575
DICE 1	37851
DICE 2	37852
AUBIESAT-1	37854
M-CUBED/EXP-1 PRIME	37855
ROBUSTA	38084
UNICUBESAT	38085
SDS-4	38339
CP5	38763
AEROCUBE 4.5A	38767
AEROCUBE 4.5B	38768
AEROCUBE 4	38769
FITSAT-1	38853
TechEdSat	38854
F-1	38855
We-Wish	38856
AAUSAT3	39087
SOMP	39134
BEESAT 2	39136
PhonesSat Bell	39142
PhoneSat Alexander	39144
PhoneSat Graham	39146
NEE 01 PEGASUS	39151
CUBEBUG 1	39153
ESTCUBE 1	39161
PHONESAT 2.4	39381
ArduSat1	39412
ArduSatX	39414
ZACUBE	39417
WNISAT 1	39423
BRITE-PL	39431
ICUBE 1	39432
HUMSAT D	39433
WREN	39435
VELOX-P 2	39438
FIRST-MOVE	39439
NEE 02 KRYSAOR	39441
PUCP-SAT 1	39442
FUNCUBE 1	39444
HINCUBE	39445
UWE-3	39446
MCUBED-2	39469
CUNYSAT-1	39470
IPEX	39471
SKYCUBE	39567
UAPSAT	39568
ArduSat2	39571
ITF-1	39573
OPUSAT	39575
KSAT2	39578
BRITE TORONTO	40020
DUCHIFAT-1	40021
NANOSAT C BR1	40024
DTUSAT-2	40030
POLYITAN 1	40042
NLS 7.1/CANX 4	40055
NLS 7.2/CANX 5	40056
FOX-1A (AO-85)	40967
ARC-1	40969
PROPCUBE 3	40973
SINOD-D 1	40974
PROPCUBE 1	40976
SINOD-D 3	40977
OUFTI-1	41458
E-ST@R-II	41459
AAUSAT-4	41460
STMSAT 1	41476
SWAYAM	41607

6.1.2. Spherical satellite database

In the early space era, calibration spheres where launched into the thermosphere mainly to calibrate radar systems by detecting the reference area from the ground and match this with the known diameter of the sphere - any offset between the observed and known area resulted in the radar system being calibrated to ensure these values were equal again. This property of calibration spheres to have an observed area irrespective of its attitude is what makes the β estimation procedure discussed in this thesis possible - the drag experienced by a sphere is uniform regardless of attitude, something that is not the case with tumbling objects of which the attitude dynamics are unknown or complex to compute.

Preferably, only perfect spheres with a known diameter, mass, and outer material are selected for the database. These perfect spheres should provide the most accurate estimation of β_S, yet selecting a database with only perfect spheres greatly reduces the number of potential spheres. Therefore, spherical satellites with an approximate spherical shape have been selected for this database as these still approach a spherical geometry and the property of attitude-independent reference area. A sphere is assumed to approximate a spherical shape when only small indentations are present yet the overall shape can still be described as a sphere.

Figure 6.4: LCS-1 (NORAD 1361)

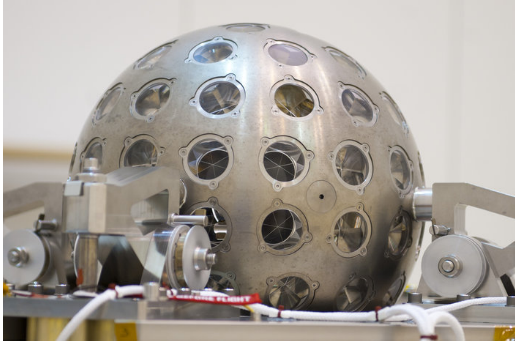


Figure 6.5: LARES (NORAD 38077)

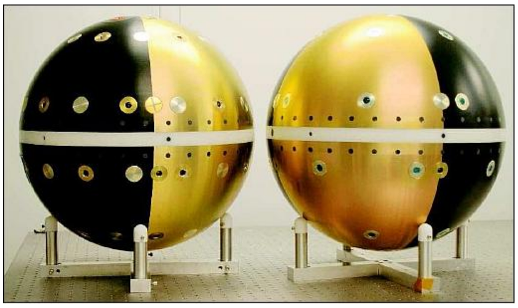


Figure 6.6: ANDE-C (NORAD 35693) and ANDE-P (NORAD 35693)

Figure 6.4 and Figure 6.5 represent the LCS-1 and LARES satellite respectively. Launched on May the 6^{th} 1965, the Lincoln Calibration Sphere 1 has been contributing to the calibration of radar systems for over 50 years now, and is one of the oldest spacecraft still in use. The Laser Relativity Satellites, or LARES, is a passive satellite launched on the 13^{th} of February 2012 and is being tracked by international laser ranging stations for measurements of the Lense-Thirring effect. Its data is also used in the fields of geodynamics and satellite geodesy.

The Atmospheric Neutral Density Experiment (ANDE) satellites depicted in Figure 6.6 have been designed and used specifically for measurements of thermospheric neutral densities and for use as test object for both radar and optical U.S. Space Surveillance Network sensors. Their orbit insertion occurred on the 30^{th} of July 2009. Three different satellite missions have been described here, yet the fact that all three have an (approximate) spherical geometry makes them suited for this thesis research.

For this research, a total of 28 spheres have been found and taken into the database. Some of them have been orbiting Earth for as long as 1965 (LCS-1), whilst others have already decayed in the atmosphere (GFZ-1). A small survey has been performed to confirm at least a couple of these spheres are within a close orbital parameter range of the two CubeSat in-orbit and re-entry sub-categories, ensuring at-least a couple of groups within both categories can be established. The algorithm used for these proximity checks are further elaborated on in subsection 5.2.2.

The spheres shall be used to estimate the β_E for the objects in the CubeSat database, and hence accurate mass and diameter values are required for these spheres. Luckily, better documentation on the spheres exist as opposed to the documentation of most CubeSats - especially since some of the spheres are used for thermospheric density derivations themselves the open-source information on the mass and diameter are more reliable. These values have been added to the database, along with the resulting reference area and β_{sphere}. For the β_E computation, a drag coefficient C_D = 2.3 has been assumed based on Figure 4.1.

Table 6.2: Spherical satellite database

NORAD CAT ID	Satellite Name	Mass [kg]	Diameter [m]	A_{ref}	β
900	CALSPHERE 1 [24]	0.98	0.356	0.0995	0.2336
902	CALSPHERE 2 [24]	9.8	0.356	0.0995	0.0234
1361	LCS 1 [25]	34	1.12	0.9852	0.0666
1512	TEMPSAT 1 [8]	9	0.356	0.0995	0.0254
1520	CALSPHERE 4(A) [24]	1.96	0.356	0.0995	0.1168
2872	SURCAL 159 [33]	3.8	0.82	0.5281	0.3196
2909	SURCAL 150B [33]	1.5	0.41	0.1320	0.2024
4168	TEMPSAT 2 [8]	9	0.356	0.0995	0.0254
5398	RIGIDSPHERE 2 [20]	34	1.12	0.9852	0.0666
7646	Starlette [42]	47.29	0.24	0.0452	0.0022
22824	Stella [26]	48	0.24	0.0452	0.0022
23558	GFZ-1 [4]	20.63	0.215	0.0363	0.0040
25398	Westpac [42]	23.8	0.245	0.0471	0.0046
25769	STARSHINE [48]	39	0.475	0.1772	0.0105
26929	STARSHINE 3 [48]	91	0.945	0.7014	0.0177
26996	STARSHINE 2 [48]	39	0.475	0.1772	0.0105
27944	Larets [26]	23.28	0.245	0.0471	0.0047
29664	ANDE-RR [42]	52.04	0.4826	0.1829	0.0081
29667	ANDE -Fcal[42]	62.7	0.448	0.1576	0.0058
35693	ANDEP [42]	50	0.48	0.1810	0.0083
35694	ANDEC [42]	25	0.48	0.1810	0.0166
35871	BLITS [42]	7.53	0.17	0.0227	0.0069
38077	LARES [42]	400	0.364	0.1041	0.0006
39267	DANDE [40]	50	0.46	0.1662	0.0076
39268	POPACS 1 [40]	1	0.1	0.0079	0.0181
39269	POPACS 2 [40]	1.5	0.1	0.0079	0.0120
39270	POPACS 3 [40]	2	0.1	0.0079	0.0090
40314	SPINSAT [32]	57	0.558	0.2445	0.0099

6.2. Propagation scenarios

In coherence with the previously discussed theory and satellite test groups, a total of four different scenarios are established. These scenarios have been created such that the contribution of the β_E estimation and varying C_D can be assessed independently. The objective of the experimental set up is to answer the hypotheses stated in chapter 1, and to subsequently assess the affectx of including Langmuir's isotherm to the energy accommodation coefficient calculations. The scenarios, summarized in Table 6.3, enable a qualitative assessment the estimation of a ballistic coefficient through analysis of historical TLE data and the inclusion of a varying drag coefficient have on orbital lifetime predictions.

Furthermore, a distinction between the assessment for the two CubeSat categories discussed at the beginning of this chapter has been made. For all CubeSats a 100 day verification orbit based on the historical TLE data is established. Consecutively scenario 1 creates an orbit propagated with the ballistic coefficient β_0 being set to 0.03125 in coherence with the 1U CubeSat dimensions assumed earlier. The reason for this scenario is to create a baseline orbit with which the other scenarios can be assessed. All scenarios shall have errors induced from the propagation model, and to be able to detect a change solely due to any of the methods described before, a baseline orbit with an unbiased β is created. Elaboration on the assessment is provided in section 6.3.

Scenario 2, scenario 3, and scenario 4 correspond to first determining the effect of the estimated ballistic coefficient β_E, followed by the orbits of the adjustment for the varying C_D with Langmuir's isotherm, and the last a scenario in which the energy accommodation coefficient is kept constant at 0.8.

Table 6.3: The eight different scenarios to which the CubeSat database is subjected to and their assessment criteria.

	In-orbit groups	Re-entry groups
Verification Orbit	[-]	[-]
Scenario 1 - Baseline orbit with β_0	Δ s.m.a. [km]	Re-entry epoch [days]
Scenario 2 - Orbits with estimated β values	Δ s.m.a. [km]	Re-entry epoch [days]
Scenario 3 - Orbits adjusted for C_D with Langmuir's isotherm	Δ s.m.a. [km]	Re-entry epoch [days]
Scenario 4 - Orbits adjusted for C_D with constant α	Δ s.m.a. [km]	Re-entry epoch [days]

The propagations shall all be performed over the same evaluation period and assessed according to identical merits. The details on the assessment are explained further in section 6.3. As will become clear in subsection 6.2.1 and subsection 6.2.2, the propagation periods for the in total ten different orbits shall all be 100 days. To keep track of the different ballistic coefficients the following subscripts have been defined:

Table 6.4: Scenario subscript clarification

Baseline	β_0 - original coefficient of 0.03125
Scenario 2	β_E - estimated coefficient
Scenario 3	$\beta_{C_D,Langmuir}$ - adjusted for C_D with Langmuir's isotherm
Scenario 4	$\beta_{C_D,\alpha=0.8}$ - adjusted for C_D with $\alpha = 0.8$

Throughout the remainder of this thesis report, these subscripts shall be used to refer to the ballistic coefficients of the four scenarios. As stated, the CubeSat groups are subdivided in two categories - the in-orbit and re-entry groups. The next sections shall elaborate on the establishment of a total of five target groups, based on the orbital element selection criteria and proximity algorithm - three in the in-orbit group and two in the re-entry group. This chapter shall conclude with a further elaboration on the performance assessment for each category.

6.2.1. In-orbit groups definition

The complete TLE database for the spheres and CubeSats have been downloaded and stored, and are subjected to the proximity algorithm described in subsection 5.2.2. The goal of this algorithm is to define a set of satellite pairs that are within the pre-defined orbital parameter criteria set and are thus experiencing similar atmospheric conditions. For the in-orbit dataset the minimum altitude has been set at 500 km. Furthermore, two different criteria sets have been researched to analyze the sensitivity of the initial orbital parameter criteria and secondary to assess the influence of individual iteration group size. The two groups that are researched, together with their orbital parameter criteria, are displayed in Table 6.5 and Table 6.6. Both groups shall be subjected to the four scenarios where the influence of an estimated β and varying C_D can be assessed. Furthermore, in section 8.1 a discussion is held on any notable discrepancies between the two groups which could lead to more insight in the sensitivity of the orbital parameter criteria selection.

Table 6.5: Orbital parameter criteria for the in-orbit groups

	In-orbit Group 1	In-orbit Group 2	In-orbit Group 3
Overall inclination range [degrees]	50 - 110	50 - 110	50 - 110
Overall altitude range [km]	500 - 1000	500 - 1000	500 - 1000
Δ_{max} altitude [km]	40	50	25
Δ_{max} inclination [degree]	10	10	15
Δ_{max} RAAN [degree]	20	20	30
Δ_{max} epoch [days]	10	10	10
# of cubes in altitude and inclination range	59	59	59
# of spheres in altitude and inclination range	23	23	23
# of cubes found with proximity algorithm	43	51	52
# of spheres found with proximity algorithm	6	5	6
# of matches from proximity algorithm	146	192	126

Table 6.6: Total satellites in within each iteration for the in-orbit groups

	In-orbit Group 1	In-orbit Group 2	In-orbit Group 3
Initial spherical satellites	6	5	6
Total 1U CubeSats	43	51	52
Iteration 1	12	16	12
Iteration 2	3	23	15
Iteration 3	19	6	7
Iteration 4	9	5	11
Iteration 5	0	1	4
Iteration 6	0	0	3

Table 6.7, Table 6.8 and Table 6.9 contain the NORAD identifiers of the CubeSats and spheres for in-orbit group 1, 2, and 3 respectively, including their iteration number based on the proximity algorithm. Additionally Figure 6.7, Figure 6.8, and Figure 6.9 represent the epoch and altitude of each CubeSat at which they are matched with their parent satellite, including the 81-day averaged F10.7 solar activity data. In accordance with Figure 4.2 it is expected to see a decrease in C_D when time progresses and the s.m.a. decreases between i and j, though expansion of the atmosphere due to an increase in solar activity would result in an expected decrease of the atomic oxygen density number. This would directly affect (increase) Langmuir's isotherm and could potentially nullify the β change.

Though an interesting field to study, no additional attention shall be given to the potential influence of the solar activity on the changing β values, though it will be mentioned again in the recommendation section.

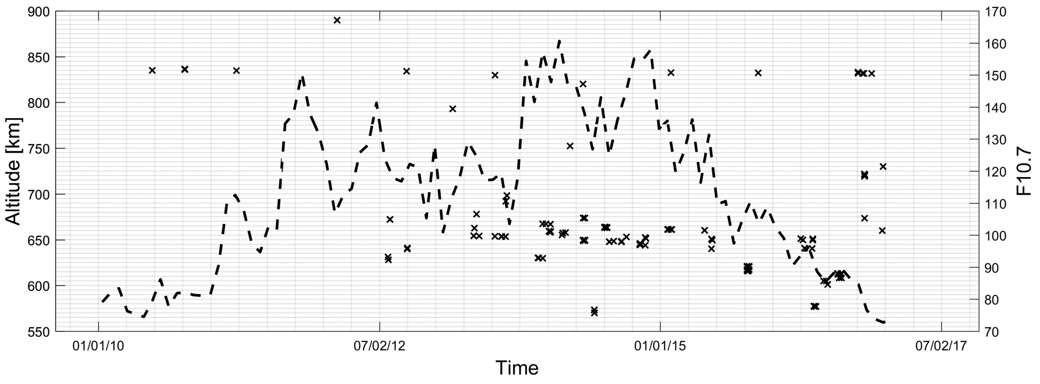


Figure 6.7: Altitude and solar activity F10.7 at epochs of matched satellites in in-orbit group 1.

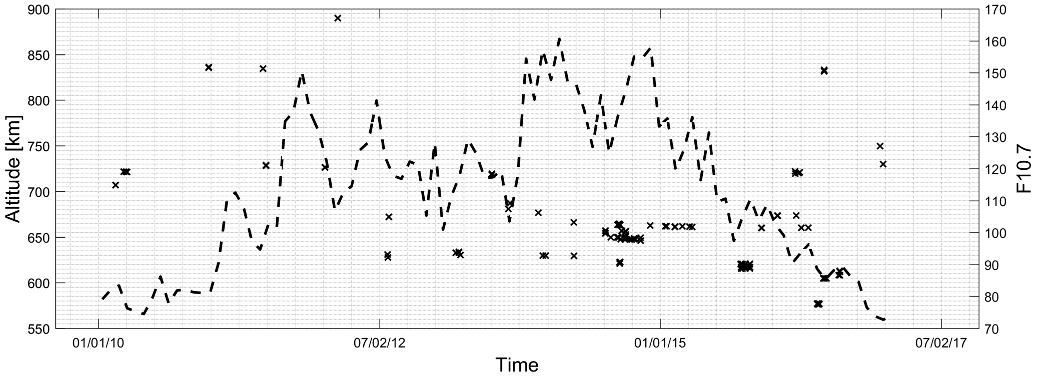


Figure 6.8: Altitude and solar activity F10.7 at epochs of matched satellites in in-orbit group 2.

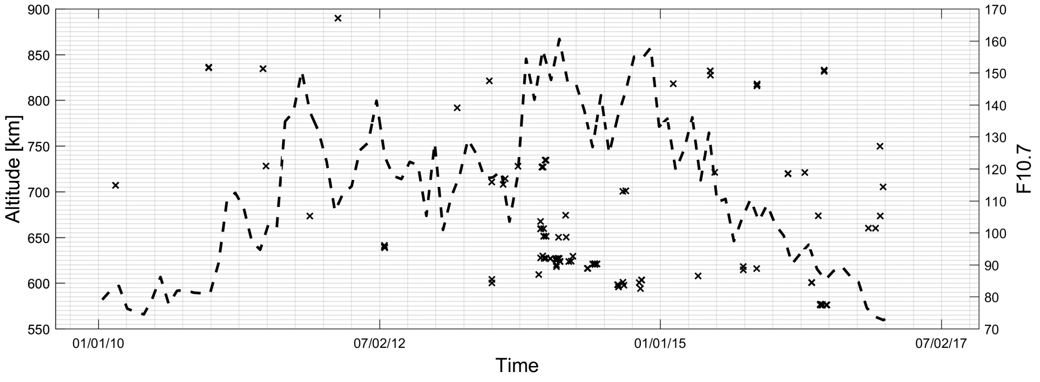


Figure 6.9: Altitude and solar activity F10.7 at epochs of matched satellites in in-orbit group 3.

Table 6.7: CubeSats and spheres for in-orbit group 1 - sorted on their iteration.

Satellite composition in-orbit group 1			
Spheres	Surcal 150B (2909)	RigidSphere-2 (5298)	Westpac (25398)
	Larets (27944)	Blits (35871)	Lares (38077)
Cubes iteration 1	DTUSAT (27842)	CUTE-1 (27844)	AAU CubeSat (27846)
	CANX-1 (27847)	XI-4 (27848)	SDS-1 (33497)
	SwissCube (35932)	BeeSat (35933)	Uwe-2 (35934)
	ITUPSAT-1 (35935)	AAUSat-3 (39087)	Brite-PL (39431)
Cubes iteration 2	Dice-1 (37851)	Dice-2 (37852)	WNISAT-1 (39423)
Cubes iteration 3	EgyptSat-1 (31117)	Compass-1 (32787)	AAUSAT-2 (32788)
	SEEDS (32791)	AubieSat-1 (37854)	M-Cubed (37855)
	ZaCube (39417)	ICube-1 (39432)	HumSat-D (39433)
	PUCP-Sat-1 (39442)	FunCube-1 (39444)	HinCube (39445)
	DuchiFat-1 (40021)	C-BR1 (40024)	DTUSat-2 (40030)
	PolyItan-1 (40042)	UFTI-1 (41458)	E-Star-1 (41459)
	AAUSat-4 (41460)		
Cubes iteration 4	Pegasus-1 (39151)	CubeBug-1 (39153)	EstCube-1 (39161)
	Velox-P-2 (39438)	First-Move (39439)	KRYSAOR (39441)
	Uwe-3 (39446)	CANX-4 (40055)	CANX-5 (40056)

Table 6.8: CubeSats and spheres for in-orbit group 2 - sorted on their iteration.

Satellite composition in-orbit group 2			
Spheres	Surcal 150B (2909)	RigidSphere-2 (5298)	Larets (27944)
	Blits (35871)	Lares (38077)	
Cubes iteration 1	DTUSAT (27842)	CUTE-1 (27844)	AAU CubeSat (27846)
	CANX-1 (27847)	XI-4 (27848)	Uwe-1 (28892)
	XI-5 (28895)	SDS-1 (33497)	SwissCube (35932)
	BeeSat (35933)	Uwe-2 (35934)	ITUPSAT-1 (35935)
	Dice-1 (37851)	Dice-2 (37852)	AubieSat-1 (37854)
	M-Cubed (37855)		
Cubes iteration 2	EgyptSat-1 (31117)	Compass-1 (32787)	AAUSAT-2 (32788)
	SEEDS (32791)	Pegasus-1 (39151)	CubeBug-1 (39153)
	EstCube-1 (39161)	ZaCube (39417)	ICube-1 (39432)
	HumSat-D (39433)	Velox-P-2 (39438)	First-Move (39439)
	PUCP-Sat-1 (39442)	FunCube-1 (39444)	HinCube (39445)
	Uwe-3 (39446)	DuchiFat-1 (40021)	C-BR1 (40024)
	DTUSat-2 (40030)	PolyItan-1 (40042)	UFTI-1 (41458)
	E-Star-1 (41459)	AAUSat-4 (41460)	
Cubes iteration 3	CSTB-1 (31122)	CP-4 (31132)	AeroCube-2 (31133)
	KRYSAOR (39441)	CANX-4 (40055)	CANX-5 (40056)
Cubes iteration 4	Libertad-1 (31128)	CP-3 (31129)	Cape-1 (31130)
	SDS-4 (38339)	Brite-PL (39431)	
Cubes iteration 5	WNISAT-1 (39423)		

Table 6.9: CubeSats and spheres for in-orbit group 3 - sorted on their iteration.

Satellite composition in-orbit group 3

Spheres	Surcal 150B (2909)	RigidSphere-2 (5298)	Westpac (25398)
	Larets (27944)	Blits (35871)	Lares (38077)
Cubes iteration 1	DTUSAT (27842)	CUTE-1 (27844)	AAU CubeSat (27846)
	CANX-1 (27847)	XI-4 (27848)	Uwe-1 (28892)
	XI-5 (28895)	SDS-1 (33497)	SwissCube (35932)
	BeeSat (35933)	Uwe-2 (35934)	AAUSAT-3 (39087)
Cubes iteration 2	EgyptSat-1 (31117)	ItupSat-1 (35935)	Pegasus-1 (39151)
	CubeBug-1 (39153)	EstCube-1 (39161)	ZaCube (39417)
	WNISAT-1 (39423)	Velox-P-2 (39438)	First-Move (39439)
	KRYSAOR (39441)	FunCube-1 (39444)	HinCube (39445)
	Uwe-3 (39446)	CANX-4 (40055)	CANX-5 (40056)
Cubes iteration 3	CSTB-1 (31122)	SEEDS (32791)	SDS-4 (38339)
	Brite-PL (39431)	ICube-1 (39432)	HumSat-D (39433)
	PUCP-Sat-1 (39442)		
Cubes iteration 4	Libertad-1 (31128)	CP-3 (31129)	Cape-1 (31130)
	CP-4 (31132)	AeroCube-2 (31133)	Compass-1 (32787)
	AAUSAT-2 (32788)	Dice-1 (37851)	Dice-2 (37852)
	AubieSat-1 (37854)	M-Cubed (37855)	
Cubes iteration 5	DuchiFat-1 (40021)	UFTI-1 (41458)	E-Star-1 (41459)
	AAUSat-4 (41460)		
Cubes iteration 6	C-BR1 (40024)	DTUSat-2 (40030)	PolyItan-1 (40042)

6.2.2. Re-entry groups definition

Due to the lack of spheres in this region, three CubeSats which have had their β estimated manually have been added to the initial database of reference satellites. These cubesats are ITF-1 (39573), F-1 (38855), and AeroCube 3 (35005). Without the inclusion of these three objects to the initial spherical satellite database no more than 2 CubeSats were found with the orbital parameter criteria in Table 6.11. The increase in potential test subjects from 2 to between 17 and 19 justify the choice of adding these CubeSats to the spherical satellite database. Note though that the expected accuracy of these ballistic coefficients are thus lower than those of other spherical satellites used in this research.

Table 6.10: Orbital parameter criteria for the re-entry groups

	AeroCube 3 [35005]	F-1 [38855]	ITF-1 [39573]
Ballistic coefficient	0.078571429	0.03125	0.025028

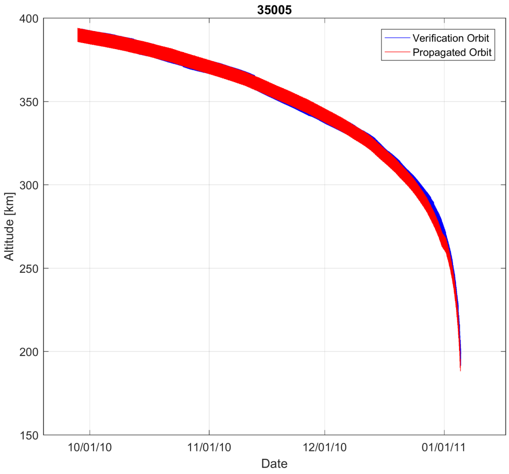


Figure 6.10: Propagated orbit for AeroCube 3 compared to verification orbit.

