POLITECNICO DI TORINO

MASTER's Degree in Aerospace Engineering

MASTER's Degree Thesis

CubeSats deployed from the ISS: development of a code to predict the orbital decay

Supervisors

Prof. Lorenzo CASALINO

Candidate

Giuseppe ROCCASALVA

April 2023

Summary

In the last decade the number of satellites per year deployed from the International Space Station (ISS) is rapidly increasing. Despite the many pros of this strategy, the main cons is the low initial altitude of the ISS orbit which implies a shorter operational life. Indeed, a satellite deployed in that orbit, without a propulsion system, is usually meant to re-enter into the Earth atmosphere over a period of time ranging from few weeks to 3 years. This work is performed in collaboration with Nanoracks Space Outpost Europe which serves satellite deployment on board the ISS. The main objective of the study is to develop a software tool to predict the in-orbit behaviour of satellites deployed from the ISS, visualising the trend of the orbital elements up to the re-entry into the atmosphere and with a main focus on the altitude variation. The method, followed to examine the effect of perturbations on the satellite orbit, is a general perturbation technique, the solution of which is achieved through numerical integration; hence, the approach is semianalytical. The simulations show results for 1U, 2U and 3U CubeSats with an average relative error in the lifetime calculation of 15%. In conclusion, the work carried out could be continued by implementing some modifications to the code to strive more accurate results.

Table of Contents

List of Tables

List of Figures

Acronyms

DRAMA Debris Risk Assessment and Mitigation Analysis

ECI Earth-Centered Inertial

ISS International Space Station

JEM Japanese Experiment Module

JEMRMS JEM Remote Manipulator System

J-SSOD JEM -Small Satellite Orbital Deployer

LEO Low Earth Orbit

MPEP Multi-Purpose Experiment Platform

NRAL Nanoracks Airlock

NRCSD Nanoracks CubeSat Deployer

NRKDS Nanoracks Kaber Deployment System

NRSOE Nanoracks Space Outpost Europe

OLET Orbit Lifetime Estimation Tool

SGP Simplified General Perturbations

SPDM Special Purpose Dexterous Manipulator

SSRMS Space Station Remote Manipulator System

STK Systems Tool Kit

Chapter 1

Introduction

This study consists of developing a code to estimate the time it takes to a LEO (Low Earth Orbit) satellite to reenter the Earth's atmosphere and to show how its orbital elements vary in the meantime. The work was performed within a company, Nanoracks Space Outpost Europe (NRSOE), which serves satellite deployment from the International Space Station (ISS). Therefore, among the various satellites populating the LEO, the main focus was on those released from the ISS.

1.1 About Nanoracks Space Outpost Europe and its services

Nanoracks Space Outpost Europe, the space company at which this thesis was conducted, deals with logistics on board the ISS. Particularly, it hosts internal and external payloads and provides satellite deployments; given the study case, the focus is on the latter. The company offers three options depending on the properties of the satellite to be deployed [1] [2] [3].

Nanoracks CubeSat Deployer (NRCSD). Of the three options it is the smallest deployer and in the single wide configuration it is capable to host a maximum volume of 6U, i.e. 10 cm x 10 cm x 60 cm, 12U in the double wide configuration. The NRCSD is suitable for CubeSats, a class of satellites conforming to a specific criteria that controls factors such as its shape, size, and weight. A CubeSat is based on a 10 cm cube unit with a mass of approximately 1 to 1.33 kg, called 1U [4]. The satellite is integrated inside NRCSD at the ground, then, the assembly is launched to the ISS, lastly, the position for deploying is reached through a robotic arm, the JEMRMS (Japanese Experiment Module Remote Manipulator System). NRCSD interfaces with the JEMRMS via the Multi-Purpose Experiment Platform (MPEP), which can also mount another CubeSat deployer, the J-SSOD (JEM -Small Satellite Orbital Deployer) [5]. As figure 1.1 shows, eight single wide NRCSDs can be used at once, for a total volume of 48U.

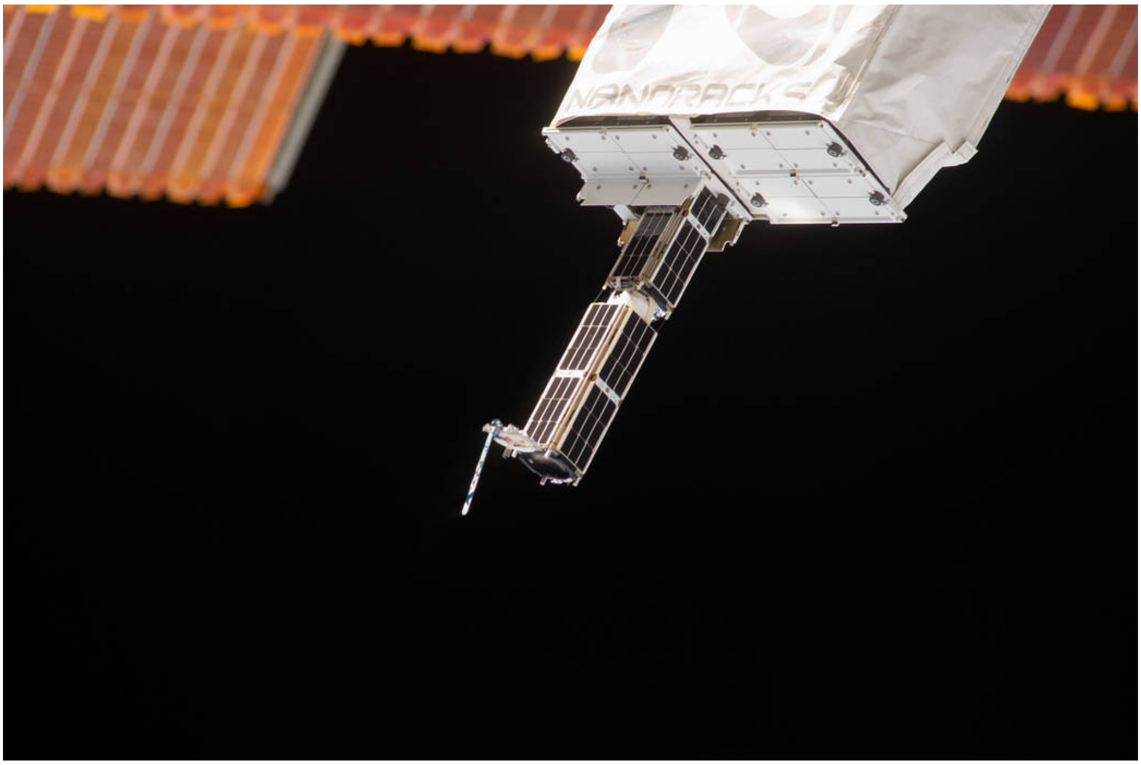

Figure 1.1: Nanoracks Cubesat Deployer (credit NASA [6])

Nanoracks Kaber Deployment System (NRKDS). It consists of two systems: the Nanoracks Kaber Deployer and the Nanoracks Separation System. The NRKDS aims to serve microsatellite with a mass lower than 90 kg and size lower than 106 x 81 x 63 cm. It utilises as a staging facility the JEM, but unlike NRCSD, it is positioned through the Canadian robotic arm SPDM (Special Purpose Dexterous Manipulator).

Nanoracks Airlock (NRAL). It is also called Bishop. Unlike the NRCSD and the NRKDS, the NRAL not only performs satellite deployments but can accommodate payloads too, either internally or externally. It is maneuvered through the SSRMS (Space Station Remote Manipulator System) and can deploy satellite with mass up to about 320 kg.

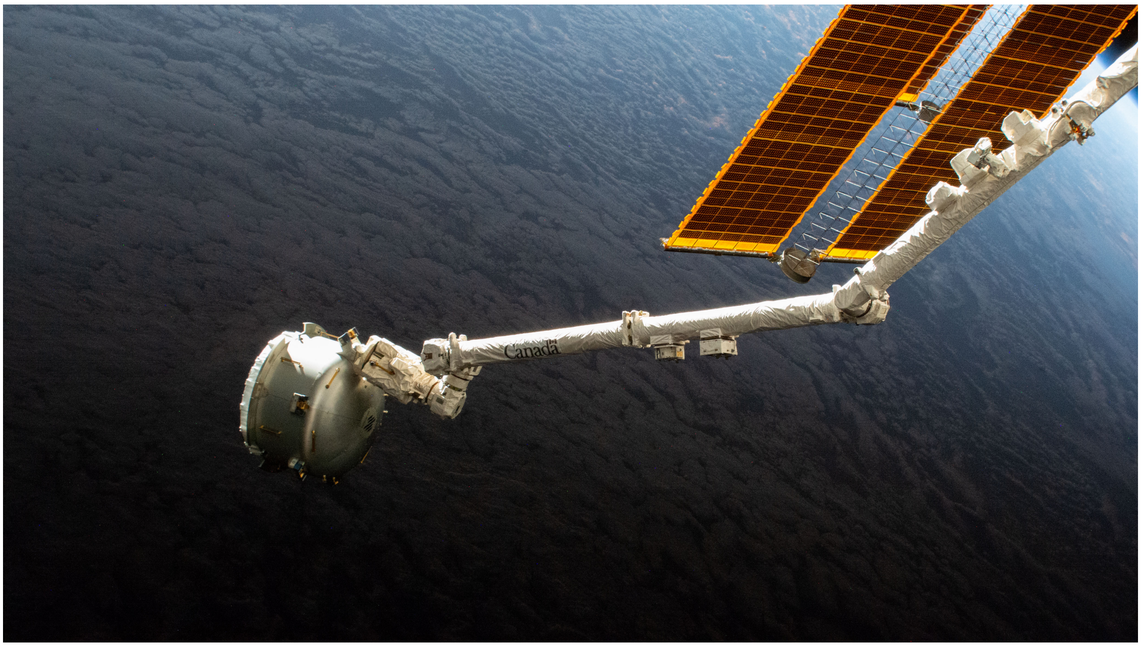

Figure 1.2: Nanoracks Bishop Airlock (credit NASA [7])

1.2 Satellite deployment from the ISS: advantages and disadvantages

The traditional method to place in orbit a small satellite is through the rideshare. Whereby, when a rocket can launch extra-weight other than its main payload, it also hosts smaller payloads. However, the alternative method of deploying from orbiting platforms has recently been successfully pursued, exploiting the presence of the ISS in LEO. In fact, in the last decade more than 200 satellites were launched from the space station [8]. There are three main reasons for these remarkable numbers [9]:

Flexible launch windows. First of all, from six to nine launches to the ISS are usually programmed each year [10]. Thus, unlike the typical rideshare, missing the launch window may only result in the mission being rescheduled over the following three months. This point is crucial for CubeSat developers, such as small companies, research centers and universities, who could find it difficult to keep up with deadlines.

Launch Environment. Launching with rideshare solutions means that the satellite is treated as a secondary payload, and it is directly mounted on the internal rocket structure. It must withstand the challenging mechanical loads in the harsh launch environment, mainly vibrations and shocks. Also, it must survive to the depressurisation load inside the rocket fairing; consequently, several expensive test are required. On the other hand, deploying from the ISS implies that the satellite is launched toward the station inside a cargo resupply ship. Therefore, a pressurised environment with ambient temperature is maintained. Moreover, the satellite is bubble wrapped and placed inside Cargo Transfer Bags. This considerably reduces the magnitude of shock and vibration loads and avoids expensive test, increasing development speed.

Quality check. Lastly, the crew onboard can check before the deployment whether the satellite was damaged during the launch.

On the other hand, the satellite could be deployed after some months it reached the station, depending on the ISS scheduling. Furthermore, the orbit of deployment usually has an altitude of 400 km. Thus, the atmospheric drag will make it decay, unless the satellite is equipped with a propulsion system compliant with ISS safety requirements. Depending on the satellite properties, it could reenter the Earth's atmosphere in a range of time of few weeks to years. Certainly this is a good point concerning the debris mitigation, but this lifetime could not be enough to accomplish the mission set by the customer.

1.3 Code Development

Relying on the context previously discussed, NRSOE commissioned a study with the aim of investigating dynamics that cause satellites to reenter the Earth's atmosphere after the deployment from ISS. In fact, this is usually a crucial point on which contracting takes place when the company negotiates with a customer. Within the company the task was approached in two ways, carried out separately: a *parametric study* and a *code development*. The first concerns how the lifetime of a satellite could change varying several parameters, e.g. initial altitude and drag coefficient. That study exploits the existing mission analysis tools (STK, DRAMA etc.) and is reported in [11]; instead, code development is dealt with in this document.

Thus, the aim of this work is to design an intuitive software tool that quickly estimates the lifetime of a satellite deployed from the ISS. Also the trend over time of the satellite orbital elements should be visualised, with a main focus on the altitude variation. The altitude, indeed, is a parameter allowing to know when the satellite has reentered the Earth's atmosphere. A code was developed to achieve this objective, it was called *Orbit Lifetime Estimation Tool* (OLET). The approach implemented to investigate the effects of perturbations is semianalytical, hence, equations are obtained with *general perturbations techniques* and then numerically integrated. The general approach is also applied in well-known orbit propagators, such as the SGP4 model.

The great majority of satellites launched from ISS are CubeSats, indeed, the OLET was tested on them, mainly 1Us, 2Us and 3Us. However, the code could be potentially used also on other types of satellites, if their properties are known. Also, with small changes on the equations implemented, OLET could be applied to other kind of low Earth orbits, and not only to that of the ISS (quasi-circular, 51.6° inclined, 400 km altitude).

Chapter 2

Basics of Astrodynamics

This work lays on physical principles which should be briefly introduced before dealing with the main part of the work. Initially, the laws underlying astrodynamics are presented; ideal conditions are considered to simplify the concepts as much as possible. Later, other effects are added to make the physical description adhere to reality. The theoretical treatment is not exhaustive, as the aim is to provide some notions to make this thesis easier to read.

2.1 Basic Laws

Kepler and Newton laws, dating back to 17th century, provide the description and the equations to understand the planetary motion. Thus, a brief overview is reported with reference to [12].

2.1.1 Kepler's Laws

Kepler's laws are an attempt to fit Tycho Brahe's observations of celestial bodies in a geometry. They are fundamental to understand orbital mechanics and mark an epoch in the history of mathematical science.

First Law. The orbit of each planet is an ellipse, with the Sun at a focus.

Second Law. The line joining the planet to the Sun sweeps out equal areas in equal times.

$$\frac{dA}{dt} = cost \tag{2.1}$$

Third Law. The square of the period of a planet (T) is directly proportional to the cube of its mean distance from the Sun. The semi-major axis (a) of the ellipse is also called the mean distance from the Sun.

$$T^2 = ka^3 \tag{2.2}$$

2.1.2 Newton's Laws of Motion and Universal Gravitation

Kepler's laws are still a description of the planet motion; their mathematical explanation was achieved years later by Newton. He identified the following three laws of motion, valid in an inertial reference frame.

First Law. Every body continues in its state of rest or of uniform motion in a straight line unless it is compelled to change that state by forces impressed upon it.

Second Law. The time-rate change of momentum is proportional to the force impressed and is in the same direction as that force.

$$\frac{d(m\vec{v})}{dt}) = \sum \vec{F} \tag{2.3}$$

Third Law. To every action there is always opposed an equal reaction.

When the mass of the body remain constant over time, the (2.3) can be expressed as follows:

$$m\frac{d\vec{v}}{dt} = m\vec{a} = \sum \vec{F} \tag{2.4}$$

Moreover, Newton also formulated the *law of gravity*, mathematically expressed in (2.5), in which the force applied to the mass m by the mass M is considered.

Law of Universal Gravitation. Any two bodies attract one another with a force (F_g) proportional to the product of their masses (M, m) and inversely proportional to the square distance (r) between them.

$$\vec{F}_g = -G\frac{mM}{r^2}\frac{\vec{r}}{r} \tag{2.5}$$

Where, $\vec{r}$ is the position vector of m with respect to M. G is the *universal gravitation constant* and has the approximate value of 6.673 × 10^{-20} $km^3/kg \cdot s^2$. The minus sign in (2.5) indicate that the gravitational force is attractive.

2.2 The Two-Body Problem

Most of the time when a complex orbital study is approached, taking into account all the effects acting on the system from the very beginning may be difficult. Thus, the *two-body problem* is often a starting point. In this section, the purpose is only to provide the equations, used further on. For a detailed understanding and mathematical demonstrations, please refer to [12] and [13].

2.2.1 Two-Body Problem Assumptions

The following assumptions are necessary to develop the two-body equation, a useful tool to comprehend the orbital mechanics but not enough to estimate accurately the motion of an orbiting body.

1. The two-body system consists of a primary body and one secondary (figure 2.1). The mass of the latter (m) is negligible compared to that of the former (M).
2. The coordinate system chosen is inertial.
3. The two bodies are spherically symmetrical, with uniform density. This allows to treat each as a point mass.
4. No other forces act on the system except for gravitational forces that act along a line joining the centers of the two bodies.

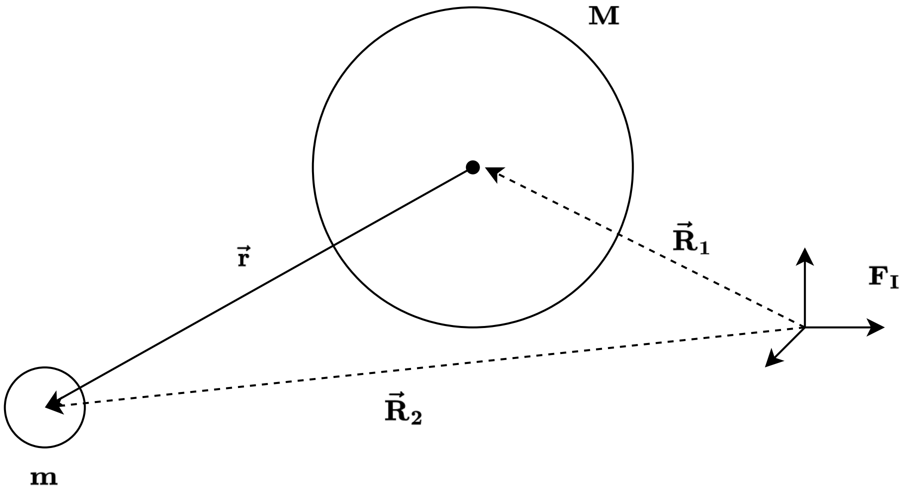

Figure 2.1: Two-body scheme and inertial frame

2.2.2 The Equation of Relative Motion

Basing on assumptions of subsection 2.2.1, the second law of motion (2.4) and the law of gravitation (2.5) are evaluated for each body and manipulated to get the *two-body equation*:

$$\ddot{\vec{r}} = -\frac{\mu}{r^2}\frac{\vec{r}}{r} \tag{2.6}$$

It is a second-order, nonlinear, vector, differential equation. μ is called *gravitational parameter* and it is defined as the product between the universal gravitation constant and the primary mass of the system, $\mu = GM$. $\vec{r}$ is the position vector of the secondary body with respect to the primary; hence, this form of the equation is *relative*.

2.2.3 The Trajectory Equation

Despite the introduction of the two-body motion equation, still some useful information about the nature of orbital motion are missing. These elements are found in the last of two-body assumptions (subsection 2.2.1).

No other forces act on the system except for gravitational forces and the fact that the gravitational field is conservative imply that the *(specific) mechanical energy* (E) of each body is constant over time. Thus, an object moving under the influence of gravity alone does not lose or gain mechanical energy, it only exchanges *kinetic energy* for *potential energy.*

The second part of the assumption, *forces that act along a line joining the centers of the two bodies*, means that there is no tangential force acting with respect to the center of the reference frame considered in the equation of relative motion, i.e. the primary point mass. In other words, there is no torque acting on the system, consequently, the *(specific) angular momentum* ($\vec{h}$) of the secondary mass about the primary mass does not change over time. Since $\vec{h}$ is defined as the cross product between position and velocity vectors, it must be perpendicular to the plane containing $\vec{r}$ and $\vec{v}$, called orbital plane. Thus $\vec{r}$ and $\vec{v}$ must always remain in the same plane fixed in inertial space because the specific angular momentum is constant. This imply that the orbital motion is two-dimensional.

In brief, the following relations are valid under 2.2.1 assumptions.

$$\begin{aligned} \frac{dE}{dt} &= \frac{d}{dt}(K + U_g) = \frac{d}{dt}\left(\frac{v^2}{2} - \frac{\mu}{r}\right) = 0 \\ \frac{d\vec{h}}{dt} &= \frac{d}{dt}(\vec{r} \times \vec{v}) = \vec{0} \end{aligned} \tag{2.7}$$

Where E and $\vec{h}$ are *specific* parameters since they are obtained dividing mechanical energy and angular momentum for the secondary mass.

Basing on (2.7), it is possible to manipulate and integrate the two-body equation, obtaining the trajectory equation, hence, how the position change over time.

$$r = \frac{h^2/\mu}{1 + (B/\mu)\cos\nu(t)} = \frac{p}{1 + e\cos\nu(t)} \tag{2.8}$$

Where B is the module of a vector constant of integration and ν is the angle between $\vec{B}$ and $\vec{r}$. The constant of integration $\vec{B}$ is defined as a vector pointing from the primary mass to the perigee.

The (2.8) equation is mathematically identical to the general equation of a conic section written in polar coordinates, where p is the conic *semilatus rectum* and ν is the *true anomaly.* Consequently, under the two-body problem assumptions, the orbital trajectory is always a *conic* with a focus located on the center of the primary body. Therefore, the fist Kepler's law is extended to circles, parabolas, hyperbolas and also to degenerate conics, such as one or two straight lines or a single point.

2.2.4 Classical Orbital Elements

Considering a two-body system, given (2.7) and (2.8), to define the satellite trajectory the only parameters needed would be the specific angular momentum vector $\vec{h}$ and the mechanical energy, E. These quantities are provided by a position vector, $\vec{r}$, and by a velocity vector, $\vec{v}$, both consisting of three components each. Thus, six parameters completely identify the orbit and they are collected in a six components vector called *state vector*, X.

The (2.6) combined with the velocity definition as time derivative of the position vector may be integrated to obtain the satellite trajectory. These equations define a first-order differential system which has the six components of X as variables (2.9). Thus, a state vector, consisting of position and velocity vectors, could be considered as a set of initial condition.

$$\begin{cases} \vec{v} = \dot{\vec{r}} \\ \dot{\vec{v}} = -\frac{\mu}{r^3}\vec{r} \end{cases} \tag{2.9}$$

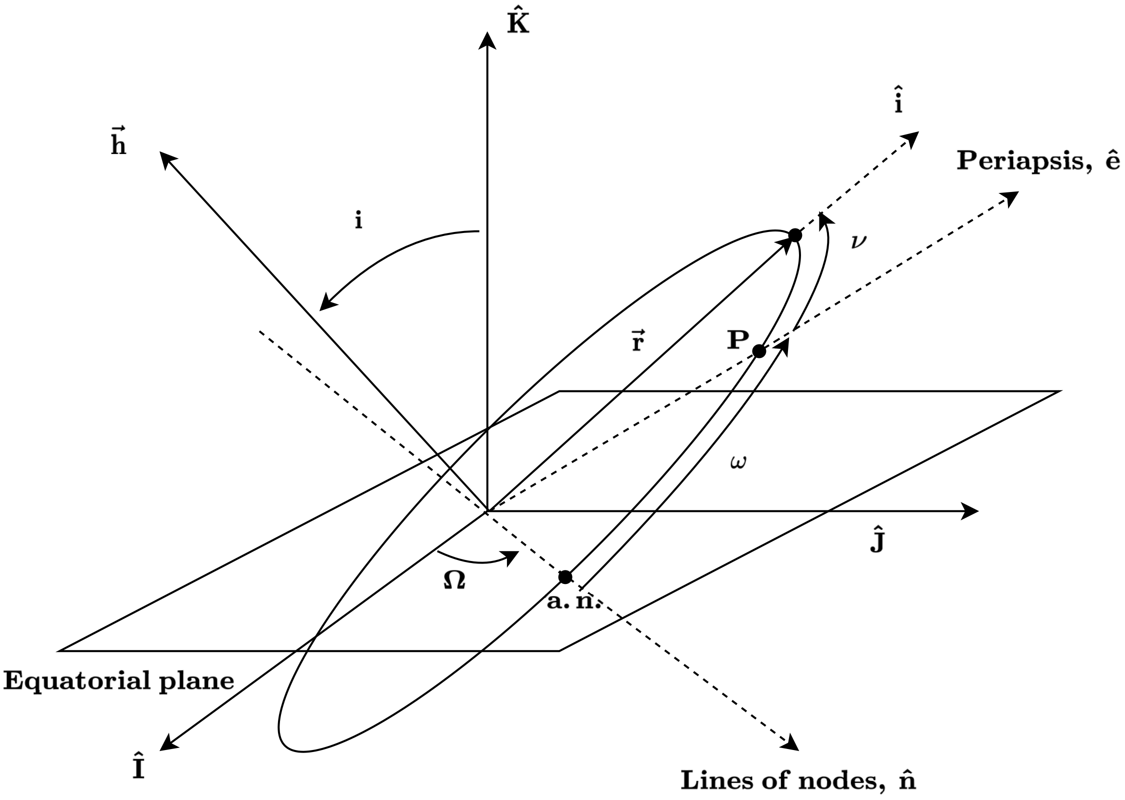

Figure 2.2: Classical Orbital Elements scheme

Defining the orbit through a state vector of position and velocity vectors is not very intuitive. Therefore, the X vector can be written in many equivalent forms, called *element sets*. The most common is the one represented by *classical orbital elements*, also named *Keplerian elements* or *osculating elements*. A scheme of an elliptical orbit is reported in figure 2.2, and some of the orbital elements and parameters needed to define them are shown. The reference system considered is geocentric and not rotating (ECI), its fundamental plane corresponds to the Earth equatorial plane. The reference frame is defined by three unit vectors, $\hat{I}$, $\hat{J}$, $\hat{K}$. Besides the ellipse, the Keplerian elements are generally adequate also for other conics (such as hyperbolas and parabolas). However, some considerations are necessaries after the definitions of the classical orbital elements below.