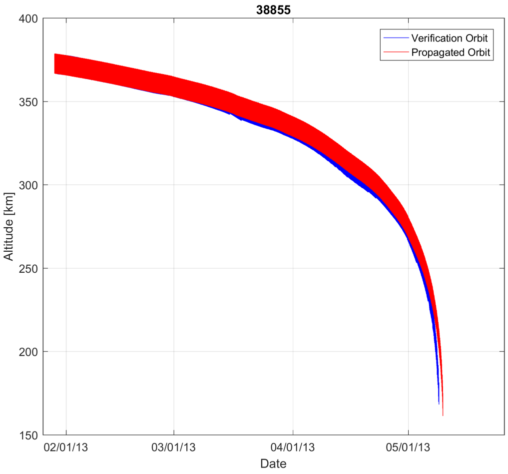


Figure 6.11: Propagated orbit for F-1 compared to verification orbit

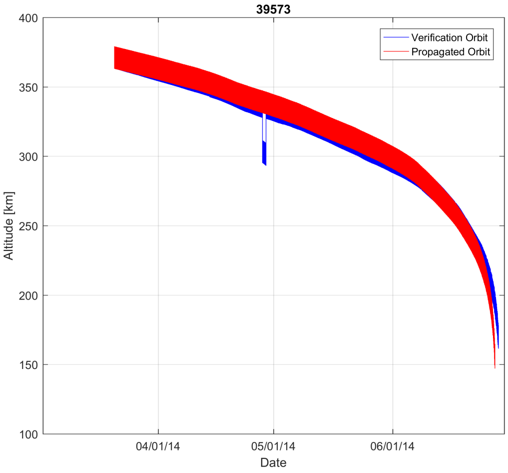


Figure 6.12: Propagated orbit for ITF-1 compared to verification orbit

Table 6.10 contains the manually optimized ballistic coefficients for the three objects to be placed in the spherical satellite database. To confirm the values are accurate, Figure 6.10, Figure 6.11, and Figure 6.12 show the overlap between the propagated orbit (red) and the verification orbit (blue). The bare visibility of the blue verification orbits already shows the ballistic coefficient values have been determined accurate enough to be placed in the other database.

With the addition of these three satellites, the proximity algorithm is run for the CubeSats which have already re-entered the atmosphere to acquire two different re-entry groups. The orbital element selection criteria for these two groups are found in Table 6.11, along with their initial altitude and inclination range.

The resulting composition of the re-entry groups is shown in Table 6.11, followed by Table 6.12 and Table 6.13 containing the NORAD ID's of the CubeSats in each respective group.

Table 6.11: Orbital parameter criteria for the re-entry groups

	Re-entry group 1	Re-entry Group 2
Overall inclination range [degrees]	0 - 110	0 - 110
Overall altitude range [km]	<500	<500
Δ_{max} altitude [km]	40	75
Δ_{max} inclination [degree]	15	20
Δ_{max} RAAN [degree]	30	30
Δ_{max} epoch [days]	10	10
# of cubes in altitude and inclination range	23	23
# of spheres in altitude and inclination range	6	6
# of cubes found with proximity algorithm	10	12
# of spheres found with proximity algorithm	4*	4*
# of matches from proximity algorithm	17	22
Initial spherical satellite*	4	4
Total 1U CubeSats	10	12
Iteration 1	7	10
Iteration 2	3	2

The number of objects in both groups are significantly lower compared to the in-orbit group due to the objects decaying faster in this regime and therefore having a smaller time period in which matches could be found. Since group 1 and 2 are not that different in composition, it was decided to keep with only two groups.

Table 6.12: CubeSats and spheres for re-entry group 1 - sorted on their iteration.

Satellite composition re-entry group 1

Spheres*	AeroCube-3 (35005)	ITF-1 (39573)	SpinSat (40314)
Cubes iteration 1	CP-6 (35003) TechEdSat (38854) OpuSat (39575)	HawkSat-1 (35004) We-Wish (38856)	FitSat-1 (38853) ArduSat-2 (39571)
Cubes iteration 2	ArduSat-1 (39412)	ArduSat-X (39414)	SkyCube (39567)

Table 6.13: CubeSats and spheres for re-entry group 2 - sorted on their iteration.

Satellite composition re-entry group 1

Spheres*	AeroCube-3 (35005)	ITF-1 (39573)	SpinSat (40314)
Cubes iteration 1	CP-6 (35003) UniCubeSat (38085) We-Wish (38856) OpuSat (39575)	HawkSat-1 (35004) FitSat-1 (38853) SkyCube (39567)	Robusta (38084) TechEdSat (38854) ArduSat-2 (39571)
Cubes iteration 2	ArduSat-1 (39412)	ArduSat-X (39414)	

Note the asterix * for when mentioning the initial spheres for the two re-entry groups. Only the SpinSat satellite is a sphere. The other three satellites are the three CubeSats mentioned above that take in the place of a sphere as their ballistic coefficient have already been established and verified.

6.3. Performance Assessment

Depending on whether the object is still in-orbit or performing a re-entry, two different assessment merits have been established. For the in-orbit target group, the difference in the mean semi-major axis of the last 10 orbits of the propagation period is analyzed. To clarify, the Cartesian coordinates of the verification and propagation orbit are used to assess its orbital velocity and subsequently its orbital period. The timespan covering the 10 final orbital periods of the verification and propagation orbit are used to determine a mean semi-major axis for both orbits. The absolute difference with the verification orbit is consecutively computed and stored for comparison with the other scenarios. The motivation behind this approach is to ensure that the orbital variation in the semi-major axis is averaged out equally for all objects.

As this research is to establish the contribution effect of the β estimation and C_D inclusion algorithms and not the orbital propagator, the difference between the verification orbit and the baseline orbit (scenario 1) shall form the baseline to which the other three scenarios are assessed. For example, an object in the in-orbit group exhibiting a s.m.a. difference of 1.3 km between verification and scenario 1 and a s.m.a. difference of 1.1 km between verification orbit and scenario 2 is evaluated to have a performance improvement of $100-(100\cdot\frac{1.1}{1.3}) = 15.4\%$.

Regarding the re-entry target group, the verification orbit's latest TLE is regarded as the re-entry epoch. At this epoch, the re-entry altitude $\text{alt}_{re-entry}$ is determined and used to remove any propagation data points below this altitude. Consecutively the epoch date of the last propagation data point, the one now thus corresponding with $\text{alt}_{re-entry}$, is used to determine the absolute difference in re-entry epoch. Similar to the in-orbit group, the difference between verification orbit and scenario 1 form the baseline of any assessment done for scenario 2, 3, and 4.

For example, the re-entry difference between verification orbit and scenario 1 is 6 days, whilst for verification orbit and scenario 2 this is 3 days. The performance improvement would then be assessed to be $100-(100\cdot\frac{3}{6}) = 50.0\%$

Table 6.14: Propagation and evaluation period for the four scenarios in-orbit group 1, 2, and 3 are subjected to.

Propagation period and assessment merits for in-orbit group 1 and group 2	
Start propagation date	21-April-2016
Propagation period	100 days
Assessment merit 1	S.m.a. difference with verification orbit [for scenario 1-2-3-4]
Assessment merit 2	S.m.a. difference with baseline orbit [for scenario 2-3-4]

Table 6.15: Propagation and evaluation period for the four scenarios re-entry group 1 and group 2 are subjected to.

Propagation period and assessment merits for re-entry group 1 and group 2	
Start propagation date	Last known TLE date minus propagation period
Propagation period	100 days
Assessment merit 1	Re-entry epoch difference with verification orbit [for scenario 1-2-3-4]
Assessment merit 2	Re-entry epoch difference with baseline orbit [for scenario 2-3-4]

For the in-orbit group, the starting propagation date is specifically set at 21-April 2016 as with this date propagation for 100 days would still be possible for all objects and have the availability of the required solar activity data.

7

Results

The experimental set-up explained the four different propagation scenarios to assess the effectiveness of estimating a ballistic coefficient and the subsequent adjustment of β_E to accommodate for the varying drag coefficient. Within the theory of computing an objects C_D a distinction can be made with the selection of the energy accommodation coefficient α - a constant value of $\alpha = 0.8$ can be assumed or an α that is related to the local presence of atomic oxygen, also known as Langmuir's isotherm. As discussed, the ballistic coefficients related to these four scenarios can be recognized with these four different subscripts: β_0, β_E, $\beta_{C_D Langmuir}$, and $\beta_{C_D,\alpha=0.8}$ for scenario one to four respectively.

The objects that are subjected to each scenario have been discussed as well, with the main distinction being between the objects being **in-orbit** or experiencing a **re-entry**. Within these two categories, three groups are created for the in-orbit category and two groups for the re-entry category, each with different orbital element selection criteria for the proximity algorithm - thus resulting in a different CubeSat composition for each group. Furthermore, both categories are assessed according to their respective merits, which have been discussed in section 6.3. In that section, the propagation periods for both categories have been discussed as well, and though the start of the propagation periods are different between the two categories, all objects in every group are propagated for an equal period of 100 days.

This chapter shall present (a summary of) the propagation results for all groups discussed in the experimental set-up. The way the data is presented is uniform irregardless of the category, and follows the following structure for each group:

- 1. Presentation of β_E values and the mean and median of the β_E values.
- Presentation of $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values and the mean and median of these values.
- Presentation of the propagation results in the format of their respective assessment merit.

As the propagation data output of the objects within each group is simply to large to display separately in each section, most of this data is stored in the appendices. However, summaries of the propagation results are of course presented in their relevant sections. These summaries have been established identically for each group, both for the in-orbit and re-entry groups, as to eventually have a consistent overview of the output data for all these groups. The results are discussed further in chapter 8, after which the conclusions and recommendations are presented in that chapter as well.

7.1. In-orbit category

This section provides the results for the in-orbit propagation groups as discussed in subsection 6.2.1. To revise, cubes and spheres orbiting above 500 km and currently still orbiting Earth have been subjected to the proximity algorithm with three different orbital parameter criteria sets. This resulted in three groups, see Table 6.7, Table 6.8, and Table 6.9, which consist largely of the same satellites though the spread in number of objects per iteration is different from each other. All objects in turn have their ballistic coefficient estimated as per procedure discussed in section 4.1. The resulting ballistic coefficients have consecutively been adjusted for the varying drag coefficient experienced between the determination epoch and start of the propagation period.

Each object is propagated for 100 days starting from 21-April-2016 until 30-July-2016 for four scenarios using the TUDAT propagation software to assess the performance of newly estimated β_E values, and the influence of the inclusion of a variable drag coefficient and Langmuir's isotherm. Reasoning behind these specific dates is that within this time period it can be ensured the latest solar and geomagnetic indices can be used in the TUDAT propagation model.

Within the in-orbit category three groups have been established with both different orbital parameter criteria for the proximity algorithm. These criteria and the resulting composition of group 1, group 2, and group 3 have been discussed before and are recapped below:

Table 7.1: Orbital parameter criteria for the in-orbit groups

	In-orbit Group 1	In-orbit Group 2	In-orbit Group 3
Overall inclination range [degrees]	50 - 110	50 - 110	50 - 110
Overall altitude range [km]	500 - 1000	500 - 1000	500 - 1000
Δ_{max} altitude [km]	40	50	25
Δ_{max} inclination [degree]	10	10	15
Δ_{max} RAAN [degree]	20	20	30
Δ_{max} epoch [days]	10	10	10
# of cubes in altitude and inclination range	59	59	59
# of spheres in altitude and inclination range	23	23	23
# of cubes found with proximity algorithm	43	51	52
# of spheres found with proximity algorithm	6	5	6
# of matches from proximity algorithm	146	192	126

For each of the three in-orbit groups, the results are presented in an consequent manner - for each group the estimated ballistic coefficients β_E of all objects from scenario 2 are presented. The average difference per iteration group β_E has with β_0 is given, as well as the mean and median β_E values. Consecutively the adjusted $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ are presented, corresponding to scenario 3 and 4 respectively.

Again an overview of the differences these values have with β_E is given, as well as the mean and median values, both in a table as graphically. The result section for that particular group is concluded with a summary of the propagation results for all scenarios, with the complete list of propagation results being presented in the appendix.

7.1.1. Group 1 - Estimated ballistic coefficients

Presented are the β_E values for **in-orbit group 1**. The procedure to acquire these values corresponds to the procedure explained in section 4.1 and using the proximity algorithm described in subsection 5.2.2. The ranges for these criteria have been discussed at the beginning of this section.

Table 7.2 contains the estimated β_E values for the in-orbit group 1. The difference in percentage with $\beta_{baseline}$ has been included as well to give a sense of the order of magnitude difference it had with the initially assumed ballistic coefficient. Though not used in any calculations henceforth, Table 6.4 contains the average difference for each iteration.

Table 7.2: Estimated β_E values for in-orbit group 1, compared to $\beta_{baseline}$.

NORAD	β_E	$\Delta\beta$	Iteration #	NORAD	β_E	$\Delta\beta$	Iteration #
DTUSAT (27842)	0.0358623	14.76	1	ICube-1 (39432)	0.0543246	73.84	3
CUTE-1 (27844)	0.0374755	19.92	1	HumSat-D (39433)	0.0697332	123.15	3
AAU CubeSat (27846)	0.0295936	-5.3	1	PUCP-Sat-1 (39442)	0.058023	85.67	3
CANX-1 (27847)	0.0297268	-4.87	1	FunCube-1 (39444)	0.0388896	24.45	3
XI-4 (27848)	0.0339202	8.54	1	HinCube (39445)	0.0373839	19.63	3
SDS-1 (33497)	0.0257445	-17.62	1	DuchiFat-1 (40021)	0.0528104	68.99	3
SwissCube (35932)	0.0671967	115.03	1	C-BR1 (40024)	0.0475302	52.1	3
BeeSat (35933)	0.0581878	86.2	1	DTUSat-2 (40030)	0.0392147	25.49	3
Uwe-2 (35934)	0.0515293	64.89	1	PolyItan-1 (40042)	0.0478618	53.16	3
ITUPSAT-1 (35935)	0.0539599	72.67	1	UFTI-1 (41458)	0.0305887	-2.12	3
AAUSat-3 (39087)	0.038552	23.37	1	E-Star-1 (41459)	0.031164	-0.28	3
Brite-PL (39431)	0.0219296	-29.83	1	AAUSat-4 (41460)	0.0433227	38.63	3
Dice-1 (37851)	0.030932	-1.02	2	Pegasus-1 (39151)	0.0532586	70.43	4
Dice-2 (37852)	0.037905	21.3	2	CubeBug-1 (39153)	0.0406202	29.98	4
WNISAT-1 (39423)	0.024921	-20.25	2	EstCube-1 (39161)	0.0401614	28.52	4
EgyptSat-1 (31117)	0.0269426	-13.78	3	Velox-P-2 (39438)	0.0389125	24.52	4
Compass-1 (32787)	0.0464152	48.53	3	First-Move (39439)	0.0773811	147.62	4
AAUSAT-2 (32788)	0.0526421	68.45	3	KRYSAOR (39441)	0.0356835	14.19	4
SEEDS (32791)	0.0443297	41.86	3	Uwe-3 (39446)	0.0434253	38.96	4
AubieSat-1 (37854)	0.0349132	11.72	3	CANX-4 (40055)	0.032113	2.76	4
M-Cubed (37855)	0.0343275	9.85	3	CANX-5 (40056)	0.037138	18.84	4
ZaCube (39417)	0.0399416	27.81	3				

Table 7.3: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for in-orbit group 1.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Average Δ between β_E and $\beta_{baseline}$ [%]	28.98	0.01	39.85	41.76

Furthermore, the average difference between β_E and $\beta_{baseline}$ is provided in Table 7.3. This difference is the average of the absolute percentage differences presented in Table 7.2 and demonstrates a significant overall change from the initially assumed $\beta_{baseline}$. The β_E values have been put in a histogram, Figure 7.1, to visualize the distribution of β_E. The mean and median values of in-orbit group 1 are present in Table 7.4.

Table 7.4: Mean and median for in-orbit group 1.

	Mean β_E	Median β_E
In-orbit group 1	0.037668	0.034856

7.1.2. Group 1 - Adjusted to accommodate for varying drag coefficient

Per procedure discussed in section 4.2 the β_E values have been directly adjusted to the $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values. These values are displayed in Table 7.5, sorted for each iteration group. The values for the average percental difference is presented in Table 7.6.

Table 7.5: Adjusted $\beta_{C_D langmuir}$ and $\beta_{C_D \alpha=0.8}$ values for in-orbit group 1, compared to β_E.

NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$	NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$
DTUSAT (27842)	0.034311	0.032451	ICube-1 (39432)	0.044467	0.041947
CUTE-1 (27844)	0.039497	0.037306	HumSat-D (39433)	0.060203	0.056772
AAU CubeSat (27846)	0.031791	0.030058	PUCP-Sat-1 (39442)	0.054057	0.050996
CANX-1 (27847)	0.032295	0.030605	FunCube-1 (39444)	0.033432	0.03158
XI-4 (27848)	0.035988	0.033977	HinCube (39445)	0.032133	0.030399
SDS-1 (33497)	0.024665	0.023371	DuchiFat-1 (40021)	0.042601	0.040178
SwissCube (35932)	0.064943	0.062098	C-BR1 (40024)	0.037968	0.035806
BeeSat (35933)	0.055392	0.052737	DTUSat-2 (40030)	0.031481	0.029690
Uwe-2 (35934)	0.048969	0.046802	PolyItan-1 (40042)	0.038554	0.036355
ITUPSAT-1 (35935)	0.051181	0.048745	UFTI-1 (41458)	0.024553	0.024348
AAUSat-3 (39087)	0.041866	0.039643	E-Star-1 (41459)	0.024939	0.024730
Brite-PL (39431)	0.023423	0.022462	AAUSat-4 (41460)	0.034783	0.034501
Dice-1 (37851)	0.025739	0.025286	Pegasus-1 (39151)	0.046748	0.044182
Dice-2 (37852)	0.031201	0.030934	CubeBug-1 (39153)	0.034600	0.032945
WNISAT-1 (39423)	0.030957	0.029492	EstCube-1 (39161)	0.034905	0.033110
EgyptSat-1 (31117)	0.025340	0.024002	Velox-P-2 (39438)	0.032549	0.030741
Compass-1 (32787)	0.033791	0.031873	First-Move (39439)	0.065237	0.061561
AAUSAT-2 (32788)	0.041938	0.039594	KRYSAOR (39441)	0.031583	0.029882
SEEDS (32791)	0.039215	0.037010	Uwe-3 (39446)	0.035976	0.034002
AubieSat-1 (37854)	0.025135	0.024858	CANX-4 (40055)	0.028948	0.027385
M-Cubed (37855)	0.024602	0.024452	CANX-5 (40056)	0.033732	0.031945
ZaCube (39417)	0.035543	0.033526			

Table 7.6: Average difference in percentage between estimated ballistic coefficient and scenario 3 and scenario 4 for in-orbit group 1.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Average Δ between $\beta_{C_D Langmuir}$ and β_E	2.43%	-4.32%	-3.5%	-1.44%
Average Δ between $\beta_{C_D \alpha=0.8}$ and β_E	-2.49%	-7.45%	-7.85%	-6.72%

The data from Table 7.5 is used to display the difference it has with the β_E values, as well as the average difference in percentage for each iteration group, in Figure 7.2. In section 8.1 the expected $\beta_{C_D Langmuir}$ and $\beta_{C_D, \alpha=0.8}$ are compared to the actual output data, where a discussion is held whether the results follow the trend which was to be expected from the theory.

Similarly to what was done for the β_E values, the mean and median have been computed and can be found in Table 7.7.

Table 7.7: Mean and median for in-orbit group 1.

	Mean $\beta_{C_D Langmuir}$	Median $\beta_{C_D Langmuir}$	Mean $\beta_{C_D \alpha=0.8}$	Median $\beta_{C_D \alpha=0.8}$
In-orbit group 1	0.037237	0.03460	0.0354496	0.032945

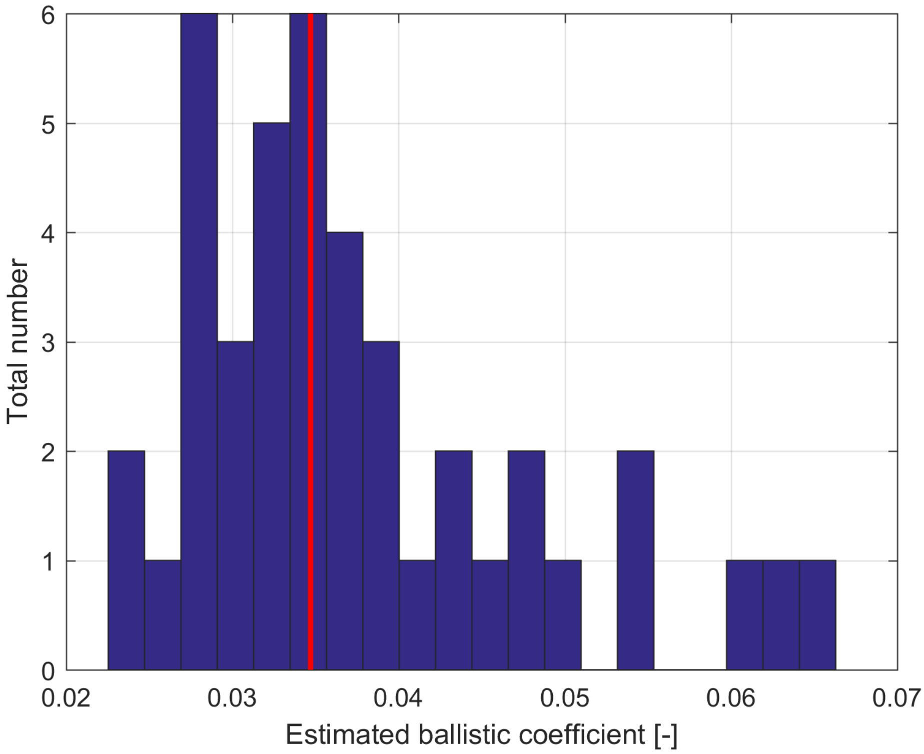


Figure 7.1: Histogram of B_E values for in-orbit group 1

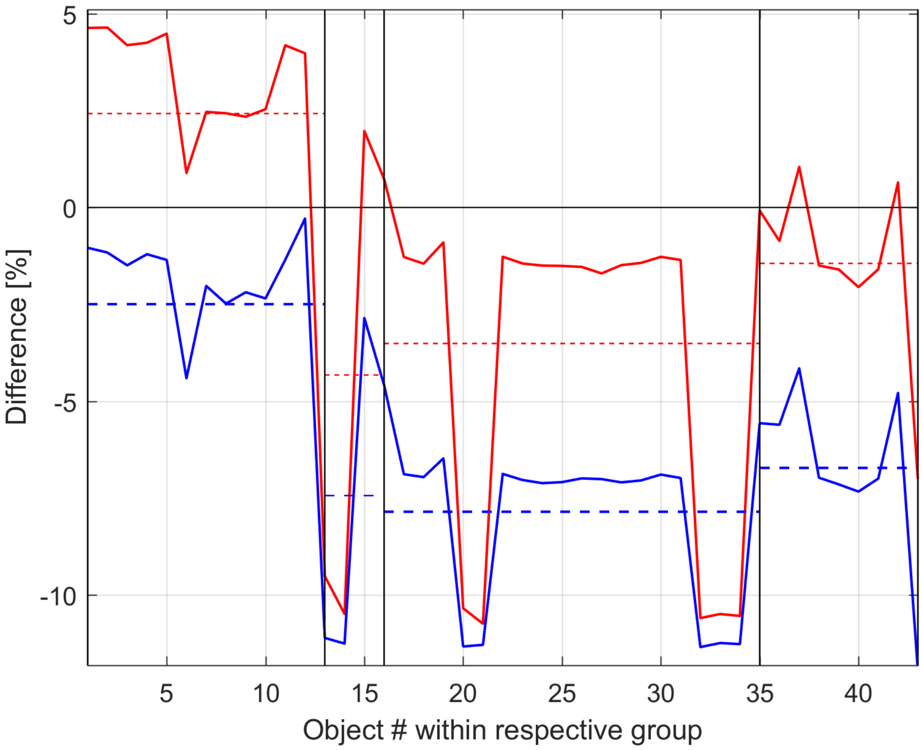


Figure 7.2: Difference of $\beta_{C_D,Langmuir}$ (red) and $\beta_{C_D,\alpha=0.8}$ (blue) values with β_E for in-orbit group 1

7.1.3. Propagation results in-orbit group 1

Having estimated and computed all required β_E and β_{C_D} values, the objects are subjected to the propagation software and assessed based on their difference in semi-major-axis of their last 10 orbits. The data is stored in Table A.1, where column 6, 7, and 8 represent the difference in kilometer between the baseline propagation and scenario 2, 3, and 4 respectively.

A histogram and probability density function for the propagation results of scenario 2, 3, and 4 has been created and can be seen in Figure 7.3, Figure 7.4, and Figure 7.5 respectively. For the propagation result data presented henceforth **outlying data points exceeding 3σ have been removed**. Note that a positive x-axis represent an improvement compared to the baseline propagation. Finally, the mean and variance of each scenario is presented in Table 7.8.

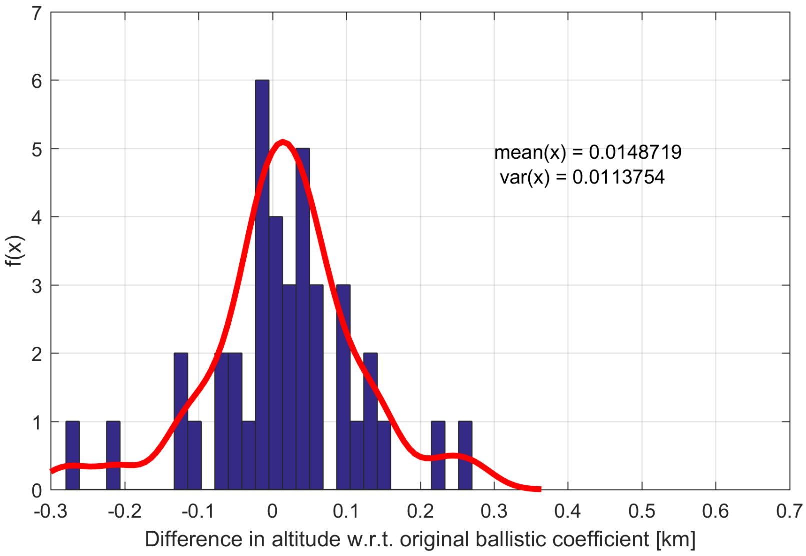


Figure 7.3: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 2 - in-orbit 1.

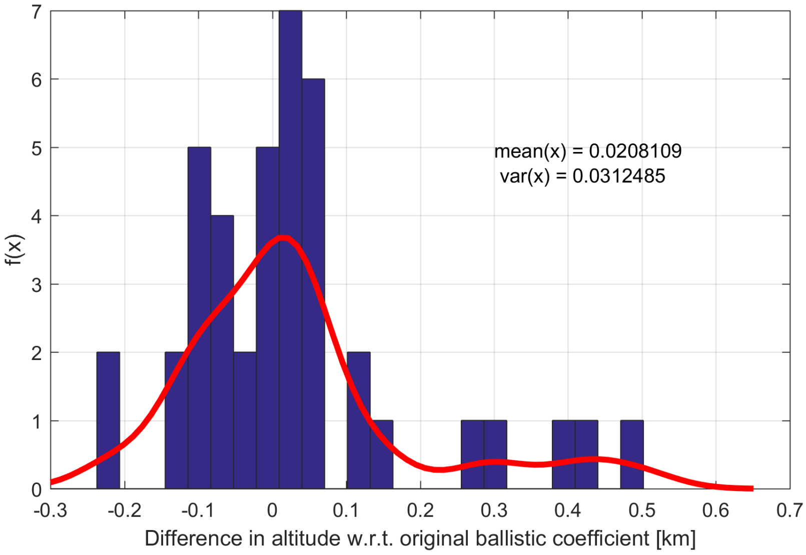


Figure 7.4: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 3 - in-orbit 1.

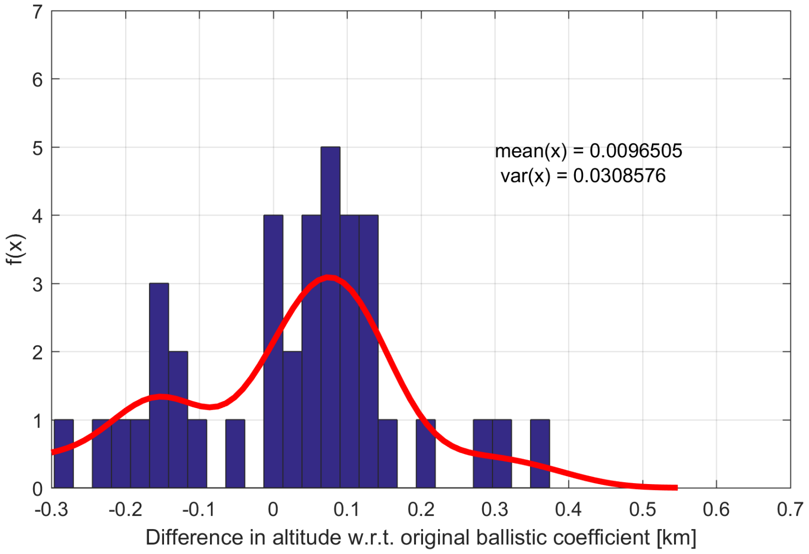


Figure 7.5: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 4 - in-orbit 1.

Table 7.8: Mean and variance for each scenario for in-orbit group 1

	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
Mean	0.01487	0.0206	0.00965
Variance	0.01137	0.03125	0.03085

Additionally, all altitude gains (or losses) compared to the baseline propagation is summarized in Table 7.9. In this table the overall altitude improvements is presented for each scenario, as well as the subsequent improvements per iteration. If interested, the reader can find the differences between the verification orbits and the propagated baseline and scenario orbits back in Table A.1.

Table 7.9: Mean propagation difference and overall improvement [km] w.r.t. baseline orbit for each scenario in in-orbit group 1. Outlying data points exceeding 3σ have been removed.

	Baseline	Scenario 2	Scenario 3	Scenario 4
Iteration 1 - Mean Δ with baseline	-	0.232	0.171	0.676
Iteration 2 - Mean Δ with baseline	-	0.104	0.464	-0.072
Iteration 3 - Mean Δ with baseline	-	-0.665	0.486	-0.701
Iteration 4 - Mean Δ with baseline	-	-0.067	-0.248	-0.092
Difference with verification orbit	10.392	10.789	9.519	10.581
Improvement w.r.t. baseline [km]	-	-0.397	0.874	-0.189
Improvement w.r.t. baseline [%]	-	-3.82	8.41	-1.82
Results with data points exceeding 3σ removed				
Difference with verification orbit	9.8823	9.2878	8.4896	9.1922
Improvement w.r.t. baseline [km]	-	0.5945	1.3927	0.6901
Improvement w.r.t. baseline [%]	-	6.01	14.09	6.98

7.1.4. Group 2 - Estimated ballistic coefficients

Table 7.10 contains the estimated β_E values for the in-orbit group 2. The difference in percentage with $\beta_{baseline}$ has been included as well to give a sense of the order of magnitude difference it had with the initially assumed ballistic coefficient. Though not used in any calculations henceforth, Table 7.11 contains the average difference for each iteration.

Table 7.10: Estimated β_E values for in-orbit group 2, compared to $\beta_{baseline}$.

NORAD	β_E	$\Delta\beta$	Iter. #
DTUSAT (27842)	0.0210409	-32.67	1
CUTE-1 (27844)	0.0292684	-6.34	1
AAU CubeSat (27846)	0.0225876	-27.72	1
CANX-1 (27847)	0.0214136	-31.48	1
XI-4 (27848)	0.0262563	-15.98	1
Uwe-1 (28892)	0.0180465	-42.25	1
XI-5 (28895)	0.0176252	-43.6	1
SDS-1 (33497)	0.0087204	-72.09	1
SwissCube (35932)	0.0148236	-52.56	1
BeeSat (35933)	0.0189423	-39.38	1
Uwe-2 (35934)	0.0116449	-62.74	1
ITUPSAT-1 (35935)	0.0172787	-44.71	1
Dice-1 (37851)	0.0381696	22.14	1
Dice-2 (37852)	0.0281322	-9.98	1
AubieSat-1 (37854)	0.0354586	13.47	1
M-Cubed (37855)	0.0363141	16.21	1
EgyptSat-1 (31117)	0.0232904	-25.47	2
Compass-1 (32787)	0.0243445	-22.1	2
AAUSAT-2 (32788)	0.0571467	82.87	2
SEEDS (32791)	0.0446805	42.98	2
Pegasus-1 (39151)	0.0340485	8.96	2
CubeBug-1 (39153)	0.0371008	18.72	2
EstCube-1 (39161)	0.0123144	-60.59	2
ZaCube (39417)	0.0382208	22.31	2
ICube-1 (39432)	0.0444745	42.32	2
HumSat-D (39433)	0.0586152	87.57	2
Velox-P-2 (39438)	0.0363764	16.4	2
First-Move (39439)	0.0797694	155.26	2
PUCP-Sat-1 (39442)	0.0479971	53.59	2
FunCube-1 (39444)	0.0446474	42.87	2
HinCube (39445)	0.0434722	39.11	2
Uwe-3 (39446)	0.0429833	37.55	2
DuchiFat-1 (40021)	0.0560602	79.39	2
C-BR1 (40024)	0.0498429	59.5	2
DTUSat-2 (40030)	0.0415419	32.93	2
PolyItan-1 (40042)	0.0504061	61.3	2
UFTI-1 (41458)	0.0357838	14.51	2
E-Star-1 (41459)	0.0330449	5.74	2
AAUSat-4 (41460)	0.0503592	61.15	2
CSTB-1 (31122)	0.0362094	15.87	3
CP-4 (31132)	0.0320172	2.45	3
AeroCube-2 (31133)	0.0285963	-8.49	3
KRYSAOR (39441)	0.0291033	-6.87	3
CANX-4 (40055)	0.0253088	-19.01	3
CANX-5 (40056)	0.0492311	57.54	3
Libertad-1 (31128)	0.0340605	8.99	4
CP-3 (31129)	0.0362997	16.16	4
Cape-1 (31130)	0.0340953	9.11	4
SDS-4 (38339)	0.0152165	-51.31	4
Brite-PL (39431)	0.0147273	-52.87	4
WNISAT-1 (39423)	0.0167363	-46.44	5

Table 7.11: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for in-orbit group 2.

Iteration	1	2	3	4	5
Average Δ between β_E and $\beta_{baseline}$	-5.01%	33.90%	6.92%	-13.92%	-46.44%

Table 7.12: Mean and median for in-orbit group 2.

	Mean β_E	Median β_E
In-orbit group 2	0.0334087	0.0340605

In Table 7.12 the median and mean β_E values are displayed. Figure 7.6 contains a histogram of the estimated ballistic coefficients for in-orbit group 2. The median is colored in red.

7.1.5. Group 2 - Adjusted to accommodate for varying drag coefficient

Per procedure discussed in section 4.2 the β_E values have been directly adjusted to the $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values. These values are displayed in Table 7.13, sorted for each iteration group.

Table 7.13: Estimated $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values for in-orbit group 2, compared to $\beta_{baseline}$.

NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$	NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$
DTUSAT (27842)	0.022113	0.020787	Velox-P-2 (39438)	0.044421	0.042204
CUTE-1 (27844)	0.030813	0.028978	First-Move (39439)	0.034143	0.032516
AAU CubeSat (27846)	0.023737	0.022314	PUCP-Sat-1 (39442)	0.037162	0.035412
CANX-1 (27847)	0.022554	0.021217	FunCube-1 (39444)	0.012443	0.011824
XI-4 (27848)	0.027583	0.025918	HinCube (39445)	0.037879	0.035947
Uwe-1 (28892)	0.018408	0.017455	Uwe-3 (39446)	0.044112	0.041795
XI-5 (28895)	0.017994	0.017068	DuchiFat-1 (40021)	0.058090	0.055022
SDS-1 (33497)	0.008793	0.008356	C-BR1 (40024)	0.035946	0.034167
SwissCube (35932)	0.015188	0.014470	DTUSat-2 (40030)	0.078776	0.074778
BeeSat (35933)	0.019391	0.018562	PolyItan-1 (40042)	0.047593	0.045093
Uwe-2 (35934)	0.011915	0.011366	UFTI-1 (41458)	0.044054	0.041986
ITUPSAT-1 (35935)	0.017709	0.016881	E-Star-1 (41459)	0.042935	0.040881
Dice-1 (37851)	0.034529	0.034186	AAUSat-4 (41460)	0.042467	0.040364
Dice-2 (37852)	0.025164	0.025154	CSTB-1 (31122)	0.037383	0.035890
AubieSat-1 (37854)	0.032003	0.031825	CP-4 (31132)	0.033145	0.031851
M-Cubed (37855)	0.032640	0.032615	AeroCube-2 (31133)	0.029619	0.028469
EgyptSat-1 (31117)	0.055458	0.052611	KRYSAOR (39441)	0.028677	0.027259
Compass-1 (32787)	0.049337	0.046803	CANX-4 (40055)	0.025476	0.024298
AAUSAT-2 (32788)	0.041190	0.039072	CANX-5 (40056)	0.049471	0.047146
SEEDS (32791)	0.049932	0.047363	Libertad-1 (31128)	0.035412	0.034115
Pegasus-1 (39151)	0.032115	0.032052	CP-3 (31129)	0.037797	0.036396
CubeBug-1 (39153)	0.029690	0.029633	Cape-1 (31130)	0.035490	0.034189
EstCube-1 (39161)	0.045237	0.045149	SDS-4 (38339)	0.015414	0.014710
ZaCube (39417)	0.023540	0.022448	Brite-PL (39431)	0.015340	0.014774
ICube-1 (39432)	0.024322	0.023010	WNISAT-1 (39423)	0.017097	0.016357
HumSat-D (39433)	0.056592	0.053737			

Table 7.14: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for in-orbit group 2.

Iteration	1	2	3	4	5
Average Δ between $\beta_{C_D Langmuir}$ and β_E	0.09%	-1.66%	1.99%	3.30%	2.15%
Average Δ between $\beta_{C_D \alpha=0.8}$ and β_E	-4.14%	-6.02%	-2.31%	-0.76%	-2.27%

Table 7.15: Mean and median for in-orbit group 2.

	Mean $\beta_{C_D Langmuir}$	Median $\beta_{C_D Langmuir}$	Mean $\beta_{C_D \alpha=0.8}$	Median $\beta_{C_D \alpha=0.8}$
In-orbit group 2	0.033103	0.03264	0.031695	0.032052

The data from Table 7.13 is used to display the difference it has with the β_E values, as well as the average difference in percentage for each iteration group, in Figure 7.7. In section 8.1 the expected $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ are compared to the actual output data, where a discussion is held whether the results follow the trend which was to be expected from the theory.

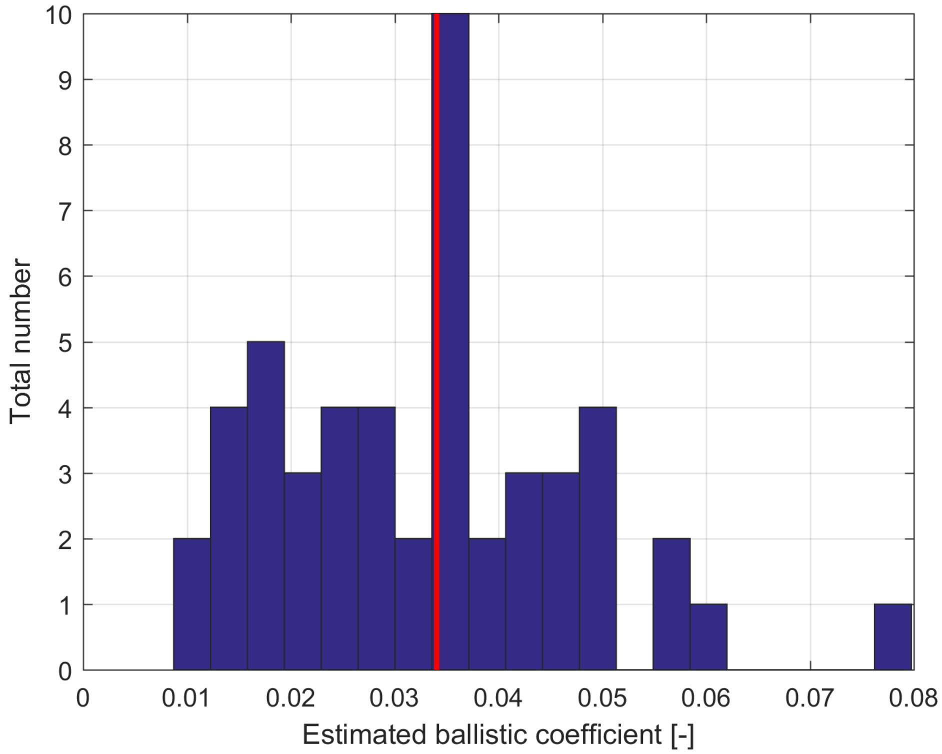


Figure 7.6: Histogram of β_E values for in-orbit group 2

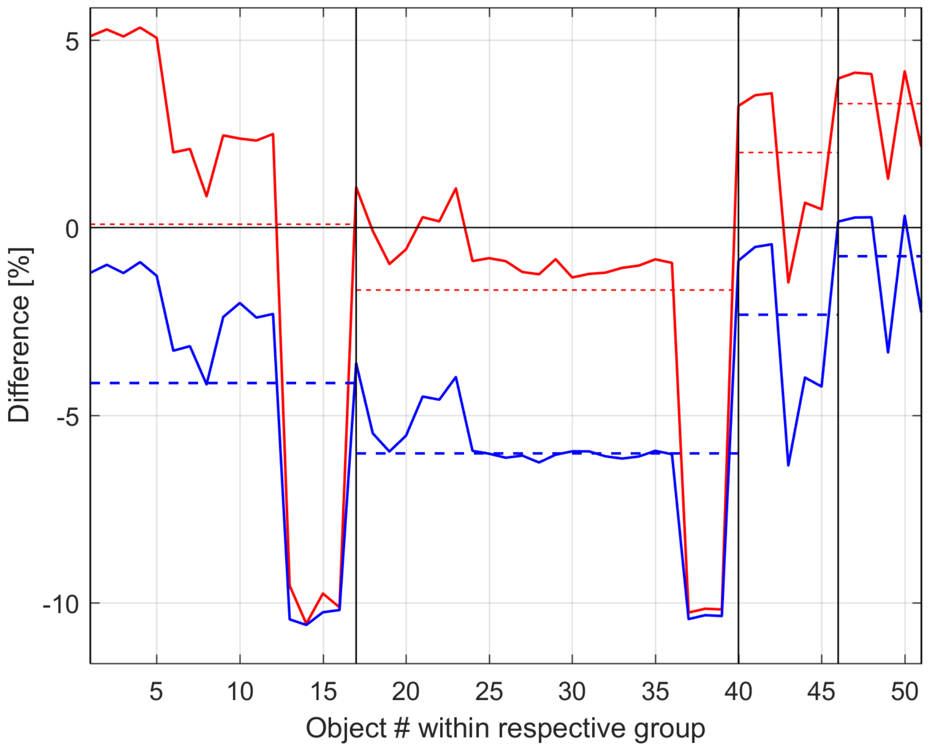


Figure 7.7: Difference of $\beta_{C_D,Langmuir}$ (red) and $\beta_{C_D,\alpha=0.8}$ (blue) values with β_E for in-orbit group 2

7.1.6. Propagation results in-orbit group 2

Having estimated and computed all required β_E and β_{C_D} values, the objects are subjected to the propagation software and assessed based on their difference in semi-major-axis of their last 10 orbits. The data is stored in section A.2, where column 6, 7, and 8 represent the difference in kilometer between the baseline propagation and scenario 2, 3, and 4 respectively.

A histogram and probability density function for the propagation results of scenario 2, 3, and 4 has been created and can be seen in Figure 7.8, Figure 7.9, and Figure 7.10 respectively. For the propagation result data presented henceforth **outlying data points exceeding 3σ have been removed**. Note that a positive x-axis represent an improvement compared to the baseline propagation. Finally, the mean and variance of each scenario is presented in Table 7.16.

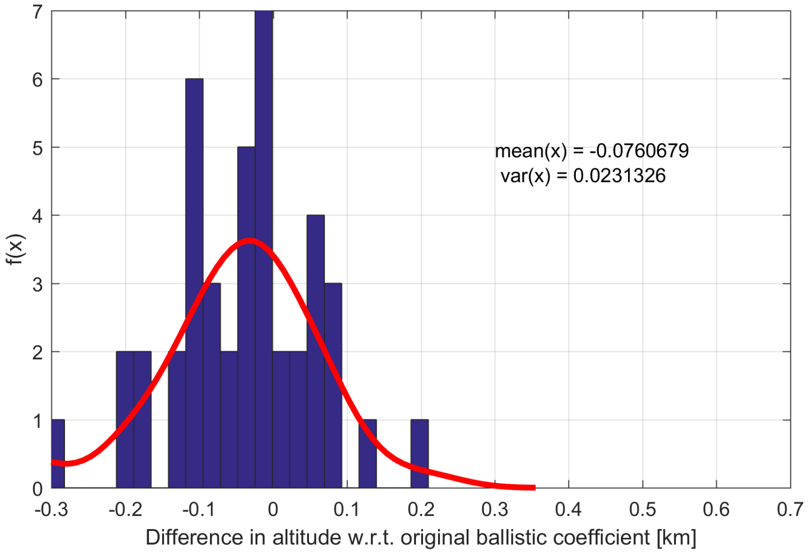


Figure 7.8: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 2 - in-orbit 2.

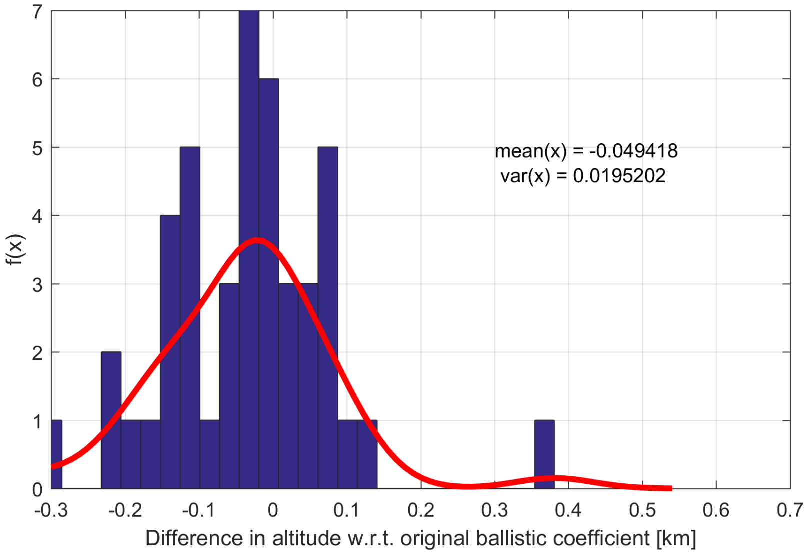


Figure 7.9: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 3 - in-orbit 2.

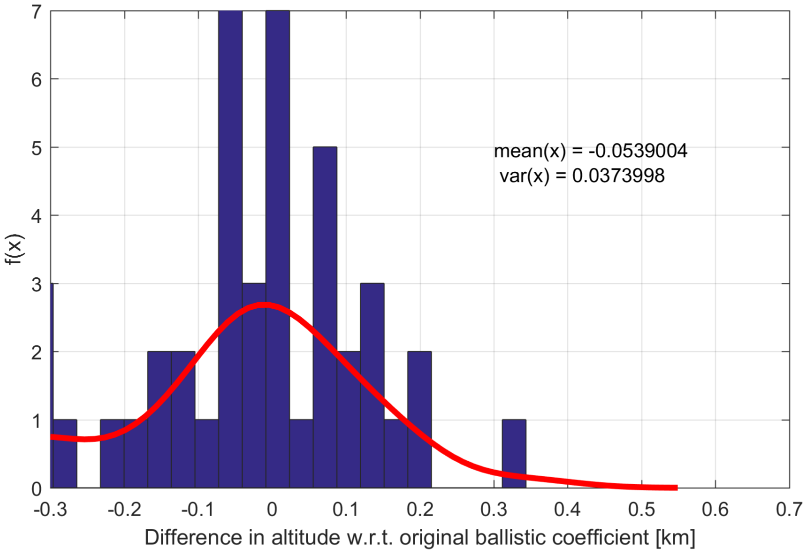


Figure 7.10: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 4 - in-orbit 2.

Table 7.16: Mean and variance for each scenario for in-orbit group 2

	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
Mean	-0.0761	-0.0494	-0.0539
Variance	0.023	0.019	0.037

Additionally, all altitude gains (or losses) compared to the baseline propagation is summarized in Table 7.17. In this table the overall altitude improvements is presented for each scenario, as well as the subsequent improvements per iteration. If interested, the reader can find all propagation results for the four scenarios back in section A.2.

Table 7.17: Mean propagation difference and overall improvement [km] w.r.t. baseline orbit for each scenario in in-orbit group 2.

	Baseline	Scenario 2	Scenario 3	Scenario 4
Iteration 1 - Mean Δ with baseline	-	-1.608	-0.408	-0.080
Iteration 2 - Mean Δ with baseline	-	-3.157	-2.737	-3.623
Iteration 3 - Mean Δ with baseline	-	-0.146	-0.095	-0.217
Iteration 4 - Mean Δ with baseline	-	-0.054	-0.098	0.403
Iteration 5 - Mean Δ with baseline	-	0.047	-0.013	-0.005
Difference with verification orbit	11.916	16.833	15.267	15.437
Improvement w.r.t. baseline [km]	-	-4.917	-3.351	-3.521
Improvement w.r.t. baseline [%]	-	-41.27	-28.13	-29.55

Results with data points exceeding 3σ removed				
Difference with verification orbit	11.4715	15.1997	13.8936	14.1136
Improvement w.r.t. baseline [km]	-	-3.7282	-2.4221	-2.6421
Improvement w.r.t. baseline [%]	-	-32.49	-21.15	-23.03

7.1.7. Group 3 - Estimated ballistic coefficients

Table 7.18 contains the estimated β_E values for the in-orbit group 3. The difference in percentage with $\beta_{baseline}$ has been included as well to give a sense of the order of magnitude difference it had with the initially assumed ballistic coefficient. Though not used in any calculations henceforth, Table 7.19 contains the average difference for each iteration. These values shall be considered later in section 8.1 to address the issue of error heritage through the iterations.

Table 7.18: Estimated β_E values for in-orbit group 3, compared to $\beta_{baseline}$.

NORAD	β_E	$\Delta\beta$	Iter. #	NORAD	β_E	$\Delta\beta$	Iter. #
DTUSAT (27842)	0.031250	0.00	1	CANX-5 (40056)	0.006590	78.91	2
CUTE-1 (27844)	0.039130	-25.22	1	CSTB-1 (31122)	0.030464	2.52	3
AAU CubeSat (27846)	0.028797	7.85	1	SEEDS (32791)	0.015653	49.91	3
CANX-1 (27847)	0.029346	6.09	1	SDS-4 (38339)	0.004446	85.77	3
XI-4 (27848)	0.033139	-6.04	1	Brite-PL (39431)	0.012414	60.28	3
Uwe-1 (28892)	0.018046	42.25	1	ICube-1 (39432)	0.009888	68.36	3
XI-5 (28895)	0.017625	43.60	1	HumSat-D (39433)	0.012381	60.38	3
SDS-1 (33497)	0.005000	84.00	1	PUCP-Sat-1 (39442)	0.010540	66.27	3
SwissCube (35932)	0.028296	9.45	1	Libertad-1 (31128)	0.023133	25.97	4
BeeSat (35933)	0.024614	21.23	1	CP-3 (31129)	0.028815	7.79	4
Uwe-2 (35934)	0.009256	70.38	1	Cape-1 (31130)	0.023525	24.72	4
AAUSAT-3 (39087)	0.072497	-131.99	1	CP-4 (31132)	0.014460	53.73	4
EgyptSat-1 (31117)	0.005600	82.08	2	AeroCube-2 (31133)	0.014551	53.44	4
ItupSat-1 (35935)	0.018851	39.68	2	Compass-1 (32787)	0.013613	56.44	4
Pegasus-1 (39151)	0.006973	77.69	2	AAUSAT-2 (32788)	0.014291	54.27	4
CubeBug-1 (39153)	0.007976	74.48	2	Dice-1 (37851)	0.008442	72.98	4
EstCube-1 (39161)	0.008711	72.12	2	Dice-2 (37852)	0.009319	70.18	4
ZaCube (39417)	0.009765	68.75	2	AubieSat-1 (37854)	0.009434	69.81	4
WNISAT-1 (39423)	0.014098	54.89	2	M-Cubed (37855)	0.008738	72.04	4
Velox-P-2 (39438)	0.007834	74.93	2	DuchiFat-1 (40021)	0.012323	60.57	5
First-Move (39439)	0.014830	52.54	2	UFTI-1 (41458)	0.009867	68.43	5
KRYSAOR (39441)	0.007375	76.40	2	E-Star-1 (41459)	0.009095	70.90	5
FunCube-1 (39444)	0.012423	60.25	2	AAUSat-4 (41460)	0.013502	56.79	5
HinCube (39445)	0.011667	62.67	2	C-BR1 (40024)	0.010959	64.93	6
Uwe-3 (39446)	0.010559	66.21	2	DTUSat-2 (40030)	0.009123	70.81	6
CANX-4 (40055)	0.006261	79.97	2	PolyItan-1 (40042)	0.011036	64.69	6

Table 7.19: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for in-orbit group 3.