Semimajor axis. It describes the size of a conic orbit and is indicated by the a symbol. The *semimajor axis* is the semisum of the distances of *perigee* and *apogee* from the orbit focus, the closest point and the furthest respectively. Moreover, it is mathematically related with mechanical energy under two body assumptions. Sometimes it could be replaced by the *semilatus rectum*, p, which defines the width at the primary focus, and its mathematical relation with h is a consequence of (2.8).

$$\begin{aligned} a &= -\frac{\mu}{2E} = \frac{r_p + r_a}{2} \\ p &= \frac{h^2}{\mu} \end{aligned} \tag{2.10}$$

Eccentricity. It defines the shape of a conic orbit and it is indicated by the e symbol. The *eccentricity* is the magnitude of the vector $\vec{e}$, that is defined through the mathematical demonstration leading to the integration of the motion equation (see [12] and [13] for further details):

$$\begin{aligned} \vec{e} &= \frac{\vec{B}}{\mu} = \frac{\vec{v} \times \vec{h}}{\mu} - \frac{\vec{r}}{r} \\ e &= |\vec{e}| \end{aligned} \tag{2.11}$$

Furthermore, the eccentricity vector points always in the same direction of the periapsis, since it is parallel to $\vec{B}$. If mechanical energy and angular momentum are known, e can also be obtained from the geometry relation between semimajor axis and semilatus rectum.

$$e = \sqrt{1 - \frac{p}{a}} \tag{2.12}$$

Semimajor axis and eccentricity ere enough to identify the orbit on the orbital plane (table 2.1). The orientation of the orbit in the orbital plane and of the orbital plane in three-dimensional space are defined by other orbital elements not yet introduced.

Table 2.1: Semimajor Axis and eccentricity influence on the orbit

Mechanical Energy	Semimajor Axis	Eccentricity	Orbit
< 0	> 0	$e = 0$	Circular
< 0	> 0	$0 < e < 1$	Elliptical
$= 0$	∞	$e = 1$	Parabolic
> 0	< 0	$e > 1$	Hyperbolic

Inclination. It is the angle between $\hat{K}$ and the angular momentum, $\vec{h}$; it is indicated with the i symbol. i refers to the inclination of the orbital plane.

$$i = \arccos\left[\hat{K} \cdot \left(\frac{\vec{h}}{h}\right)\right] \tag{2.13}$$

Table 2.2: Inclination

Inclination Values	Orbit Type
$i = 0°$	Equatorial Prograde
$0° < i < 90°$	Inclined Prograde
$i = 90°$	Polar
$90° < i < 180°$	Inclined Retrograde
$i = 180°$	Equatorial Retrograde

Longitude of the ascending node. It is the angle, in the equatorial plane, between $\hat{I}$ and the *ascending node* measured in counterclockwise when viewed from the north side of the equatorial plane. It is indicated by the Ω symbol and also named *right ascension of the ascending node (RAAN)*. The ascending node is the point on the equatorial plane at which the satellite crosses the equator from south to north, vice versa the descending node from north to south. The line connecting these two points is called *line of nodes*, which direction is defined by the $\vec{n}$ vector, pointing to the ascending node.

$$\begin{aligned} \vec{n} &= \hat{K} \times \vec{h} \\ \Omega &= \arccos\left[\hat{I} \cdot \frac{\vec{n}}{n}\right] \end{aligned} \tag{2.14}$$

Table 2.3: Longitude of the Ascending Node

$\vec{n} \cdot \hat{J}$	RAAN Values
$n_y \geq 0$	$0° \leq \Omega \leq 180°$
$n_y < 0$	$180° < \Omega < 360°$

Argument of perigee. It is the angle, in the orbital plane, between the ascending node and the perigee measured in the direction of the satellite's motion. It is indicated by the ω letter.

$$\omega = \arccos\left[\frac{\vec{n}}{n} \cdot \frac{\vec{e}}{e}\right] \tag{2.15}$$

Table 2.4: Argument of Perigee

$\vec{e} \cdot \hat{K}$	Argument of Perigee Values
$e_z \geq 0$	$0° \leq \omega \leq 180°$
$e_z < 0$	$180° < \omega < 360°$

True anomaly. It is the angle, in the orbital plane between the periapsis and the position vector. It was introduced to write the two-body trajectory equation (2.8), as ν. This parameter let to identify the satellite position in the orbit. Given (2.16) and the properties of the arccos function, the true anomaly is not defined unequivocally. To determine if the satellite is between perigee and apogee or vice versa, the *flight path angle* should be introduced (table 2.5), i.e. the angle between position and velocity vectors.

$$\nu = \arccos\left[\frac{\vec{e}}{e} \cdot \frac{\vec{r}}{r}\right] \tag{2.16}$$

Table 2.5: True anomaly

Flight path angle	True anomaly values
$\vec{r} \cdot \vec{v} \geq 0$	$0° \leq \nu \leq 180°$
$\vec{r} \cdot \vec{v} < 0$	$180° < \nu < 360°$

The equations (2.10), (2.11), (2.13), (2.14), (2.15) and (2.16) show that, starting from position and velocity vectors, it is possible to obtain a new set of parameters which defines equally the satellite trajectory. Moreover, under assumptions of subsection 2.2.1, all the orbital elements remain constant over time, except the true anomaly, which represents the time dependence of the motion.

Alternative Parameters

For some geometries, Ω, ω and ν could not be defined. For instance, a perfectly equatorial non-circular orbit has no ascending node, hence, the RAAN angle is undetermined. For this reason, an alternative parameter is introduced, the *true longitude of periapsis*, $\tilde{\omega}_{true}$, which is the angle between the $\hat{I}$ axis and the eccentricity vector, $\vec{e}$.

$$\tilde{\omega}_{true} = \arccos \hat{I} \cdot \frac{\vec{e}}{e} \tag{2.17}$$

Another interesting case is the perfectly circular non-equatorial orbit that has no periapsis. Thus, the true anomaly can not be evaluated and the *argument of latitude*, u, is defined as the angle between the ascending node and the satellite position vector measured in the motion direction.

$$u = \arccos \frac{\vec{n}}{n} \cdot \frac{\vec{r}}{r} \tag{2.18}$$

Lastly, for a circular and equatorial orbit the *true longitude*, λ_{true}, is introduced as the angle measured eastward from $\hat{I}$ axis and the satellite position vector.

$$\lambda_{true} = \arccos \hat{I} \cdot \frac{\vec{r}}{r} \tag{2.19}$$

A check on the sign of the arccos argument is needed for these parameters, similarly to what was done with Ω, ω and ν.

Eccentric and Mean Anomalies

The definition of alternative anomalies can be helpful in addressing certain astrodynamic aspects (such as the Kepler's equation). The figure 2.3 shows the definition of *eccentric anomaly*, E, through an auxiliary circle.

It can be mathematically demonstrated that the eccentric anomaly is related with the true anomaly through the following expression:

$$\cos \nu = \frac{\cos(E) - e}{1 - e\cos(E)} \tag{2.20}$$

The uniform angular motion on a circle of radius a is the *mean anomaly*, M, and it is defined by the relation below:

$$M = n(t - T) \tag{2.21}$$

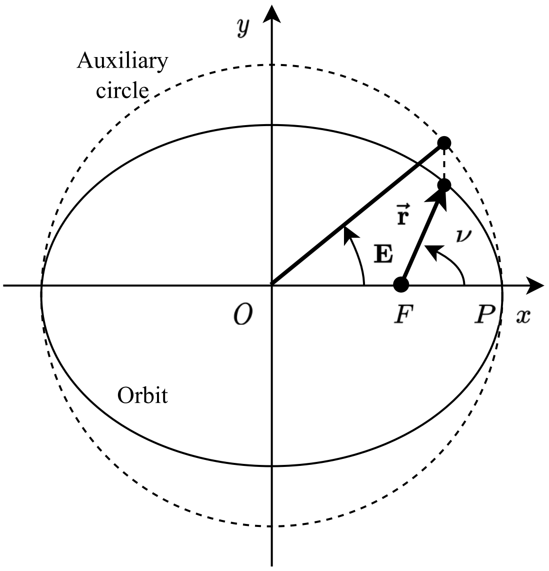

Figure 2.3: Eccentric and True Anomalies

where $n = \sqrt{\mu/a^3}$ is the *mean angular rate*, $(t - T)$ is the elapsed time since the last passage over the perigee. The mean anomaly is also related with the eccentric anomaly trough the Kepler's equation.

$$M = \sqrt{\frac{\mu}{a^3}}(t - T) = E - e\sin(E) \tag{2.22}$$

2.3 Perturbations

The problem addressed in this thesis does not fully respect the assumptions of the two-body problem, an idealized solution which only approximates the observed motion. Indeed, perturbations of other bodies (such as Sun, Moon and other planets) and additional forces not included in Kepler motion influence the trajectory by deviating from the ideal motion. The deviation may be larger or smaller depending on the case under consideration, for instance the atmospheric drag during reentry is comparable to the gravitational force of the two-body problem. Initially, three main approaches to examine the effects of perturbations on the satellite trajectory are introduced. Then, each source of perturbation is treated singularly, describing its physics and the variables behind it. The equations of forces that most influence the motion of the case study satellites will be discussed in more detail in the next chapter. The main reference for this section is [13].

2.3.1 Perturbation Techniques

The *perturbation techniques* are divided in three main classes: numerical, analytical and semianalytical. The analytical one is that developed first since the other two approaches require a computing power that did not yet exist at that time. Below is a brief description of each category.

Special Perturbation (or Numerical) Techniques integrate, numerically, the equation of motion including all necessary perturbing accelerations. Several formulation of the motion equation can be considered, such as Encke's (2.23) and Cowell's (2.24).

The fist method integrates the difference between the two-body acceleration (2.6) and the perturbed acceleration ($\ddot{\vec{r}}_P = -\frac{\mu}{r_P^3}\vec{r}_P + \vec{a}_P$), after some manipulations the following equation is obtain:

$$\delta\ddot{\vec{r}} = \vec{a}_P + \frac{\mu}{r^3}\left[\left(1 - \frac{r^3}{r_P^3}\right)\vec{r}_P - \delta\vec{r}\right] \tag{2.23}$$

$\delta\vec{r}$ is the difference between the two body position $\vec{r}$ and the perturbed position $\vec{r}_P$, while $\vec{a}_P$ is the sum of all perturbing accelerations. Nowadays Encke's formulation is not very popular since computers can integrate numerically without making any particular changes to the equations.

As computer become faster Cowell's formulation is usually preferred.

$$\begin{aligned} \vec{v} &= \dot{\vec{r}} \\ \dot{\vec{v}} &= -\frac{\mu}{r^3} + \vec{a}_P \end{aligned} \tag{2.24}$$

$\vec{r}$ is the satellite position, $\vec{v}$ is the satellite velocity with respect to an inertial reference frame and $\vec{a}_P$, as before, is the sum of all perturbing accelerations considered.

These methods are really accurate but are not without any difficulties. Such as: selecting the most appropriate integration technique, the long computing time required or obtaining the necessary data needed to evaluate the perturbing accelerations. Lastly, numerical techniques degrades over time since truncation and round-off errors due to fixed computer wordlength.

General Perturbation (or Analytical) Techniques allow to examine the effect of perturbations on the orbital elements instead of position and velocity vectors unlike previous techniques. This is achieved replacing the motion equation with analytical approximations. These formulations enable to integrate analytically truncating the series expansions of the perturbing accelerations.

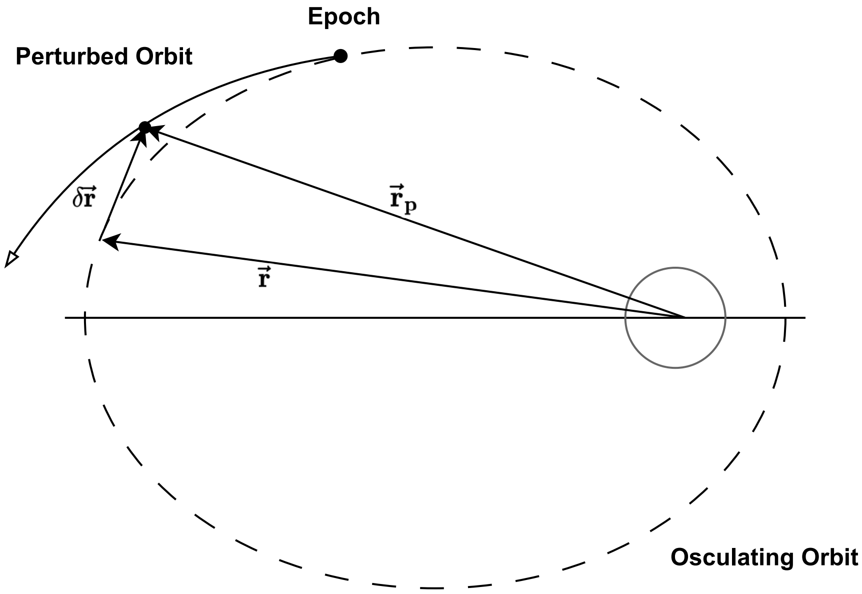

Figure 2.4: Encke's method [13]

General perturbations theories relay on some analytical formulations, i.e. the *Lagrange (and Gauss) planetary equations of motion* and the *Hamilton's formulation.* These approaches are more difficult to develop than numerical techniques, less accurate and the quality of solution degrades over time. Nevertheless, they let to better understand the perturbation source, they reduce the computing time and the results are "general", which means that any initial condition is accepted.

Semianalytical Techniques , as the name suggests, are an attempt to combine the high accuracy of numerical solutions with the high simulation speed of analytical formulations.

None of these approaches is the most suitable in every study case; hence, the choice usually depends on the problem faced.

The perturbations briefly described in the next subsections are: central-body and third-body gravitational effects, atmospheric drag and solar-radiation pressure. The first two are due to gravity, a conservative force. The total energy of the system is constant and the acceleration could be determined as the gradient of a *potential function.* Whereas the last two are non-conservative effects. A potential function is a way to express a conservative force, and, depending on the sign convention, may or may not coincide with energy potential. In this case, the potential function ($U = \mu/r$) is the negative of the potential energy ($U_g = -\mu/r$).

2.3.2 Gravity field of a Central Body

One of the assumptions of the two-body problem is that the bodies are spherically symmetric and can therefore be treated as point masses. This allows to evaluate quite easily the potential function of the attracting body :

$$U_{2-body} = \frac{\mu}{r} \tag{2.25}$$

The gradient of (2.25) provides the attractive acceleration:

$$\nabla U_{2-body} = \ddot{\vec{r}}_{2-body} \tag{2.26}$$

If the gradient is explicated, the equation of motion for the two-body problem (2.6) is obtained.

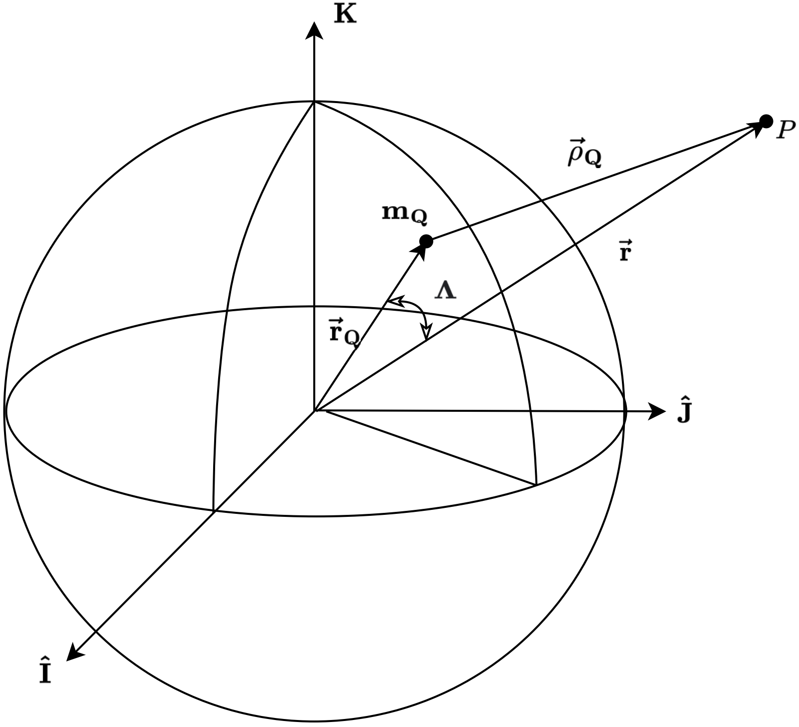

Figure 2.5: Scheme to derive the Gravitational Potential

In reality, the Earth is not exactly spherical and hence an aspherical-potential function should be evaluated. To begin with, an infinitesimal element of Earth mass $dm_{\oplus}$, at point Q, is considered. The change in potential, due to $dm_{\oplus}$, at point P, which is at a distance ρ_Q from Q, is:

$$dU = G\frac{dm_{\oplus}}{\rho_Q} \tag{2.27}$$

Let r be the geocentric distance at point P, r_Q the distance between the centre of the Earth and $dm_\oplus$, and let Λ be the angle between r and r_Q. The law of cosines is used to explicate ρ_Q as a function of r, r_Q and Λ; then, dU is integrated over the entire body:

$$U = G \int_{body} \frac{dm_\oplus}{r\sqrt{1 - 2\frac{r_Q}{r}\cos\Lambda + \left(\frac{r_Q}{r}\right)^2}} \tag{2.28}$$

Through a series of *Legendre polynomials*, P_l, the denominator is expanded:

$$U = \frac{G}{r}\int_{body} \sum_{l=0}^{\infty} \left(\frac{r_Q}{r}\right)^l P_l[\cos\Lambda]\, dm_\oplus \tag{2.29}$$

where:

$$P_l[\gamma] = \frac{1}{2^l l!}\frac{d^l\left(\gamma^2 - 1\right)^l}{d\gamma^l} \tag{2.30}$$

A geometric approach is followed using the cosine law of spherical trigonometry to relate Λ with longitudes and latitudes of P and Q. The *associated Legendre functions* are introduced:

$$P_{l,m}[\gamma] = \frac{1}{2^l l!}\left(1 - \gamma^2\right)^{m/2}\frac{d^{l+m}}{d\gamma^{l+m}}\left(\gamma^2 - 1\right)^l \tag{2.31}$$

where l and m are *degree* and *order*, respectively. Then, through the *addition theorem of spherical harmonics* and other steps, reported in [13], the aspherical-potential expression is obtained:

$$\begin{aligned} U = \frac{\mu}{r}\Bigg[1 - \sum_{l=2}^{\infty} J_l\left(\frac{R_\oplus}{r}\right)^l P_l\left[\sin(\phi_{gc_{sat}})\right] \\ + \sum_{l=2}^{\infty}\sum_{m=1}^{l}\left(\frac{R_\oplus}{r}\right)^l P_{l,m}\left[\sin(\phi_{gc_{sat}})\right]\{C_{l,m}\cos(m\lambda_{sat}) + S_{l,m}\sin(m\lambda_{sat})\}\Bigg] \end{aligned} \tag{2.32}$$

C and S are coefficient empirically determined from observations and are strictly related to the mathematical modeling for the Earth's shape using spherical harmonics. $R_\oplus$ is the Earth radius. $\phi_{gc_{sat}}$ and λ_{sat} are satellite latitude and longitude, respectively. Lastly, J_l coefficient is the negative of the C notation for *zonal harmonics*, namely:

$$J_l = -C_{l,0} \tag{2.33}$$

The trigonometric argument of the Legendre polynomials in (2.32) constitutes surface spherical harmonics, since they are periodic on the surface of a unit sphere.

As (2.33) shows, the *zonal harmonics* are defined by zeroth order (m=0) and represents bands of latitude. For instance, considering the second degree (l=2): there are two circles, made by points with same latitudes, along which P_2 is zero.

Thus, these two circles identify three zones on the unit sphere, in which the function is alternately increasing and decreasing. Among the zonal harmonics, the most influential coefficient is J_2. In fact, it is 1000 times larger than the next largest coefficient and represents the most of the Earth's oblateness. *Sectorial harmonics* occur when $l{=}m$ and represents bands of longitude. While *tesseral harmonics* represent cases in which $l \neq m \neq 0$.

2.3.3 Third-Body Perturbations

In the two-body problem the first assumption (subsection 2.2.1) consists in identify the two bodies. Typically the Earth-satellite system may be considered, neglecting the gravitational influence of other bodies on them, such as the Sun or the Moon. This effect can be significant or negligible depending on the satellite altitude (as in section 3.2 will be discussed).

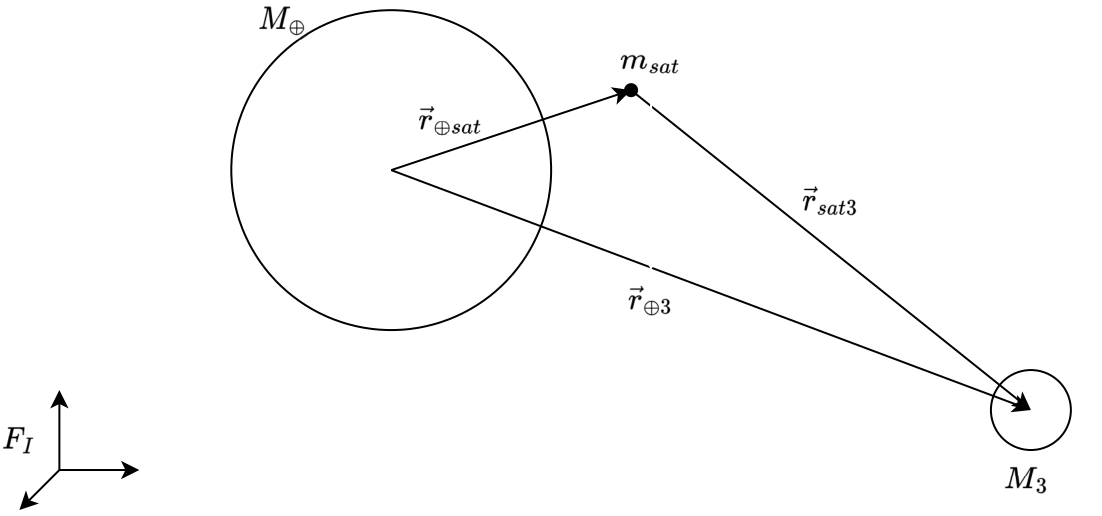

Figure 2.6: Third-body influence

The acceleration induced on an object orbiting around the Earth by a third body is evaluated developing the motion equation for a three-body system (2.34). The expression is obtained writing the equations of dynamics (2.4) for both the Earth and the satellite with respect to an inertial frame. Besides considering the two-body gravitational attraction, the attraction of the third body (figure 2.6) is also taken into account in both equations. Then, the two equations are subtracted, assuming the satellite mass negligible compared to that of the Earth and of the third body.

$$\ddot{\vec{r}}_{\oplus sat} = -\frac{\mu_\oplus \vec{r}_{\oplus sat}}{r^3_{\oplus sat}} + \mu_3 \left(\frac{\vec{r}_{sat3}}{r^3_{sat3}} - \frac{\vec{r}_{\oplus 3}}{r^3_{\oplus 3}} \right) \tag{2.34}$$

$\vec{r}_{\oplus sat}$ is the satellite position with respect to the Earth, $\vec{r}_{sat3}$ is the the third body position with respect to satellite, $\vec{r}_{\oplus 3}$ is the third body position with respect to the Earth, and lastly, μ_3 is the third body gravitational parameter.

The first term is that of the motion equation in the two-body problem, hence, the second term represents the perturbing acceleration due to the third body. Interestingly, besides the third-body attraction to the satellite itself, the perturbation is also caused by the third-body attraction to the Earth.

2.3.4 Atmospheric Drag

In LEO one of the main perturbations on satellite trajectory is determined by air particles which are not negligible at 400 km altitude and hinder the motion. This phenomenon is called *atmospheric drag* and it induces an acceleration on an orbiting body. The drag acceleration has a simple mathematical formulation.

$$\vec{a}_{drag} = -\frac{1}{2}\frac{C_D S}{m}\rho v_{rel}^2 \frac{\vec{v}_{rel}}{\vec{v}_{rel}} \tag{2.35}$$

C_D is the *coefficient of drag*, it depends by several factors, such as the body shape and the way air molecules collide with it; a typical C_D value is 2.2.

The *cross-sectional area*,indicated by S, is the satellite area normal to the relative velocity vector, $\vec{v}_{rel}$. The spacecraft attitude must be known accurately to estimate S. Therefore, if the satellite randomly tumbles, it may be very difficult to calculate the cross-section.

m is the satellite *mass*, if the spacecraft is not equipped with a propulsion system the mass can usually be assumed constant. The ratio of mass to cross-sectional area times the drag coefficient is named *ballistic coefficient*, BC (2.36). The lower is BC, the greater will be the effect of atmospheric drag on the spacecraft orbit.

$$BC = \frac{m}{SC_D} \tag{2.36}$$

The last two parameters of (2.35) are *relatively velocity*, $\vec{v}_{rel}$, and *atmospheric density*, ρ. $\vec{v}_{rel}$ is relative to the atmosphere. In fact, the air particles move due the friction with the rotating Earth and due to the winds. The latter present a great uncertainty, hence, they are not usually considered. The following expression can be used to account the atmosphere motion due to the Earth's rotation, $\omega_\oplus$:

$$\vec{v}_{rel} = \frac{d\vec{r}}{dt} - \vec{\omega}_\oplus \times \vec{r} \tag{2.37}$$

The minus sign in (2.35) indicates that the force acts along the opposite direction to that of the relative velocity.

ρ is probably the parameter presenting the greatest uncertainty, the causes are dealt with forward.

In conclusion, although the drag acceleration expression is definitely simpler than that obtained to evaluate the potential of an aspherical body (2.32), all the parameters within the (2.35) formula are affected by a great uncertainty [13] [14].

Density Evaluation

Besides altitude, the density values in the upper atmosphere are mainly influenced by three factors which interacts each other:

Molecular composition, i.e. what gases the atmosphere is composed of, their molecular weight and whether they are electrically charged or not.

Solar flux, its main influence on the upper atmosphere density is due to the incoming solar radiation, in particular to the *Extreme Ultra-Violet*, EUV. This kind of radiation can not reach the Earth's surface due to the presence of the atmosphere. Thus, the EUV radiation flux, F_{EUV} can not be detected from the ground. However, the incoming solar radiation with a 10.7 cm wavelength, which originates in the same layers of the Sun's chromosphere and corona as the EUV, can be measured from the Earth's surface. Consequently, flux measurements of the 10.7 cm wavelength radiation, $F_{10.7}$, allow to deduce those of F_{EUV} without requiring any instrument outside of the atmosphere. The solar flux is measured in Solar Flux Unit, SFU, and $F_{10.7}$ values are usually between 70 SFU and 300 SFU.

$$1\,SFU = 10^{-22}\,\frac{W}{m^2 Hz} \tag{2.38}$$

The solar activity has 11-year cycle that follows the Sun-spot cycle and influence the amount of incoming solar radiation on Earth (figure 2.8).