Iteration	1	2	3	4	5	6
Average Δ between β_E and $\beta_{baseline}$	10.13%	68.10%	56.21%	51.03%	64.17%	66.81%

Table 7.20: Mean and median for in-orbit group 3.

	Mean β_E	Median β_E
In-orbit group 3	0.0162	0.0124

Figure 7.11 contains a histogram of the estimated ballistic coefficients for in-orbit group 3. The median is colored in red.

7.1.8. Group 3 - Adjusted to accommodate for varying drag coefficient

Per procedure discussed in section 4.2 the β_E values have been directly adjusted to the $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values. These values are displayed in Table 7.21, sorted for each iteration group.

Table 7.21: Estimated $\beta_{C_D Langmuir}$ and $\beta_{C_D \alpha=0.8}$ values for in-orbit group 3, compared to $\beta_{baseline}$.

NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$	NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$
DTUSAT (27842)	0.032653	0.030941	CANX-5 (40056)	0.006621	0.006293
CUTE-1 (27844)	0.040921	0.038832	CSTB-1 (31122)	0.031347	0.029929
AAU CubeSat (27846)	0.030096	0.028416	SEEDS (32791)	0.0155	0.014687
CANX-1 (27847)	0.030726	0.029138	SDS-4 (38339)	0.004486	0.004269
XI-4 (27848)	0.034644	0.032693	Brite-PL (39431)	0.012888	0.012343
Uwe-1 (28892)	0.018416	0.017431	ICube-1 (39432)	0.009758	0.009227
XI-5 (28895)	0.018003	0.017046	HumSat-D (39433)	0.012174	0.011536
SDS-1 (33497)	0.005041	0.004795	PUCP-Sat-1 (39442)	0.010404	0.009832
SwissCube (35932)	0.028992	0.027637	Libertad-1 (31128)	0.024044	0.023038
BeeSat (35933)	0.025207	0.024027	CP-3 (31129)	0.029914	0.028648
Uwe-2 (35934)	0.009476	0.009035	Cape-1 (31130)	0.024443	0.023417
AAUSAT-3 (39087)	0.075508	0.071659	CP-4 (31132)	0.01502	0.014362
EgyptSat-1 (31117)	0.005637	0.005361	AeroCube-2 (31133)	0.015156	0.014467
ItupSat-1 (35935)	0.019342	0.018435	Compass-1 (32787)	0.01369	0.012996
Pegasus-1 (39151)	0.006967	0.006613	AAUSAT-2 (32788)	0.014045	0.013342
CubeBug-1 (39153)	0.007965	0.00756	Dice-1 (37851)	0.007852	0.007763
EstCube-1 (39161)	0.008799	0.008376	Dice-2 (37852)	0.008269	0.008294
ZaCube (39417)	0.009634	0.009123	AubieSat-1 (37854)	0.008455	0.008389
WNISAT-1 (39423)	0.014354	0.013657	M-Cubed (37855)	0.007786	0.007772
Velox-P-2 (39438)	0.007725	0.007314	DuchiFat-1 (40021)	0.012121	0.011486
First-Move (39439)	0.014598	0.01382	UFTI-1 (41458)	0.008886	0.008875
KRYSAOR (39441)	0.007254	0.006865	E-Star-1 (41459)	0.008163	0.008145
FunCube-1 (39444)	0.012253	0.011603	AAUSat-4 (41460)	0.01212	0.012094
HinCube (39445)	0.011508	0.010897	C-BR1 (40024)	0.0108	0.010218
Uwe-3 (39446)	0.010409	0.009856	DTUSat-2 (40030)	0.008991	0.008517
CANX-4 (40055)	0.006289	0.005978	PolyItan-1 (40042)	0.010867	0.010298

Table 7.22: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for in-orbit group 3.

Iteration	1	2	3	4	5	6
Average Δ between $\beta_{C_D Langmuir}$ and β_E	3.07%	-0.03%	0.79%	-1.62%	-1.55%	-1.47%
Average Δ between $\beta_{C_D \alpha=0.8}$ and β_E	-2.19%	-5.09%	-4.14%	-4.93%	-7.78%	-6.69%

The data from Table 7.21 is used to display the difference it has with the β_E values, as well as the average difference in percentage for each iteration group, in Figure 7.12. In section 8.1 the expected $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ are compared to the actual output data, where a discussion is held whether the results follow the trend which was to be expected from the theory.

Table 7.23: Mean and median for in-orbit group 3.

	Mean $\beta_{C_D Langmuir}$	Median $\beta_{C_D Langmuir}$	Mean $\beta_{C_D \alpha=0.8}$	Median $\beta_{C_D \alpha=0.8}$
In-orbit group 3	0.0167	0.0122	0.0159	0.0115

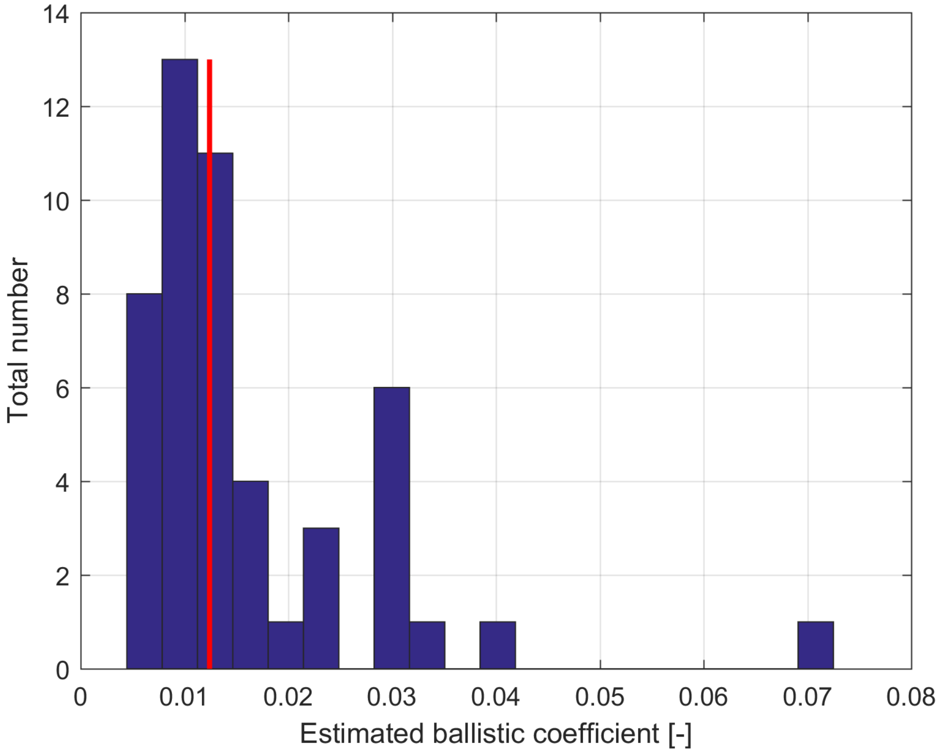


Figure 7.11: Histogram of β_E values for in-orbit group 3

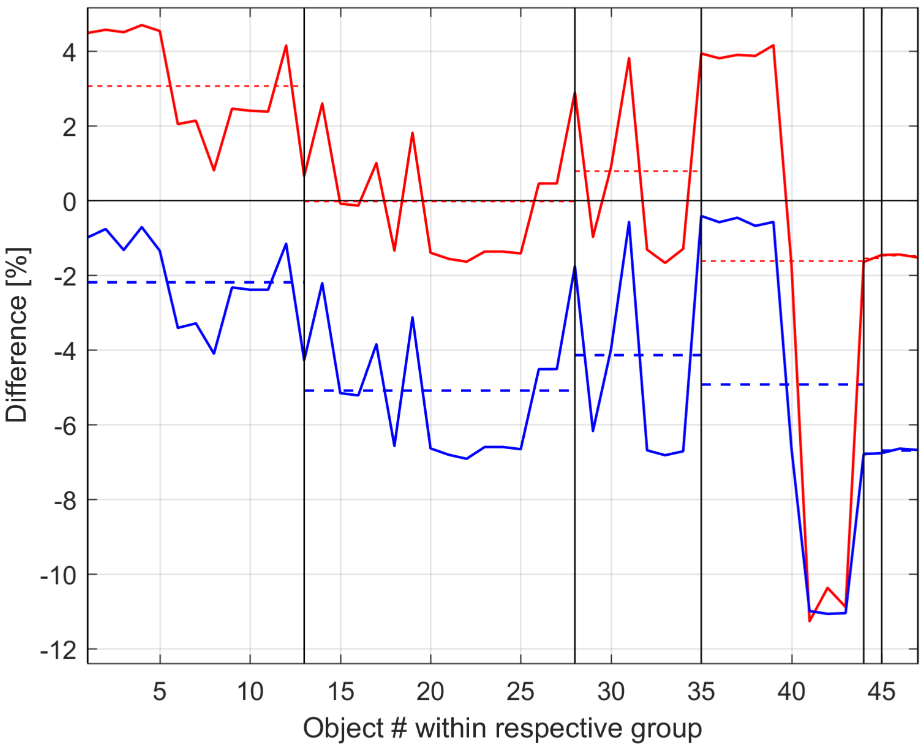


Figure 7.12: Difference of $\beta_{C_D,Langmuir}$ (red) and $\beta_{C_D,\alpha=0.8}$ (blue) values with β_E for in-orbit group 3

7.1.9. Propagation results in-orbit group 3

Having estimated and computed all required β_E and β_{C_D} values, the objects are subjected to the propagation software and assessed based on their difference in semi-major-axis of their last 10 orbits. The data is stored in section A.3, where column 6, 7, and 8 represent the difference in kilometer between the baseline propagation and scenario 2, 3, and 4 respectively.

A histogram and probability density function for the propagation results of scenario 2, 3, and 4 has been created and can be seen in Figure 7.13, Figure 7.14, and Figure 7.15 respectively. For the propagation result data presented henceforth **outlying data points exceeding 3σ have been removed**. Note that a positive x-axis represent an improvement compared to the baseline propagation. Finally, the mean and variance of each scenario is presented in Table 7.24.

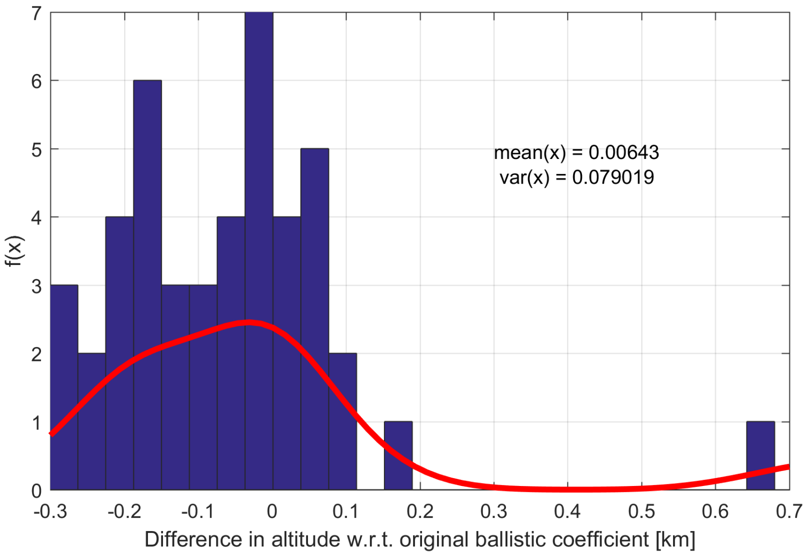


Figure 7.13: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 2 - in-orbit 3.

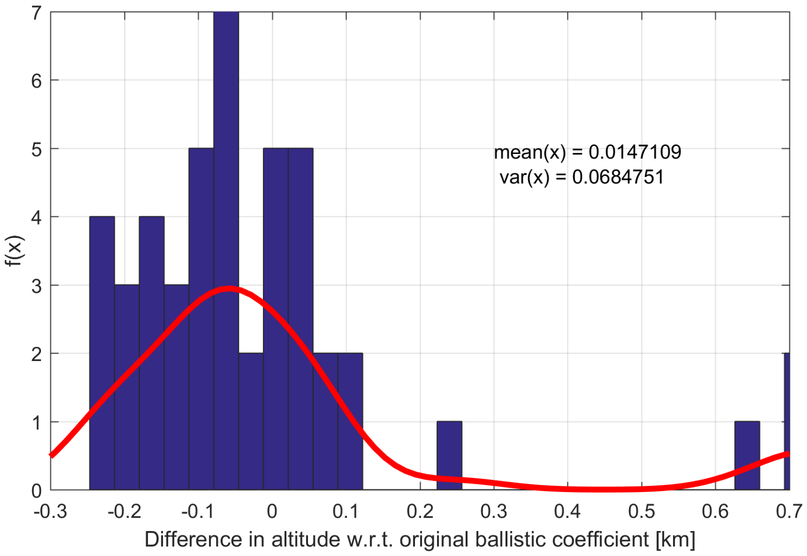


Figure 7.14: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 3 - in-orbit 3.

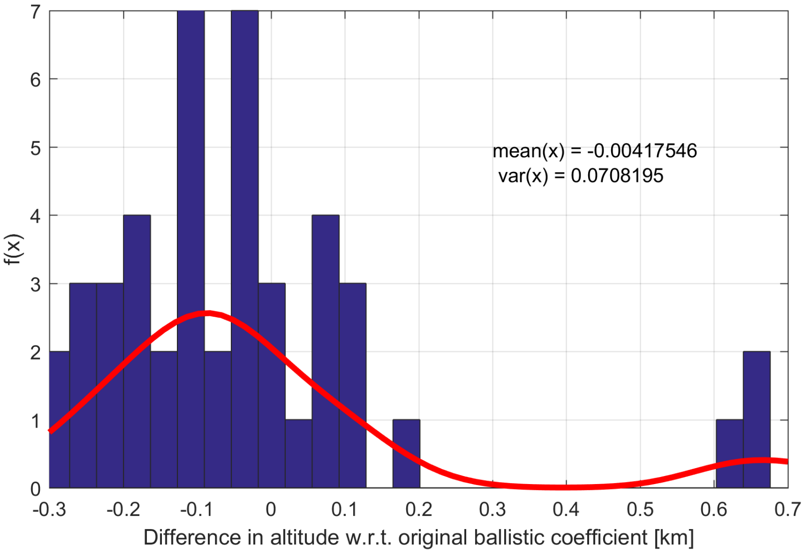


Figure 7.15: Histogram and probability density function (pdf) of the difference in propagated altitude between scenario 1 and scenario 4 - in-orbit 3.

Table 7.24: Mean and variance for each scenario for in-orbit group 3

	Scenario 2	Scenario 3	Scenario 4
Mean	0.00643	0.0147	-0.0042
Variance	0.0790	0.0684	0.0708

Additionally, all altitude gains (or losses) compared to the baseline propagation is summarized in Table 7.25. In this table the overall altitude improvements is presented for each scenario, as well as the subsequent improvements per iteration. The complete list of propagation results for all four scenarios of in-orbit group 3 can be found back in section A.3.

Table 7.25: Mean propagation difference and overall improvement [km] w.r.t. baseline orbit for each scenario in in-orbit group 3. Outlying data points are included, yet are removed in the discussion in chapter 8.

	Baseline	Scenario 2	Scenario 3	Scenario 4
Iteration 1 - Mean Δ with baseline	-	-0.211	-0.125	-0.315
Iteration 2 - Mean Δ with baseline	-	-1.569	-1.340	-1.423
Iteration 3 - Mean Δ with baseline	-	-0.541	-0.358	-0.473
Iteration 4 - Mean Δ with baseline	-	2.707	2.700	2.172
Iteration 5 - Mean Δ with baseline	-	1.112	1.098	0.855
Iteration 6 - Mean Δ with baseline	-	-0.009	-0.105	-0.024
Difference with verification orbit	12.1175	10.6281	10.247	11.325
Improvement w.r.t. baseline [km]	-	1.48	1.87	0.791
Improvement w.r.t. baseline [%]	-	12.29	15.43	6.53
Results with data points exceeding 3σ removed				
Difference with verification orbit	10.7206	10.3996	9.9865	10.9305
Improvement w.r.t. baseline [km]	-	0.3210	0.7341	-0.2099
Improvement w.r.t. baseline [%]	-	2.99	6.847	-1.95

7.2. Re-entry category

The results for the re-entry group are presented in this section. Similar to the in-orbit group, two different orbital parameter criteria sets have been used to establish two separate re-entry groups. The details on these criteria as well as the resulting NORAD satellites can be found in Table 7.26. As can be seen, the group sizes are significantly lower than those of the in-orbit groups, this is due to the smaller concentration of objects in this regime and the normally fast decay period. Since the objects decay relatively fast, there are a lot less matches possible between concentrations of objects. Nonetheless, the two groups shall still be used to assess the performance of β estimation and inclusion of a variable drag coefficient during the period leading towards re-entry.

Table 7.26: Orbital parameter criteria for the re-entry groups

	Re-entry group 1	Re-entry Group 2
Overall inclination range [degrees]	0 - 110	0 - 110
Overall altitude range [km]	<500	<500
Δ_{max} altitude [km]	40	75
Δ_{max} inclination [degree]	15	20
Δ_{max} RAAN [degree]	30	30
Δ_{max} epoch [days]	10	10
# of cubes in altitude and inclination range	23	23
# of spheres in altitude and inclination range	6	6
# of cubes found with proximity algorithm	17	19

As the objects are assessed in their re-entry period, there is no set evaluation period such as was done for the in-orbit groups. For the case of a subject in the re-entry groups, all objects are propagated over the last 100 days of their existence - thus before they re-enter the atmosphere. As evaluating the performance based on a difference on semi-major axis does not make sense when considering re-entry, the objects are assessed with the re-entry date as assessment merit, see section 6.3.

7.2.1. Group 1 - Estimated ballistic coefficients

Table 7.27 contains the estimated β_E values for the re-entry group 1. The difference in percentage with $\beta_{baseline}$ has been included as well to give a sense of the order of magnitude difference it had with the initially assumed ballistic coefficient.

These are the estimated β_E values that are to be used for the 100 day propagation in TUDAT to assess scenario 1 for **re-entry group 1**. Contrary to the in-orbit group, the propagation period are set to be the last 100 days before re-entry of the object.

Table 7.27: Estimated β_E values for re-entry group 1, compared to $\beta_{baseline}$.

NORAD	β_E	$\Delta\beta$	Iter. #	NORAD	β_E	$\Delta\beta$	Iteration #
CP-6 (35003)	0.030978	0.87	1	ArduSat-2 (39571)	0.045133	-44.43	1
HawkSat-1 (35004)	0.034675	-10.96	1	OpuSat (39575)	0.019861	36.44	1
FitSat-1 (38853)	0.024785	20.68	1	ArduSat-1 (39412)	0.033868	-8.38	2
TechEdSat (38854)	0.032023	-2.48	1	ArduSat-X (39414)	0.041739	-33.57	2
We-Wish (38856)	0.043613	-39.56	1	SkyCube (39567)	0.037333	-19.47	2

The average the average difference between β_E and $\beta_{baseline}$ for iteration 1 is +5.63% and for iteration 2 +20.47%. Moreover, mean β_E for re-entry group 1 equals 0.0344 and the median 0.0343 - close to the β_0 of 0.03125. A histogram of the objects is depicted in Figure 7.16.

7.2.2. Group 1 - Adjusted to accommodate for varying drag coefficient

Presented are the adjusted ballistic coefficients $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ for re-entry group 1. The differences that $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ have with the β_E values are presented in Figure 7.17, where the red line corresponds to $\beta_{C_D Langmuir}$ and the blue line with $\beta_{C_D,\alpha=0.8}$.

Table 7.28: Estimated β_E values for re-entry group 1, compared to $\beta_{baseline}$.

NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$	NORAD	$\beta_{C_D Langmuir}$	$\beta_{C_D \alpha=0.8}$
CP-6 (35003)	0.034282	0.0315809	ArduSat-2 (39571)	0.0399127	0.045899
HawkSat-1 (35004)	0.0386658	0.0353342	OpuSat (39575)	0.0181943	0.0203921
FitSat-1 (38853)	0.022694	0.0254269	ArduSat-1 (39412)	0.0330025	0.0350643
TechEdSat (38854)	0.029411	0.0328339	ArduSat-X (39414)	0.0405117	0.0432511
We-Wish (38856)	0.0431707	0.0446761	SkyCube (39567)	0.0348971	0.0384859

Table 7.29: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for re-entry group 1.

Iteration	1	2
Average Δ between $\beta_{C_D Langmuir}$ and β_E	-2.25%	-4.00%
Average Δ between $\beta_{C_D \alpha=0.8}$ and β_E	2.41%	3.41%

Referring back to Figure 4.2, the deviation $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ have with β_E is to be expected. In the lower regions of the atmosphere in which re-entry group 2 resides one can clearly see that the expected C_D using langmuir's adsorption model lies around 2.5, whilst for $\alpha = 0.8$ this value hovers slightly above 2.8. Remember that β_E was considered to have a C_D of 2.8. The resulting lower average value for $\beta_{C_D Langmuir}$ and the slightly higher average value for $\beta_{C_D,\alpha=0.8}$ are thus as expected.

Table 7.30: Mean and median for re-entry group 1.

	Mean $\beta_{C_D Langmuir}$	Median $\beta_{C_D Langmuir}$	Mean $\beta_{C_D \alpha=0.8}$	Median $\beta_{C_D \alpha=0.8}$
Re-entry group 1	0.0334	0.0346	0.0353	0.0352

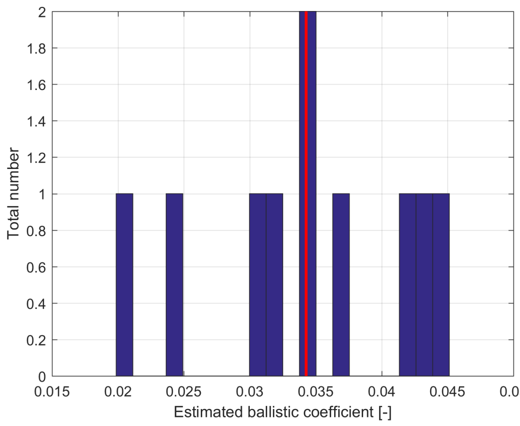


Figure 7.16: Histogram of β_E values for re-entry group 1

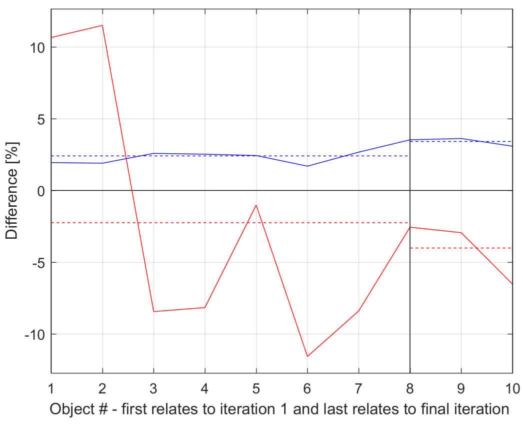


Figure 7.17: Difference of $\beta_{C_D,Langmuir}$ (red) and $\beta_{C_D,\alpha=0.8}$ (blue) values with β_E for re-entry group 1

7.2.3. Propagation results re-entry group 1

Due to the small amount of objects in re-entry group 1 and 2, there shall be no histogram presented of the difference in decay epoch relative to the baseline orbits. Instead, the complete propagation data for all four scenarios is presented in table format as opposed to the in-orbit group where that data has been stored in the Appendices. Also due to the small object size no data points exceeded 3σ and thus no points have been eliminated. The propagation results and the respective gains in days for re-entry group 1 is presented in Table 7.31.

Iteration 1	**Difference [days] with verification orbit**				**Difference [days] with initial orbit**		
NORAD	Baseline	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
CP-6 (35003)	5.918	6.545	10.177	4.575	-0.6272	-4.259	1.343
HawkSat-1 (35004)	17.439	10.428	13.798	8.759	7.010	3.641	8.679
FitSat-1 (38853)	21.336	5.815	15.576	1.927	15.521	5.760	19.409
TechEdSat (38854)	3.047	0.958	6.372	0.912	2.090	-3.325	2.135
We-Wish (38856)	34.214	0.202	5.035	2.524	34.012	29.180	31.690
ArduSat-2 (39571)	50.853	8.003	0.269	12.406	42.850	50.584	38.447
OpuSat (39575)	39.229	8.088	18.488	4.391	31.140	20.740	34.838
ABS. SUM	172.036	40.039	69.175	35.495	131.997	102.321	136.542

Iteration 2	**Difference [days] with verification orbit**				**Difference [days] with initial orbit**		
NORAD	Baseline	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
ArduSat-1 (39412)	1.927	9.604	3.129	12.490	-7.677	-1.202	-10.563
ArduSat-X (39414)	2.121	26.238	21.146	28.672	-24.116	-19.025	-26.551
SkyCube (39567)	30.637	39.401	35.912	40.635	-8.765	-5.275	-9.999
ABS. SUM	34.685	75.243	60.187	81.798	-40.558	-25.502	-47.113

Table 7.32: Mean propagation difference and overall improvement [km] w.r.t. baseline orbit for each scenario in re-entry group 1. Outlying data points are included, yet are removed in the discussion in chapter 8.