Geomagnetic activity, indeed, the Earth's magnetic field presents variations. These influence density since electrically charged air particles interact with the geomagnetic field. The geomagnetic activity can be measured trough the *geomagnetic planetary index*, k_p, a value averaged over twelve stations around the world and detected every three hours. It is a quasi-logarithmic index ranging from 0, low activity, to 9, extreme activity. The k_p linear equivalent is the *geomagnetic planetary amplitude*, a_p (2.39), which is measured every three hours, like k_p. The average daily value of a_p is A_p and its range is from 0 to 400, although values usually do not exceed 100.

$$a_p = \exp\frac{k_p + 1.6}{1.75} \tag{2.39}$$

The geomagnetic planetary amplitude is measured in *gammas*:

$$1\,gamma = 10^{-9}\,T = 10^{-9}\,\frac{kg\,s}{m} \tag{2.40}$$

As the solar activity, the geomagnetic activity has a 11-year cycle, but it is not the only periodic variation. In fact, the A_p values are also influenced by the solar wind position relative to the Earth's magnetosphere, which has a semi-annual cycle.

Generally, these three effects influence upper atmosphere density by increasing or decreasing the number of particle collisions. Figure 2.7 and figure 2.8 show A_p and $F_{10.9}$ past observations and the periodicity of this parameters. Although the several observations in the recent years, it is really complicate to evaluate accurately ρ. In fact, there is no atmospheric model which is best for all applications since none of them is able to remove all errors that may occur.

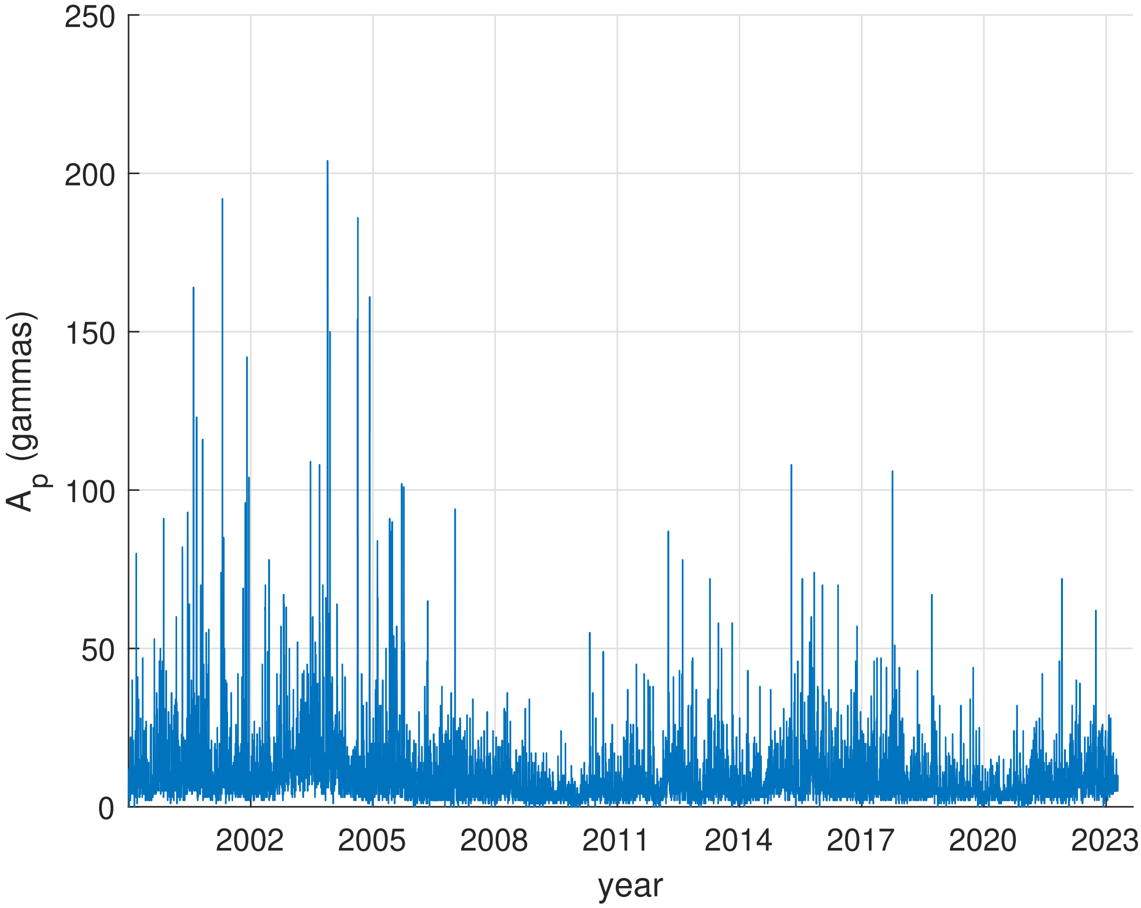

Figure 2.7: Geomagnetic Planetary Amplitude variations

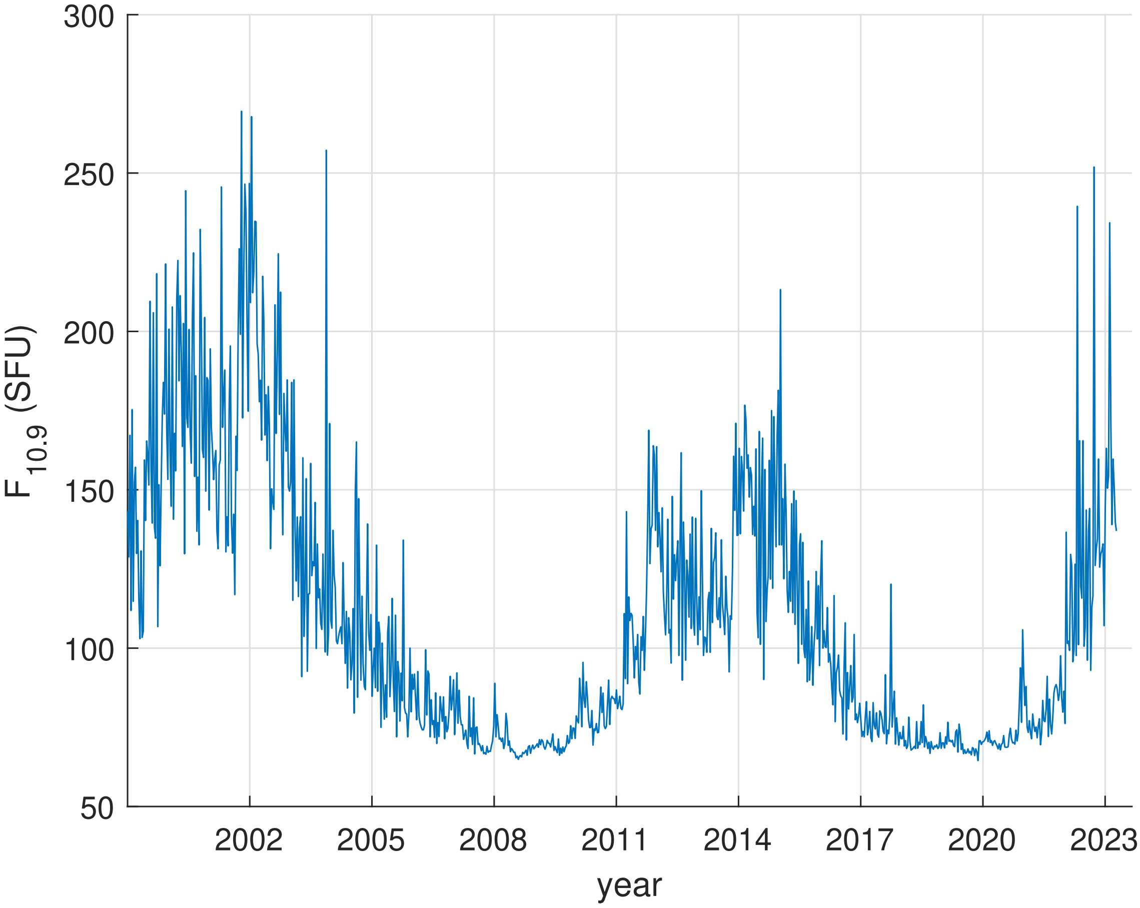

Figure 2.8: Solar flux variations

Model Atmospheres

Many atmospheric models have been developed to achieve accurate density values, but one of the main obstacles is the difficulty in predicting $F_{10.9}$ and A_p.

Some of them try to bypass this uncertainty by following a *static* approach, achieving great simplicity but at the same time being very imprecise. These models may also take into account some variations such as latitudinal and longitudinal. The former correspond to a change in altitudes at different latitudes due to the asphericity of the Earth for circular inclined orbits. The latter considers changes in density, temperature and wind due to the region (e.g. ocean, mountains etc.) over which the satellite is orbiting. Following, some static approach models are briefly described:

Exponential Model . It ranges from 0 to 1000 km, and it is based on two assumptions: the density decreases exponentially with increasing altitude and there is a spherical distribution of particles.

$$\rho = \rho_0 \exp\left[- \frac{h_{ellp} - h_0}{H} \right] \tag{2.41}$$

where ρ_0 is a reference density, h_0 is the reference altitude, H is the scale height and h_{ellp} is the altitude above the ellipsoid, indeed the Earth is not perfectly spherical. Some values are reported in table 2.6.

Standard Atmosphere . It was released by the U.S. Committee on Extensions to the Standard Atmosphere (COESA). As the Exponential Model, it ranges from 0 to 1000 km altitude. The Standard Atmosphere is a rough representation of year-round and midlatitude (45°) conditions. It is assumed that the air obey to the perfect gas law and the hydrostatic equation. Furthermore, the atmosphere is considered to rotate with the Earth and its values are an average over the diurnal cycle, semi-annual variation, cycle of Sun spots and cycle of geomagnetic activity [15].

Neither of them are suitable for the purpose of this work since they neglect many real world effects. However, at the beginning of writing the code, their simplicity, particularly that of the *Exponential Model*, was helpful in providing an easily derived input.

Table 2.6: Exponential Atmospheric Model

h_{ellp}	h_0	ρ_0	H	h_{ellp}	h_0	ρ_0	H
(km)	(km)	(kg/m^3)	(km)	(km)	(km)	(kg/m^3)	(km)
0-25	0	1.225	7.249	150-180	150	2.070x10^{-9}	22.523
25-30	25	3.899×10^{-2}	6.349	180-200	180	5.464×10^{-10}	29.740
30-40	30	1.774×10^{-2}	6.682	200-250	200	2.789×10^{-10}	37.105
40-50	40	3.972×10^{-3}	7.554	250-300	250	7.248×10^{-11}	45.546
50-60	50	1.057×10^{-3}	8.382	300-350	300	2.418×10^{-11}	53.628
60-70	60	3.206×10^{-4}	7.714	350-400	350	9.158×10^{-12}	53.298
70-80	70	8.770×10^{-5}	6.549	400-450	400	3.725×10^{-12}	58.515
80-90	80	1.905×10^{-5}	5.799	450-500	450	1.585×10^{-12}	60.828
90-100	90	3.396×10^{-6}	5.382	500-600	500	6.967×10^{-13}	63.822
100-110	100	5.297×10^{-7}	5.877	600-700	600	1.454×10^{-13}	71.835
110-120	110	9.661×10^{-8}	7.263	700-800	700	3.614×10^{-14}	88.667
120-130	120	2.438×10^{-8}	9.473	800-900	800	1.170×10^{-14}	124.64
130-140	130	8.484×10^{-9}	12.636	900-1000	900	5.245×10^{-15}	181.05
140-150	140	3.845×10^{-9}	16.149	1000-	1000	3.019×10^{-15}	268.00

To achieve better accuracy, *time-varying models* are required, but at the cost of greatly increasing the complexity. In fact, they may consider several effects such as: density variations due to the local time, namely *diurnal variations*, *semi-annual variations*, related to the distance of the Earth from the Sun, *solar activity cycle*, and many others. Some of the best known models are briefly discussed below.

Jacchia-Roberts Atmospheric Model . Jacchia divided the upper atmosphere in two regions: one from 90 km to 125 km altitude and the other above 125 km. This choice was due by assuming that mixing is predominant between 90 km and 100 km, and since diffusion equilibrium was assumed between 100 km and 125 km. An empirical profile of temperature, as a function of altitude and exospheric temperature, is defined for each region. Temperature values are substituted in the barometric and diffusion equations [16], to obtain the density. These differential equations are numerically integrated over the altitude range (90 km - 2000 km) for various constant values of exospheric temperature. Fixed boundary conditions are assumed at the lower altitude limit. In this way the solar activity, the geomagnetic activity, semiannual variations, seasonal variations and diurnal variation are considered. The numerical results are tabulated. Although the densities are accurate, these tables are not easy to compute in mechanizations since they require big storage, resulting in slow running. For these reasons, Roberts developed analytical expressions that made possible implementing more easily the Jacchia model in computing machines. For more details, see [16] [17] [18].

NRLMSISE-00 . It was developed by the *United States Naval Reasearch Laboratory*, or NRL. MSIS stands for *Mass Spectrometer and Incoherent Scatter Radar*, which are the two main data sources. The *E* indicates that the model extends from the ground to space, whereas previous versions covered only the upper atmosphere. *00* is reported in the name since the model was published in the 2000. NRLMSISE-00 is an improvement of the early MSIS versions since, in addition to mass spectrometer and radar measurements, it assimilates density values determined from drag on satellites and other orbiting objects. As Jacchia-Roberts, it accounts time-varying effects such as solar and geomagnetic activity, but it appears to provide advantages on the Jacchia model for estimating total mass density [19].

In conclusion, this subsection shows the great complexity in determining the atmospheric drag due to the uncertainty of its parameters. Particularly, the density is influenced by many variables that are difficult to predict.

2.3.5 Solar-Radiation Pressure

The solar-radiation could be impacting or negligible depending on several variables, such as the solar activity and the altitude. The incoming radiation from the Sun has a mean intensity, called *solar flux*, of 1353 W/m^2. When the radiation hits an exposed surface of a satellite, it exchanges momentum with it. The change of momentum is expressed through a *solar pressure* value:

$$p_{SR} = 4.51 \times 10^{-6} \frac{N}{m^2} \tag{2.42}$$

Dimensionally, the intensity the solar-radiation force on the satellite is defined as the product between the solar pressure and the surface exposed to the Sun ($A_\odot$). This section is not the same used for aerodynamic drag calculations. Following, it is reported a vectorial expression of the acceleration due the solar-radiation which takes into account the reflectivity (c_R) of the satellite:

$$\vec{a}_{SR} = \frac{p_{SR} c_R A_\odot}{m_{sat}} \frac{\vec{r}_{\odot sat}}{|\vec{r}_{\odot sat}|} \tag{2.43}$$

$\vec{r}_{\odot sat}$ is the satellite position vector with respect to the Sun and it identifies the direction along which the force acts. c_R is a value between 0.0 and 2.0, which indicates how the satellite reflects the incoming radiation. A value of 0.0 means that the object is translucent, hence the light pass through it without any absorption and reflection. 1.0 means that the radiation is all absorbed and 2.0 indicates that the radiation is totally reflected. For further details see [13].

Chapter 3

Assumptions and Methods

Once that the theoretical aspect has been explained, the following step is to establish the method for pursuit the work objective. Also, it is necessary to indicate the assumptions and justify them, e.g. which are the perturbation considered and why. Thus, this chapter still includes theory but, at the same time, assessments are carried out to understand how the code was developed.

3.1 Choice of Approach

As was briefly described in subsection 2.3.1, the main methods to evaluate the effects of perturbation on satellite motion are: *special perturbation (numerical)*, *general perturbation (analytical)* and *semianalytical techniques*.

The purpose of this study is understanding which perturbation source influences the satellite trajectory and showing the results in an intuitive way. Therefore, analytical techniques should be more appropriate. Instead, numerical techniques are not considered suitable because their output is usually a set of positions and velocities. Moreover, although special techniques could achieve a great accuracy, the result would be affected by the uncertainty behind the input data required to define perturbing accelerations. Lastly, the three-dimensional numerical integration of motion equation takes a long time to analyse long-duration orbit lifetime cases [14].

3.1.1 Variation of Parameters

General perturbation techniques usually relay on the *variation of parameters* (VOP) form of the equations of motion, developed by Euler and improved by Lagrange. In fact, unlike the two-body problem, the orbital elements are not constant when the system is perturbed. Thus, these elements are more properly *osculating elements* associated with an *osculating orbit* at a particular instant of time. VOP equations describe their rates of change through a set of first-order differential equations:

$$\frac{d\vec{c}}{dt} = f(\vec{c}, t) \tag{3.1}$$

Parameters can change over time in two main ways: *secular* and *periodic*. Secular variations represent a linear change over time, or proportional to a power of time. Periodic variations, on the other hand, are due to effects that recur over a period of time, which could be short (order of the satellite period or less) or long (considerably longer than one period).

The next step is to make explicit the relationship between the rates of change of the orbital elements and perturbing acceleration. The position vector is a function of six parameters (state vector), such as orbital elements, and time:

$$\vec{r} = \vec{x}(\vec{c}, t) = \vec{x}(c_1, c_2, c_3, c_4, c_5, c_6, t) = \vec{x}(a, e, i, \Omega, \omega, M, t) \tag{3.2}$$

The velocity, hence the derivative over time of (3.2), is expressed in a perturbed case as follow:

$$\vec{\dot{r}} = \vec{\dot{x}}(\vec{c}, t) = \frac{d\vec{x}(\vec{c}, t)}{dt} = \frac{\partial\vec{x}(\vec{c}, t)}{\partial t} + \sum_{i=1}^{6} \frac{\partial\vec{x}(\vec{c}, t)}{\partial c_i}\frac{dc_i}{dt} \tag{3.3}$$

In an unperturbed system the second term of the right member would be zero since in that case the orbital element rates of change are zero ($dc_i/dt = 0$). To guarantee that each position and velocity vector defines an osculating ellipse, the *condition of osculation* is imposed:

$$\sum_{i=1}^{6} \frac{\partial\vec{x}(\vec{c}, t)}{\partial c_i}\frac{dc_i}{dt} \equiv \vec{0} \tag{3.4}$$

Substituting this expression in (3.3), the velocity is the same for an unperturbed system and for one in which the osculation condition is imposed. However, it should be noted that the latter implies only that the sum of the rates of change is zero, and not each term.

After that, the velocity is derived over time, to obtain the acceleration, and the result is substituted in the motion equation of a perturbed system (such as the second equation of (2.24)). Lastly, the result is set up with the condition of osculation to have a system of six equations in the six element rates:

$$\begin{bmatrix} \sum_{i=1}^{6} \frac{\partial\vec{x}(\vec{c}, t)}{\partial c_i}\frac{dc_i}{dt} \\ \sum_{i=1}^{6} \frac{\partial\dot{\vec{x}}(\vec{c}, t)}{\partial c_i}\frac{dc_i}{dt} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{a}_{pert} \end{bmatrix} \tag{3.5}$$

This expression does not have the same form of (3.1) and needs further development to make computations easier.

The main VOP forms are the *Lagrangian* and the *Gaussian*. The former is only suited for conservative effects, such as gravity field of an aspherical central-body and third-body gravitational attraction. The Gaussian VOP, on the other hand, works for both conservative and non conservative effects. For further information about VOP equations, see [13].

3.1.2 Lagrangian Variation of Parameters

Lagrange was the first to provide equations in the (3.1) form for all six orbital elements. The development of these mathematical relations lays on considering the perturbing acceleration of (3.5) as the gradient of the *perturbing potential function*, R. This explains why the Lagrangian form is only appropriate for conservative effects. In fact, it is not possible to define a potential function, U, for non-conservative effects.

$$R = U - U_{2-body} \tag{3.6}$$

After substituting ∇R in (3.5), the equation is further developed (details reported in [13]) and the relations for the variation of parameters, also named *Lagrangian planetary equations*, are obtained:

$$\begin{aligned}
\frac{da}{dt} &= \frac{2}{na}\frac{\partial R}{\partial M_0} \\
\frac{de}{dt} &= \frac{1-e^2}{na^2 e}\frac{\partial R}{\partial M_0} - \frac{\sqrt{1-e^2}}{na^2 e}\frac{\partial R}{\partial \omega} \\
\frac{di}{dt} &= \frac{1}{na^2\sqrt{1-e^2}\sin{(i)}}\left\{\cos{(i)}\frac{\partial R}{\partial \omega} - \frac{\partial R}{\partial \Omega}\right\} \\
\frac{d\omega}{dt} &= \frac{\sqrt{1-e^2}}{na^2 e}\frac{\partial R}{\partial e} - \frac{\cot{(i)}}{na^2\sqrt{1-e^2}}\frac{\partial R}{\partial i} \\
\frac{d\Omega}{dt} &= \frac{1}{na^2\sqrt{1-e^2}\sin{(i)}}\frac{\partial R}{\partial i} \\
\frac{dM_0}{dt} &= \frac{1-e^2}{na^2 e}\frac{\partial R}{\partial e} - \frac{2}{na}\frac{\partial R}{\partial a}
\end{aligned} \tag{3.7}$$

The derivatives of R with respect to the classical orbital elements appear in the (3.7) equations. Thus, it would be more appropriate to express the perturbing potential as a function of these parameters .

3.1.3 Gaussian Variation of Parameters

The main feature of Gaussian VOP is that the orbital element rates are explicitly expressed in terms of specific forces. Therefore this approach is suitable for both conservative and non-conservative forces.

To obtain a general expression: the (3.5) is considered and manipulated (see [13]):

$$\frac{dc_j}{dt} = \frac{\partial c_j}{\partial \dot{\vec{x}}} \cdot \vec{a}_{pert} \tag{3.8}$$

It should be highlighted that the partial derivative with respect to the velocity and the perturbing acceleration must be in the same frame, i.e. the RSW system (figure 3.1) which is defined by the following right-hand set of axis:

$\hat{\mathbf{R}}$: along the radius vector.

$\hat{\mathbf{S}}$: orthogonal to $\hat{R}$, lying on the orbital plane in the motion direction.

$\hat{\mathbf{W}}$: perpendicular to orbital plane and its direction coincides with that of the angular momentum vector.

RSW system is not fixed and its axis rotate because of the satellite motion. As a starting point, the specific forces, position and velocity should be written in this coordinate frame.

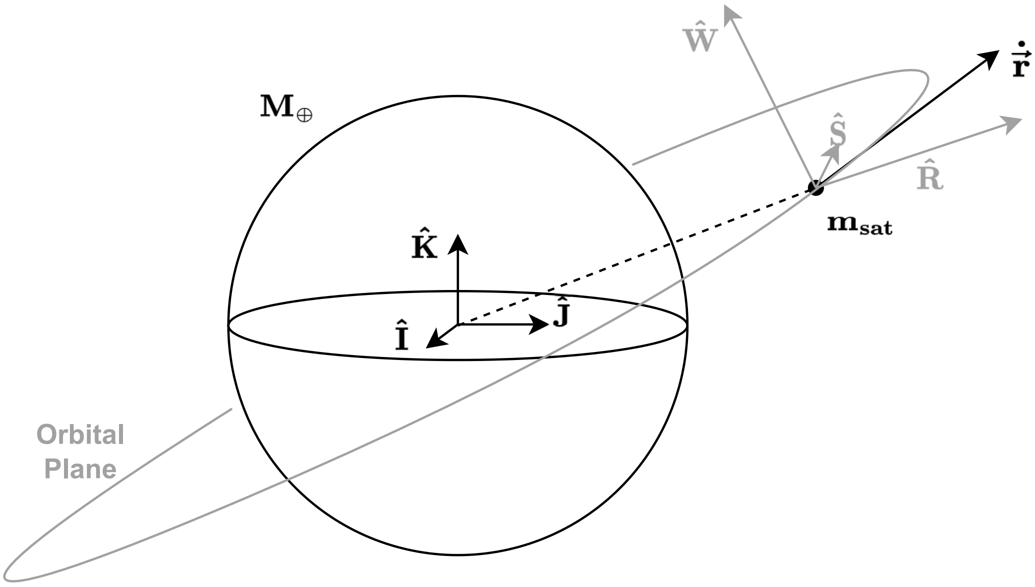

Figure 3.1: RSW coordinate system

The (3.8) is still too general, whereby each orbital element variation is treated singularly to obtain a more explicit expression relating the rates of change with specific perturbing forces.

$$
\begin{aligned}
\frac{da}{dt} &= \frac{2}{n\sqrt{1-e^2}}\left\{e\sin(\nu)F_R + \frac{p}{r}F_S\right\}\\
\frac{de}{dt} &= \frac{\sqrt{1-e^2}}{na}\left\{\sin(\nu)F_R + \left[\cos(\nu) + \frac{e+\cos(\nu)}{1+e\cos(\nu)}F_S\right]\right\}\\
\frac{di}{dt} &= \frac{r\cos(u)}{na^2\sqrt{1-e^2}}F_W\\
\frac{d\Omega}{dt} &= \frac{r\sin(u)}{na^2\sqrt{1-e^2}\sin i}F_W\\
\frac{d\omega}{dt} &= \frac{\sqrt{1-e^2}}{nae}\left\{-\cos(\nu)F_R + \sin(\nu)\left(1+\frac{r}{p}\right)F_S\right\} - \frac{r\cot(i)\sin(u)}{h}F_W\\
\frac{dM_0}{dt} &= \frac{1}{na^2e}\left\{(p\cos(\nu) - 2er)F_R - (p+r)\sin(\nu)F_S\right\}
\end{aligned}
\tag{3.9}
$$

It should be underlined that F_R, F_S and F_W are not forces but specific forces, hence accelerations.

The Gaussian VOP are limited mainly to elliptical orbits due to the $\sqrt{1-e^2}$ term in almost all expressions. This term imposes the condition that the eccentricity values must be less than 1. Furthermore, the presence of e and $\sin(i)$ terms in the denominator of some equations identifies singularities for circular or equatorial orbits.

The (3.9) are usually further developed to study singularly the effect of each perturbation. In fact, the perturbing acceleration are substituted and the expression may be manipulated and expressed as power series. Exploiting power series to express rates of change is an entirely analytical approach. However, this method was not followed in this study. Indeed, the solution of (3.9) was obtained through numerical integration, consequently the approach was semianalytical. The choice was motivated by the desire not to lose much accuracy through further analytical development. Also, the code is executed with the possibility of using some computing power to integrate numerically (3.9).

3.2 LEO Perturbations

After defining the VOP form of motion equation, the order of magnitude for each perturbation is determined, in order to identify which perturbing force influences mostly the satellite trajectory, between those listed in section 2.3. As far as the gravitational effect of a central-body is concerned, all periodic effects (tesseral and sectorial) are ignored and it is considered only the J2 term (see subsection 2.3.2).

Two cases have been evaluated: one with a satellite mass of 1 kg and one with 10 kg, both with a 0.01 m^2 cross-section (drag area and area exposed to the Sun). As figures 3.2 and 3.3 show, the atmospheric drag decreases with the altitude because of the density trend, whereas the order of magnitude of other effects is about constant over the altitude range represented.

In both cases the force with the higher order of magnitude is the J2 effect for almost all the altitudes, except for those below about 150 km, where atmospheric drag becomes predominant. The main difference between the two cases is that, for the same cross-section area and altitude, the heavier satellite present more intense perturbing effect than the lighter one, except for drag. Thus, for a 1 kg satellite the atmospheric drag becomes negligible with respect to the other effects at higher altitude than the 10 kg satellite.

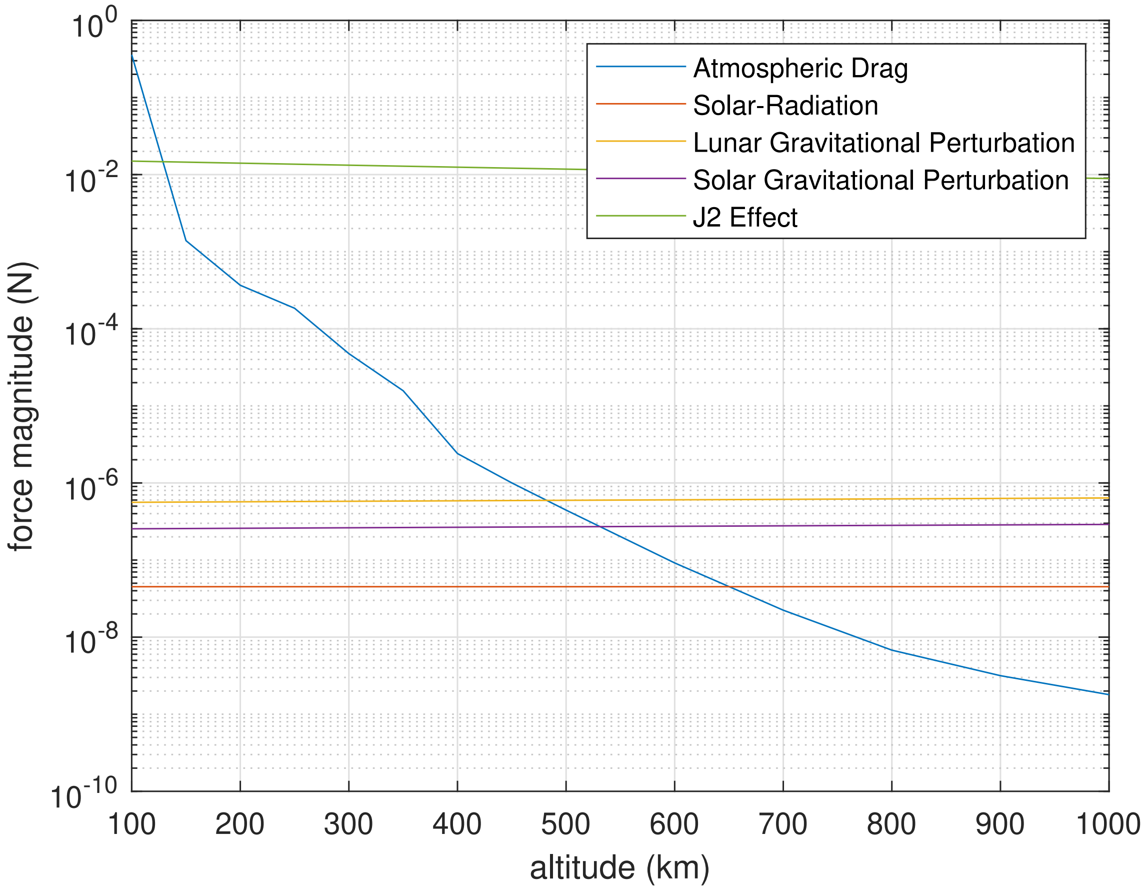

Figure 3.2: Perturbing force orders of magnitude in LEO $m_{sat} = 1kg$

As explained in Chapter 1, the study case is on 1U, 2U and 3U CubeSats deployed from the ISS. Consequently, the starting condition is usually about 400-420 km altitude with a 51.6 degree orbit inclination. Satellites studied have a weight to cross-section ratio closer to the first case. Thus, as a matter of simplicity, the two main effects for a satellite with 1 kg mass and 0.01 m^2 cross-section in the 100-400

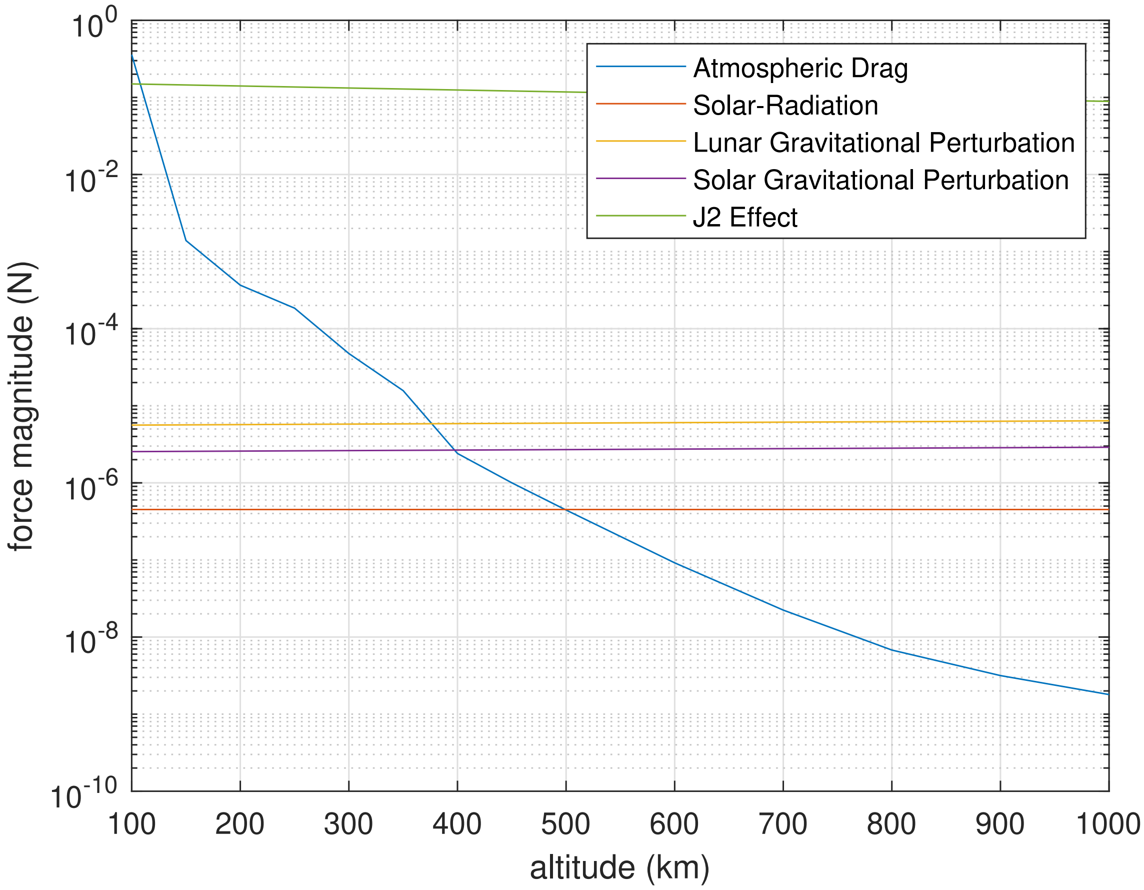

Figure 3.3: Perturbing force orders of magnitude in LEO $m_{sat} = 10kg$

km range are the only ones considered in the code, i.e. J2 effect and atmospheric drag.

3.3 Atmospheric Drag Analysis

The most complex aspect in developing the code was that regarding the atmospheric drag. The starting point was to identify equations suitable for the study case and the input parameters for them, the latter is that with major difficulties. Indeed, since the uncertainty of parameters, it is necessary to assess complex atmospheric models and evaluate assumptions on satellite attitude.

3.3.1 Drag Equations

An analysis is carried out starting from the Gaussian VOP equations, (3.9), to study the effects of the atmospheric drag on the motion of a satellite. Gaussian form is chosen because drag is a non-conservative force. Initially, the focus is on the rate of change of the semi-major axis (choice motivated at the end of the subsection).

$$\frac{da}{dt} = \frac{2}{n\sqrt{1-e^2}} \left\{ e \sin(\nu) F_R + \frac{p}{r} F_S \right\} \tag{3.10}$$

Firstly, the drag acceleration (2.35) is decomposed in RSW coordinate system and substituted in (3.10). It can be noticed that the drag component along R will result in periodic variations, while the component along S in secular variations. The expression is manipulated (further details in [13]) and the following relation is obtained:

$$\frac{da}{dt} = -\rho \frac{C_D S}{m_{sat}} v_{rel}^2 \left(\frac{\sqrt{1 + e^2 + 2e\cos(\nu)}}{n\sqrt{1-e^2}} \right) \tag{3.11}$$

Then, v_{rel} is written in function of orbital parameters and the equation is manipulated again to get a rate of change of a with respect to the true anomaly:

$$\frac{da}{d\nu} = -\rho \frac{C_D S a^2 (1 + e^2 + 2e\cos(\nu))^{3/2}}{m_{sat}(1 + e\cos(\nu))^2} \tag{3.12}$$

To proceed analytically, the true anomaly, ν is converted in eccentric anomaly, E, and the expression is integrated over one period. Lastly, the result is expanded in power series of $e\cos(E)$ basing on an exponential atmospheric model for ρ. The equations for the other parameters are also given for completeness:

$$\Delta a_{rev} = -2\pi\delta a^2 \rho_p \left\{ J_0 + 2eJ_1 + \frac{3e^2}{4}(J_0 + J_2) + \frac{e^3}{4}(3J_1 + J_3) + O(e^4) \right\} \exp(-c) \tag{3.13}$$

$$\Delta e_{rev} = -2\pi\delta a \rho_p \left\{ J_1 + \frac{e}{2}(J_0 + J_2) - \frac{e^2}{8}(5J_1 + J_3) - \frac{e^3}{16}(5J_0 + 4J_2 - J_4) + O(e^4) \right\} \exp(-c) \tag{3.14}$$

$$\Delta i_{rev} = -\frac{\pi a \omega_\oplus \delta \rho_p}{2n\sqrt{Q}} \sin(i) \left\{ J_0 - 2eJ_1 + (J_2 - 2eJ_1)\cos(2\omega) + O(e^2) \right\} \exp(-c) \tag{3.15}$$

$$\Delta\Omega_{rev} = -\frac{\pi a \omega_\oplus \delta \rho_p}{2n\sqrt{Q}} \left\{ J_2 - 2eJ_1 + O(e^2) \right\} \sin(2\omega) \exp(-c) \tag{3.16}$$

$$\Delta\omega_{rev} = -\Delta\Omega_{rev} \cos(i) \tag{3.17}$$

$\omega_\oplus$ is the Earth rotation rate, ρ_p is density at perigee. $c = ae/H$ where H is the scale height, $\delta = QSC_D/m_{sat}$ where Q is a factor which include the rotation of atmosphere. J_j are modified Bessel functions with c as argument:

$$J_j(c) = \frac{1}{2\pi} \int_0^{2\pi} \cos(j\theta) \exp(c\cos\theta)\, d\theta \tag{3.18}$$

Referring to [20], a formulation for inclination rate of change, derived from Gauss equations, could be also considered.

$$\frac{di}{dt} = -\frac{1}{4}\sqrt{\frac{a^3}{\mu}}\frac{\rho S C_D}{m} v_{rel}\omega_E \sin{(i)} \tag{3.19}$$

Summarising, the main variations are on a and e, while i, Ω and ω present small changes due to the cross-force, perpendicular to the orbital plane, created by atmospheric rotation (see [21]). Thus, the main driver of the orbital decay for a satellite is the Earth atmosphere through the aerodynamic drag. This force is the cause of the gradual decreasing of the semi-major axis in LEO and has the strongest influence on the satellite's lifetime. This motivates the choice to initially focus only on the a variation at the beginning of this subsection.

For the same reason, in the development of OLET, the drag effect is studied only on the semi-major axis variation. As a matter of simplicity, the approach followed is semianalytical, and the (3.12) equation, obtained in an analytical way, is integrated numerically bypassing the computation of modified Bessel functions.

Besides a, e and ν, the (3.12) equation needs as inputs atmospheric density, drag coefficient, cross-section and satellite mass. The m_{sat} is usually well known and it could be assumed constant since satellites analysed in this study usually do not have a propulsion system. The only mission phase when this hypothesis is not in accordance with reality is the last operative moments when the satellite is disintegrating itself in the atmosphere. The C_D calculation is derived by the gas-surface interaction in orbit and it is not easy to have an accurate value because it is influenced by many variables, mainly altitude but also sunspot cycle [22]. A good estimation could be a constant 2.2 value, assuming the flat plate model. Lastly, ρ and S, as C_D are both uncertain parameters. The atmospheric density is evaluated through models while the cross-section is dependent by the satellite dimensions and attitude.

3.3.2 Atmospheric Model Implementation

As seen in subsections 2.3.4 and 3.3.1, the atmospheric density is strongly influenced by many variables, consequently, its calculation introduces many difficulties. From the very beginning of writing the code, to test it, it was necessary a ρ input for the equations. Initially an exponential model was used for its simplicity. After that, striving for accuracy it was implemented a more complex atmospheric model, NRLMSISE-00.

Exponential Atmospheric Model

The Exponential Atmospheric Model was introduced in subsection 2.3.4. The expression that determines the atmosphere density, (2.41), needs as input only the satellite altitude, allowing to identify the other input, i.e. ρ_0 and H, from table 2.6. Thus, the implementation consists in just developing a function which reads a text file, replicating table 2.6. Then, depending on the altitude input, the function obtains reference density and scale height. These data allow to determine the density value represented in figure 3.4.

NRLMSISE-00

The NRLMSISE model, as explained in 2.3.4, takes into account many variables, consequently, it is greatly complex. It is implemented in the code through an existing library written in C language by Brodowski [23].

Table 3.1: NRLMSISE-00 library: input and output

Structure	Variable	Description
flags	switch	Array with 24 elements (switches) to turn on or off particular variations. Standard values are 0 for switch 0 and 1 for switches 1 to 23, however they could be set by the user
	sw	24 element array set internally
	swc	24 element array set internally
input	year	Current year
	doy	Current day of year
	sec	Current seconds in day (Universal Time)
	alt	Altitude in kilometers
	g lat	Geodetic latitude, angle between the equatorial plane and the normal to the surface of the ellipsoid
	g long	Geodetic longitude
	lst	Local apparent solar time (hours). It should be consistent with sec and g long through the following relation: $lst = sec/3600 + g_{long}/15$
	f107A	81 day average of F10.7 flux (centered on doy)
	f107	Daily F10.7 flux for previous day
	ap	Daily magnetic index (amplitude)
	ap array	Array containing ap values for the current time and for 3, 6 and 9 hours earlier, in addition also averaged values until two days before are considered
output	d	Densities of each main element composing the atmosphere and the overall density
	t	Exospheric temperature

The two main subroutine are GTD7 and GTD7D, both require as input two variable structures, *flag* and *input*, as shown in table 3.1, and both give the same *output* structure. They differ only because GTD7D considers the influence of the *anomalous oxygen* in the ρ calculation. The anomalous oxygen component accounts for the presence of appreciable hot atomic oxygen (O_h) or atomic oxygen ions (O^+) near the exobase under some conditions but does not explicitly distinguish contributions by the two species [24]. O_h and O^+ increase the thermospheric total mass density, indeed, observational evidence shows an additional component to drag at altitudes above 500 km with high latitudes in the summer hemisphere. In the code GTD7 subroutine is usually recall since the study case altitude is below 420 km and the anomalous oxygen is negligible.

As far as input and flags are concerned, the *switch* array is set with standard values, see table 3.1. To get the input time variables *year*, *doy* and *sec*, a function, capable of updating the date starting from an initial date and a Δt, was implemented. Then, satellite geodetic latitude is obtained, resolving the following non-linear equation [13] through the Newton method:

$$\tan(\phi_{gd}) = \frac{r_k + C_\oplus e_\oplus^2 \sin(\phi_{gd})}{r_{ij}} \tag{3.20}$$

Where $e_\oplus$ is the *eccentricity of the Earth*, r_{ij} is the projection of the satellite position on the equatorial plane, r_k is the component orthogonal to the equatorial plane and $C_\oplus$ is defined as follow:

$$C_\oplus = \frac{R_\oplus}{\sqrt{1 - e_\oplus^2 \sin^2(\phi_{gd})}} \tag{3.21}$$

Consequently, the altitude (on the ellipsoid) is calculated:

$$h_{ellp} = \frac{r_{ij}}{\cos(\phi_{gd})} - C_\oplus \tag{3.22}$$

Given the satellite position vector with respect to the geocentric frame, the geodetic longitude is easily obtained as:

$$\lambda = \alpha - \alpha_G \tag{3.23}$$

where $\alpha = \arccos(r_i/r_{ij})$ and α_G is the α angle for Greenwich meridian.

Solar flux values and magnetic amplitudes are taken from *Space Weather* file, released by *celestrack* website [25], which contains several data organized as in table 3.2. From the fourth column to the eleventh, there are a_p measured every three hours, the twelfth column is the a_p daily average, the thirteenth column is the 10.7-cm Solar Radio Flux observed and in the last column the Centered 81-day arithmetic average of $F_{10.7}$ value. Measurements are collected since 1957 until the present day and predictions are from tomorrow until 2040. However the predictions are greatly uncertain, hence, estimating accurately when the reenter of a satellite will happen is indeed difficult.

Table 3.2: Simplified Space Weather file structure

y	m	d	ap	ap	ap	ap	ap	ap	ap	ap	Ap	F10.7	Ctr81
2018	01	01	18	22	9	9	12	4	4	5	10	66.8	69.3
2018	01	02	6	3	2	3	6	4	7	4	4	67.2	69.3
2018	01	03	0	3	4	3	2	4	2	0	2	68.3	69.4
2018	01	04	3	0	4	0	3	2	5	4	3	67.2	69.4
2018	01	05	6	5	3	3	5	4	7	4	11	67.0	69.5
2018	01	06	5	0	3	0	2	0	2	2	2	67.1	69.4
2018	01	07	4	0	0	0	2	3	3	2	2	67.6	69.4
2018	01	08	2	3	9	6	22	15	7	6	9	68.0	69.4

Lastly, the *output* structure provided by the NRLMSISE library is made by the main atmospheric species densities, overall ρ included (see Table 3.3), and by two values of temperatures, the first one is a global average value for altitudes below 120 km and the latter is the corresponding temperature at the input altitude. However, of all these outputs, the developed code only collects the total mass density, which serves as an input for the (3.12) equation.

Table 3.3: NRLMSISE-00 library output densities

Output Variable	Species	Description
d[0]	ρ_{He}	Helium number density (cm^{-3})
d[1]	ρ_{O}	Oxygen number density (cm^{-3})
d[2]	ρ_{N_2}	Molecular Nitrogen number density (cm^{-3})
d[3]	ρ_{O_2}	Molecular Oxygen number density (cm^{-3})
d[4]	ρ_{Ar}	Argon number density (cm^{-3})
d[5]	ρ	Total mass density (g/cm^3)
d[6]	ρ_{H}	Hydrogen number density (cm^{-3})
d[7]	ρ_{N}	Nitrogen number density (cm^{-3})
d[8]	ρ_{O_h,O^+}	Anomalous Oxygen number density (cm^{-3})

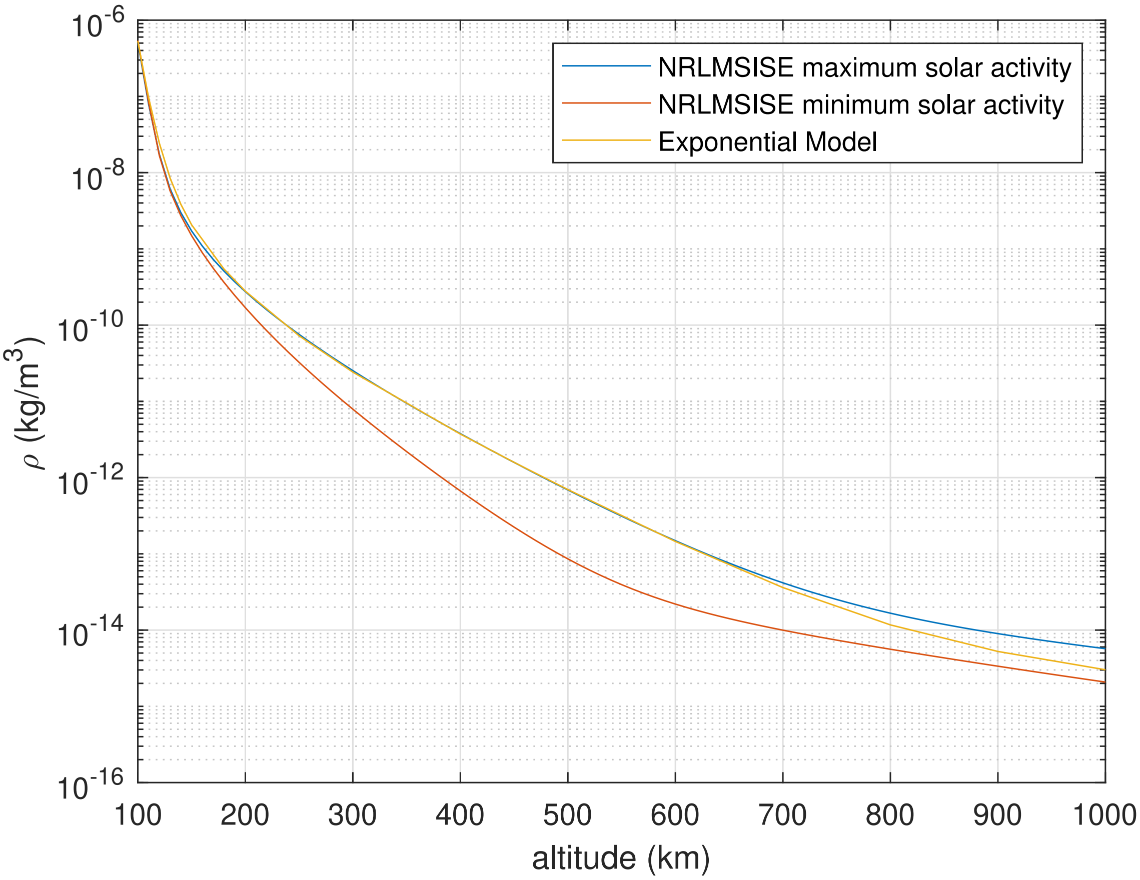

Figure 3.4: Comparison between NRLMSISE-00 and Exponential models, fixed longitude, latitude and time of the day

3.3.3 Cross-Sectional Area Estimation

The cross-section in (2.35) and (3.12) equations is defined as the satellite area perpendicular to the velocity relative to the Earth's atmosphere. This surface is determined by the satellite dimensions and its attitude. Since the satellite objects of this study are Cubesats, they usually do not present an attitude control system or passive stabilization, whereby they randomly tumble.

For a tumbling spacecraft the averaged cross-section could be used to evaluate the ballistic coefficient (2.36). The easiest way to calculate it is considering a straightforward two-point average of the minimum and maximum cross-sectional areas. However, this method could be improved considering the sections for all the anticipated tumbling attitudes, integrating them and dividing the result for the number of the tumbling attitudes. [14]

This approach is more accurate than the first one, but, at the same time, more complex. Nevertheless, there are some opens source code, which, provide this kind of

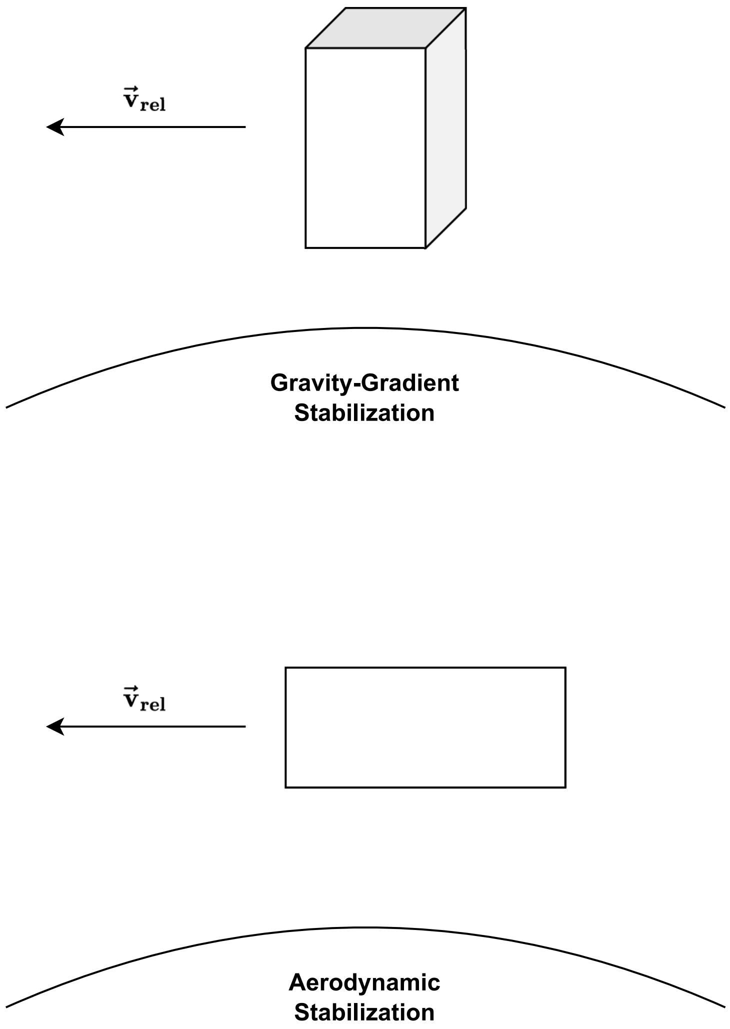
$\vec{\mathbf{v}}_{\mathbf{rel}}$
Gravity-Gradient Stabilization
$\vec{\mathbf{v}}_{\mathbf{rel}}$
Aerodynamic Stabilization

Figure 3.5: Attitude influence on cross-section determination

calculation. To get some valid cross-section values needed for testing the OLET code, the ESA's software, DRAMA (Debris Risk Assessment and Mitigation Analysis), was used, particularly its CROC (Cross-section Of Complex bodies) tool.