	Baseline	Scenario 2	Scenario 3	Scenario 4
Mean	-	9.1441	7.682	8.943
Difference with verification orbit	206.721	115.282	129.902	117.292
Improvement w.r.t. baseline [days]	-	91.439	76.822	89.43
Improvement w.r.t. baseline [%]	-	44.23	37.16	43.26

Clearly, the overall performance w.r.t. the baseline is shown to have increased significantly - close to an 45% increase for scenario 2. However, when looking more closely in the iterations themselves, it can be seen that this increase can only be attributed to first iteration. Interestingly, when only considering the first iteration, the best scenario would be scenario 4, where the energy accommodation coefficient is kept constant at $\alpha = 0.8$. chapter 8 shall go more in-depth on these results, though it can already be clearly stated that for just a single iteration the performance increases drastically for scenario 2 and 4.

7.2.4. Group 2 - Estimated ballistic coefficients

Table 7.2 contains the estimated β_E values for the in-orbit group 1. The difference in percentage with $\beta_{baseline}$ has been included as well to give a sense of the order of magnitude difference it had with the initially assumed ballistic coefficient. Though not used in any calculations henceforth, Table 6.4 contains the average difference for each iteration upon which in chapter 8 conclusions are being drawn from.

These are the estimated β_E values that are to be used for the 100 day propagation in TUDAT to assess scenario 1 for **re-entry group 1**. Contrary to the in-orbit group, the propagation period are set to be the last 100 days before re-entry of the object.

Table 7.33: Estimated β_E values for re-entry group 2, compared to $\beta_{baseline}$.

NORAD	β_E	$\Delta\beta$	Iter. #	NORAD	β_E	$\Delta\beta$	Iteration #
CP-6 (35003)	0.030971	0.89	1	We-Wish (38856)	0.04368	-39.78	1
HawkSat-1 (35004)	0.034802	-11.37	1	SkyCube (39567)	0.017204	44.95	1
Robusta (38084)	0.032421	-3.75	1	ArduSat-2 (39571)	0.045804	-46.57	1
UniCubeSat (38085)	0.032527	-4.09	1	OpuSat (39575)	0.019803	36.63	1
FitSat-1 (38853)	0.024912	20.28	1	ArduSat-1 (39412)	0.029697	4.97	2
TechEdSat (38854)	0.031971	-2.31	1	ArduSat-X (39414)	0.029932	4.22	2

The average the average difference between β_E and $\beta_{baseline}$ for iteration 1 is -0.63% and for iteration 2 +4.58%. Moreover, mean β_E for re-entry group 1 equals 0.0312 and the median 0.0315 - close to the β_0 of 0.03125. A histogram of the objects is depicted in Figure 7.18. Compared to the re-entry group 1, lower β_E values are observed for group 2 - which started with more relaxed orbital element selection criteria. The influence this has on the propagations is presented in subsection 7.2.6 and further discussed in chapter 8.

7.2.5. Group 2 - Adjusted to accommodate for varying drag coefficient

The ballistic coefficients adjusted for the varying drag coefficient (including a distinction between Sesam's theory and $\alpha = 0.8$) for re-entry group 2 is presented on this page, as well as their differences per iteration w.r.t. β_E, which is also visualized in Figure 7.19 where the red line represents $\beta_{C_{D}Langmuir}$ and blue $\beta_{C_D,\alpha=0.8}$.

Table 7.34: Estimated β_E values for re-entry group 2, compared to $\beta_{baseline}$.

NORAD	$\beta_{C_{D}Langmuir}$	$\beta_{C_{D}\alpha=0.8}$	NORAD	$\beta_{C_{D}Langmuir}$	$\beta_{C_{D}\alpha=0.8}$
CP-6 (35003)	0.032217	0.031439	We-Wish (38856)	0.038082	0.044404
HawkSat-1 (35004)	0.036217	0.035329	SkyCube (39567)	0.015574	0.017623
Robusta (38084)	0.028887	0.032987	ArduSat-2 (39571)	0.039696	0.046308
UniCubeSat (38085)	0.02841	0.032934	OpuSat (39575)	0.01798	0.020358
FitSat-1 (38853)	0.023998	0.025545	ArduSat-1 (39412)	0.028655	0.030665
TechEdSat (38854)	0.027885	0.032487	ArduSat-X (39414)	0.029119	0.030894

Table 7.35: Average difference in percentage between estimated ballistic coefficient and the baseline ballistic coefficient for re-entry group 2.

Iteration	1	2
Average Δ between $\beta_{C_{D}Langmuir}$ and β_E	-7.65 %	-3.11%
Average Δ between $\beta_{C_{D}\alpha=0.8}$ and β_E	0.24%	3.24%

Referring back to Figure 4.2, the deviation $\beta_{C_{D}Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ have with β_E is to be expected. In the lower regions of the atmosphere in which re-entry group 2 resides one can clearly see that the expected C_D using Langmuir's adsorption model lies around 2.5, whilst for $\alpha = 0.8$ this value hovers slightly above 2.8. Remember that β_E was considered to have a C_D of 2.8. The resulting lower average value for $\beta_{C_{D}Langmuir}$ and the slightly higher average value for $\beta_{C_D,\alpha=0.8}$ are thus as expected.

Table 7.36: Mean and median for re-entry group 2.

	Mean $\beta_{C_{D}Langmuir}$	Median $\beta_{C_{D}Langmuir}$	Mean $\beta_{C_{D}\alpha=0.8}$	Median $\beta_{C_{D}\alpha=0.8}$
Re-entry group 2	0.0289	0.0288	0.0313	0.0320

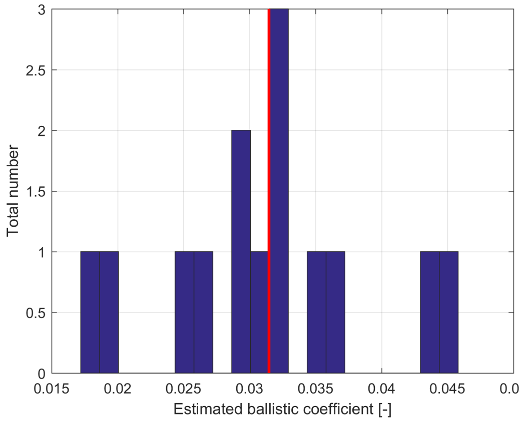


Figure 7.18: Histogram of β_E values for re-entry group 2

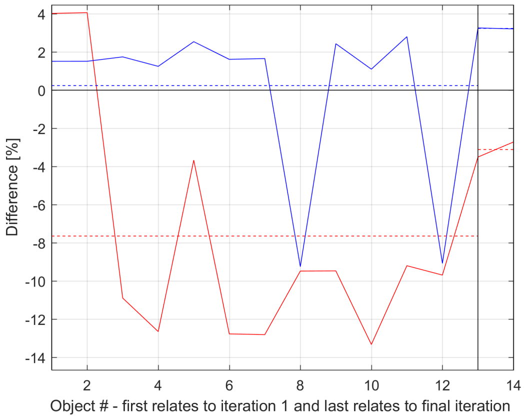


Figure 7.19: Difference of $\beta_{C_D,Langmuir}$ (red) and $\beta_{C_D,\alpha=0.8}$ (blue) values with β_E for re-entry group 2

7.2.6. Propagation results re-entry group 2

The propagation results for re-entry group 2 are presented in Table 7.37. As opposed to the in-orbit groups, there shall be no histograms of the relative difference in decay epoch per scenario as the number of objects within the re-entry group allow for a better overview when the data is displayed in said table. Also due to the small object size no data points exceeded 3σ and thus no points have been eliminated. Similarly, Table 7.38 provides a summary of the performance of the three scenarios.

Iteration 1	**Difference [days] with verification orbit**				**Difference [days] with initial orbit**		
NORAD	Baseline	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
CP-6 (35003)	5.918	6.603	10.400	4.830	-0.685	-4.482	1.088
HawkSat-1 (35004)	17.439	10.123	13.584	8.558	7.316	3.854	8.880
Robusta (38084)	15.189	18.796	7.929	21.424	-3.607	7.260	-6.235
UniCubeSat (38085)	12.013	16.292	4.912	19.016	-4.279	7.100	-7.003
FitSat-1 (38853)	21.336	5.087	14.798	1.470	16.249	6.538	19.867
TechEdSat (38854)	3.047	1.595	7.191	0.806	1.452	-4.144	2.241
We-Wish (38856)	34.214	0.004	4.925	2.764	34.210	29.289	31.450
SkyCube (39567)	30.637	40.863	34.528	38.630	-10.226	-3.892	-7.994
ArduSat-2 (39571)	9.466	10.135	1.848	14.165	-0.668	7.618	-4.699
OpuSat (39575)	2.947	2.750	27.527	18.243	0.197	-24.580	-15.295
ABS. SUM	152.206	112.248	127.644	129.906	39.959	24.562	22.301

Iteration 2	**Difference [days] with verification orbit**				**Difference [days] with initial orbit**		
NORAD	Baseline	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$	β_E	$\beta_{C_D Langmuir}$	$\beta_{C_D,\alpha=0.8}$
ArduSat-1 (39412)	1.927	2.750	5.056	2.726	-0.823	-3.129	-0.799
ArduSat-X (39414)	3.564	4.223	8.101	3.363	-0.658	-4.536	0.201
ABS. SUM	5.491	6.973	13.157	6.089	-1.482	-7.665	-0.598

Table 7.38: Mean propagation difference and overall improvement [km] w.r.t. baseline orbit for each scenario in re-entry group 2. Outlying data points are included, yet are removed in the discussion in chapter 8.

	Baseline	Scenario 2	Scenario 3	Scenario 4
Mean	-	3.2062	1.408	1.809
Difference with verification orbit	157.698	119.221	140.801	135.995
Improvement w.r.t. baseline [days]	-	38.477	16.897	21.703
Improvement w.r.t. baseline [%]	-	24.399	10.715	13.763

As can be seen, re-entry group 2 has a positive overall performance for scenario 2, though this improvement comes solely from the first iteration. Furthermore it is shown that for scenario 4 performance better than scenario 3 in again iteration 1, though that positive performance is greatly reduced when taking into account iteration 2. A potential error source could be the error heritage of the parents in iteration 1 flowing through to the children in iteration 2. A discussion on this is held in chapter 8.

7.3. Summary propagation results

The individual performances per group have been discussed and elaborated on on a per-iteration level. The big question is how the three scenarios hold up on an overall performance, and whether anything can be derived from the group's composition. Since however the in-orbit and re-entry groups are still vastly different, these two shall be presented separately in Table 7.39 and Table 7.40 respectively.

Table 7.39: Propagation result summary of the in-orbit groups

		In-orbit group 1		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	9.8823	9.2878	8.4896	9.1922
Improvement w.r.t. baseline [km]	-	0.5945	1.3927	0.6901
Improvement w.r.t. baseline [%]	-	6.01	14.09	6.98
		In-orbit group 2		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	11.4715	15.1997	13.8936	14.1136
Improvement w.r.t. baseline [km]	-	-3.7282	-2.4221	-2.6421
Improvement w.r.t. baseline [%]	-	-32.49	-21.15	-23.03
		In-orbit group 3		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	10.7206	10.3996	9.9865	10.9305
Improvement w.r.t. baseline [km]	-	0.3210	0.7341	-0.2099
Improvement w.r.t. baseline [%]	-	2.99	6.847	-1.95
		Overall		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	32.0744	34.8871	32.3697	34.2363
Improvement w.r.t. baseline [km]	-	-2.8127	-0.2953	-2.1619
Improvement w.r.t. baseline [%]	-	-8.06	-0.92	-6.74

The results presented here are elaborated on further in the discussion section, chapter 8, though most apparent findings on the influence of Langmuir's adsorption model in the adjustment for the varying drag coefficient are still held in this section. An important note, within these results **outlying data points have been removed** for points exceeding 3σ.

Table 7.40: Propagation result summary of the re-entry groups

		Re-entry group 1		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	206.721	115.282	129.902	117.292
Improvement w.r.t. baseline [days]	-	91.439	76.822	89.43
Improvement w.r.t. baseline [%]	-	44.23	37.16	43.26
		Re-entry group 2		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	157.698	119.221	140.801	135.995
Improvement w.r.t. baseline [days]	-	38.477	16.897	21.703
Improvement w.r.t. baseline [%]	-	24.399	10.715	13.763
		Overall		
	Baseline	Scenario 2	Scenario 3	Scenario 4
Difference with verification orbit	364.42	234.50	270.70	253.29
Improvement w.r.t. baseline [days]	-	129.92	93.72	111.13
Improvement w.r.t. baseline [%]	-	35.65	25.72	30.50

Starting with the in-orbit category, it is apparent group 1 and 3 perform adequate, with group 1 demonstrating an improvement of 14.09% for scenario 3. The trend that can be observed for all groups, irrespective of the level of improvement, is that scenario 3, thus the β_E adjusted for the variable drag coefficient using Langmuir's adsorption model does result in a positive improvement compared to scenario 2 - the estimated ballistic coefficients. Scenario 4, using a constant energy accommodation coefficient of 0.8, does not demonstrate a clear improvement compared to scenario 2. Therefore, the first conclusion to be drawn from the in-orbit category is that the adjustment for the variable drag coefficient using Langmuir's adsorption model is a useful method to integrate.

The poor results from group 2 are however still bothersome, as solely this group drags the overall improvement of the in-orbit category to a negative value. In chapter 8 a further investigation on this is performed, though already consider that group 2 had the most relaxed initial orbital element selection criteria for the proximity algorithm. These loose criteria could have resulted in matches being made in regions too far away from each other - resulting in wrongly estimated β_E values that could have propagated further in the later iterations. This so called error heritage is further discussed in chapter 8.

Interestingly, the re-entry category demonstrates amazing result compared to the in-orbit category - with a 44% performance increase in group 1. However, the trend observed in the in-orbit category regarding the adjustment for varying drag coefficient can not be observed in the re-entry group - neither for a constant α nor for Langmuir's adsorption model. The initially estimated values in scenario 2 provide the best β estimate, where adjusting these values for the changing C_D in scenario 3 and 4 worsened the results - though here a constant $\alpha = 0.8$ does seem to be a better adjustment than Langmuir's adsorption model , which at these altitudes quickly converges to $\alpha = 1.0$.

Again, the re-entry group with the looser selection criteria performs worst - in this case group 2. Further discussed in chapter 8, it already seems to become apparent that these selection criteria have a large effect on the estimated ballistic coefficient values, which is of course a logical follow-up from the theory. Loose criteria allows for matches being made under atmospheric conditions that start to deviate from the near-similar conditions advocated in the theory. However, even with loose criteria the re-entry category demonstrated the best improvements, and with the data from this experiment it can be advised that this method performs best for lower orbits - and can therefore be a proper addition to methods focused on re-entry predictions.

Due to the apparent sensitivity of the these selection criteria and the expected error heritage resulting from inaccurate estimates propagating their errors to the next iterations, a sensitivity analysis is held later in chapter 8. Furthermore, a sanity check is performed on the ballistic coefficient values for all groups and scenarios to determine if indeed the found β affect the orbital decay as they are expected to do. The discussions held there are concluded with a deeper analysis on the above propagation performance results, after which recommendations for future research are presented. These recommendations range from advised modification for this specific research to advise on the approach when one considers to expand this research and start incorporating (part of) this research into a global ballistic coefficient estimation model.

8

Conclusion and Recommendations

With all the results presented in the previous chapter and with extended data provided in the appendices, the experimental set-up as described in chapter 6 has come to an end. In this chapter, the data shall be analyzed with as goal to deduct performance changes that can be specifically attributed to one or more theories described earlier in this report. Key in this analysis are the performance indicators for the in-orbit and re-entry groups, and data that could indicate why certain unexpected outputs occurred - though this is still an analysis based on statistical data, it is still of interest to be able to deduct why certain outliers came to be as to hopefully prevent them in the future.

The discussion shall aim towards answering the hypotheses stated in the introduction and the sub-questions related to this hypotheses. Any arguments made either in favor or against the statement are to be supported by any conclusions derived from the data analysis. This chapter shall conclude with a section on recommendations for any continuation of this research and on matters that, in hind-sight, would have been a more suited approach to this research. It is believed that the theory described in this report, the required methodology to investigate this theory, and the described experimental set-up required to perform this research, resulted in adequate and meaningful output data. Interpretation of this data and investigation into potential areas of risk, improvement, or success, can hopefully in the future lead to follow-up research being conducted on this topic.

8.1. Discussion

Leading up to any conclusion, the data is analyze and discussed in this section. A focus shall be put between first the **sensitivity analysis** and followed by the **data analysis**. They both have their respective sub-discussion as displayed below:

- **Sensitivity analysis** - The error heritage from parent to child shall be investigated here, as well as the sensitivity a group has with regards to the orbital element criteria for the proximity algorithm as well as the sensitivity of the selected ballistic coefficients of the initial spherical satellites. The sensitivity analysis is concluded with a discussion the size and composition of the initial database have on the outcome.
- **Data analysis** - Assessment of the performance merits for both the in-orbit and re-entry groups. Furthermore, a discussion on the expectancy versus outcome regarding the ballistic coefficients is held. The data analysis is concluded with an overall advice regarding the application of the three scenarios.

The thought process behind this structure is to give the reader an understanding of the sensitivity of the described processes and to understand what data can be considered as believable and what data could be regarded as outlying data. As shall become clear, there are many circumstances under which the described theory can perform exceedingly well or, in other cases, under-perform. It is therefore recommended to thoroughly comprehend the sensitivity analysis and sanity checks before drawing own conclusions on the data.

8.1.1. Sensitivity analysis - Orbital element criteria and satellite database

For each group, whether it was classified as in-orbit or as re-entry, a set of orbital element criteria had been established for the proximity algorithm. This algorithm computed any set of matching satellites experiencing near-similar atmospheric conditions from the initial two (sphere and cube) satellite databases. Already mentioned in subsection 5.2.2, the two prime selection criteria were considered to come be epoch date range and altitude range. Table 8.1 presents a quick overview of the criteria sets for all five groups.

Table 8.1: Orbital parameter criteria for all in-orbit and re-entry groups.

		In-orbit		**Re-entry**	
	Group 1	Group 2	Group 3	Group 1	Group 2
Δ_{max} altitude [km]	40	50	25	40	75
Δ_{max} inclination [degree]	10	10	15	15	20
Δ_{max} RAAN [degree]	20	20	30	30	30
Δ_{max} epoch [days]	10	10	10	10	10

The main difference that is easily spotted is the variation in altitude range - both for the in-orbit as for the re-entry groups. Remember that the initial satellite database for all groups were identical, see Table 6.1. Based on these criteria, the proximity algorithm computed the number of satellites it could find, and the total number of matches. Again, an overview of this is provided, see Table 8.2. A quick note on the difference between number of matches and total number of found satellites - some satellites have been matched multiple times within an iteration, that is to say, they had multiple parents out of which their ballistic coefficient was estimated.

Table 8.2: Summary of proximity algorithm output

		In-orbit		**Re-entry**	
	Group 1	Group 2	Group 3	Group 1	Group 2
CubeSats found	43/59	51/59	52/59	10/23	12/23
# of iterations	4	5	6	2	2
# of matches from proximity algorithm	146	192	126	17	22

A few interesting aspects can already be derived from these two overviews, namely that the orbital element selection criteria have a direct effect on the total number of found satellites from an identical initial satellite database, its composition, and is also related to the number of parents an object can have. For reference, a list of the parents satellites for each object is given in section B.1,section B.2 ,section B.3 and. Though only three variations for the in-orbit category and just two variations for the re-entry category have been established, it can already be observed that the group composition is a directly related to these selection criteria, and it seems to be mostly influenced by the **altitude range**. Furthermore, a relaxation of the Right Ascension of the Ascending Node (RAAN) resulted in a large gain in total number of satellites. Whether epoch and inclination have such a large effect can not be observed from this dataset - additional parameter optimization research purely on initial database and selection criteria would have to be conducted for that.

Without getting too much ahead with the performance analysis, the first few observations purely on the selection criteria and initial database can be seen. First of all, let's compare the overall results of the five groups with their selection criteria. Especially the in-orbit groups are beneficial for this comparison, as the deviation in the altitude ranges allow for an expected best group (group 3), an intermediate group (group 1), and an expected worst group (group 3), due to the respective relaxation of the selection criteria. Without going in detail, clearly group three is performing best overall for the in-orbit category, and group 1 for the re-entry category, and, as expected, group 1 with its altitude range of 50 km is greatly under-performing in the in-orbit category.

To go more in depth on this argumentation, the three images below contains graphical representations of the propagation difference of scenario 1 w.r.t. the baseline of the objects in the in-orbit groups and the distance in kilometers that those objects had with their match(es). To clarify, the standard deviation σ for a group has been computed and the propagation differences exceeding 3σ have been excluded from the dataset. From that dataset, a regression line has been made in the form y = ax + b, which provides a clear indication that performance deteriorates with growing distance between the matches.

Now, this does not mean that suddenly a complete group, such as group 2, can be considered defect or even neglected - there is still a lot of relevant data within that group - though it does provide the first clue as to the sensitivity of the orbital element selection criteria. Logically, with a similar satellite database there should also be a lower bound to the selection criteria. After all, the more tightened the criteria become, the fewer matches can be observed, and as shall become apparent in the next section, too much dependency on just a few parent satellites can occur, increasing the probability of error heritage within the iterations. On the contrary, an increase in objects in the initial satellite database would in turn allow for tight criteria yet still with a healthy number of matches. The discussion on the definition of a healthy number of matches shall be resumed in subsection 8.1.2.

Concluding the first part of the sensitivity analysis, the influence and dynamics of the initial satellite database size and the orbital element selection criteria have on the healthiness of a group should have become apparent. A small extension on this is mentioned in the recommendations, though basically a proper performing group is already partially defined upon defining the satellite database and selection criteria. The aim would be to increase the database size and to tighten the selection criteria, most primarily for the altitude range. Follow-up research is encouraged to examine the effect on the tightening of the other three criteria in more depth, for instance by keeping three criteria constant and alternate only one, and perform a similar analysis as was performed for the graphs above. It would be interesting to see which criteria demonstrates the steepest (negative) regression lines and is thus most influential for the performance of the overall group.

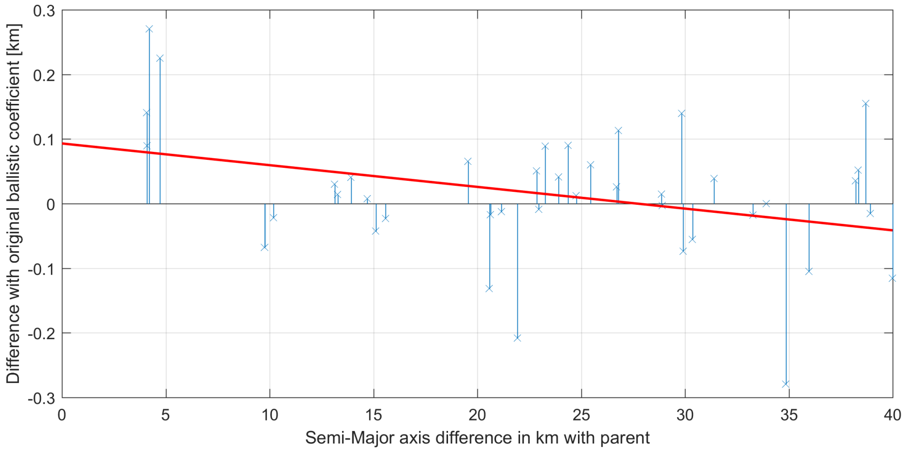


Figure 8.1: Regression line from object performance vs matched s.m.a. difference - in-orbit 1.

8.1.2. Sensitivity analysis - Error heritage

The global performance of a group when solely looking at the initial database and the orbital element selection criteria has been discussed, though within the ballistic coefficient estimation procedure there is another sensitive aspect - error heritage. Remember that an object gets its ballistic coefficient estimated relative to the ballistic coefficient of its parent(s). Therefore, if an object has a parent with a badly estimated ballistic coefficient that error propagates to the child, which in turn gives it to the next iteration and so forth. To highlight this problem, this subsection shall elaborate on this negative feature by providing an overall analysis of this error propagation for the in-orbit groups, specifically chosen as the re-entry group do not have enough iterations to perform a proper error propagation analysis.

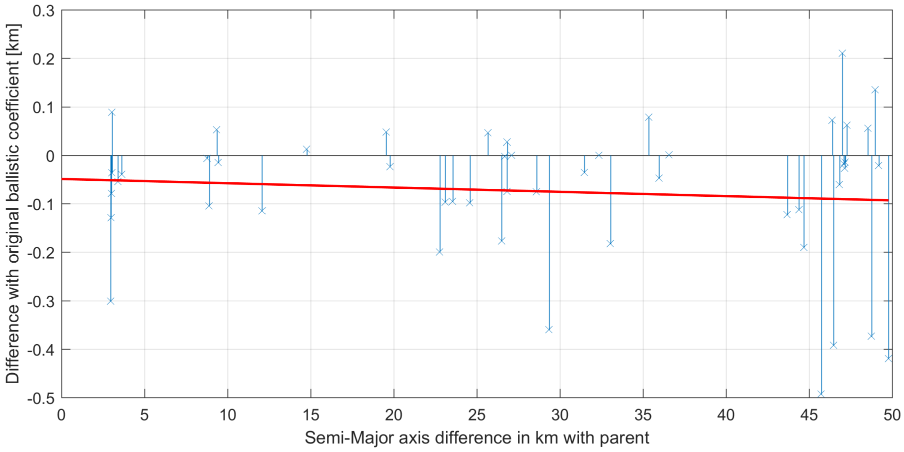


Figure 8.2: Regression line from object performance vs matched s.m.a. difference - in-orbit 2.