Table 3.4 shows some cross-sectional area calculation and it is evident the difference between the value obtained with a straightforward two-point average and the one calculated by the CROC tool. The values shown in table 3.4 are for Cubesats with a regular shape and without any kind of extension, such as solar panels and antennas. In that cases the cross-section should be adjust accordingly.

Lastly, as was introduced before, some satellites are designed to achieve a certain type of passive stabilization. When it is so the section estimation is certainly easier. For example, a spacecraft with a large length to diameter ratio could be

Table 3.4: Average cross-sectional area

Category	S_{min} (m^2)	S_{max} (m^2)	$(S_{min} + S_{max})/2$ (m^2)	S_{avg} (m^2)
1U	0.010	0.0173	0.0137	0.0149
2U	0.010	0.0298	0.0199	0.0248
3U	0.010	0.0435	0.0268	0.0347

stabilized for gravity gradient. In this case it would be quite correct assuming the cross section coincides with the maximum value, although some uncertainties still remain. Similarly with drag-induced passive attitude stabilization, it is not a wrong assumption considering the cross section coincides with the minimum cross section value (figure 3.5).

3.4 Central-Body Analysis

The central-body has the largest perturbing effect on satellites orbiting in LEO (figure 3.2). Concepts discussed in this section were yet introduced in subsection 2.3.2. Alone secular effects (zonal harmonics) are accounted, hence tesseral and sectorial harmonics, which represent periodic effect, are ignored. Moreover, since J2 is three order of magnitude bigger than the other coefficients, only the Legendre polynomial (2.30) of second degree is considered.

Basing on these assumptions, the (2.32) equation is evaluated subtracting the two-body potential to obtain the perturbing potential function.

$$R = -\frac{\mu J_2}{r}\left(\frac{R_\oplus}{r}\right)^2 \frac{3}{2}\left(\sin^2(\phi_{gc}) - \frac{1}{3}\right) \tag{3.24}$$

This equation is developed, substituting the latitude with i, ω and ν, and with further steps [13] an averaged form of R over one revolution is provided. This form depends on the classical orbital elements and it can be substituted in (3.7) equation.

$$R_{avg} = -\frac{\mu J_2}{a}\left(\frac{R_\oplus}{a}\right)^2 \left(\frac{3}{4}\sin^2(i) - \frac{1}{2}\right)\left(\frac{1}{(1-e^2)^{3/2}}\right) \tag{3.25}$$

As (3.25) shows, of all six parameters only three influence the potential function: semimajor axis, eccentricity and inclination. Therefore, referring to the Lagrangian VOP (3.7), a, e and i are not affected by the Earth's oblateness since:

$$\frac{\partial R}{\partial M_0} = \frac{\partial R}{\partial \omega} = \frac{\partial R}{\partial \Omega} = 0 \tag{3.26}$$

Thus, substituting (3.25) in the Lagrangian planetary equations (3.7) the three parameters influenced by J_2 are Ω, ω and M_0.

$$\begin{aligned}
\frac{d\Omega}{dt} &= -\frac{3nR_\oplus^2 J_2}{2p^2}\cos(i) \\
\frac{d\omega}{dt} &= \frac{3nR_\oplus^2 J_2}{4p^2}\left[4 - 5\sin^2(i)\right] \\
\frac{dM_0}{dt} &= -\frac{3nR_\oplus^2 J_2\sqrt{1-e^2}}{4p^2}\left[3\sin^2(i) - 2\right]
\end{aligned} \tag{3.27}$$

Integrating on one round and assuming that p, i and e remain constant over one orbit period:

$$\Delta_{2\pi}\Omega = -3\pi\frac{nR_\oplus^2 J_2}{p^2}\cos(i) \tag{3.28}$$

$$\Delta_{2\pi}\omega = 3\pi\frac{nR_\oplus^2 J_2}{2p^2}\left[4 - 5\sin^2(i)\right] \tag{3.29}$$

$$\Delta_{2\pi}M_0 = -3\pi\frac{nR_\oplus^2 J_2\sqrt{1-e^2}}{2p^2}\left[3\sin^2(i) - 2\right] \tag{3.30}$$

These relations indicate that, all other equation parameters being equal, Ω, ω and M_0 variations are bigger for smaller orbit because of the p at the denominator.

As far as Ω is concerned, the $\cos(i)$ term in (3.28) shows that low inclinations imply great variations of the longitude of the ascending node. This phenomenon is called *regression of the node line*, and it is absent for polar orbits. For a LEO orbit nearly equatorial, but still with i other than 0 deg or 180 deg, the longitude of the ascending node changes of about 9 degree per day.

The (3.29) provides the *apsidal precession*. This phenomenon does not occur at 63.4 deg and 116.6 deg, that are the inclination values chosen for *Molnyia*. This kind of orbit, indeed, requires that the apogee remains fixed at high-latitudes.

3.5 OLET Code

Basing on what has been previously discussed, in this section the code features are described. The code has been called *Orbit Lifetime Estimation Tool*, OLET, since its aim of predicting the time taken by a satellite to decay and reenter the Earth's atmosphere. Initially, the main equations implemented are briefly summarised; then, the way to impose initial conditions is explained. After this, the choice of the numerical integration method is justified and, lastly, a recap of the several OLET version is reported. A part of the last version of the OLET code is reported in Appendix B.

3.5.1 Equations

As assessed in section 3.2 the two main perturbations to be taken into account are atmospheric drag and J2 effect. As far as atmospheric drag is concerned, in subsection 3.3.1 some equations have been identified and the main secular variation is on the semi-major axis, (3.12). For central body gravitational perturbations only the J2 effect and its influence on Ω and ω are considered, section 3.4. Variations on M are neglected since they do not significantly impact the lifetime calculation.

$$\begin{aligned}
&\frac{da}{d\nu} = -\rho\frac{C_D S a^2 (1+e^2+2e\cos{(\nu)})^{3/2}}{m_{sat}(1+e\cos{(\nu)})^2} \\
&\frac{de}{dt} = 0 \\
&\frac{di}{dt} = 0 \\
&\Delta_{2\pi}\Omega = -3\pi\frac{nR_\oplus^2 J_2}{p^2}\cos(i) \\
&\Delta_{2\pi}\omega = 3\pi\frac{nR_\oplus^2 J_2}{2p^2}\left[4-5\sin^2(i)\right]
\end{aligned} \tag{3.31}$$

The drag effect on eccentricity is to make circular the orbit, but the initial orbit of the study case is yet almost circular, indeed, it usually has e values from 0.0001 to 0.001. Thus, the eccentricity variation is neglected. Similarly, basing on equation (3.19), the variation on orbit inclination are ignored since they are very small and do not influence intensively the satellite lifetime.

Initially, in the first OLET versions, the semimajor axis equation was considered integrated over one round, assuming the semimajor axis constant. This form was used to test numerical methods of composite quadrature, discussed in subsection 3.5.4.

$$\begin{aligned}
&\Delta_{2\pi}a = -\frac{C_D S a^2}{m_{sat}}\int_0^{2\pi}\frac{\rho(1+e^2+2e\cos{(\nu)})^{3/2}}{(1+e\cos{(\nu)})^2}\,d\nu \\
&\frac{de}{dt} = 0 \\
&\frac{di}{dt} = 0 \\
&\Delta_{2\pi}\Omega = -3\pi\frac{nR_\oplus^2 J_2}{p^2}\cos(i) \\
&\Delta_{2\pi}\omega = 3\pi\frac{nR_\oplus^2 J_2}{2p^2}\left[4-5\sin^2(i)\right]
\end{aligned} \tag{3.32}$$

3.5.2 Initial Conditions

To begin the simulation, some starting values of the orbital elements are needed. Basically, given the study case of CubeSats deployed from the ISS, the initial conditions may be the typical orbital elements of space station (table 3.5). However, when testing the simulator on satellites yet reentered, these fixed initial conditions may not be enough to validate OLET. In fact, the ISS altitude ranges between 390 km and 420 km (figure 3.6) and this strongly influences the satellite lifetime.

Table 3.5: Example of ISS typical orbital elements

$a(km)$	e	$i(°)$	$\Omega(°)$	$\omega(°)$	$\nu(°)$
400	0.002	51.64	243.10	230.44	200.69

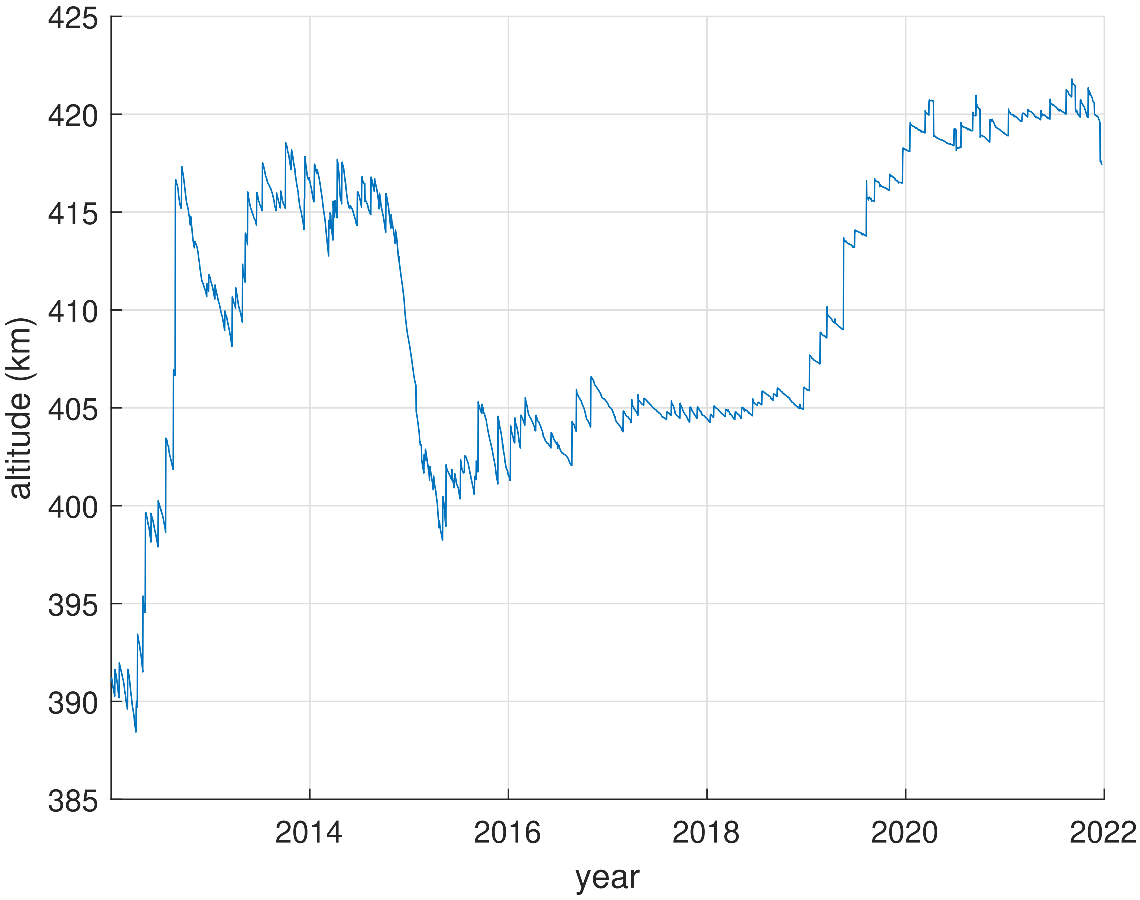

Figure 3.6: ISS altitude

Table 3.6: Two-Line Element set: First Line

Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of the launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Day of the year and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion
45-52	Second Time Derivative of Mean Motion (Leading decimal point assumed)
54-61	BSTAR drag term (Leading decimal point assumed)
63	Ephemeris type
65-68	Element number
69	Checksum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Table 3.7: Two-Line Element set: Second Line

Column	Description
01	Line Number of Element Data
03-07	Satellite Number
08	Classification (U=Unclassified)
09-16	Inclination (°)
18-25	Right Ascension of the Ascending Node (°)
27-33	Eccentricity (Leading decimal point assumed)
35-42	Argument of Perigee (°)
44-51	Mean Anomaly (°)
53-63	Mean Motion (Revs per day)
64-68	Revolution number at epoch (Revs)
69	Checksum (Modulo 10)

Thus, the initial conditions are made to coincide with those of the ISS, basing on its TLE on the current day of deployment. TLE set, i.e. *Two-Line Element*, was developed by the NORAD for SGP4/SDP4 orbital model. It consists of two 69-character lines of data to identify a satellite and determine its state (position and velocity) at a given time. In tables 3.6 and 3.7, the fields for line 1 and 2 are defined [26].

Table 3.7 shows that e, i, Ω, ω and M are directly defined by the TLE. On the other hand the semimajor axis must be derived from the number of revolutions in one day, i.e. 64-68 column of the second line.

Table 3.8: Example of ISS Two-Line Element set

1 25544U 98067A 12001.23210845 .00014903 $00000+0$ $19296-3$ 0 7342
2 25544 51.6432 244.6790 0024050 229.4243 226.7947 15.58805023751829

The ISS TLEs are obtained from celestrack website [27] and inserted in a text file containing TLEs from 2012 to 2022. Then, this file is read by OLET to impose the inital conditions at the beginning of the simulation.

3.5.3 Input Data

The input data required by OLET changed during the development of the code. Initially, when the atmospheric model implemented was the exponential one, in addition to satellite mass and cross-section (drag coefficient set at 2.2 internally), the only input was the orbital elements set to identify initial conditions. Then, implementing the NRLMSISE-00 atmospheric model the initial date variable was introduced (see table 3.9).

Table 3.9: Input data (imposing manually initial conditions)

Variable	Description
day	Day of start date
mo	Month of start date
yr	Year of start date
$\vec{x}$	Set of initial orbital elements (six variables)
m	Satellite mass in kg
S	Satellite cross-sectional area in m^2

To test the code on satellites deployed from ISS, further modifications were made. In fact, basing on what was discussed in subsection 3.5.2, the initial conditions are imposed by ISS TLE. Consequently, giving only the deployment date of the satellite as an input, a dedicated function in OLET provides the initial orbital element set. This allows to remove the $\vec{x}(a, e, i, \Omega, \omega, \nu)$ from the input data (table 3.10). Furthermore, the real lifetime of the satellite is required as an additional input to compare it with the lifetime value calculated by OLET.

Table 3.10: Input data for code testing

Variable	Description
day	Day of satellite deployment
mo	Month of satellite deployment
yr	Year of satellite deployment
sat_{life}	Actual value of satellite lifetime in years
m	Satellite mass in kg
S	Satellite cross-sectional area in m^2

Since the number of the testing satellites (Appendix A) is more than thirty, it was preferred to avoid to enter manually input data for each satellite and for each simulation. Thus, the process was automated developing OLET in such a way that it can receive as input a text file containing data for more than one satellite (table 3.11). Consequently, more satellites can be evaluated in one simulation .

Table 3.11: Example of a structure for automatic data entry

Sat. Number	day	mo	yr	sat-life (yr)	m (kg)	S (m^2)
1	11	02	2014	0.323	5.00	0.069
2	04	03	2015	0.411	4.25	0.051
3	17	09	2015	1.186	3.99	0.030
4	16	05	2016	0.973	3.50	0.039
5	16	05	2016	0.636	4.00	0.069
6	16	05	2017	1.384	4.00	0.069
7	11	05	2018	2.088	4.00	0.038
8	31	01	2019	2.723	3.56	0.029
9	17	06	2019	2.334	2.60	0.038
10	04	07	2019	2.104	3.00	0.038
11	14	06	2021	0.748	3.36	0.120

3.5.4 Numerical Analysis

Several numerical approaches are implemented in the OLET code, hence in this subsection some of them are briefly described. For a better understanding of the topic, refer to [12] [13] [28].

Numerical Integration

The easiest way to handle equations in the (3.33) form introduced in subsection 3.5.1 is through *Runge-Kutta (RK) methods.* These are a family of integration methods of varying orders derived from Taylor series. The order is defined by the highest power of the *step size*, *h*, in the equivalent Taylor series expansion.

$$\frac{dx}{dt} = f(x,t) \tag{3.33}$$

The RK method of *first-order* (*Euler method*) is the simplest one but usually the most inaccurate. It was used to solve (3.32) with a 2π step size.

$$x_{k+1} = x_k + hf(t_k, x_k) \tag{3.34}$$

The *fourth-order* Runge Kutta method, instead, was implemented to solve the semimajor axis equation of (3.31). The standard formulation is the following:

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.35}$$

where:

$$\begin{aligned} k_1 &= f(t_n, x_n) \\ k_2 &= f(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}) \\ k_3 &= f(t_n + \frac{h}{2}, x_n + \frac{k_2}{2}) \\ k_1 &= f(t_n + h, x_n + k_3) \end{aligned} \tag{3.36}$$

In this approach, the derivative is evaluated at four points along the estimated trend of the parameter, two of them are intermediate points. Also higher order RK methods may be used, with an improved accuracy but an increased complexity in implementation. In those cases it would be better referring to existing libraries.

Summarising, the standard fourth-order RK is easy to implement, stable, does not require a starting procedure, presents a small truncation error and, being a single step method, the step size could be easily changed. But, for the same reasons it is difficult to identify the truncation error and, consequently, the proper step size.

These disadvantages may be overcome with variable-step methods, which exploit the history of the function being integrated to predict and eventually correct the function value of the next step, or with higher-order Runge-Kutta methods. However, as discussed in section 3.3, many uncertainties are present within the problem. Thus, it was retained unnecessary to go too deep into the implementation of complex numerical integration methods.

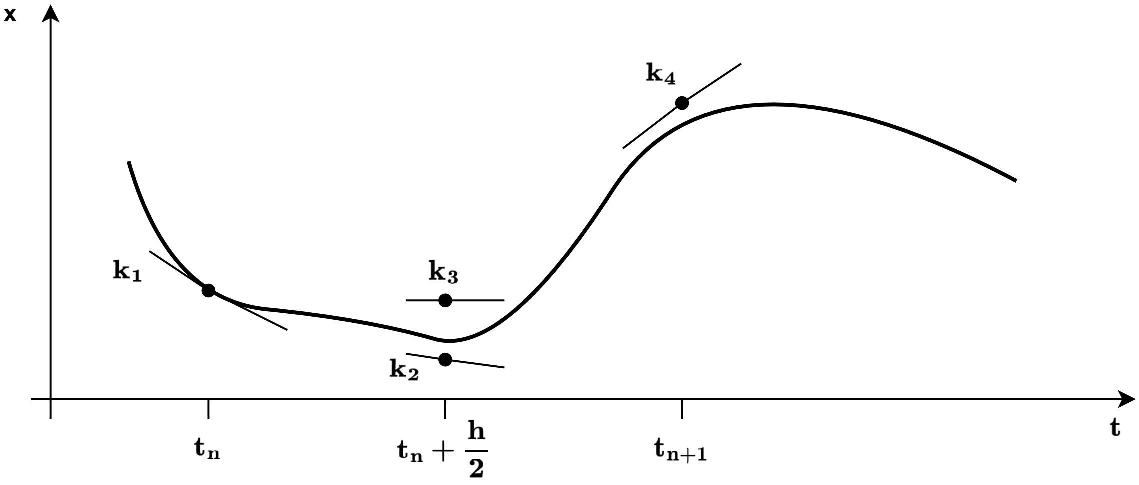

Figure 3.7: 4th order Runge-Kutta [13]

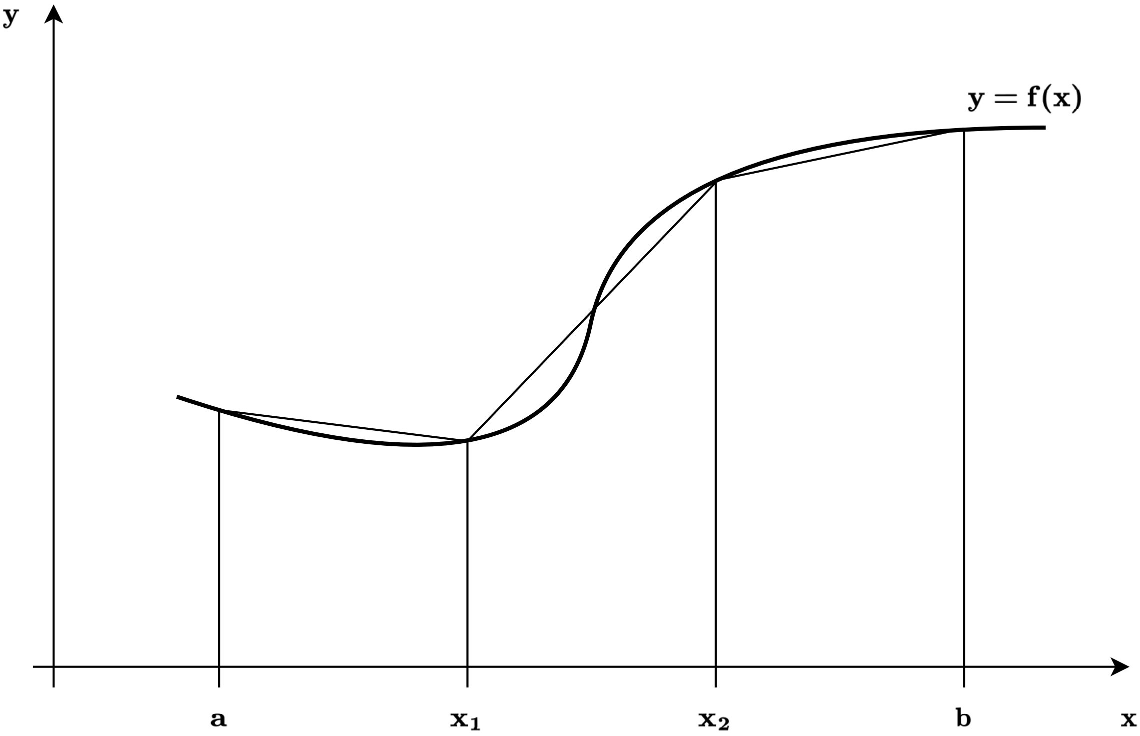

Figure 3.8: Composite Trapezoidal Rule

Numerical Quadrature

In semimajor axis equation of (3.32), there is an integral, the basic method to approximate a definite integral is called *numerical quadrature.* Without going too far in the numerical analysis a simple method was chosen and implemented, i.e. the *composite trapezoidal rule*. A formulation is reported below for n subintervals. The size of subintervals is $h = (a - b)/n$.

$$\int_a^b f(x)\,dx = \frac{h}{2}\left[f(a) + \sum_{j=1}^{n-1} f(x_j) + f(b)\right] \tag{3.37}$$

Figure 3.8 shows graphically the composite trapezoidal rule. In OLET code, the minimum size of subintervals, h, is of 10°. In fact, for lower amplitude there is few improvement of the simulator accuracy and the simulation time considerably increases. However, increasing h, the error increases slightly but the simulation time decreases considerably, see Chapter 4.

Table 3.12: OLET main versions

Version	Description
OLET 0.2	Equations (3.13) and (3.14). 1st order RK method. Exponential atmospheric model. Manual input data entry. Initial conditions set from ISS TLE.
OLET 0.4	Equations from [20]. 1st order RK method. Jacchia atmospheric model [13]. Manual input data entry. Initial conditions imposed: circular, 400 km altitude, 51° inclined orbit.
OLET 0.6	Equations from [20]. 1st order RK method. NRLMSISE-00 atmospheric model. Manual input data entry. Initial conditions set from ISS TLE.
OLET 0.8	Equations (3.32). 1st order RK method. NRLMSISE-00 atmospheric model. Automatic input data entry. Initial conditions set from ISS TLE.
OLET 1.1	Equations (3.31). 4th order RK method. NRLMSISE-00 atmospheric model. Automatic input data entry. Initial conditions set from ISS TLE.

3.5.5 Versions

Several OLET versions were developed, changing the effects involved, the equations, atmospheric models, the way to enter input data and initial conditions. The main versions of the code are reported in table 3.12. However, the equations and the results of the majority of them are not discussed in this document to keep the focus on the latest versions. Nevertheless, each OLET update was essential to achieve the two final versions: OLET 0.8 and OLET 1.1. Particularly, results of OLET 1.1 are discussed in Chapter 4.

3.5.6 Output Data

The OLET code provides the lifetime, expressed in years for each satellite analysed, as on-screen output. Moreover, two text files are given; the first shows, for each satellite, the orbital elements for each orbit elapsed (table 3.13) until the altitude is below 100 km. Whereas the second reports the lifetime values (actual and estimated) with the relative error (table 3.14). These two files are then reprocessed through a Matlab script allowing to show a graphical perspective of results (see Chapter 4).

Table 3.13: First output data file structure

a (km)	e	i (°)	Ω (°)	ω (°)	ν (°)	Elapsed time (days)	Elapsed orbits
6793.47	0.00183	51.648	304.50	126.00	0.000	0.0644	1
6793.46	0.00183	51.648	304.18	126.16	0.000	0.1289	2
6793.45	0.00183	51.648	303.86	126.32	0.000	0.1934	3
6793.44	0.00183	51.648	303.54	126.48	0.000	0.2579	4
6793.43	0.00183	51.648	303.22	126.64	0.000	0.3224	5
6793.42	0.00183	51.648	302.90	126.79	0.000	0.3869	6
6793.41	0.00183	51.648	302.58	126.95	0.000	0.4514	7
6793.40	0.00183	51.648	302.26	127.11	0.000	0.5159	8
6793.39	0.00183	51.648	301.94	127.27	0.000	0.5804	9

Table 3.14: Second output data file structure

Sat. Number	OLET lifetime value (yr)	Actual lifetime value (yr)	Normalised OLET lifetime on the actual
1	1.552881	1.792000	0.866563
2	2.106056	1.789000	1.177225
3	1.634980	1.732000	0.943984
4	2.083202	1.921000	1.084436
5	1.300764	1.699000	0.765606

Chapter 4

Results

Some data of satellites deployed by ISS have been collected (Appendix A), to test the accuracy of the *Orbit Lifetime Estimation Tool* (code in Appendix B) . The following information were searched for each satellite.