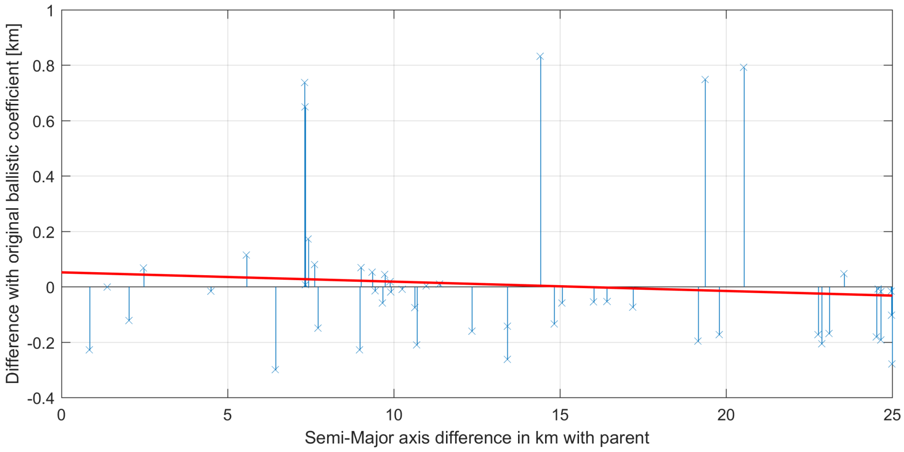


Figure 8.3: Regression line from object performance vs matched s.m.a. difference - in-orbit 3.

To perform a proper analysis of the supposed error heritage the following procedure has been performed. First, the data that is analyzed is the absolute difference β_E has with the verification orbit, corresponding to column C, or 3, in the propagation results, see Appendix A. The reason behind selecting this column and not the difference between scenario 1 and 2 is because here the interest lies in determining if a poor β_E does propagate to the next iteration, not whether it is better than β_0. Secondly, section B.1 and the following 2 appendices are established containing all objects in each group and their parents. Here it already becomes obvious that certain parent combinations are strongly influencing the later iterations.

From here, for each object (or, child) the difference with verification orbit is stored, with next to it a column with the mean difference of the parent(s) of that child. Since the interest lies in the error propagation amongst the estimated ballistic coefficients, all children from spherical satellites are removed - that is basically iteration 1 for each group. The reasoning to do so is to purely look at error heritage in β_E and not the error induced by the accuracy of β^T. Finally, outlying data points are removed by computing the standard deviation σ from the remaining children group and exclude all objects exceeding 3σ. The data for the three in-orbit groups are presented respectively in Figure 8.4, Figure 8.5, and Figure 8.6, in which also a regression line of the data is presented.

As expected, the regression line shows that with increasing difference of the verification orbit of the parents, the child feels this effect as well - or short, positive mean parent error (x-axis) correlates to positive child error (y-axis). Furthermore, the dependency of a group of just a few parents is also shown in the graphs. When

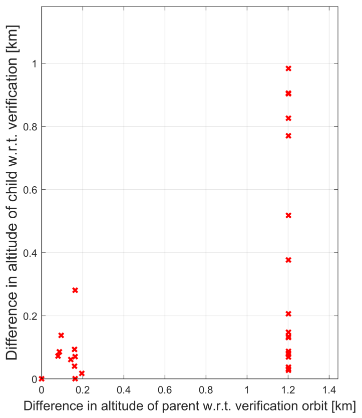


Figure 8.4: X-axis represent mean km difference of parent(s) with the verification orbit and the y-axis the difference of the child with the verification orbit - in-orbit group 1.

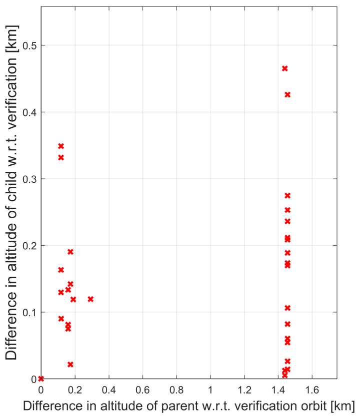


Figure 8.5: X-axis represent mean km difference of parent(s) with the verification orbit and the y-axis the difference of the child with the verification orbit - in-orbit group 2.

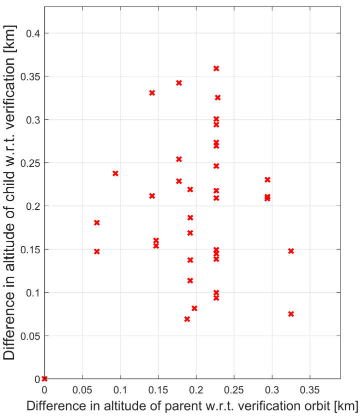


Figure 8.6: X-axis represent mean km difference of parent(s) with the verification orbit and the y-axis the difference of the child with the verification orbit - in-orbit group 3.

looking closely, especially at in-orbit group 1 and group 2, one can see two large columns at the end of the x-axis. In group 1, this corresponds to the mean absolute difference from satellite 37851 and 37852, for group 2 these objects are 37851, 37851, 37854, and 37855. The overall group can thus be considered to be sensitive to the just a few select parents.

Going a little deeper, let us zoom in on Figure 8.4, representing the data of the in-orbit group 1. The highest point, or the one effected the most by the errors of its parents, correlates to a x-axis of 1.2 and a y-axis of around 1.0. Noting that this 1.2 on the x-axis corresponds to the two 37851 and 37852 objects, the propagation results in chapter 7 can be used to determine which object can be associated with the y-axis. The only possibility would be object 41460, which indeed shows a difference of almost 1 km with the verification orbit. This object is affected the most by the error heritage, and this also comes back when looking up the corresponding performance results - a loss of 0.5 km is seen between scenario 1 and scenario 2, directly making this object the worst of its group. A similar case happens in Figure 8.5, where again the worst performing is object 41460 after having the highest error heritage.

To conclude the error heritage examination, let us take a closer look whether any error heritage comes from the spherical satellites. Again using the propagation results and the child-parent tables in the appendices, it can be derived that object 27944, or the LARETS spacecraft, is the parent of most satellites in iteration 1 for all three groups. No verification orbit have been created for the spheres, nor has their performance based on the originally assumed β from section 6.1 been analyzed. However, when looking at the propagation performance of LARETS children, something interesting occurs - most of the poor performance in iteration 1 from group 2 and group 3 can be attributed to objects descending from LARETS, whilst for group 1 there is no indication of poor performance due any LARETS error heritage. The LARETS spacecraft looks almost identical to Figure 6.5.

Though literature[46] often uses the same β^T for LARETS as is done in this research ($\beta = 0.004658$), it could be that its ballistic coefficient might have to be reevaluated. The non-perfect spherical shape of LARETS, representing the shape of a golf ball could possess a lower C_D value than the originally assumed 2.3, as is suggested M. Rutkowska's work[45]. Furthermore, in accordance to the work of B. Bowman[4], spherical drag coefficients differences between computed and observed values would sometimes be around % lower than initially computed. This reduction in the real experienced C_D could indicate initially assumed C_D values for the spheres - errors that can have propagated towards the child objects.

Error heritage has, as logically expected, shown to influence further iterations down the line. A healthy group would be a group where poor β_E are not propagated through in time, and where the entirety of the group does not depend on just a select few objects. Preferably a merit is created capable of either assessing the quality of β_E upon which a decision can be made whether or not a new match should be formed with that value, or an object must have a minimum number of parents to average out any errors induced by a single poor β_E. A quick ad-hoc solution to this problem would be to largely increase the satellite database with variety in orbits - this way multiple matches per object should be made possible, selection criteria could be tightened, and most likely the group would not be dependent on a few objects or a certain iteration group.

8.1.3. Data analysis - Sanity check

The ballistic coefficients resulting from the estimation algorithms are approximations of the true ballistic coefficient and, inherent to the approach of acquiring β_E, still septic to aspects such as bad TLE observation data and to deviations in the computed local atmospheric density error. It is therefore that the resulting ballistic coefficients within a group do not always demonstrate a uniform accuracy, where some objects have gotten their β_E computed accurately, other objects might actually have received worse β_E values as when compared to their baseline orbits. The sensitivity of an object's accurate ballistic coefficient computation has been discussed before, where it was shown that the orbital element selection criteria, satellite database, and the error propagation, are potential causes of an object's extraordinary good or poor behavior. However, these error would have been induced only once during the estimation procedure, and the resulting ballistic coefficients, whether good or bad, can still be used to analyze the performance of the drag coefficient adjustment algorithm and the expected performance of the propagator in the form of a sanity check.

The question whether the computation of $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ has been properly executed can be evaluated fairly straightforward. Figure 4.2 displays the drag coefficient of a 1U CubeSat as a function of both Langmuir's isotherm and an energy accommodation coefficient of $\alpha = 0.8$, both for solar maximum and solar minimum conditions. For both categories, the adjustment of β_E to obtain $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ has been analyzed to see if they do indeed follow the trend from section 4.2. This has been done by simply subtracting $\beta_{C_D Langmuir}$ from $\beta_{C_D,\alpha=0.8}$, thus scenario 4 - scenario 3. The differences for all objects in the in-orbit groups and re-entry groups are visualized in Figure 8.7 and Figure 8.8 respectively. Note that the total object size of the in-orbit category is thus 43 + 51 + 52 = 146 and that of the re-entry category is 10 + 12 = 22.

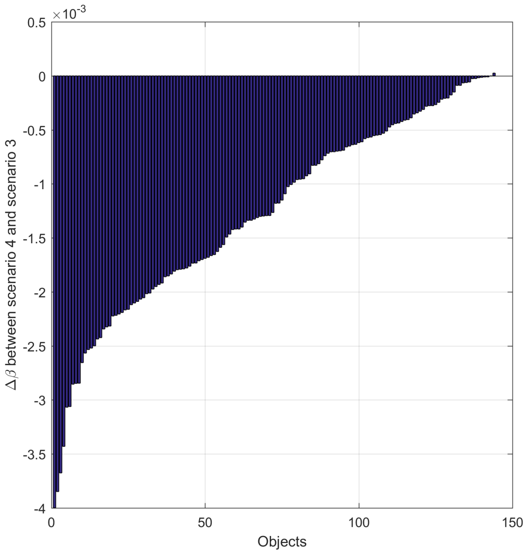


Figure 8.7: Ballistic coefficient difference between scenario 4 and scenario 3 of all re-entry objects. Positive y-axis indicates $\beta_{C_D,\alpha=0.8}$ to exceeds $\beta_{C_D Langmuir}$.

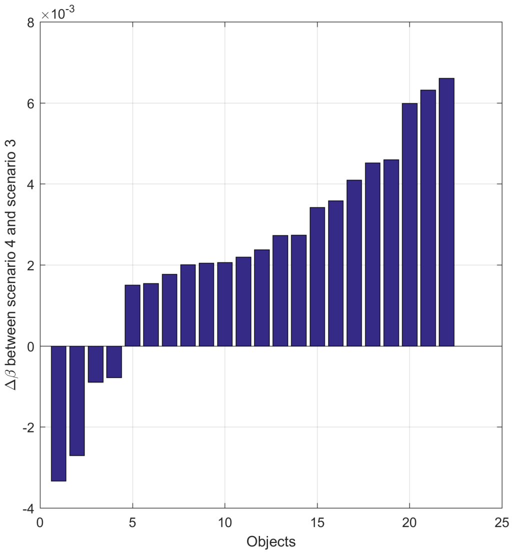


Figure 8.8: Ballistic coefficient difference between scenario 4 and scenario 3 of all re-entry objects. Positive y-axis indicates $\beta_{C_D,\alpha=0.8}$ to exceeds $\beta_{C_D Langmuir}$.

It instantly becomes apparent that the in-orbit group have (almost) all objects which have lower $\beta_{C_D,\alpha=0.8}$ values than $\beta_{C_D Langmuir}$, whilst for the re-entry group this is the opposite. Referencing back to Figure 4.2, this was to be expected - for higher altitudes and lower solar activity Langmuir's adsorption model computes a significantly higher C_D than when the energy accommodation coefficient α is assumed constant at 0.8. For the lower orbits, thus for the re-entry category, the opposite is expected and proven to be the case. This simple sanity check demonstrates that the ballistic coefficient conversions for scenario 3 and scenario 4 do indeed follow the expected trend.

Another aspect one would logically expect to see is that with a decreasing ballistic coefficient, the orbit would decay slower. Opposite holds true too - the higher the ballistic coefficient, the more it is affected by the atmosphere and thus the faster it decays. With so many ballistic coefficients having been computed, this simple reasoning should be shown back in the data by analyzing whether an object's difference with the verification orbit either increased or decreased compared to object's other computed ballistic coefficient.

To illustrate how this analysis was performed, remember that for each object four ballistic coefficients have been computed, one for each scenario. The baseline scenario 1 always had a ballistic coefficient of 0.03125 assigned to the object. The difference $\delta\beta$ between for instance β_0 and β_E corresponds to a δ altitude gain w.r.t. the verification orbit. If for instance β_0 corresponds to a difference with the verification orbit of -0.2 km and β_E corresponds to a difference of +0.3 km, it is expected that β_E is smaller than β_0.

Due to the large sample size of the in-orbit groups, this analysis has been performed for these three groups in a similar fashion as was done with the sensitivity analysis for the selection criteria. For all objects in each group $\delta\beta$ between scenario 1 and scenario 2 and the corresponding altitude difference w.r.t. the verification orbit has been computed. After that, the resulting data has had its standard deviation σ calculated and any data exceeding 3σ was regarded as outlying data and discarded. The results for the three in-orbit groups are presented in Figure 8.9, Figure 8.10, and Figure 8.11 respectively. As was expected, for all three groups it can be demonstrated that with an increasing ballistic coefficient the altitude w.r.t. the verification orbit drops.

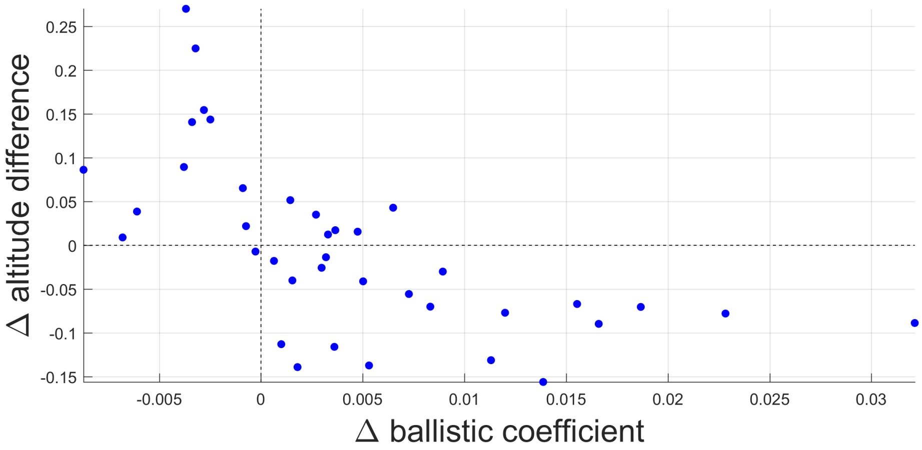


Figure 8.9: Altitude loss w.r.t. $\Delta\beta$ between scenario 1 and 2 for in-orbit 1.

8.1.4. Data analysis - Performance assessment

With the sensitivity discussion being held and having given a closer look at how the data was expected and how it actually performed in the sanity check, a few expectations w.r.t. the performance assessment can already be stated. First of all, with the sensitivity to the orbital element selection criteria sets, it can be expected that the group with tighter criteria shall performed best, as long as the group composition is healthy - that is, there are multiple parents for the objects, and not just a few ancestors through which errors can potentially propagate. Furthermore, the expected difference between $\beta_{C_D Langmuir}$ and $\beta_{C_D,\alpha=0.8}$ seems to follow the predicted trend. Therefore, any conclusions made between scenario 3 and scenario 4 can thus be directly related to the theory and, more importantly, to either Langmuir's adsorption model or the use of a constant energy accommodation coefficient.

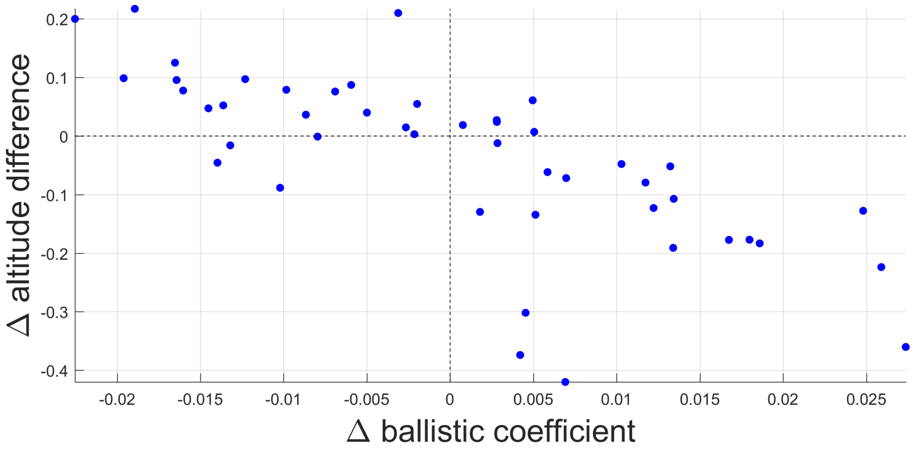


Figure 8.10: Altitude loss w.r.t. $\Delta\beta$ between scenario 1 and 2 for in-orbit 2.

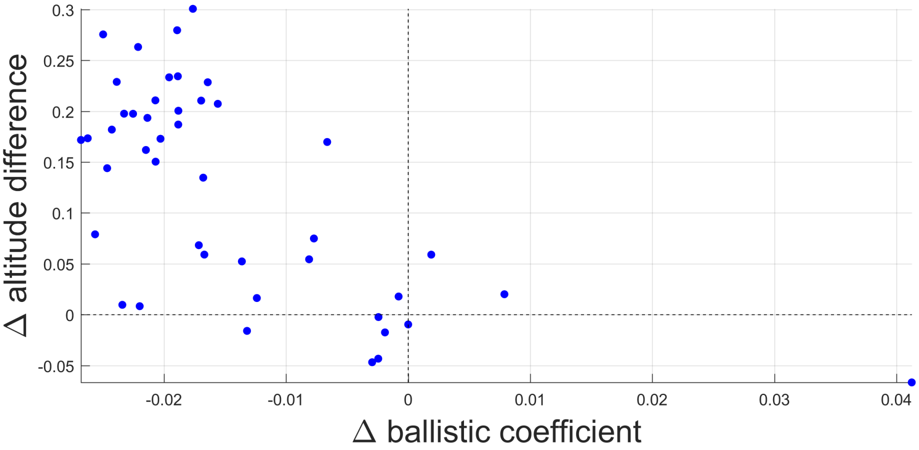


Figure 8.11: Altitude loss w.r.t. $\Delta\beta$ between scenario 1 and 2 for in-orbit 3.

Before any assessment is performed, the propagation results, and most specifically the propagation errors between the baseline and scenario 2, are checked for outlying data points in coherence with the methods used for the sensitivity analysis before. That is, over Column J of the data results, the standard deviation σ is computed and all outlying data points exceeding 3σ are discarded. The impact this has on each group and on the overall performance is presented in Table 8.3. When comparing this data with the original propagation results, no change for the re-entry group is observed, though overall the in-orbit group has a slight increase in performance.

Starting with the in-orbit groups, it can be seen that sadly the overall performance is negative, which can be solely attributed to the group with the most relaxed selection criteria, group 2. Without the negative -32.49% from group 2, the overall performance would have been positive for all scenarios - the first indication that indeed the initial selection criteria hugely affect the performance of a group. Before going into detail regarding the performance of the three in-orbit groups, it is interesting to note that when discarding the outlying data, the best performing in-orbit group has shifted from group 3 to group 1. This is not that unexpected when taking a closer look at the individual performance of the objects in group 3. Objects 37852, 37854, and 37855 both performed exceedingly well in iteration 4, which made them according to the 3σ boundary outliers and were thus discarded. Hence, group 1 turned out to be the best performing group, which immediately raises a new question - is maybe the Right Ascension of the Ascending Node another key selection criteria, together with epoch and altitude? After all, the only two variations between 1 and 3 are the RAAN and altitude range.

So let's zoom in on three groups, starting with group 1. The discarding of outliers created a new image of this group, one in line with that expected from the theory. Scenario 1 with its estimated ballistic coefficients proved to form a slight better overall performance, though the real outstanding data comes when considering scenario 3. Accounting for the variation of the drag coefficient computed with Langmuir's isotherm shows an improvement of little above 14%, remarkably close to double the accuracy gain compared to scenario 2! Consider again scenario 4, and the performance drops closer to values that of scenario 2. Thus, for in-orbit group 1, with a total of 43 test subjects, an altitude range of 40 km and a RAAN range of 20 degrees, it can be stated that compared to the baseline an undeniable improvement has been achieved, especially in scenario 3. Furthermore, Langmuir's adsorption model for these orbits seems to outperform usage of a constant energy accommodation coefficient of $\alpha = 0.8$.

Unsettlingly group 2 behaves extremely poor - not only does it reduce the overall in-orbit group performance to nothing, it demonstrates that cases can exist where the overall performance may drop well below that of a constant ballistic coefficient of 0.03125. Clearly, in the estimation procedure for group 2 there have been either misfits, ranges that exceeded the consideration of experiencing near-similar atmospheric conditions, simply error heritage, or a combination of all these. Though sadly the estimation procedure proved to have its faults, it doesn't render the data set invalid - quite the contrary. Apart from demonstrating the necessity of parameter research, it also shows that even with bad β_E, improvements are made in scenario 3, again proving the importance of accounting for the variable drag coefficient. And even though just barely, scenario 3 outperforms group 4 again, strengthening the resolution that Langmuir's isotherm does indeed prove valuable for higher orbits.

Concluding the in-orbit category is group 3 - the former best performer though with the removal of outliers brought back to a second place. With only a 3% gain in performance for the estimated ballistic coefficient, scenario 2 does not seem to be that much of an improvement. However, the trend identified in group 1 and group 2 continuous in this group as well; scenario 3 almost doubles the performance and again leaves scenario 4 way behind. With this final group, it can be confidently said that Langmuir's adsorption model is a requirement when adjusting for the variable drag coefficient. Furthermore, though having tight criteria yet not have the best performance, group 3 could be the clearest indication of the error heritage described earlier. The mean β_E is extremely low compared to group 1 and 2, and even though this does sometimes provided with good propagation results, it seems that low ballistic coefficient values were estimated in iteration 2 and propagated itself towards the other iterations. Though not a conclusive observation, additional research in this group might have exposed new relations between error heritage and performance. Finally, group 3 also demonstrated that unexpectedly the RAAN is a more vital element in the proximity algorithm than was initially assumed.

Upon examining the data from the re-entry group, the results are as were to be expected. In the lower region of the atmosphere, the predicted thermospheric density values are expected to have a higher accuracy opposed to those at the higher orbits such as in the in-orbit group. Therefore, the expectancy was that the estimated ballistic coefficients β_E demonstrates a significant increase as opposed to the baseline, which can be seen back in the data ; an overall 35.65% increase between β_E and the baseline orbits. Though huge improvements have been made in the re-entry group, let's try to understand why this group so greatly outperformed the in-orbit group.

Re-entry group 1, with the tightest selection criteria of the two, shows an initial improvement of in scenario 2 of close to 45%. Contrary to the in-orbit groups, this value drops to 37% after which it dips up again in scenario 4 to nearly the same value as in scenario 2, 42%. Interestingly, the approach with computing the drag coefficient with Langmuir's adsorption model does not work that great for lower orbits. Potentially this is due to the fact that with Langmuir's isotherm, the energy accommodation coefficient approaches 1.0 in the lower regions of the atmosphere, and that this convergence to 1.0 is not the most accurate solution. Moreover, the huge increase in performance itself could maybe also be attributed to more accurate TLE observation data being created for objects in lower orbit (and in orbits close to the ISS), as well as that the output of NRLMSISE-00 is generally more accurate for lower regions [38]. Still showcasing a notable increase in performance for all scenarios, re-entry group 2 is barely demonstrating half the increase compared to group 1. Especially sce-

Table 8.3: Propagation result summary with discarded 3σ data points.

	Absolute propagation sum			
	Baseline	Scenario 2	Scenario 3	Scenario 4
In-orbit 1 [km]	9.8823	9.2878	8.4896	9.1922
In-orbit 2 [km]	11.4715	15.1997	13.8936	14.1136
In-orbit 3 [km]	10.7206	10.3996	9.9865	10.9305
Re-entry 1 [days]	206.7210	115.2822	129.9021	117.2923
Re-entry 2 [days]	157.6978	119.2211	140.8006	135.9945
	Performance increase w.r.t. baseline orbit, absolute			
	Baseline	Scenario 2	Scenario 3	Scenario 4
In-orbit 1 [km]	-	0.5945	1.3927	0.6901
In-orbit 2 [km]	-	-3.7282	-2.4221	-2.6421
In-orbit 3 [km]	-	0.3210	0.7341	-0.2099
Re-entry 1 [days]	-	91.4388	76.8189	89.4287
Re-entry 2 [days]	-	38.4767	16.8972	21.7033
	Performance increase w.r.t. baseline orbit, percentage			
	Baseline	Scenario 2	Scenario 3	Scenario 4
In-orbit 1 [%]	-	6.01	14.09	6.98
In-orbit 2 [%]	-	-32.49	-21.15	-23.03
In-orbit 3 [%]	-	2.99	6.847	-1.95
Re-entry 1 [%]	-	44.23	37.16	43.26
Re-entry 2 [%]	-	24.39	10.71	13.76
	Overall performance per category, percentage			
	Baseline	Scenario 2	Scenario 3	Scenario 4
In-orbit [%]	-	-8.06	-0.92	-6.74
Re-entry [%]	-	35.65	25.71	30.4957

nario 3 makes a large drop to 10%, again showing that using Langmuir's adsorption model for lower orbits might not be as suited as it was for the higher in-orbit groups.

Overall, the propagation results for all groups highlight some expected outcomes such as lower performance with relaxed selection criteria. On the other hand it also highlighted unexpectedly the enormous performance increase for the re-entry groups and also notably, the influence the altitude of an object has w.r.t. whether to use Langmuir's adsorption model or not. Initially it was also assumed that the two key selection cirteria were epoch and altitude, yet the difference between group 1 and group 3 demonstrated the importance of the RAAN as well.

8.2. Conclusion

The procedures described in this thesis are a promising way to establish ballistic coefficients for large groups of objects in space, though it has been shown that the criteria on which the procedures are conducted, and the sample size of the initial database, have a significant impact on the outcome. Though having found relations that were initially not considered and getting both expected and unexpected results, it is first required to link back to the hypothesis stated at the beginning of this thesis, the hypothesis on which this entire research was initiated on:

How do estimated ballistic coefficients adjusted for atmospheric density errors and the inclusion of an ambient gas-composition dependent C_D affect orbital lifetime predictions?

To answer this question, four scenarios had been established of which scenario 1 took the role of baseline scenario and the other three were assessed based on this baseline. As discussed in the previous report, a ballistic coefficient can be estimated by analyzing historical TLE observation data and the corresponding atmospheric gas composition. The estimated ballistic coefficient would normally still incorporate any density errors associated with the used atmospheric model, in this case NRLSMISE-00. To mitigate this, Equation 4.3 showcases how an object with an accurately known β^T experiencing near-similar atmospheric conditions can be used to obtain a more accurate model-based estimated ballistic coefficient for an object.

Defining these near-similar conditions required a set of orbital element criteria and an initial database, which has proven to be one of the first and most sensitive steps towards the final propagation. With small sample size satellite databases, tight selection criteria could create groups that either have a low amount of objects in it, or have a large dependency on just a few objects through which error heritage can become an issue. Nonetheless, for both in-orbit (500 - 1000 km) and re-entering objects notable performance improvements have been observed w.r.t. the baseline orbits - in the most optimistic case even leading to a 44% increase in re-entry group 1.

Table 8.4: Propagation result summary with discarded 3σ data points.

	Performance gains per scenario			
Group	Baseline	Scenario 2	Scenario 3	Scenario 4
In-orbit 1	-	6.01	14.09	6.98
In-orbit 2	-	-32.49	-21.15	-23.03
In-orbit 3	-	2.99	6.847	-1.95
Re-entry 1	-	44.23	37.16	43.26
Re-entry 2	-	24.39	10.71	13.76

Continuing with the adjustment for the varying drag coefficient, two additional scenarios had been established to which all groups (three in-orbit and two re-entry) were subjected. The difference in these scenarios lied in the fundamental computation of C_D, more specifically whether to adopt Langmuir's adsorption model (scenario 3) or keep a constant energy accommodation coefficient of $\alpha = 0.8$ (scenario 4). Interestingly, for the in-orbit groups the estimated ballistic coefficients from proved to become more accurate when adopting Langmuir's isotherm as opposed to keep a constant α. For example, the performance of group 1 increased from 6% in scenario 2 to 14% in scenario 3. Yet for the re-entry group the opposite proved to be the case, with scenario 3 actually worsening the gained performance from scenario 2. Also scenario 4 showcased no increase in the performance of the re-entry groups compared to scenario 2.

Additionally, a set of sub-question had been established to provide a better understanding of the underlying principles of which the hypothesis is composed of. Though having been answered throughout the report already, the next page lists these four sub-questions and summarizes the answers to them. The conclusion section shall be ended with a link to the real-life impact this research could have, which is followed by a final recommendation section on the additional research that can be performed to extend the accuracy of the methods described in this thesis.

How can variations in C_D as function of gas composition be accurately modeled for calibration spheres and tumbling CubeSats?
Starting with subquestion 1, the methods discussed in section 4.2 demonstrated a performance increase for the in-orbits groups. These models have been verified to confidently be exploited in scenario 3 and 4. However, the fact that positive performance has been observed does not automatically indicate that the modeled C_D variations has been done in the most accurate manner. For instance, the model does not take into account a tumbling CubeSat, only a straightforward computation of a 1U box of 10 x 10 x 10 cm dimensions is considered. Furthermore, no exotic geometries such as deployed solar arrays or antennae are considered, nor is the material of the CubeSat considered to be a function input to the model. The idea of including an analytical model capable of computing an objects drag coefficient as a function of ambient gas composition has been proven to be a good one - to what extend the accuracy of the model can be increased has not been researched in depth in this thesis.

Is Langmuir's adsorption model suited for thermospheric energy accommodation coefficients?
Langmuir's adsorption model is interestingly both suited and unsuited as input to the drag coefficient models. Objects in orbits between 500 and 1000 kilometers, or the in-orbit category, demonstrated a significant improvement in performance when using Langmuir's isotherm. However, for the re-entry objects a worsening of the performance was observed. A potential cause could be that the objects in the re-entry group still have atomic oxygen attached to them from when they were orbiting in higher orbits, thereby reducing the energy accommodation coefficient to values well below the ones established through Langmuir's model for the altitudes of propagation.

In what regions do this method perform best, and how sensitive are the results to the initial sample size?
The greatest improvements were definitely in the re-entry regime, where on average a 35% improvement compared to the baseline has been achieved. Two out of the three in-orbit groups saw a positive performance increase as well, though not as significant as for the re-entry groups. The in-orbit groups did however highlight the sensitivity of the initial sample size and the orbital element selection criteria. Suggested is to read the sensitivity analysis for the details, but the bottom line is that a larger initial database allows for more tightened selection criteria, which has demonstrated to have a positive effect on the quality of estimated ballistic coefficients.

Can this method be extended for satellites with more complex geometries?
Expected is that the methods described in this thesis should indeed be possible to extend to more satellites as well as satellites with more complex geometries. In fact, it is recommended to start increasing the initial database, as for the estimation of the ballistic coefficients no C_D adjustment algorithm is required yet. As long as the objects with known ballistic coefficients β^T, in this research the spherical satellites, are not reduced increasing the total objects of interest is expected to only be beneficial. Remember, a larger sample size allows for tighter selection criteria and results in, overall, a better performance. Regarding the complex shape, if one is intending to also incorporate the adjustment for the varying C_D, which is recommended to do in general anyway according to the results from this research, it is key to also update the drag coefficient computation model for these shapes. Elongated objects tend to have a much larger drag coefficient than simple cubes, and extruding parts such as antennae or solar arrays only increase this drag coefficient even more. Though it would definitely be possible, it does add to the complexity.

8.2.1. Potential impact on real-life applications

The performance accuracy potential for orbital lifetime predictions using estimated ballistic coefficients an adjusting for the varying drag coefficient has shown to be quite significant, even though there are still lots of kinks the in cable that needs additional research - more on this in the recommendations. Irregardless of the required additional research, the benefits for real-life applications could be quite substantial for both space institutions and spacecraft operators. When working in its fullest potential this method could be used to estimate the ballistic coefficients of thousands of objects orbiting low-Earth orbit - satellites, space debris, as long as there is TLE observation data present it can be subjected to this method. Space debris prediction models from for instance the Space Situational Awareness (SSA) department of ESA or the SSA department of the EU located in Madrid could instantly update their models by assigning more accurate ballistic coefficients to their debris database - most likely resulting in a better understanding of the actual rate at which debris is decaying towards Earth. Scenario 2 described in this thesis would be most suited for this, as scenario 3 and 4 would need complex C_D algorithms in this case. Moreover, attitude-controlled satellites with an accurate and well-known ballistic coefficient could be added to the β^T database, which most likely results in additional matches being found. Though being computationally extremely intensive, it would provide institutions around the world with a massive and continuously updated ballistic coefficient database.

Extending the application discussed above, spacecraft operators could wish to add custom C_D algorithms and upgrade the accuracy of the ballistic coefficients found for their satellite. A more individual approach, it would allow for operators around the world to get better estimates of their predicted time left in-orbit. Potentially it could indicate that a mission is actually estimated to be in-orbit for 5 years instead of the initially assumed 4 years - in that case the operator could already start planning additional finances for that extra year, as well as plan extra science missions or commercial operations for their spacecraft. The other way around holds true too, if an estimation indicates only 3 years instead of the assumed 4, the operator may want to re-prioritize the planned science missions, or re-evaluate the costing they want to associate with their commercial services.

A highly ideal image is being described here, yet the results from this thesis does allow for such ideas to be created. In the recommendation section a set of research areas identified to be needed for this image to become reality has been listed. There would still be a long way to go before a global ballistic coefficient estimation model can become reality, though investing in working towards this goal could reap great benefits for both space institutions and spacecraft operators.

8.3. Recommendations

Throughout the 9-month thesis work a continuous stream of new information appeared, most of which were the results of unexpected intermediate results that could be quickly added into the experiment. Though some questions remained unanswered, and upon analyzing the results a couple of ideas arose that could prove to be beneficial for the accuracy of this experiment. This final section shall discuss recommendations that could open pathways to a better understanding of the science discussed in this thesis. Though the results in this research are definitely useful, it is not a conclusive research - in fact, it might even have opened doors to more follow-up research on this topic. If given the chance and resources, the author would like to recommended the following:

- **In-depth orbital element sensitivity analysis -** the performance has been shown to be highly dependent on the proximity algorithm and the accompanying orbital element selection criteria. The four criteria used in this research have not seen a parameterization optimization, an aspect that is highly recommended though. When this research would be conducted, advised is to also include the effect of the initial satellite database size.

- **Accuracy of β^T for the (non-)spherical satellites -** the β^T in this research have been defined with a simple equation based on open-source information. An investigation in the actual accuracy of these initial values is recommended, as well as a research into the addition of attitude-controlled objects into that database, provided their ballistic coefficient is well-known.

- **Energy accommodation coefficient for lower orbits -** higher orbits proved to follow Langmuir's model, yet for re-entry this model was unsuited. It is suspected that atomic oxygen adhered to the objects at higher orbits and was still present during re-entry, thereby lowering the energy accommodation coefficient. Whether this assumption is correct would have to be clarified, and potentially a hybrid energy accommodation coefficient model can be established following both Langmuir's adsorption model for higher orbits yet also considers the orbital history of an object when assessing lower orbits.

- **Extension on drag coefficient model -** the C_D model for this research was advanced in the way that it computed C_D as a function of a simple geometry, the freestream velocity, and the ambient gas composition. However, the model did not account for more complex geometries, nor did it account for a tumbling object. The creation of a model that is still computationally realistic yet does allow for more complex geometries is considered to be one of the first steps towards a large global ballistic coefficient estimation model.

- **Filtering criteria and assessment of local matches -** even with tightened selection criteria, objects have been identified that did not had an increase in performance. How these objects got their ballistic coefficient estimated poorly is unknown, and answers to this could allow for filtering criteria for certain matches. For instance, when a match occurs during a geomagnetic storm, is the accuracy of this model dropping significantly or not? Or are there certain time- or altitude range in which NRLMSISE-00 might not have been a suitable candidate? Answers to these questions could allow for certain matches to be discarded and not have their under-performing ballistic coefficient affect the generations after it.

- **Investigate merits reducing error heritage -** when considering a global model, it would be inevitable that somewhere in the model an object will have a poorly estimated ballistic coefficient. To prevent this 'bad apple' from infecting too many other objects, special care will have to be taken for not allowing large groups of objects to be dependent of just a handful of parent objects. Perhaps an object can only be allowed to enter into a model when either the parent has a confirmed healthy β_E or that the object has at least a certain amount of parents - multiple parents would level-out the negative effect of a single parent and thereby reduce the error propagation throughout the model. Investigating exactly how and when a decision is made to either include or exclude an object based on the quality of and number of parents could reduce the effect of error propagation throughout a global model.

- **Accuracy assessment of TLE -** all the input objects have had their TLE observation data inserted into the model. However, TLE datasets are not always as accurate - some objects were found to have lots of TLE epochs within the 30 day β_E period, and with many TLE epochs assessments can be run to determine of a TLE is inappropriate for usage and be discarded. Other objects however had few TLE epochs, which made it harder to detect whether or not to incorporate all TLE epochs for the β_E period. Potentially in combination with recommendation item 5 and 6 this could lead to an advanced filtering system where objects are assessed on their available TLE data, the number and healthiness of their parents, and the epoch and altitude of the matches.

The above seven items could be a good step towards a global and accurate ballistic coefficient estimation model, and together with an early investment in getting computationally efficient software the creation of a preliminary global ballistic coefficient estimation model could be achieved in the near future. In parallel with the operation of that model upgrades can be created that for instance integrate advanced filtering algorithms and incorporates C_D computations for complex geometries. Though it has not been mentioned too often in this report, it shall be stressed here one more time - even for the small database in this experiment the computation was quite intensive. The choice of a combination between Matlab and C++ was a justified one for the prototyping phase, yet this would not suffice for larger operations. Script rewritting and usage of servers should be able to mitigate this and is therefore advised to be done as early as possible when work towards a global model is started.

A

Propagation Results

A.1. Propagation results in-orbit group 1

Iteration 1	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
DTUSAT (27842)	0.154	0.114	0.103	0.014	0.040	0.051	0.140
CUTE-1 (27844)	0.196	0.239	0.219	0.072	-0.043	-0.023	0.124
AAU CubeSat (27846)	0.153	0.174	0.100	-0.019	-0.022	0.052	0.134
CANX-1 (27847)	0.117	0.110	0.171	-0.011	0.007	-0.054	0.105
XI-4 (27848)	0.072	0.059	0.127	0.073	0.014	-0.055	-0.001
SDS-1 (33497)	0.053	0.062	0.056	0.013	-0.009	-0.003	0.040
SwissCube (35932)	0.175	0.086	0.016	-0.019	0.089	0.160	0.156
BeeSat (35933)	0.064	-0.014	0.024	0.032	0.050	0.040	0.032
Uwe-2 (35934)	0.194	0.104	0.172	0.145	0.090	0.022	0.049
ITUPSAT-1 (35935)	0.065	-0.005	0.014	-0.009	0.060	0.051	0.056
AAUSat-3 (39087)	0.202	0.172	0.181	0.189	0.030	0.021	0.013
Brite-PL (39431)	-0.006	0.080	0.098	0.178	-0.074	-0.091	-0.171
ABS. SUM	1.451	1.219	1.280	0.775	0.232	0.171	0.676

Iteration 2	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
Dice-1 (37851)	-1.397	-1.242	-0.957	-1.198	0.155	0.440	0.199
Dice-2 (37852)	-1.046	-1.162	-1.010	-1.178	-0.116	0.036	-0.132
WNISAT-1 (39423)	-0.137	-0.072	-0.149	-0.276	0.065	-0.012	-0.139
ABS. SUM	2.580	2.476	2.116	2.652	0.104	0.464	-0.072

Iteration 3	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
EgyptSat-1 (31117)	-0.173	-0.134	-0.130	-0.057	0.039	0.043	0.1152
Compass-1 (32787)	0.113	0.087	0.073	0.024	0.026	0.039	0.0887
AAUSAT-2 (32788)	0.054	-0.077	-0.163	-0.147	-0.023	-0.109	-0.0933
SEEDS (32791)	0.001	-0.069	-0.084	-0.167	-0.068	-0.083	-0.1659
AubieSat-1 (37854)	-1.130	-0.905	-0.726	-0.756	0.225	0.404	0.3742
M-Cubed (37855)	-1.173	-0.903	-0.671	-1.047	0.270	0.502	0.1261
ZaCube (39417)	0.132	0.148	0.196	-0.061	-0.016	-0.064	0.0713
ICube-1 (39432)	0.026	-0.131	-0.154	-0.359	-0.105	-0.129	-0.3334
HumSat-D (39433)	-0.066	-0.518	-0.488	-0.650	-0.452	-0.422	-0.5841
PUCP-Sat-1 (39442)	-0.097	-0.376	-0.311	-0.496	-0.279	-0.214	-0.3992
FunCube-1 (39444)	-0.062	-0.027	-0.029	0.036	0.035	0.033	0.0260
HinCube (39445)	-0.089	-0.037	-0.072	-0.095	0.052	0.017	-0.0064
DuchiFat-1 (40021)	0.046	-0.031	-0.091	-0.196	0.014	-0.046	-0.1499
C-BR1 (40024)	-0.025	-0.081	-0.126	-0.186	-0.056	-0.101	-0.1607
DTUSat-2 (40030)	-0.188	-0.206	-0.065	-0.079	-0.018	0.124	0.1092
UFTI-1 (41458)	-0.859	-0.770	-0.579	-0.560	0.089	0.280	0.2998
E-Star-1 (41459)	-0.966	-0.826	-0.657	-0.690	0.141	0.309	0.2759
AAUSat-4 (41460)	-0.444	-0.983	-0.542	-0.739	-0.539	-0.097	-0.2949
ABS. SUM	5.644	6.309	5.158	6.345	-0.665	0.486	-0.701

Iteration 4	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
Pegasus-1 (39151)	0.027	-0.040	-0.139	-0.226	-0.013	-0.112	-0.198

CubeBug-1 (39153)	0.076	0.093	0.046	-0.009	-0.017	0.030	0.067
EstCube-1 (39161)	-0.098	-0.086	-0.092	-0.021	0.012	0.005	0.077
Velox-P-2 (39438)	0.140	0.000	0.023	-0.040	0.139	0.116	0.100
First-Move (39439)	0.072	-0.281	-0.284	-0.305	-0.209	-0.212	-0.233
KRYSAOR (39441)	0.130	0.017	0.122	-0.042	0.113	0.008	0.088
Uwe-3 (39446)	0.067	-0.070	-0.026	-0.063	-0.003	0.041	0.004
CANX-4 (40055)	-0.006	0.138	0.125	0.059	-0.131	-0.119	-0.053
CANX-5 (40056)	0.102	0.061	0.107	0.045	0.041	-0.005	0.057
ABS. SUM	0.718	0.785	0.966	0.810	-0.067	-0.248	-0.092

A.2. Propagation results in-orbit group 2

Iteration 1	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
DTUSAT (27842)	0.154	0.066	0.154	-0.067	0.088	0.001	0.087
CUTE-1 (27844)	0.196	0.251	0.234	0.073	-0.055	-0.038	0.122
AAU CubeSat (27846)	0.153	0.189	0.164	0.004	-0.036	-0.011	0.149
CANX-1 (27847)	0.117	0.196	0.126	-0.022	-0.079	-0.010	0.095
XI-4 (27848)	0.072	0.112	0.094	0.050	-0.040	-0.021	0.022
Uwe-1 (28892)	-0.237	-0.253	-0.225	0.030	-0.016	0.012	0.206
XI-5 (28895)	-0.198	-0.145	-0.125	0.028	0.052	0.072	0.170
SDS-1 (33497)	0.053	0.253	0.194	0.113	-0.200	-0.141	-0.059
SwissCube (35932)	0.175	0.271	0.299	0.124	-0.096	-0.123	0.052
BeeSat (35933)	0.064	0.161	0.104	0.189	-0.097	-0.040	-0.125
Uwe-2 (35934)	0.194	0.293	0.304	0.248	-0.099	-0.110	-0.054
ITUPSAT-1 (35935)	0.065	0.020	0.086	0.137	0.046	-0.021	-0.072
Dice-1 (37851)	-1.397	-1.817	-1.605	-1.761	-0.420	-0.208	-0.364
Dice-2 (37852)	-1.046	-0.836	-0.665	-0.856	0.210	0.381	0.189
AubieSat-1 (37854)	-1.130	-1.504	-1.168	-1.191	-0.374	-0.038	-0.060
M-Cubed (37855)	-1.173	-1.667	-1.284	-1.612	-0.493	-0.111	-0.439
ABS. SUM	6.423	8.031	6.831	6.503	-1.608	-0.408	-0.080

Iteration 2	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
EgyptSat-1 (31117)	-0.173	-0.174	-0.105	-0.085	-0.001	0.068	0.088
Compass-1 (32787)	0.113	0.189	0.236	0.122	-0.076	-0.123	-0.009
AAUSAT-2 (32788)	0.054	-0.170	-0.343	-0.354	-0.116	-0.289	-0.300
SEEDS (32791)	0.001	-0.106	-0.038	-0.180	-0.105	-0.037	-0.179
Pegasus-1 (39151)	0.027	0.054	-0.009	0.014	-0.027	0.018	0.013
CubeBug-1 (39153)	0.076	0.014	0.096	-0.114	0.062	-0.020	-0.038
EstCube-1 (39161)	-0.098	0.119	0.011	0.147	-0.022	0.087	-0.049
ZaCube (39417)	0.132	0.060	0.040	-0.076	0.072	0.092	0.056
ICube-1 (39432)	0.026	-0.026	-0.161	-0.318	-0.001	-0.135	-0.292
HumSat-D (39433)	-0.066	-0.426	-0.446	-0.682	-0.360	-0.381	-0.616
Velox-P-2 (39438)	0.140	0.005	-0.001	-0.059	0.134	0.138	0.080
First-Move (39439)	0.072	-0.465	-0.491	-0.393	-0.393	-0.419	-0.321
PUCP-Sat-1 (39442)	-0.097	-0.275	-0.275	-0.425	-0.177	-0.178	-0.328
FunCube-1 (39444)	-0.062	-0.253	-0.245	0.005	-0.191	-0.183	0.057
HinCube (39445)	-0.089	-0.211	-0.131	-0.176	-0.123	-0.042	-0.088
Uwe-3 (39446)	0.067	-0.012	-0.016	-0.117	0.055	0.051	-0.050
DuchiFat-1 (40021)	0.046	-0.082	-0.198	-0.412	-0.036	-0.152	-0.367
C-BR1 (40024)	-0.025	-0.208	-0.233	-0.249	-0.183	-0.208	-0.224
DTUSat-2 (40030)	-0.188	-0.236	-0.289	-0.235	-0.048	-0.100	-0.047
UFTI-1 (41458)	-0.859	-1.161	-0.932	-1.000	-0.302	-0.072	-0.141
E-Star-1 (41459)	-0.966	-1.096	-0.889	-0.956	-0.130	0.078	0.011
AAUSat-4 (41460)	-0.444	-1.633	-1.374	-1.323	-1.189	-0.929	-0.879
ABS. SUM	3.820	6.977	6.557	7.443	-3.157	-2.737	-3.623

Iteration 3	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
CSTB-1 (31122)	0.129	0.190	0.179	0.119	-0.061	-0.049	0.011
CP-4 (31132)	0.003	0.021	-0.033	-0.010	-0.019	-0.030	-0.007

AeroCube-2 (31133)	0.127	0.142	0.115	0.243	-0.015	0.012	-0.115
KRYSAOR (39441)	0.130	0.133	0.136	-0.139	-0.003	-0.006	-0.008
CANX-4 (40055)	-0.006	0.081	0.065	0.060	-0.075	-0.059	-0.054
CANX-5 (40056)	0.102	-0.075	-0.064	-0.144	0.027	0.038	-0.042
ABS. SUM	0.498	0.643	0.593	0.714	-0.146	-0.095	-0.217

Iteration 4	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
Libertad-1 (31128)	0.105	0.130	0.172	0.093	-0.024	-0.066	0.013
CP-3 (31129)	0.156	0.163	0.169	0.099	-0.007	-0.013	0.057
Cape-1 (31130)	0.361	0.349	0.315	0.017	0.012	0.046	0.344
SDS-4 (38339)	-0.409	-0.332	-0.327	-0.260	0.078	0.082	0.150
Brite-PL (39431)	-0.006	0.119	0.152	0.165	-0.113	-0.146	-0.159
ABS. SUM	1.038	1.092	1.136	0.634	-0.054	-0.098	0.403

Iteration 5	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
WNISAT-1 (39423)	-0.137	-0.090	-0.150	-0.142	0.047	-0.013	-0.005
ABS. SUM	0.137	0.090	0.150	0.142	0.047	-0.013	-0.0049

A.3. Propagation results in-orbit group 3

Iteration 1	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
DTUSAT (27842)	0.154	0.144	0.201	0.186	0.010	-0.047	-0.032
CUTE-1 (27844)	0.196	0.216	0.153	0.204	-0.020	0.043	-0.008
AAU CubeSat (27846)	0.153	0.109	0.145	0.061	0.043	0.008	0.092
CANX-1 (27847)	0.117	0.099	0.101	0.168	0.017	0.016	-0.051
XI-4 (27848)	0.072	0.131	0.114	0.178	-0.059	-0.042	-0.106
Uwe-1 (28892)	-0.237	-0.253	-0.200	-0.179	-0.016	0.036	0.058
XI-5 (28895)	-0.198	-0.145	-0.136	-0.134	0.052	0.061	0.064
SDS-1 (33497)	0.053	0.227	0.208	0.256	-0.173	-0.155	-0.202
SwissCube (35932)	0.175	0.129	0.186	0.180	0.047	-0.010	-0.005
BeeSat (35933)	0.064	0.234	0.101	0.181	-0.170	-0.037	-0.117
Uwe-2 (35934)	0.194	0.202	0.247	0.240	-0.008	-0.053	-0.046
AAUSAT-3 (39087)	0.202	0.135	0.147	0.162	0.067	0.054	0.040
ABS. SUM	1.814	2.024	1.939	2.129	-0.211	-0.125	-0.315

Iteration 2	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
EgyptSat-1 (31117)	-0.173	-0.094	-0.113	-0.057	0.079	0.060	0.116
ItupSat-1 (35935)	0.065	0.081	0.114	0.102	-0.016	-0.049	-0.037
Pegasus-1 (39151)	0.027	0.209	0.266	0.272	-0.182	-0.239	-0.245
CubeBug-1 (39153)	0.076	0.273	0.153	0.331	-0.198	-0.077	-0.256
EstCube-1 (39161)	-0.098	0.100	0.051	0.010	-0.002	0.047	0.088
ZaCube (39417)	0.132	0.294	0.331	0.327	-0.162	-0.199	-0.195
WNISAT-1 (39423)	-0.137	-0.069	-0.035	-0.024	0.068	0.102	0.113
Velox-P-2 (39438)	0.140	0.149	0.205	0.153	-0.010	-0.065	-0.014
First-Move (39439)	0.072	0.300	0.319	0.287	-0.228	-0.247	-0.215
KRYSAOR (39441)	0.130	0.359	0.290	0.345	-0.229	-0.160	-0.215
FunCube-1 (39444)	-0.062	0.138	0.108	0.163	-0.076	-0.046	-0.101
HinCube (39445)	-0.089	0.145	0.177	0.113	-0.056	-0.088	-0.024
Uwe-3 (39446)	0.067	0.218	0.147	0.120	-0.150	-0.080	-0.053
CANX-4 (40055)	-0.006	0.269	0.242	0.263	-0.263	-0.236	-0.256
CANX-5 (40056)	0.102	0.246	0.165	0.231	-0.144	-0.062	-0.129
ABS. SUM	1.375	2.945	2.715	2.798	-1.569	-1.340	-1.423