Deployment Date is needed to impose the initial conditions of the satellite basing on the ISS TLEs. Also, it is helpful to extract space weather data.

Decay Date allows to know the real lifetime, which then will be compared with the one calculated by the simulator.

Mass is one of the inputs for the equations used in the simulator.

Dimensions to categorize the satellite (e.g. 1U CubeSat); at the same time they are a starting point to evaluate the cross-section.

Attitude combined with other data allows to estimate the cross-section; however, most of the time it is unknown so in that case the satellite is considered tumbling randomly.

Panels and Extensions influence the estimated cross-section value.

The satellites were divided into groups to better understand the behaviour of the OLET when tested on similar satellites.

- 1U CubeSats;
- 2U CubeSats;
- 3U CubeSats;
- Satellites with sails or that do not fall into the previous categories (Other).

	m (kg)	S (m^2)	S/m (m^2/kg)	lifetime (y)
1U	1.15	0.012	0.0104	1.66
2U	2.07	0.020	0.0096	1.79
3U	3.75	0.054	0.0144	1.36
Other	11.9	0.390	0.0328	1.44

Table 4.1: Satellite categories and average data from Appendix A

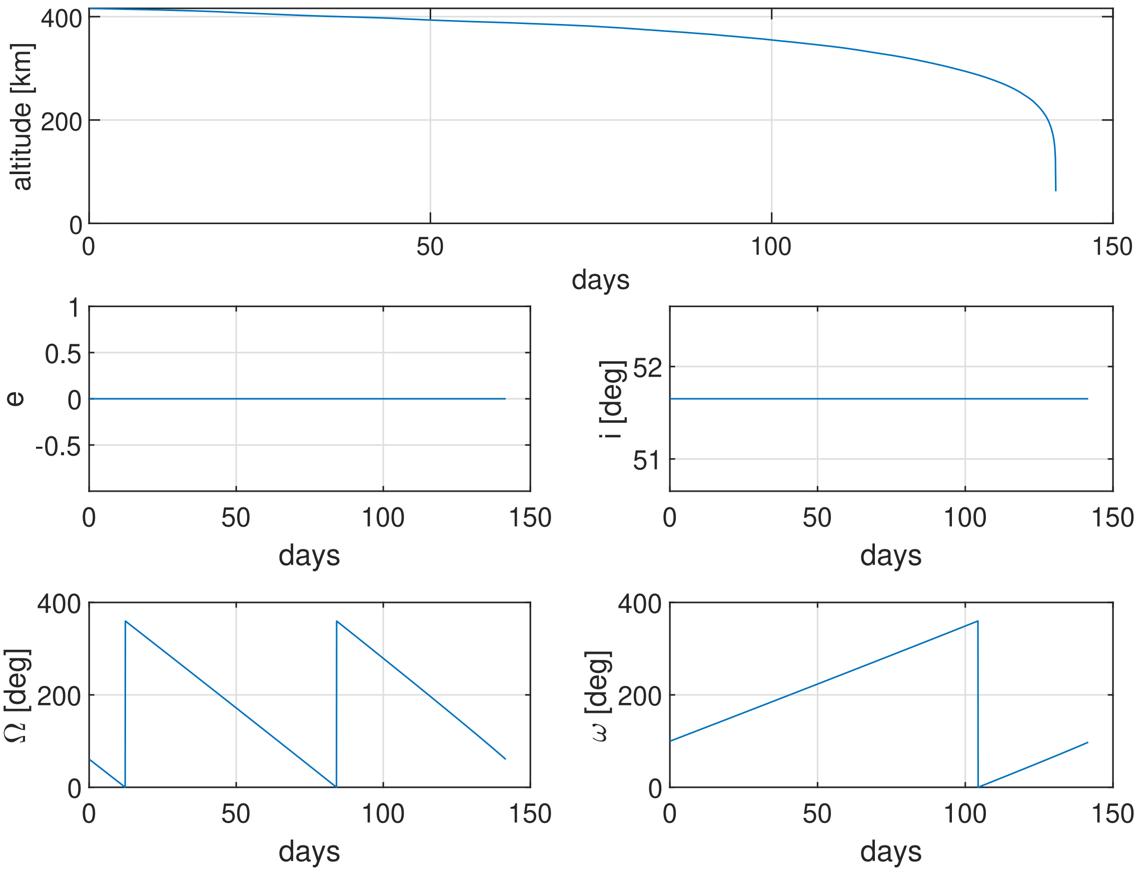

Figure 4.1: Ardusat orbital element variation

Their features are shown in the table 4.1, where the average satellite data collected in Appendix A are taken into account. These data indicate that the higher the cross-section to mass ratio of the category, the lower the lifetime. The trend is not fully respected by the *Other* class, probably because of different sizes and forms which may influence the drag coefficient.

In the next sections, a representative case is considered for each category. Then, the error between lifetime values calculated by OLET for all satellites of each category, and the actual values is shown on a diagram and commented on. Some deductions about the simulations are carried out and results of OLET are compared with those of SGP4 [29]. In fact, as discussed in Chapter 1, the OLET code approaches the perturbation influence with a similar method to that of SGP4. SGP4 is a mathematical model for prediction of satellite position and velocity, it was developed by Ken Cranford in 1970 and is used for near-Earth satellites [30]. SGP stands for *Simplified General Perturbations.*

4.1 1U CubeSats

1U Cubesats are a class of satellites fitting in a box of about 10 cm x 10 cm x 10 cm, called 1U form factor. Given their small size they usually present a simple design, without deployable solar panels or accurate AODCS. For this analysis a group of twelve 1U satellites (table A.2) were chosen. The Ardusat (table 4.2) could be taken as an example, his real lifetime is 148 days, while OLET underestimates by about one week (figure 4.1). For completeness, trends of the other orbital elements are reported in the same figure. Inclination and eccentricity are constant because of the previous assumptions, while longitude of the ascending node and argument of perigee are influenced by the J2 effect.

Deployment date	Lifetime (years)	Mass (kg)	Estimated Cross-section (m^2)	S/m (m^2/kg)
19/11/2013	0.405	1.00	0.014	0.014

Table 4.2: Ardusat

Considering all the 1U satellite class, some simulations were performed to investigate the size of the integrating step (figure 4.2 and 4.3). Consequently, a comparison is carried out between OLET 0.8 and 1.1 versions, both mentioned in subsection 3.5.5. First, below a step size of about 180° the relative error between the OLET lifetime and the actual one settles at about 10 %, whereas the average simulation time for each satellites increases rapidly. It should be highlighted that for OLET 0.8 is not fully correct to speak about an integration step but more properly of a size of the subintervals used for the composite trapezoidal method. Summarising, figures 4.2 and 4.3 show that the relative error values and trend and simulation time are about the same for the two OLET versions 0.8 and 1.1. Thus, from this point onwards, only results of the 1.1 version are discussed.

Basing on the outcome of figure 4.2, the integration step size, chosen to conduct analysis on 1U CubeSats, is 180°. This value ensures an enough accurate and quick estimation of when the CubeSat studied will reenter the Earth's atmosphere.

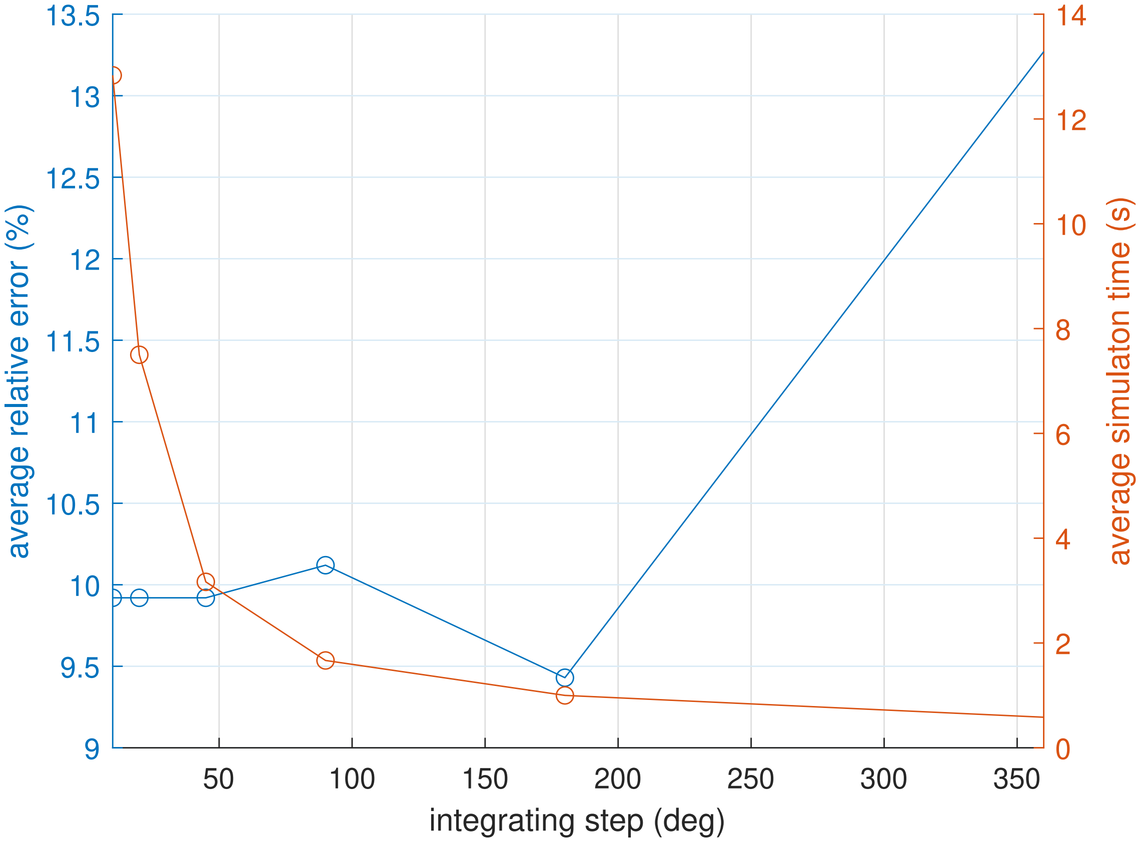

Figure 4.2: 1U simulation analysis, OLET v.1.1

The 1U category has the lowest average error in the lifetime calculation by OLET (figure 4.2) compared to the other categories. In fact, the percentage relative error is about 9.5 % with a step size of half orbit (180°). This outcome may be explained since 1U CubeSats do not usually have deployable solar panels which introduce a major uncertainty on the cross-sectional surface estimation. However, they could have other kind of extensions, such as antennas, which influence lightly the estimated cross-sectional area.

As mentioned above, to have a yardstick for the quality of OLET results, the orbit of satellites is also propagated with SGP4. Figure 4.4 shows for each satellite the lifetime calculations, by OLET and SGP4, normalised on its real lifetime value. Thus, ideally, all the columns should tend to 1. Obviously, this is not the case since the uncertainties on the input data (satellite mass and cross-section), in the density calculation and in neglecting certain effects. Nevertheless, OLET results are promising.

The input data for OLET simulations were taken by table A.2. Whereas simulations with SGP4 were conducted imposing a 1.00 kg mass and a 0.010 m^2 cross-section

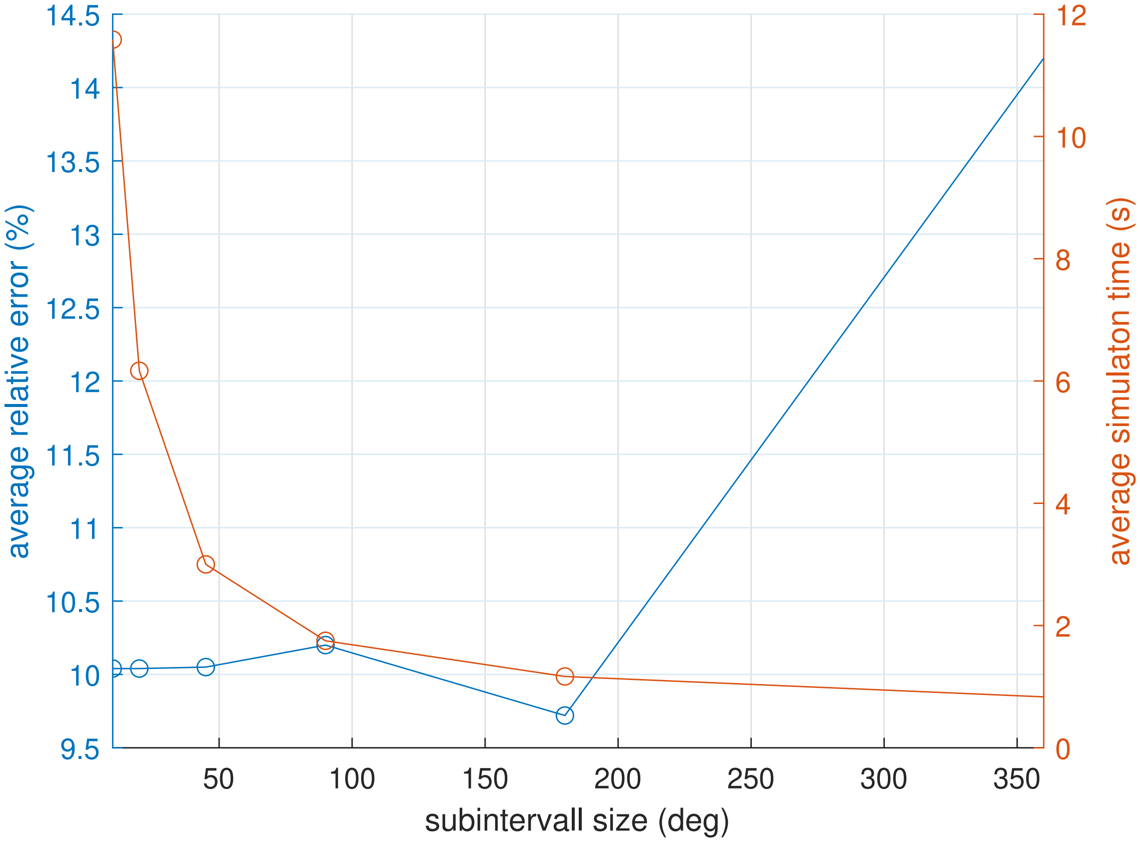

Figure 4.3: 1U simulation analysis, OLET v.0.8

for all 1U CubeSats not considering the table in appendix. In SGP propagations, the initial conditions for each satellite were imposed basing on the ISS TLE of the current day of deployment, as for OLET.

As far as the comparison between OLET and SGP4 is concerned (figure 4.4), interestingly, the lifetime evaluated by the two models for FITSAT, F-1, TechEdSat and Ardusat are pretty similar. Probably, this could be explained because the SGP4 model is based on an exponential atmospheric model [30][31]. In fact, the solar maxima occurring in 2012-2014 (figure 2.8), the years when those four satellites were deployed, is reproduced similarly by the exponential model (SGP4) and NRLMSISE (OLET), see figure 3.4. The remaining 1Us are deployed during a solar minima instead, and as figure 3.4 shows, the two atmospheric models envisage density values considerably different.

Thus, since NRLMSISE is a more accurate atmospheric model than the exponential one, OLET provides results closer to reality. However, it should be noted that the NRLMSISE accuracy strongly depends on the quality of $F_{10.9}$ and a_p predictions.

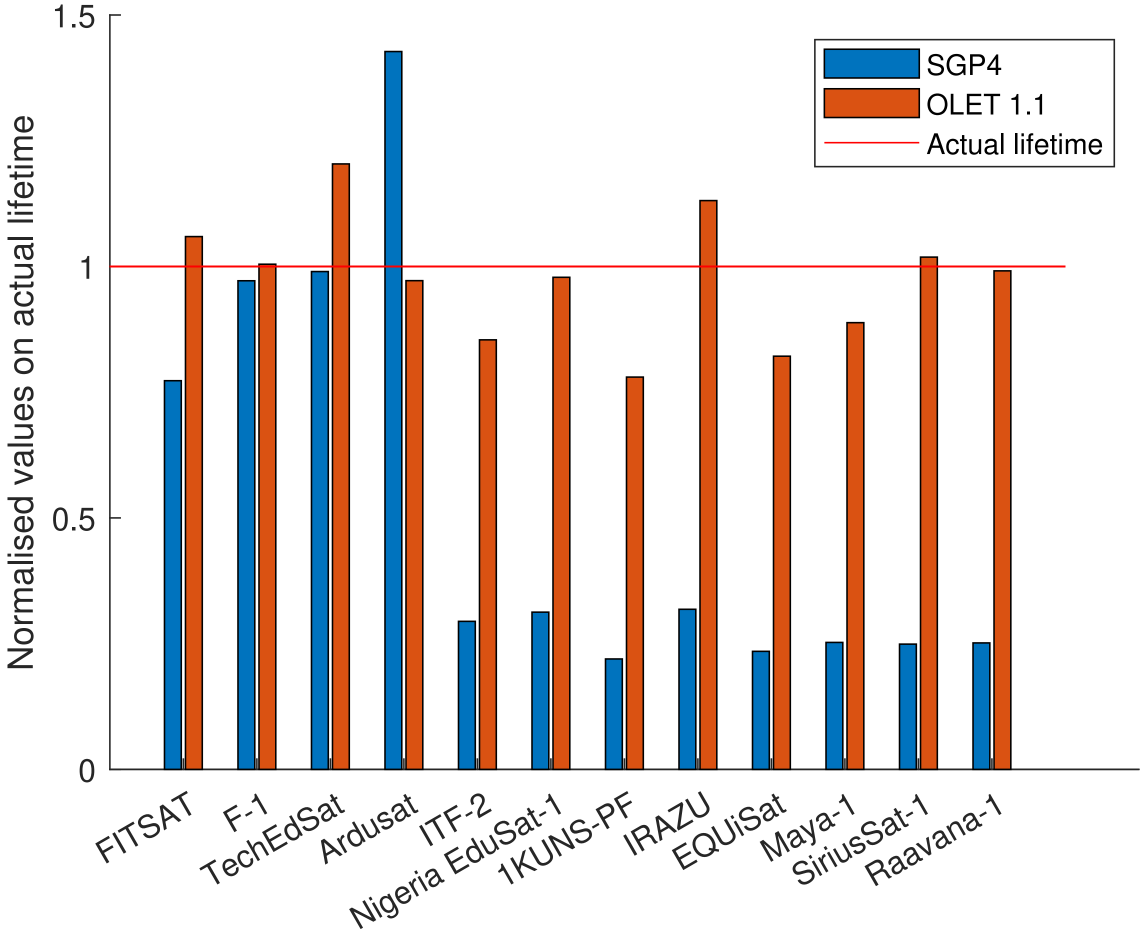

Figure 4.4: 1U CubeSat results

Deployment date	Lifetime (years)	Mass (kg)	Estimated Cross-section (m^2)	S/m (m^2/kg)
26/05/2017	1.921	2.26	0.020	0.0089

Table 4.3: BeEagleSat

4.2 2U CubeSats

Since 2Us have bigger dimensions than 1U CubeSats, usually 20 *cm x* 10 *cm x* 10 *cm*, they are somewhat more elaborated. Some of them may present deployable solar panels and an active attitude control. A set of 5 satellites was evaluated (table A.1). The BeEagleSat (table 4.3) is considered as a representative case and the trend of its orbital elements is shown in figure 4.5. The error between the actual lifetime and the one of calculated by the simulator is about 20 days. The considerably long time between deployment and re-entry, of about 700 days, hence almost 2 years, may be motivated because of the low cross-section to mass ratio of the satellite. Also, the deployment date from ISS corresponds to a period of solar minimum activity. The RAAN angle and ω variations are about the same as that of the satellite evaluated in the previous section. Eccentricity and inclination also maintain same values, particularly the 51.6° value for inclination which is obviously the same as that of the ISS, because of the assumptions of the previous chapter.

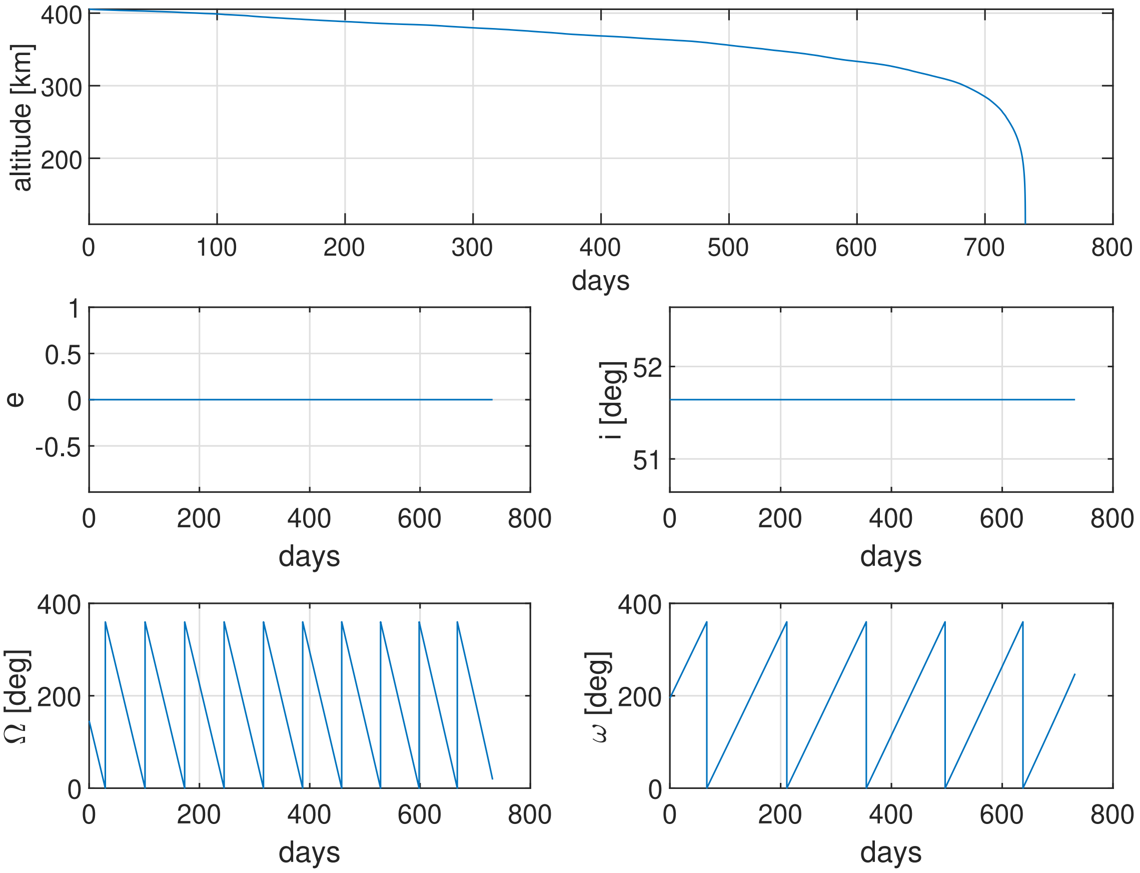

Figure 4.5: BeEagleSat orbital element variation

The 2U CubeSat relative error on the lifetime calculation, is about the same as 1Us. Furthermore, figure 4.6 shows, unlike 1Us, that the relative error decrease with increasing step size, and consequently decreasing the duration of the simulation. However, this result should be taken with a grain of salt since the satellite population on which OLET was tested is considerably smaller than the other categories. To maintain the same set up of the simulator as that for the 1U class, the integration step chosen is 180°, hence, the relative error is about 12 %.

The SGP4 was performed on the satellites of table A.1 imposing a mass of 2.00 kg and a cross-section of 0.02 m^2. As above, the initial conditions were the same of that in OLET simulations, hence, the ISS TLE on the date of deployment.

The outcome of SGP4 simulation is that the lifetime calculated is quite underestimated for all satellites of the 2U category, figure 4.7. In fact, referring to deployment date of table A.1, these CubeSats are all deployed during a solar minima (years 2017-2019). Therefore, the exponential model implemented in SGP4 provides density values almost one order of magnitude higher than that of a more accurate model, such as NRLMSISE, during a period of minimum solar activity, see figure 3.4.

Again, having as input for the NRLMSISE model some well-known measurements of $F_{10.9}$ and A_P, the OLET provides rather accurate results with respect to that of SGP4. However, as already mentioned, given the low number of satellites considered in this category, this results is less reliable than that for 1Us.