Iteration 3	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
CSTB-1 (31122)	0.129	0.147	0.230	0.244	-0.018	-0.100	-0.115
SEEDS (32791)	0.001	0.208	0.148	0.113	-0.207	-0.147	-0.112
SDS-4 (38339)	-0.409	-0.238	-0.156	-0.229	0.172	0.253	0.181
Brite-PL (39431)	-0.006	0.181	0.185	0.107	-0.174	-0.178	-0.101
ICube-1 (39432)	0.026	0.219	0.161	0.203	-0.193	-0.135	-0.177
HumSat-D (39433)	-0.066	0.169	0.136	0.122	-0.103	-0.071	-0.056
PUCP-Sat-1 (39442)	-0.097	0.114	0.077	0.189	-0.016	0.020	-0.092
ABS. SUM	0.734	1.275	1.092	1.207	-0.541	-0.358	-0.473

Iteration 4	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8

Libertad-1 (31128)	0.105	0.160	0.202	0.196	-0.054	-0.096	-0.091
CP-3 (31129)	0.156	0.154	0.132	0.215	0.002	0.024	-0.059
Cape-1 (31130)	0.361	0.436	0.484	0.391	-0.075	-0.123	-0.030
CP-4 (31132)	0.003	0.137	0.083	0.174	-0.135	-0.081	-0.171
AeroCube-2 (31133)	0.127	0.186	0.129	0.270	-0.059	-0.002	-0.143
Compass-1 (32787)	0.113	0.413	0.321	0.422	-0.301	-0.208	-0.309
AAUSAT-2 (32788)	0.054	0.264	0.174	0.232	-0.210	-0.120	-0.178
Dice-1 (37851)	-1.397	0.229	0.261	0.395	1.168	1.136	1.002
Dice-2 (37852)	-1.046	0.254	0.317	0.398	0.792	0.729	0.648
AubieSat-1 (37854)	-1.130	0.382	0.413	0.414	0.748	0.717	0.717
M-Cubed (37855)	-1.173	0.342	0.448	0.388	0.831	0.725	0.786
ABS. SUM	5.665	2.958	2.965	3.493	2.707	2.700	2.172

Iteration 5	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
DuchiFat-1 (40021)	0.046	0.325	0.283	0.351	-0.280	-0.238	-0.306
UFTI-1 (41458)	-0.859	0.211	0.213	0.236	0.649	0.647	0.624
E-Star-1 (41459)	-0.966	0.230	0.206	0.324	0.736	0.761	0.642
AAUSat-4 (41460)	-0.44437	0.4377363	0.516023	0.549486	0.007	-0.072	-0.105
ABS. SUM	2.316	1.204	1.218	1.461	1.112	1.098	0.855

Iteration 6	**Difference [km] with verification orbit**				**Difference [km] with initial orbit**		
NORAD	initial	estimated	adjusted	a = 0.8	estimated	adjusted	a = 0.8
C-BR1 (40024)	-0.025	0.148	0.223	0.144	-0.123	-0.198	-0.119
DTUSat-2 (40030)	-0.188	0.075	0.095	0.093	0.113	0.093	0.095
ABS. SUM	0.213	0.223	0.319	0.238	-0.009	-0.105	-0.024

B

Parent-Child

B.1. Parent-Child in-orbit group 1

Child	# of Parents	Parent(s)
DTUSAT (27842)	3	5398, 25398, 35871
CUTE-1 (27844)	3	5398, 25398, 35871
AAU CubeSat (27846)	3	5398, 25398, 35871
CANX-1 (27847)	3	5398, 25398, 35871
XI-4 (27848)	3	5398, 25398, 35871
EgyptSat-1 (31117)	2	37851, 37852
Compass-1 (32787)	2	37851, 37852
AAUSAT-2 (32788)	2	37851, 37852
SEEDS (32791)	2	37851, 37852
SDS-1 (33497)	1	27944
SwissCube (35932)	1	27944
BeeSat (35933)	1	27944
Uwe-2 (35934)	1	27944
ITUPSAT-1 (35935)	1	27944
Dice-1 (37851)	1	33497
Dice-2 (37852)	1	33497
AubieSat-1 (37854)	2	37851, 37852
M-Cubed (37855)	2	37851, 37852
AAUSat-3 (39087)	1	5398
Pegasus-1 (39151)	10	31117, 32787, 32788, 32791, 39417, 39432, 39433, 39442, 39444, 39445
CubeBug-1 (39153)	10	31117, 32787, 32788, 32791, 39417, 39432, 39433, 39442, 39444, 39445
EstCube-1 (39161)	5	31117, 32787, 39417, 39444, 39445
ZaCube (39417)	2	37851, 37852
WNISAT-1 (39423)	1	39431
Brite-PL (39431)	1	5398
ICube-1 (39432)	1	37851
HumSat-D (39433)	2	37851
Velox-P-2 (39438)	9	32787, 32788, 32791, 39417, 39432, 39433, 39442, 39444, 39445
First-Move (39439)	9	32787, 32788, 32791, 39417, 39432, 39433, 39442, 39444, 39445
KRYSAOR (39441)	7	31117, 39417, 39432, 39433, 39442, 39444, 39445
PUCP-Sat-1 (39442)	2	37851, 37852
FunCube-1 (39444)	2	37851, 37852
HinCube (39445)	2	37851, 37852
Uwe-3 (39446)	9	32787, 32788, 32791, 39417, 39432, 39433, 39442, 39444, 39445
DuchiFat-1 (40021)	2	37851,3 7852
C-BR1 (40024)	2	37851, 37852
DTUSat-2 (40030)	2	37851, 37852
PolyItan-1 (40042)	2	37851, 37852
CANX-4 (40055)	5	31117, 39417, 39432, 39444, 39445
CANX-5 (40056)	6	31117, 39417, 39432, 39442, 39444, 39445
UFTI-1 (41458)	2	37851, 37852
E-Star-1 (41459)	2	37851, 37852
AAUSat-4 (41460)	2	37851, 37852

B.2. Parent-Child in-orbit group 2

Child	**# of Parents**	**Parent(s)**
DTUSAT (27842)	2	25398,35871
CUTE-1 (27844)	2	25398,35871
AAU CubeSat (27846)	2	25398,35871
CANX-1 (27847)	2	25398,35871
XI-4 (27848)	2	25398,35871
Uwe-1 (28892)	1	27944
XI-5 (28895)	1	27944
EgyptSat-1 (31117)	4	37851,37852,37854,37855
CSTB-1 (31122)	1	31117
Libertad-1 (31128)	3	31122,31132,31133
CP-3 (31129)	3	31122,31132,31133
Cape-1 (31130)	3	31122,31132,31133
CP-4 (31132)	1	31117
AeroCube-2 (31133)	1	31117
Compass-1 (32787)	4	37851,37852,37854,37855
AAUSAT-2 (32788)	4	37851,37852,37854,37855
SEEDS (32791)	4	37851,37852,37854,37855
SDS-1 (33497)	1	27944
SwissCube (35932)	1	27944
BeeSat (35933)	1	27944
Uwe-2 (35934)	1	27944
ITUPSAT-1 (35935)	1	27944
Dice-1 (37851)	1	27944
Dice-2 (37852)	1	27944
AubieSat-1 (37854)	1	27944
M-Cubed (37855)	1	27944
SDS-4 (38339)	3	31122,31132,31133
Pegasus-1 (39151)	4	37851,37852,37854,37855
CubeBug-1 (39153)	4	37851,37852,37854,37855
EstCube-1 (39161)	1	35934
ZaCube (39417)	4	37851,37852,37854,37855
WNISAT-1 (39423)	1	39431
Brite-PL (39431)	1	31122
ICube-1 (39432)	4	37851,37852,37854,37855
HumSat-D (39433)	4	37851,37852,37854,37855
Velox-P-2 (39438)	3	37851,37852,37855
First-Move (39439)	3	37851,37852,37855
KRYSAOR (39441)	16	31117,32787,32788,32791,39151,39153,39161,39417,39432, 39433,39438,39439,39442,39444,39445,39446
PUCP-Sat-1 (39442)	4	37851,37852,37854,37855
FunCube-1 (39444)	4	37851,37852,37854,37855
HinCube (39445)	4	37851,37852,37854,37855
Uwe-3 (39446)	3	37851,37852,37855
DuchiFat-1 (40021)	4	37851,37852,37854,37855
C-BR1 (40024)	4	37851,37852,37854,37855
DTUSat-2 (40030)	4	37851,37852,37854,37855
PolyItan-1 (40042)	4	37851,37852,37854,37855
CANX-4 (40055)	16	31117,32787,32788,32791,39151,39153,39161,39417,39432, 39433,39438,39439,39442,39444,39445,39446
CANX-5 (40056)	16	31117,32787,32788,32791,39151,39153,39161,39417,39432, 39433,39438,39439,39442,39444,39445,39446

UFTI-1 (41458)	4	37851,37852,37854,37855
E-Star-1 (41459)	4	37851,37852,37854,37855
AAUSat-4 (41460)	4	37851,37852,37854,37855

B.3. Parent-Child in-orbit group 3

Child	# of Parents	Parent(s)
DTUSAT (27842)	3	5398,25398,35871
CUTE-1 (27844)	3	5398,25398,35871
AAU CubeSat (27846)	3	5398,25398,35871
CANX-1 (27847)	3	5398,25398,35871
XI-4 (27848)	3	5398,25398,35871
Uwe-1 (28892)	1	27944
XI-5 (28895)	1	27944
EgyptSat-1 (31117)	1	33497
CSTB-1 (31122)	1	39423
Libertad-1 (31128)	1	31122
CP-3 (31129)	1	31122
Cape-1 (31130)	1	31122
CP-4 (31132)	2	31122,38339
AeroCube-2 (31133)	2	31122,38339
Compass-1 (32787)	5	32791,39432,39433,39435,39442
AAUSAT-2 (32788)	5	32791,39432,39433,39435,39442
SEEDS (32791)	1	39417
SDS-1 (33497)	1	27944
SwissCube (35932)	1	27944
BeeSat (35933)	1	27944
Uwe-2 (35934)	1	27944
ItupSat-1 (35935)	4	33497,35932,35933,35934
Dice-1 (37851)	4	32791,39432,39433,39442
Dice-2 (37852)	4	32791,39432,39433,39442
AubieSat-1 (37854)	4	32791,39432,39433,39442
M-Cubed (37855)	4	32791,39432,39433,39442
SDS-4 (38339)	1	31117
AAUSAT-3 (39087)	1	5398
Pegasus-1 (39151)	1	33497
CubeBug-1 (39153)	1	33497
EstCube-1 (39161)	1	33497
ZaCube (39417)	1	33497
WNISAT-1 (39423)	3	35932,35933,35934
Brite-PL (39431)	1	39423
ICube-1 (39432)	6	31117,39151,39153,39417,39444,39445
HumSat-D (39433)	6	31117,39151,39153,39417,39444,39445
Velox-P-2 (39438)	1	33497
First-Move (39439)	1	33497
KRYSAOR (39441)	1	33497
PUCP-Sat-1 (39442)	6	31117,39151,39153,39417,39444,39445
FunCube-1 (39444)	1	33497
HinCube (39445)	1	33497
Uwe-3 (39446)	1	33497
DuchiFat-1 (40021)	1	37851
C-BR1 (40024)	1	40021
DTUSat-2 (40030)	1	40021
PolyItan-1 (40042)	1	40021
CANX-4 (40055)	1	33497
CANX-5 (40056)	1	33497
UFTI-1 (41458)	5	32788,37851,37852,37854,37855
E-Star-1 (41459)	5	32788,37851,37852,37854,37855

AAUSat-4 (41460)	5	32788,37851,37852,37854,37855

B.4. Parent-Child re-entry group 1 and group 2

Table B.4: Parent - Child table for re-entry group 1

Child	# of Parents	Parent(s)
CP-6 (35003)	1	35005
HawkSat-1 (35004)	1	35005
FitSat-1 (38853)	1	38855
TechEdSat (38854)	1	38855
We-Wish (38856)	1	38855
ArduSat-1 (39412)	1	39571
ArduSat-X (39414)	1	39571
SkyCube (39567)	1	39571
ArduSat-2 (39571)	1	39573
OpuSat (39575)	1	39573

Table B.5: Parent - Child table for re-entry group 2

Child	**# of Parents**	**Parent(s)**
CP-6 (35003)	1	35005
HawkSat-1 (35004)	1	35005
Robusta (38084)	1	40314
UniCubeSat (38085)	1	40314
FitSat-1 (38853)	1	38855
TechEdSat (38854)	1	38855
We-Wish (38856)	1	38855
ArduSat-1 (39412)	2	39567,39571
ArduSat-X (39414)	2	39567,39571
SkyCube (39567)	1	39573
ArduSat-2 (39571)	1	39573
OpuSat (39575)	1	39573

C

NRLMSISE-00 Model

Table C.1: Input parameters for the Matlab function of NRLMSISE-00.

Input	**Description**
H	An array of M altitude in meters.
LAT	An array of M geodetic latitude in degrees
LON	An array of M geodetic longitude in degrees.
YEAR	An array of M year.
DOY	An array of M day of year. Day of year ranges from 1 to 365 (or,366).
SEC	An array of M seconds in day in universal time (UT)
LST	An array of M local apparent solar time (hours). To obtain a physically realistic value, LST is set to (SEC/3600 + LON/15) by default.
F107A	An array of M 81 day average of F10.7 flux (centred on doy). If F107A is input, F107 and APH must also be input. The effects of F107A are neither large nor well established below 80,000 meters, therefore the default value is set to 150.
F107	An array of M daily F10.7 flux for previous day. If F107 is input, F107A and APH must also be input. The effects of F107 are neither large nor well established below 80,000 meters, therefore the default value is set to 150.
APH	An array of M-by-7 of magnetic index information. If APH is input, F107A and F107 must also be input. This information consists of daily magnetic index (AP), 3 hour AP for current time, 3 hour AP for 3 hours before current time, 3 hour AP for 6 hours before current time, 3 hour AP for 9 hours before current time, average of eight 3 hour AP indices from 12 to 33 hours prior to current time, and average of eight 3 hour AP indices from 36 to 57 hours prior to current time. The effects of daily magnetic index are neither large nor well established below 80,000 meters, therefore the default value is set to 4.
FLAGS	A numerical array of 23 values for setting particular variations in calculation the output. Setting a value to 0.0 removes that value's effect on the output. Setting a value to 1.0 applies the main and the cross term effects of that value on the output. Setting a value to 2.0 applies only the cross term effect of that value on the output.
OTYPE	A string specifying if the total mass density output will include anomalous oxygen ('Oxygen') or not ('NoOxygen'). The default is 'NoOxygen'.
ACTION	A string to determine action for out of range input. Specify if out of range input invokes a 'Warning', 'Error', or no action,('None'). The default is 'Warning'.

Table C.2: Output parameters for the Matlab function of NRLMSISE-00.

Output	**Description**
T	An array of M-by-2 values of temperatures. These values are exospheric temperature in Kelvin and temperature at altitude in Kelvin.
RHO	An array of M-by-9 values of densities. These values are HE number density in meters^-3, O number density in meters^-3, N2 number density in meters^-3, O2 number density in meters^-3, AR number density in meters^-3, total mass density in kilogram per meters cubed, H number density in meters^-3, N number density in meters^-3, and Anomalous oxygen number density in meters^-3.

Table C.3: Flag settings for Matlab NRLMSISE function *atmosnrlmsise*

FLAG	**Description**
FLAGS(1)	F10.7 effect on mean
FLAGS(2)	Time independent
FLAGS(3)	Symmetrical annual
FLAGS(4)	Symmetrical semi-annual
FLAGS(5)	Asymmetrical annual
FLAGS(6)	Asymmetrical semi-annual
FLAGS(7)	Diurnal
FLAGS(8)	Semi-diurnal
FLAGS(9)	Daily AP
FLAGS(10)	All UT, longitudinal effects
FLAGS(11)	Longitudinal
FLAGS(12)	UT and mixed UT, longitudinal
FLAGS(13)	Mixed AP, UT, longitudinal
FLAGS(14)	Ter-diurnal
FLAGS(15)	Departures from diffusive equilibrium
FLAGS(16)	All exospheric temperature variations
FLAGS(17)	All variations from 120,000 meter temperature (TLB)
FLAGS(18)	All lower thermosphere (TN1) temperature variations
FLAGS(19)	All 120,000 meter gradient (S) variations
FLAGS(20)	All upper stratosphere (TN2) temperature variations
FLAGS(21)	All variations from 120,000 meter values (ZLB)
FLAGS(22)	All lower mesosphere temperature (TN3) variations
FLAGS(23)	Turbopausescale height variations

Bibliography

[1] J. M. Aristoff and A. B. Poore. Implicit Runge-Kutta Methods for Orbit Propagation. *AIAA/AAS Astrodynamics Specialist Conference*, 2012.

[2] K. Berry and K. Schatten. Orbit decay prediction sensitivity to solar flux variations. *AIAA/AAS Astrrodynamics Specialist Conference*, 2008.

[3] G.A. Bird. *Molecular gas dynamics and the direct simulation of gas flows.* Clarendon Press, Oxford, 1994.

[4] B. R. Bowman and K. Moe. Drag Coefficient Variability at 175-500 km from the Orbit Decay Analyses of Spheres. *AAS/AIAA Astrodynamics Conference*, 2005.

[5] W. K. Bowman, B. R .and Tobiska, F. A. Marcos, C. Y. Huang, C. S. Lin, and W. J. Burke. A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices. *Cospar*, 2008.

[6] S. Bruinsma. The DTM-2013 thermosphere model. *Journal of Space Weather and Space Climate*, 2015.

[7] F. Cain. *How many satellites are in space?* [online] Available at: https://www.universetoday.com/ [Accessed] 03, May 2017.

[8] CelesTrak. *International Designator 1965-065.* [online] Available at: https://www.celestrak.com/satcat/1965/1965-065.asp [Accessed], January 2017.

[9] D. A. Vallado. *Fundamentals of Astrodynamics and Applications.* 1998.

[10] J. C. Dolado-Perez, C. Pardini, and L. Anselmo. Review of uncertainty sources affecting the long-term predictions of space debris evolutionary models, 2015.

[11] E. Doornbos. *Thermospheric Density and Wind Determination from Satellite Dynamics.* 2011.

[12] E. Doornbos, S. Bruinsma, M. D. Pilinski, and B. Bowman. The need for a standard for satellite drag computation to improve consistency between thermosphere density models and data sets. *6th European Conference on Space Debris*, 2013.

[13] J. T. Emmert. A long-term data set of globally averaged thermospheric total mass density. *Journal of Geophysical Research*, 2009.

[14] J.T. Emmert. Thermospheric mass density: A review. *Advances in Space Research*, 2015.

[15] O. Golberg. Adaptive Stepsize Numerical Methods for Solving Ordinary Differential Equations. 2007.

[16] I. Harris and W. Priester. Time-Dependent Structure of the Upper Atmosphere, 1962.

[17] D. H. Hathaway and R. M. Wilson. Geomagnetic activity indicates large amplitude for sunspot cycle 24. *Geophysical Research Letters*, 2006. doi: 10.1029/2006GL027053.

[18] F. R. Hoots and R. L. Roehrich. Spacetrack Report No. 3–Models for Propagation of NORAD Elements Sets. *Spacetrack Report*, 1980.

[19] D.V. Hoyt and K. H. Schatten. A discussion of plausible solar irradiance variations:1700-1992, 1993.

[20] InfoSatellites. *RigidSphere2.* [online] Available at: http://www.infosatellites.com/rigidsphere2-lcs4-satellite-information-norad-5398.html [Accessed], January 2017.

[21] L. G. Jacchia. Revised Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles. Technical report, 1971.

[22] D. G. King-Hele. The Earth's neutral upper atmosphere. *Reviews of Geophysics*, 1978.

[23] G. Koppenwallner. Energy accommodation coefficient and momentum transfer modeling. Technical report, HTG, Katlenburg Lindau, 2009.

[24] G.D. Krebs. *Calsphere 1, 2, 3, 4*. [online] Available at: http://space.skyrocket.de/doc_sdat/calsphere-1.htm [Accessed], January 2017.

[25] G.D. Krebs. *LCS 1, 2, 3, 4*. [online] Available at: http://space.skyrocket.de/doc_sdat/lcs-1.htm [Accessed], January 2017.

[26] D. Kucharski. Spin parameters of Low Earth Orbiting satellites Larets and Stella determined from Satellite Laser Ranging data. *Advances in Space Research*, 2014.

[27] C. Levit and W. Marshall. Improved orbit predictions using two-line elements. *Advances in Space Research*, 2011.

[28] NASA Lyndon, B. Johnson Space Center. Debris Assessment Software User's Guide Astromaterials Research and Exploration Science. 2012.

[29] P. M. Mehta, A. Walker, E. Lawrence, R. Linares, D. Higdon, and J. Koller. Modeling satellite drag coefficients with response surfaces. *Advances in Space Research*, 2014.

[30] K. Moe and M. M. Moe. Gas-surface interactions and satellite drag coefficients. *Planetary and Space Science*, 2005.

[31] O. Montenbruck and E. Gill. *Satellite Orbits*. Springer, Oberpfaffenhofen, 2001.

[32] A. Nicholas, T. Finne, I. Galysh, A. Mai, J. Yen, W. Sawka, J. Ransdell, and S. Williams. SSC13-I-3 SpinSat Mission Overview Conference on Small Satellites. *27th Annual AIAA/USU Conference on Small Satellites*, 2013.

[33] Arecibo Observatory. *Radar calibration (via satellites)*. [online] Available at: https://www.naic.edu/p̃hil/hardware/radarcal/radarcal.html [Accessed], January 2017.

[34] Natonal Oceanic and Atmospheric Administration (NOAA). *F10.7 cm radio emissions*. [online] Available at: http://www.swpc.noaa.gov/phenomena/f107-cm-radio-emissions [Accessed] 30, November 2017.

[35] D. L. Oltrogge, K. Leveque, and M. Park. An Evaluation of CubeSat Orbital Decay. *25th Annual AIAA/USU Conference on Smallsats*, 2011.

[36] J. N. Pelton. *New Solutions for the Space Debris Problem*. Springer, 2015.

[37] W. D. Pesnell. Predictions of solar cycle 24. *Solar Physics*, 2008.

[38] J. M. Picone. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research*, 2002.

[39] J. M. Picone, J. T. Emmert, and J. L. Lean. Thermospheric densities derived from spacecraft orbits: Accurate processing of two-line element sets. *Journal of Geophysical Research: Space Physics*, 2005.

[40] M. D. Pilinski, B. A. Bowman, S. E. Palo, J. M. Forbes, B. L. Davis, R. G. Moore, C. Koehler, and B. Sanders. Comparative Analysis of Satellite Aerodynamics and Its Application to Space-Object Identification.

[41] M. D. Pilinski, B. M. Argrow, and S. E. Palo. Semiempirical Model for Satellite Energy-Accommodation Coefficients. *Journal of Spacecraft and Rockets*, 2010.

[42] Earth Observation Portal. *eoPortal Directory*. [online] Available at: https://directory.eoportal.org/ [Accessed], January 2017.

[43] N. R. Rigozo, M. P. Souza Echer, H. Evangelista, D. J R Nordemann, and E. Echer. Prediction of sunspot number amplitude and solar cycle length for cycles 24 and 25. *Journal of Atmospheric and Solar-Terrestrial Physics*, 2011. URL `http://dx.doi.org/10.1016/j.jastp.2010.09.005`.

[44] International Earth Rotation and Reference Systems Service. *Polar motion, 2001-2006.* [IERS] Available at: https://www.iers.org/IERS/EN/Science/EarthRotation/PolarMotionPlot.html [Accessed] 13, February 2017.

[45] M. Rutkowska. The accuracy of orbit estimation for the low-orbit satellites LARETS and WESTPAC. *Advances in Space Research*, 2005.

[46] J. Sang and J. C. Bennett. Achievable debris orbit prediction accuracy using laser ranging data from a single station. *Advances in Space Research*, 2014.

[47] J. Sang, J. C. Bennett, and C. H. Smith. Estimation of ballistic coefficients of low altitude debris objects from historical two line elements. *Advances in Space Research*, 2013.

[48] A. Saunders, G.G. G Swinerd, and H.G. G Lewis. Deriving Accurate Satellite Ballistic Coefficients from Two-Line Element Data. *Journal of Spacecraft and Rockets*, 49(1):175–184, 2012. ISSN 0022-4650. doi: 10.2514/1.A32023. URL `http://eprints.soton.ac.uk/185181/`.

[49] T. Scholz, C. O. Asma, and A. Aruliah. Recommended Set of Models and Input Parameters for the Simulations of Orbital Dynamics of the Qb50 Cubesats. 2012.

[50] J. H. Seago and D. A. Vallado. The coordinate frames of the US space object catalogs. 2000.

[51] L. H. Sentman. *Sentman, Lee H. (1961), Free molecule flow the- ory and its application to the determination of aerodynamic forces.* Lockheed Missiles Space Company, 1961.

[52] International Standard. ISO 27852: Space systems - Estimation of orbit lifetime. 2011, 2011.

[53] E. K. Sutton. Normalized Force Coefficients for Satellites with Elongated Shapes. *Journal of Spacecraft and Rockets*, 2009.

[54] M. Swartwout. The first one hundred CubeSats : A statistical look. *Journal of Small Satellites*, 2013.

[55] D. A. Vallado and P. Crawford. SGP4 Orbit Determination. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008.

[56] D. A. Vallado and D. Finkleman. A critical assessment of satellite drag and atmospheric density modeling. *Acta Astronautica*, 2014.

[57] P. J. van der Houwen, E. Messina, and J. J. B. de Swart. Parallel Störmer–Cowell methods for high-precision orbit computations. *Applied Numerical Mathematics*, 1999.

[58] J. Verheijen. *International Terrestrial Reference Frame 2014 (ITRF2014).* [online] Available at: https://confluence.qps.nl/ [Accessed], February 2017.

[59] K. F. Wakker. *Astrodynamics.* TU Delft, Delft, 2014.

[60] W. E. Wiesel. Spaceflight Dynamics. 1997.

[61] M. P. Wilkins, K. T. Alfriend, S. L. Coffey, and A. M. Segerman. Transitioning from a General Perturbations to a Special Perturbations Space Object Catalog. *AIAA Paper 2000-4238, AIAA/AAS Astrodynamics Specialist Conference*, 2000.