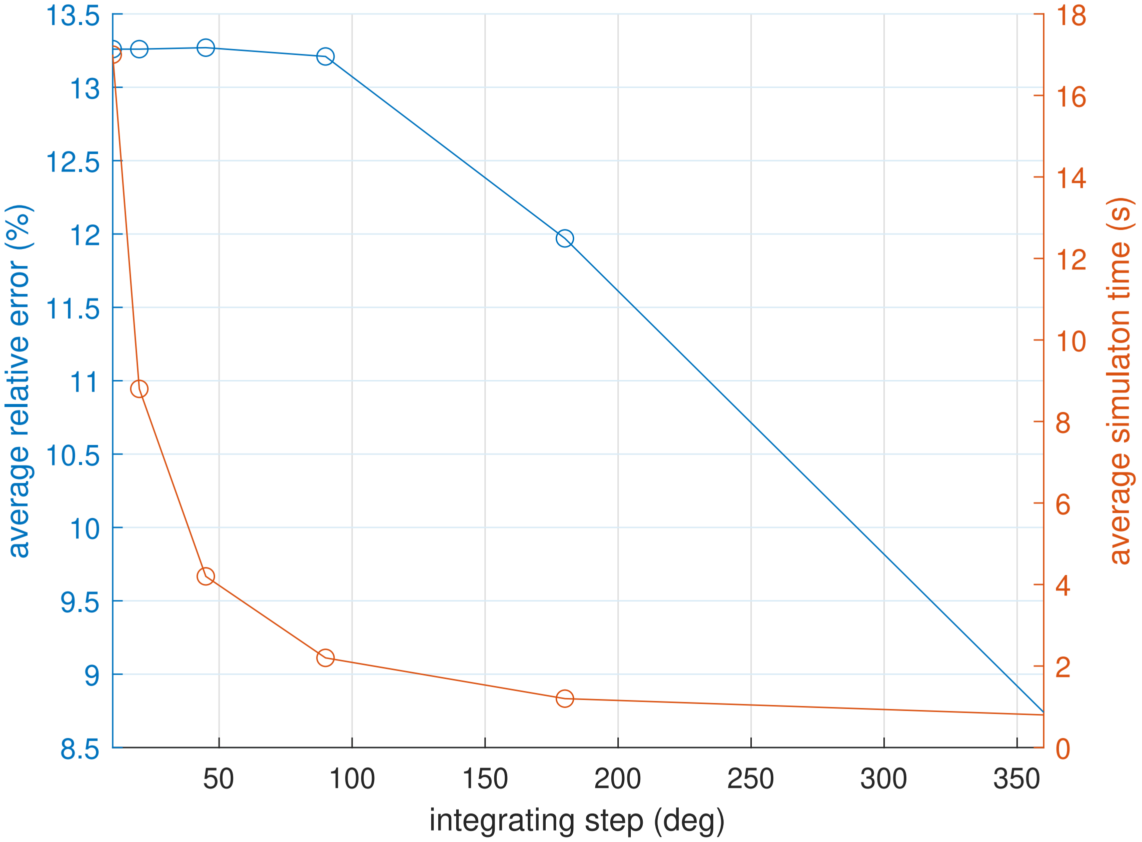

Figure 4.6: 2U simulation analysis, OLET v1.1

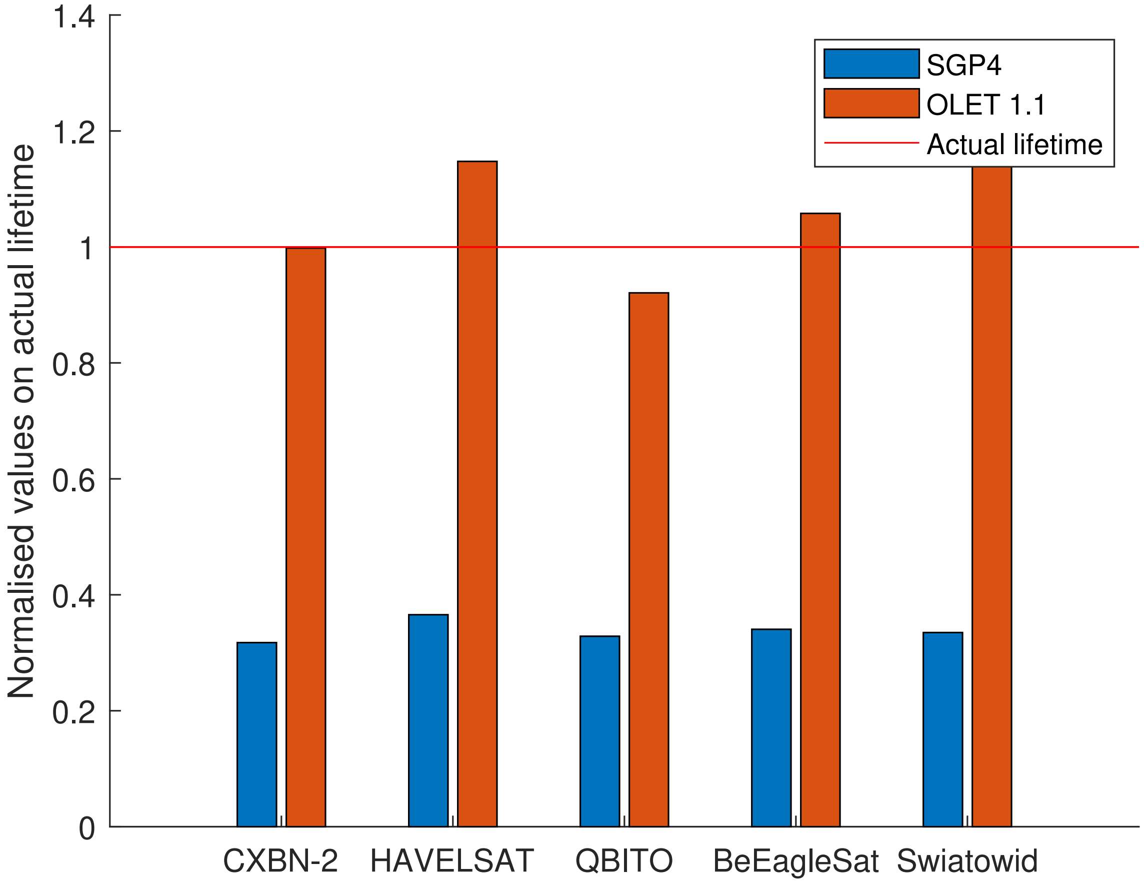

Figure 4.7: 2U CubeSat results

Deployment date	Lifetime (years)	Mass (kg)	Estimated Cross-section (m^2)	S/m (m^2/kg)
31/01/2019	2.723	3.56	0.029	0.0081

Table 4.4: UNITE

4.3 3U CubeSats

Eleven 3U CubeSats were considered (table A.3). Since their higher cross-sectional area to mass ratio, they usually tend to stay in orbit less than 1Us (table 4.1). On the other hand, because of the many variables involved, such as when the satellite is deployed, this is not always the case. Taking the orbital element variations of the UNITE CubeSat (table 4.4), it can be seen, in figure 4.8, that the altitude drops below

300 km after about 800 days. While in the case of ArduSat, the satellite re-enters the Earth's atmosphere in less than 150 days (figure 4.1). In fact, ArduSat mission was executed during a solar maxima whereas UNITE during a solar minima (figure 2.8). In contrast, the rate of change of Ω and ω, not influenced by the atmospheric density, is about the same as that of the Ardusat. In this case the UNITE lifetime is underestimated by OLET, since the real value is about 994 days (table 4.4) while the one calculated is almost 900 days (figure 4.8).

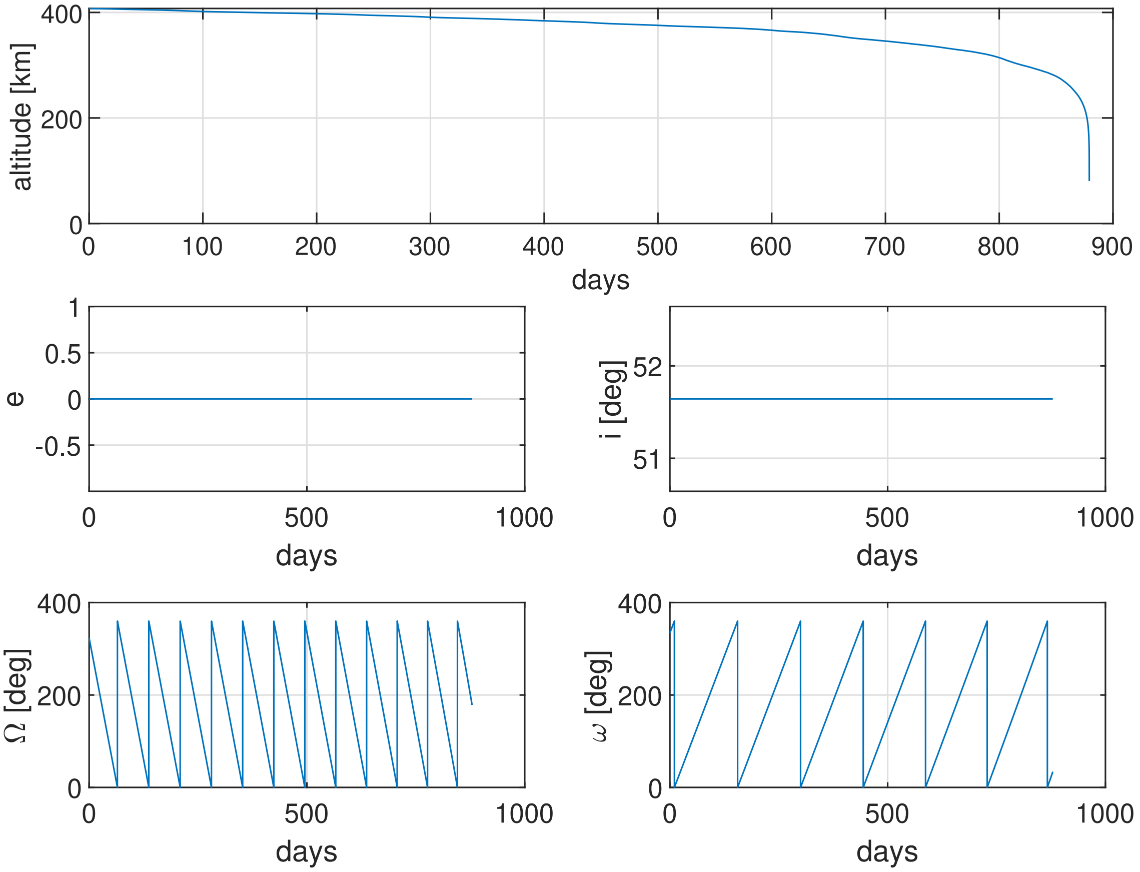

Figure 4.8: UNITE orbital element variation

For 3U CubeSats the average percentage relative error in the lifetime calculation by the OLET 1.1 (figure 4.9) is slightly greater than that for 1Us and 2Us. This is due to the higher uncertainty in the estimation of the cross-sectional area. Indeed, 3Us may present extension not negligible, furthermore the difference between the maximum and the minimum cross-sectional surfaces is considerable. It is no coincidence that the highest error is that of Icecube (figure 4.10), which has big solar panels (table A.3). In figure 4.9, a similar trend to that of figure 4.2 is shown, so the larger the step size, the larger the error and the shorter the simulation time. Choosing an integration step of 180° the OLET average relative error in calculating the lifetime of a 3U satellite is about 12.5 %.

SGP4 propagations are executed considering for each CubeSat a 4 kg mass and 0.03 m^2 cross-section, while initial conditions are imposed as for the categories above. Concerning Flock 1-1 and MicroMAS-1 (figure 4.10), the relative error is quite high because the cross-section entered as input departs considerably from the real values. Both satellites present indeed big solar panels. Instead, the simulation for SOAR CubeSat is quite accurate because of two factors canceling each other out. The period of solar minima leads SGP4 to underestimate the lifetime, whereas the cross-section underestimated value (indeed solar panels are neglected), leads SGP4 to overestimate the lifetime. Similar comments could be carried out for S-CUBE, MinXSS-1, CADRE and Icecube. In conclusion, considering UBAKUSAT, UNITE, Spooky-1 and EntrySat, all deployed during a period of solar minimum activity and without any kind of extension strongly influencing the cross-section. They actually remain in orbit longer than predicted by SGP4 due to the density overestimated by the exponential model.

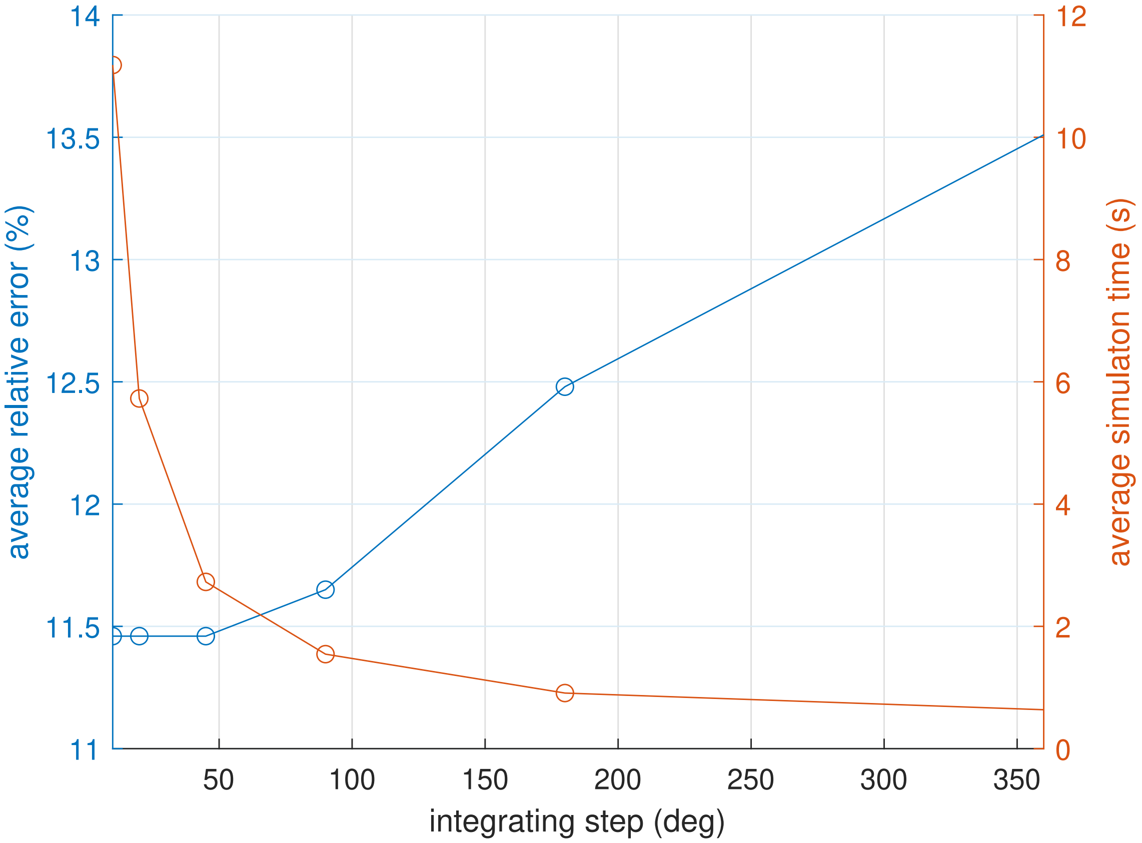

Figure 4.9: 3U simulation analysis, OLET v1.1

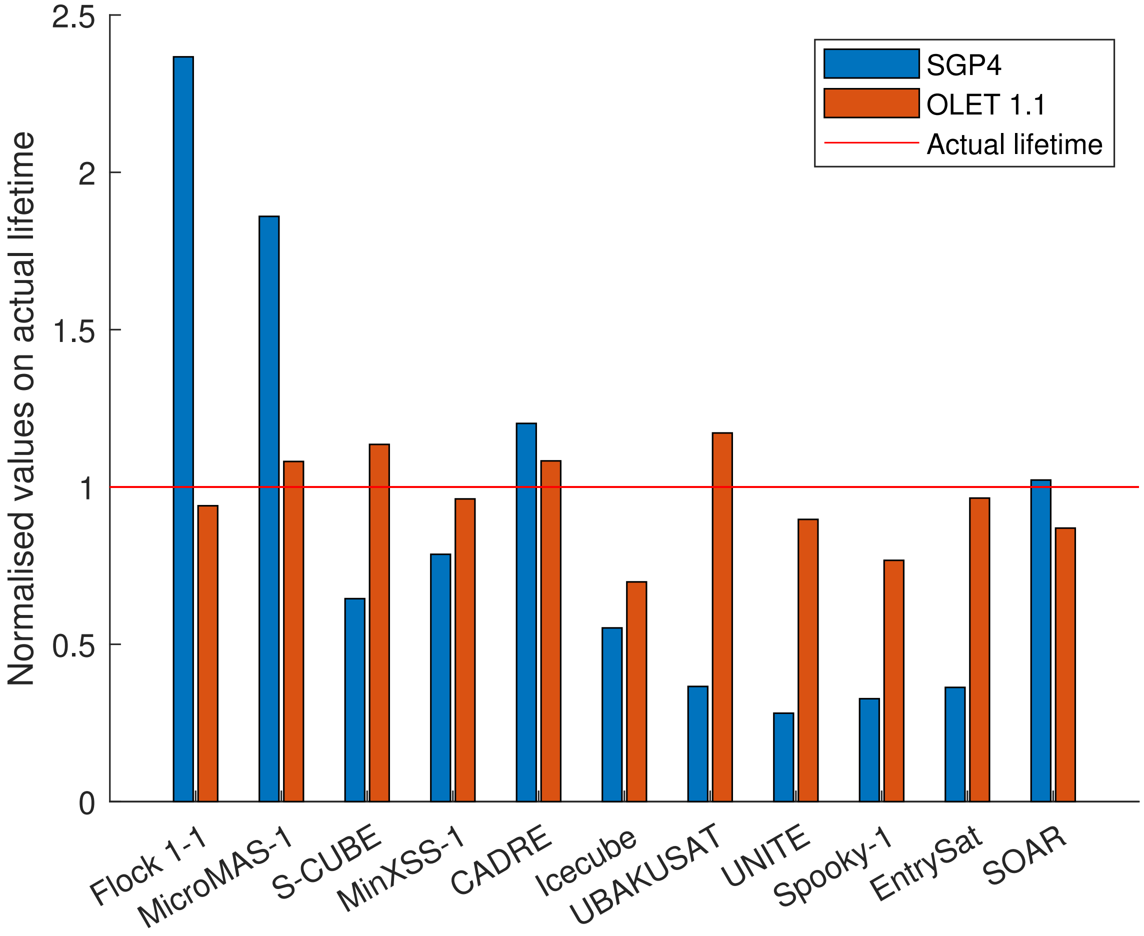

Figure 4.10: 3U CubeSats results

4.4 Other Satellites

Several satellites, which don't fall within previous categories, are reported in this last section (table A.4). These are many different in shape and size and some of them may have a drag sail or big antennas. These features contribute to make more complicate the cross-section estimation since it is also difficult to simulate when, during the mission, the extensions are fully deployed. The Tancredo-1 satellite (table 4.5) is taken as an example, unlike CubeSats it has a cylindrical shape. OLET underestimates the lifetime value of about 25 days (figure 4.11).

As far as this class of satellites is concerned, OLET is not reliable, as figure 4.12 shows, the average relative error is almost 40 %. This may be explained not only for the difficulties on the surface estimation but also for the drag coefficient selected. In fact, for CubeSats a C_D of 2.2 is considered quite successfully and for simplicity this assumption is extended also to satellites of this last category. However, until

Deployment date	Lifetime (years)	Mass (kg)	Estimated Cross-section (m^2)	S/m (m^2/kg)
16/01/2017	0.753	0.570	0.013	0.023

Table 4.5: Tancredo-1

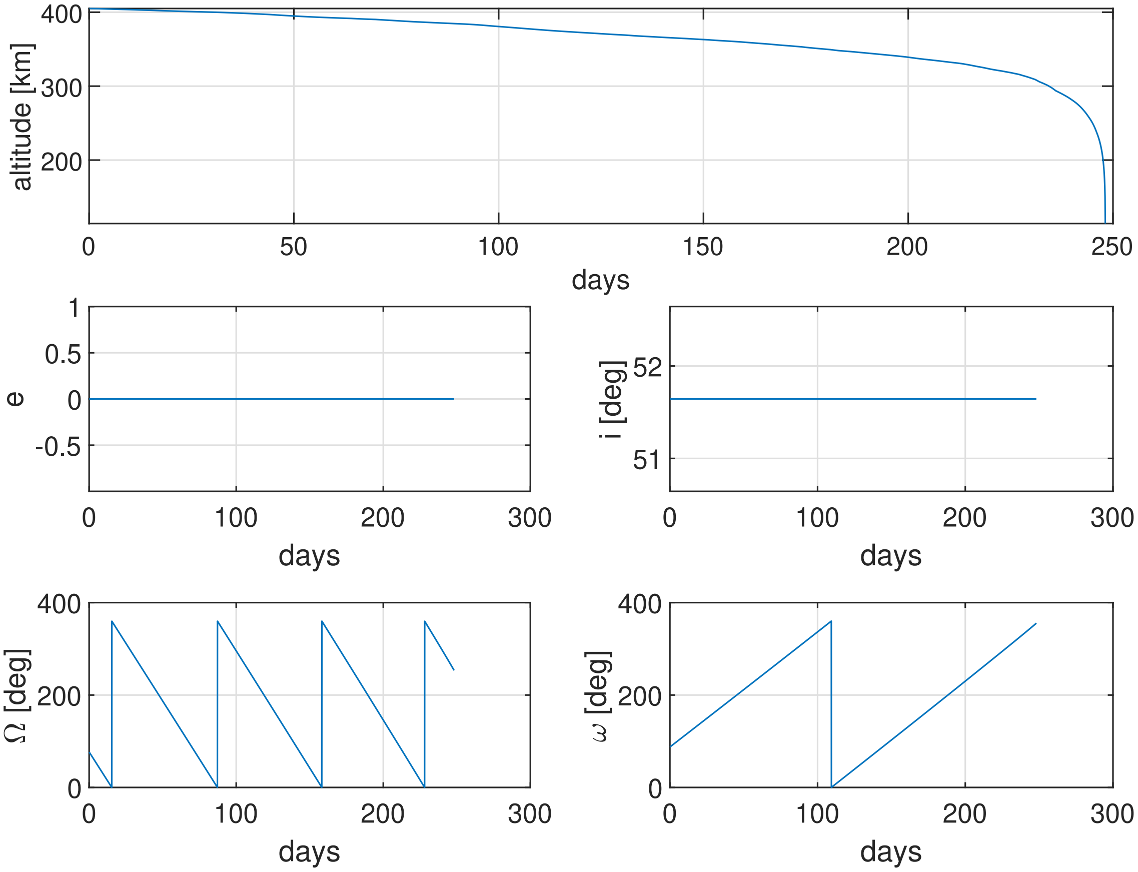

Figure 4.11: Tancredo-1 orbital element variation

the satellites had about the same shape (CubeSats) imposing a 2.2 value for C_D was a good assumption, this condition is less suitable for the category dealt with in this section.

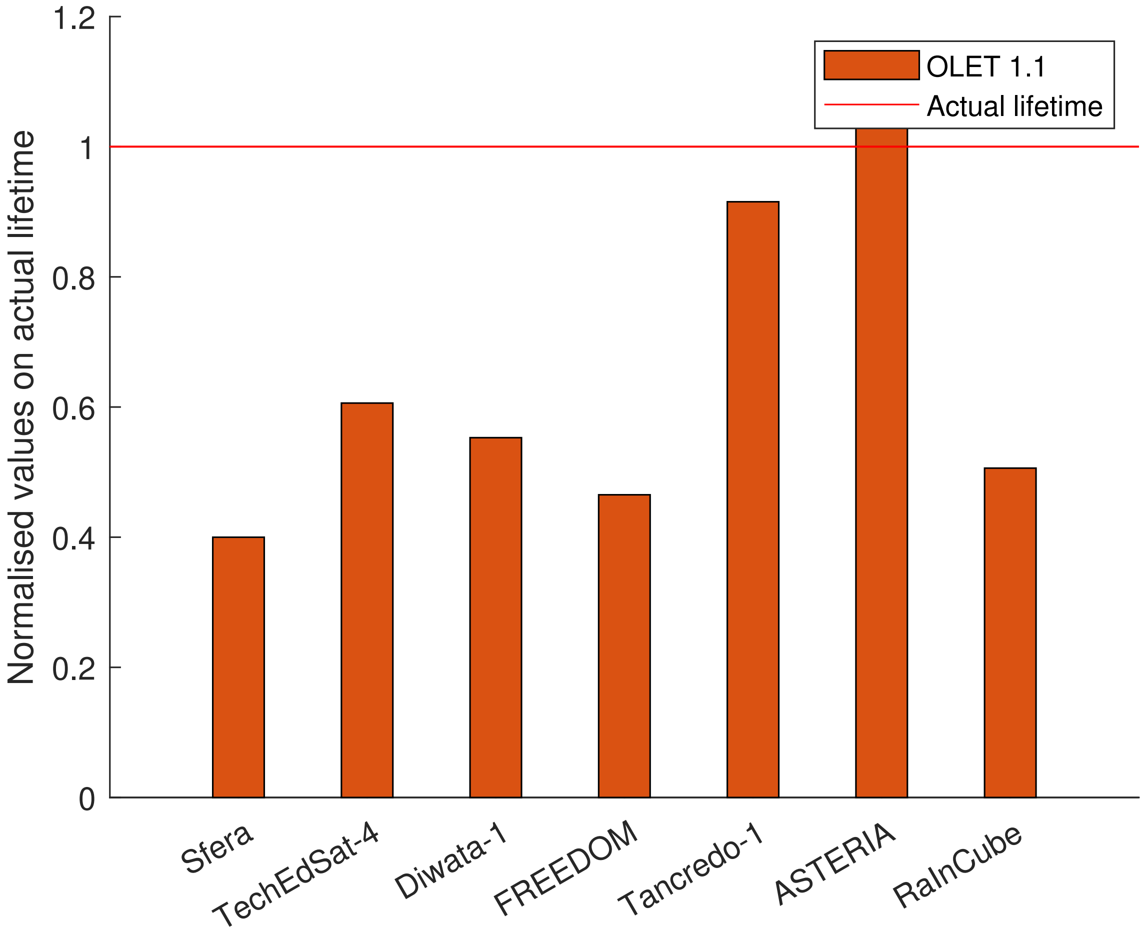

Figure 4.12: Others results

Chapter 5

Conclusions

Relying on what has been discussed in chapters above, conclusions are drawn. Initially, the work is summarised and commented to compare what has been achieved with the intended purpose (section 5.1). Following, some future developments to improve the quality of the work are identified and some final judgements are expressed (section 5.2).

5.1 Final work evaluations

To recap briefly, the work consisted in an initial bibliographic research to identify critical aspects, equations and approaches suitable to the study case, such as analytical and semianalytical techniques to investigate secular effects. Subsequently, the research explored the topic regarding the atmospheric drag, hence atmospheric models and methods to estimate the satellite cross-sectional area. Based on what was understood, the writing of the code followed, starting, initially, with simple and inaccurate models in the first versions. Then, the approach is made more complex and accurate by updating the new versions. To test OLET, a database of satellites, deployed from ISS and yet reentered the Earth's atmosphere, was realized through several searches, therefore obtaining some realistic input data to enter in the simulator. Lastly, the OLET simulations were performed, and a comparison with an other existing model, SGP4, was carried out too. Bibliographic research, development of the code, testing and comparison were steps following each other in a loop. A scheme of the work method is shown in figure 5.1.

OLET proved to be a tool performing simulations quickly, i.e. about 1 second to simulate the orbital decay of a satellite (tables 4.2, 4.6, 4.9). Whereas the duration of STK simulations in [11] is approximately 30 seconds. Also, OLET provides as an output a trend over time of satellite orbital elements. Their plotted variations are influenced by the main secular effects acting in the low Earth orbit. Consequently, the code can easily estimate the lifetime of a satellite. At the same time, the code is

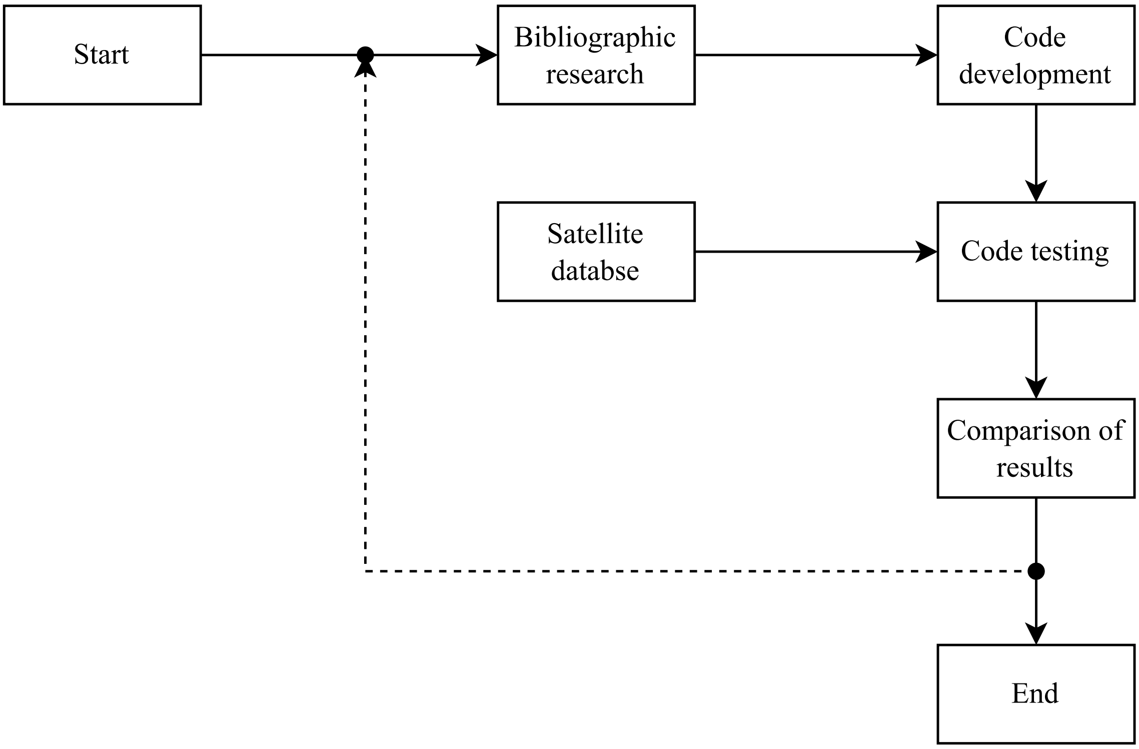

Figure 5.1: Method

quite intuitive to use, requiring as input few data, namely deployment date, satellite mass and cross-sectional area. Moreover, OLET allows to analyse multiple satellites, one after another, in one simulation. In fact, the user can write the input data of the satellites in an input file all at once (table 3.11). Thus, the aim introduced at the beginning of this document is achieved, although some improvements could still be made (section 5.2).

The results of the OLET test are rather positive. In fact, the average relative percentage error, regarding the lifetime calculation of CubeSats already reentered and reported in Appendix A, is generally below 15 %. Particularly, for 1Us the error is about 9.5 %; for 2Us is 12 %; and for 3Us 12.5 % (table 5.1). Although these errors could appear pretty high, it should be noted that the problem is full of uncertainties. Therefore, an achievable goal is only an approximate estimation of the time for which the CubeSat will be able to operate. The relative error for predictions would be higher than those obtained when testing OLET on satellites already decayed because the uncertainty is greater. In fact, one can no longer rely on solar flux and geomagnetic index measurements but only on forecasts. In table 5.1 a comparison between OLET and STK results is reported for completeness; STK is set according to [11]. Even though the main target of this study were regular CubeSats, also satellites with sails or different form were considered and grouped in the *other* satellites category. As far as this category is concerned, the relative percentage error is much more higher than 1Us, 2Us and 3Us. Probably, this could be explained since satellites of this class present different size and form, consequently, the uncertainty on the cross-section is higher. Moreover, drag coefficient should be evaluate differently than for CubeSats categories, in which was assessed a constant value of 2.2.

Analysing these results and comparing them with other simulators, the strength of OLET is the accuracy of the atmospheric model implemented, i.e. NRLMSISE-00, and the low simulation time. Other tools, such as SGP4, present similar simulation time as OLET, and this is dictated by the general techniques approach used. However, these models usually implement an exponential atmospheric model, which is certainly simpler but less accurate. Thus, although NRLMSISE is complex and require much more input than the exponential model, using it may be worth it to have more accurate results.

Table 5.1: OLET and STK comparison

Category	Number of satellites	OLET Relative percentage error	STK Relative percentage error
1U	12	9.50 %	42.7 %
2U	5	12.0 %	13.4 %
3U	11	12.5 %	52.8 %
Other	7	37.4 %	-

5.2 Future Developments

Despite the purpose of the work was achieved, as discussed above, still some improvements may be made to OLET. First of all, using a larger number of satellites to test the simulator may be helpful to detect any bugs in the code. Ideally the data of these satellites, such as mass, dimensions, attitude, should be well known.

Another interesting aspect to take into account could be the ΔV of deployment impressed by the deployer onboard ISS on the satellite. The assumption, that at the first instant of the simulation the orbital elements of satellite and ISS are exactly the same, would be no longer valid. However, it should be assessed whether this consideration would influence the satellite lifetime even slightly, or whether this aspect is entirely negligible.

A future development of OLET may be implementing an integration method other than Runge-Kutta, also using existing libraries. Thus, an assessment could be carried out to study the increase or decrease of accuracy and simulation time.

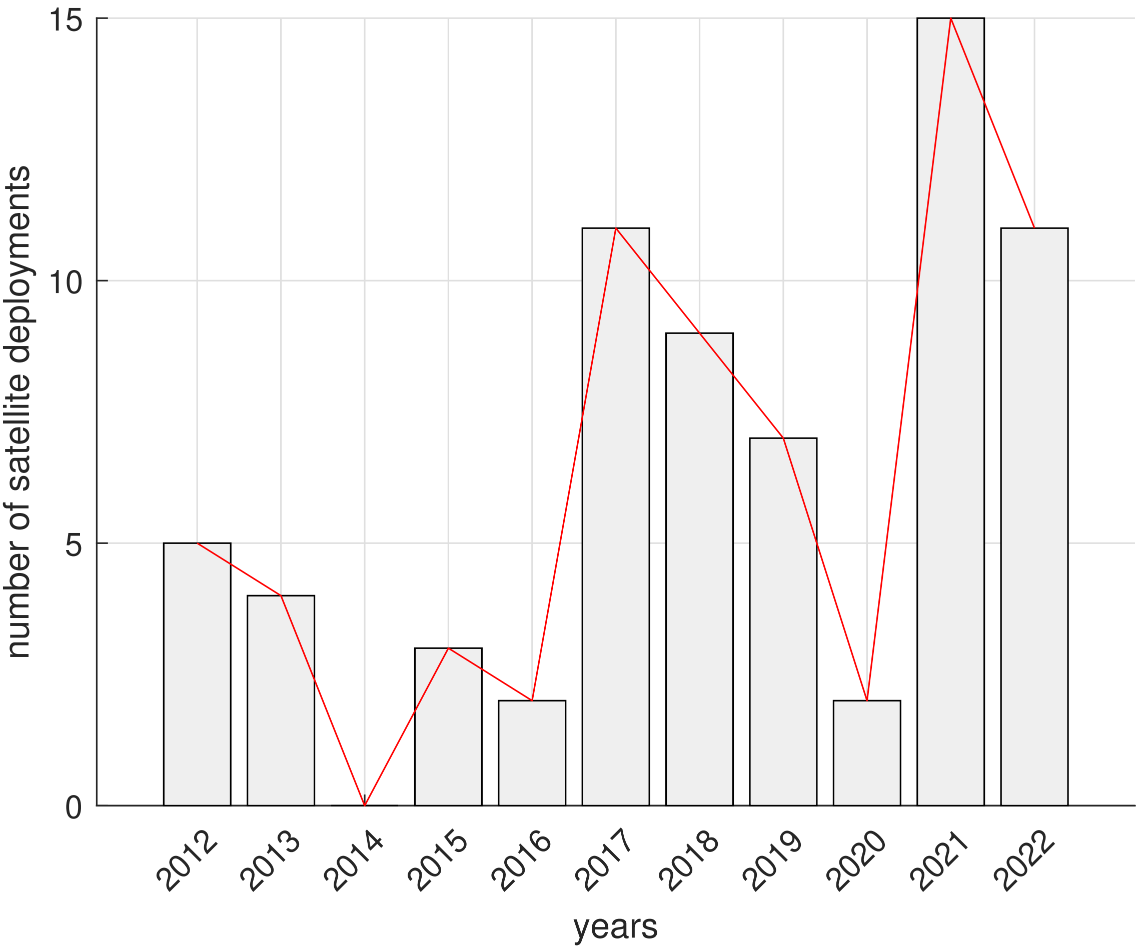

Figure 5.2: Number of satellites deployed with J-SSOD

The assumptions made in Chapter 3 neglect variations on the orbit eccentricity and inclination. Also, the trend of the mean anomaly due to J_2 is not considered in OLET. Lastly, only the main secular effects are taken into account neglecting all other periodic effects. These hypothesis were derived from the initial conditions, namely the ISS orbit, and the main objective to determine satellite lifetime. Thus, the equations implemented in the code may be changed to study deeper the variations on the orbital elements. This changes may allow to evaluate not only quasi-circular orbit, dictated from the deployment onboard ISS, but also other orbit with higher eccentricity and still with an altitude in the LEO range (below 2000 km). Therefore, in this case, also the gravitational effects due to the Moon and the Sun should be taken into account.

In conclusion, it might be interesting to make some improvements to OLET. Indeed, it could be a very useful tool in the years to come, since the increasing number of deployed satellites. As an example of this trend, figure 5.2 shows the number of deployments occurred over the past decade through the J-SSOD onboard ISS [32].

Appendix A

Satellite data

Tables reported in this appendix are the result of a bibliographic research. The aim is to provide an input to test OLET. When some data regarding mass and dimensions were missing, a standard value for that category was imposed. For example, 1 *kg* mass and 10cmx10cmx10cm size for 1Us.

Table A.1: 2U CubeSats

Name	Depl. date	lt (yrs)	m (*kg*)	Dim. (*cm*)	Attitude	Exten.	Ref.
CXBN-2	16/05/2017	1.792	2.00	10x10x20	Sun-Pointing	Solar Panels	[33] [34]
HAVEL-SAT	16/05/2017	1.789	2.30	10x10x20	Unknown	Small antenna	[34] [35]
QBITO	25/05/2017	1.732	1.80	10x10x20	Aligned to V-direction	Small antenna	[33]
BeEagle Sat	26/05/2017	1.921	2.26	10x10x20	Unknown	Small antenna	[34] [36]
Swia-towid	03/07/2019	1.699	2.00	10x10x20	Active Control	Solar Panels	[34]

Table A.2: 1U CubeSats

Name	Depl. date	lt (yrs)	m (kg)	Dim. (cm)	Attitude	Exten.	Ref.
FITSAT-1	04/10/2012	0.748	1.33	10x10x10	Permanent magnet	Small antenna	[33]
F-1	04/10/2012	0.595	1.00	10x10x10	Unknown	Small antenna	[37]
TechEd-Sat	04/10/2012	0.584	1.20	10x10x11	Unknown	Small antenna	[33]
Ardusat	19/11/2013	0.405	1.00	10x10x10	Unknown	Small antenna	[38]
ITF-2	16/01/2017	1.964	1.20	10x10x10	Unknown	No	[34]
Nigeria EduSat-1	07/07/2017	1.849	1.00	10x10x10	Unknown	No	[34]
1KUNS -PF	11/05/2018	2.633	1.00	10x10x11	Magnetic stab.	Small Antenna	[39] [40]
IRAZU	11/05/2018	1.816	1.00	10x10x10	Unknown	Small Antenna	[33]
EQUi-Sat	13/07/2018	2.463	1.00	10x10x10	Unknown	Small Antenna	[38]
Maya-1	10/08/2018	2.290	1.00	10x10x11	Permanent magnet	Small Antenna	[41] [42]
Sirius-Sat-1	15/08/2018	2.321	1.45	10x10x10	Unknown	Instruments	[40] [43]
Raavana-1	17/06/2019	2.299	1.10	10x10x11	Unknown	Small Antenna	[44]

Table A.3: 3U CubeSats

Name	Depl. date	lt (yrs)	m (kg)	Dim. (cm)	Attitude	Exten.	Ref.
Flock 1-1	11/02/2014	0.323	5.00	10x10x34	Magnetic stab.	Solar panels	[33] [45]
Micro-MAS-1	04/03/2015	0.411	4.25	10x10x34	Aligned to V-direction	Solar panels	[33] [34]
S-CUBE	23/11/2016	1.186	3.99	10x10x30	G.G. stab.	Solar panels	[33] [46]
MinXSS-1	16/05/2016	0.973	3.50	10x10x34	Sun-pointing	Solar panels	[33]
CADRE	16/05/2016	0.636	4.00	10x10x34	Unknown	Solar panels	[34] [33]
Icecube	16/05/2017	1.384	4.00	10x10x34	USun-pointing	Solar panels	[33]
UBAKU-SAT	11/05/2018	2.088	4.00	10x10x30	Unknown	Small Antenna	[34]
UNITE	31/01/2019	2.723	3.56	10x10x34	Unknown above 300km	Small fins	[47]
Spooky-1	17/06/2019	2.334	2.60	10x10x34	Passive control	Small Antenna	[34] [48]
EntrySat	04/07/2019	2.104	3.00	10x10x34	Magnetic stab.	No	[34] [49]
SOAR	14/06/2021	0.748	3.36	10x10x30	Aligned to V-direction	Solar panels	[34] [50]

Table A.4: Other satellites

Name	Depl. date	lt (yrs)	m (kg)	Dim. (cm)	Attitude	Exten.	Ref.
Sfera	23/08/2012	0.271	13.0	53	Unknown	No	[37]
TechEd Sat-4	03/03/2015	0.085	2.68	65.5	Aligned to V-direction	Sail	[33]
Diwata-1	27/04/2016	4.044	50.0	55x35x55	Active control	No	[33]
FREEDOM	16/01/2017	0.055	1.30	500	Unknown	Sail	[34] [36]
Tancredo-1	16/01/2017	0.753	0.57	9,25x12,7	Unknown	Small antenna	[34]
ASTERIA	20/11/2017	2.427	10.2	24x12x37	Active control	Solar panels	[33] [38]
RaInCube	13/07/2018	2.452	5.50	25x22x10	Nadir-Pointing	Antenna and solar panels	[33]

Appendix B

Code

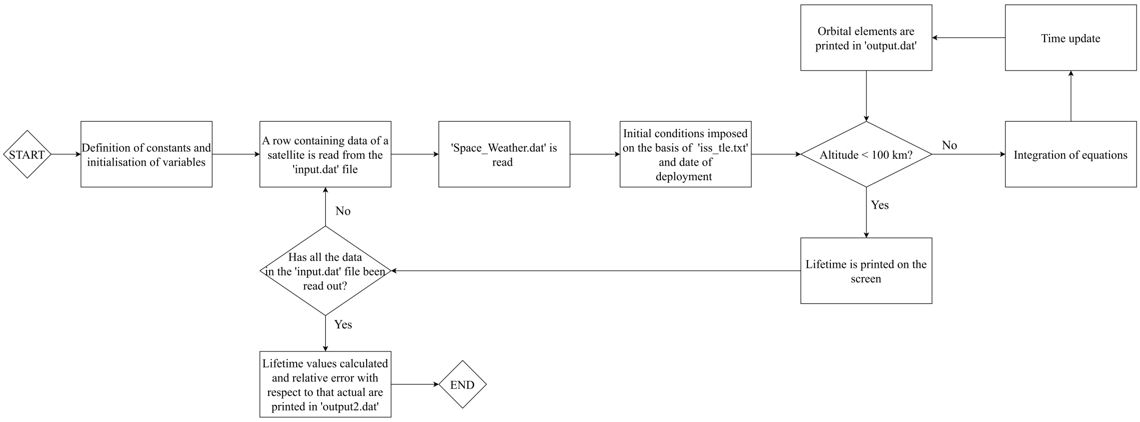
START
Definition of constants and initialisation of variables
A row containing data of a satellite is read from the 'input.dat' file
'Space_Weather.dat' is read
Initial conditions imposed on the basis of 'iss_tle.txt' and date of deployment
Altitude < 100 km?
No
Integration of equations
Time update
Orbital elements are printed in 'output.dat'
Yes
Lifetime is printed on the screen
Has all the data in the 'input.dat' file been read out?
No
Yes
Lifetime values calculated and relative error with respect to that actual are printed in 'output2.dat'
END

Figure B.1: OLET scheme

```

/* OLET (1.1)
This code was developed to study the long term behavior of satellites in the LEO environment; the focus was on satellites deployed from ISS. Given the nature of the problem, a study was carried out on the secular variation of orbital elements. Two body problem, perturbations are considered (drag and J2 effect). Atmospheric model: NRLMSISE-2000 (Dominik Brodowski's library). According to deployment date and time, ISS TLE is identified. NORAD data base is used. ISS orbital elements are obtained from TLE and are set as the initial condition of the satellite studied. Gauss'form of Lagrange planetary equations is used [2]. Results are printed in a .dat file. Furthermore, a Matlab script "output.m" can be run to plot the altitude decay and others orbital elements.

```

```
References
[1]------------------> Approximate optimal LEO transfers with J2 perturbation and dragsail. Authors: Casalino, Forestieri
[2]------------------> Vallado; Fundamentals of Astrodynamics and Applications
[3]------------------> Bate,Muller; Fundamentals of Astrodynamics
[4]------------------> Cornelisse; Rocket Propulsion and Spaceflight Dynamics
*/

#include "orbit_simulator_functions.c"

/*CONSTANTS*/
#define k_lim 20000          /* Max number of elapsed orbits
#define num_var 6            /* Number of orbital elements


/*MAIN*/
int main ()
{

    /*
    fp ------------------------------------------> .dat output file to show the trend of orbital elements
    fp2 ----------------------------------------> .dat output file to show the error between the actual lifetime and the one calculated by Orbit Simulator
    x[num_var] --------------------------------> [a(km),e,i(deg),RAAN(deg),omega(deg),ni(deg)]
    X[num_var][2] ---------------------------> [a[k], e[k], i[k], RAAN[k], omega[k], ni[k];a[k+1], e[k+1], i[k+1], RAAN[k+1], omega[k+1], ni[k+1]]
    delta2piX[num_var]-------------------------> X variation in one orbit
    integral ----------------------------------> variable which contains the integral value of the semi-axis formula from [2]
    t ------------------------------------------> time counter (days)
    sat_dat[ROW4][COL4] --------------------> matrix which contains satellite data: deployment date, actual lifetime, mass, cross section
    yr,mo,day,hr,min,sec --------------------> date counters
    dyr ----------------------------------------> day of the year
    T ------------------------------------------> orbit period
    weather[ROW3][COL3] --------------------> space weather matrix
    rho ----------------------------------------> density [kg/m3]
    m,S,CD ------------------------------------> satellite mass (kg), satellite cross-sectional area (m2), drag coefficient
    k,i,sat ------------------------------------> counters
    lifetime[ROW4] ---------------------------> lifetime calculated by the Orbit Simulator for each satellite evaluated
    error[ROW4] ------------------------------> error between the actual lifetime and the one calculated by the Orbit Simulator for each satellite evaluated
```

```
      output[2] ----------------------> delta_semi_axis_RK4 function output, the first element is the semi-major axis in one orbit, the second is
                                                    that orbit period
     */

     FILE* fp;
     FILE* fp2;
     double x[num_var],X[num_var][2],delta2piX[num_var],rho=1,t=0, weather[ROW3][COL3],sat_dat[ROW4][COL4],error[ROW4],CD,S,m,dyr=0, sec,T,lifetime[ROW4],output[2];
     int k,i,yr,mo,day,hr,min,sat;
     CD = 2.2;


     /*"output.dat" file is open*/
     fp = fopen("output.dat","w");
     /*
     "output.dat" structure:
     a(km)  e  i(deg)  RAAN(deg)  omega(deg)  ni(deg)  t(day)  k(orbit number)
     */
     if (fp == NULL)
     {
         printf("FILE OPENING ERROR");
     }
     else
     {
     /*satellites data are read from an input file*/
         read_satellite_data(sat_dat);
         for(sat=0;sat<ROW4;sat++)
         {
             day = (int)sat_dat[sat][1];
             mo = (int)sat_dat[sat][2];
             yr = (int)sat_dat[sat][3];
             m = sat_dat[sat][5];
             S = sat_dat[sat][6];
             hr = 0; // by default
             min = 0; // by default
             sec = 0; // by default
             fprintf(fp,"SATELLITE %d \n",(int)sat_dat[sat][0]);



/*Space weather and ISS TLE files are read*/
             read_Space_Weather(weather,yr);
             read_ISS_tle(x, yr, mo, day); //Initial orbital elements are set according to ISS TLE

```

```

/*Initial orbital elements are passed to matrix X*/
            for (i=0;i<6;i++)
            {
                X[i][0] = x[i];
            }

/*Propagation*/
            k = 0;
            t = 0;
            T = 0;
            dyr = 0;
            time_update(T,&t,&yr,&mo,&day,&dyr,&hr,&min,&sec); //initial day of year (dyr) is calculated
            rho = density(x,weather,dyr,yr,mo,day,hr,min,sec); //density calculation through NRLMSISE-00 model (initial value)
            delta2piX[0] = 0;
            while (k<k_lim)
            {
                for (i=0;k>0&&i<6;i++)
                {
                    x[i] = X[i][0]; //matrix elements are passed to the x vector which will be used in the following functions
                }
                k = k+1; //counter update


                if (rho<0) //if rho<0 it means that the satellite altitude is below 100 km, the propagation ends
                {
                    k = k_lim;
                    lifetime[sat] = t/360;
                    error[sat] = lifetime[sat]/sat_dat[sat][4];
                    printf("\nLifetime Satellite %d: %f years", (int)sat_dat[sat][0],lifetime[sat]);
                }

                for (i=0;i<6&&k<k_lim;i++)
                {
                    switch (i) //Gauss' planetary equations according to [2]
                    {
                        case 0: //semi-major axis equation
                            delta_semi_axis_RK4(output,x,weather,dyr,yr,mo,day,hr,min,sec,CD,S,m); //RungeKutta4 on semi-major axis equation
                            delta2piX[i] = output[0]; //semi-major axis variation in one orbit
                            T = output[1]; //time elapsed in one orbit
```

```c
                        time_update(T,&t,&yr,&mo,&day,&dyr,&hr,&min,&sec); //time is updated considering the orbit period T
                        if(delta2piX[i]==-1e5) //if delta2piX[i]=-1e5 it means that the satellite altitude is below 100 km
                        {
                            i = 6;
                            rho = -1;
                            delta2piX[i] = 0;
                        }
                        else
                        {
                            X[i][1] = X[i][0]+ delta2piX[i]; //orbital element is updated
                            if(X[0][1]>=6428)
                            {
                                fprintf(fp,"%f ",X[i][1]);
                            }
                            else //the satellite altitude is below 50 km
                            {
                                i = 6;
                                rho = -1;
                            }
                        }
                        break;
                    case 1: //eccentricity
                        delta2piX[i] = 0;
                        X[i][1] = X[i][0]+ delta2piX[i]; // orbital element is updated
                        fprintf(fp,"%f ",X[i][1]);
                        break;
                    case 2: //inclination
                        delta2piX[i] = 0;
                        X[i][1] = X[i][0]+ delta2piX[i]; // orbital element is updated
                        X[i][1] = X[i][1] - (floor(X[i][1]/360)*360); //Angles are imposed between 0 and 360 degrees
                        fprintf(fp,"%f ",X[i][1]);
                        break;
                    case 3: //RAAN
                        delta2piX[i] = -3*pi*J2*pow((R/X[0][0]),2)*cos(X[2][0]*(pi/180))*(180/pi);
                        X[i][1] = X[i][0]+ delta2piX[i]; // orbital element is updated
                        X[i][1] = X[i][1] - (floor(X[i][1]/360)*360); //Angles are imposed between 0 and 360 degrees
                        fprintf(fp,"%f ",X[i][1]);
                        break;
                    case 4: //argument of periapsis
```

```
                            delta2piX[i] = (3/2)*pi*J2*pow((R/X[0][0]),2)*((5*pow(cos(X[2][0]*(pi/180)),2))-1)*(180/pi);
                            X[i][1] = X[i][0]+ delta2piX[i]; // orbital element is updated
                            X[i][1] = X[i][1] - (floor(X[i][1]/360)*360); //Angles are imposed between 0 and 360 degrees
                            fprintf(fp,"%f ",X[i][1]);
                            break;
                        case 5: //true anomaly
                            delta2piX[i] = 0;
                            X[i][1] = X[i][0]+ delta2piX[i]; // orbital element is updated
                            X[i][1] = X[i][1] - (floor(X[i][1]/360)*360); //Angles are imposed between 0 and 360 degrees
                            fprintf(fp,"%f %f %d\n",X[i][1],t,k);
                            break;
                        default:
                            printf("main switch error");
                    }

                    X[i][0] = X[i][1]; // Matrix update

                }
            }
        }
        fclose(fp);
    }
/*lifetime errors are printed in the second output file*/
    fp2 = fopen("output2.dat","w");
    for(sat=0;sat<ROW4;sat++)
    {
        fprintf(fp2,"%f %f %f\n",lifetime[sat],sat_dat[sat][4],error[sat]);
    }
    fclose(fp2);
    printf("\n Now you can run 'output.m'");

return 0;
}
```

Bibliography

[1] NanoRacks CubeSat Deployer. «Interface Definition Document (IDD)». In: *NanoRacks LLC* (2018) (cit. on p. 1).

[2] Nanoracks Kaber Deployment System. «Interface Definition Document (IDD)». In: *NanoRacks LLC* (2016) (cit. on p. 1).

[3] Nanoracks Airlock. «Interface Definition Document (IDD)». In: *NanoRacks LLC* (2021) (cit. on p. 1).

[4] N CubeSat and L CubeSat Initiative. *101: Basic concepts and processes for first-time CubeSat developers. 2017* (cit. on p. 1).

[5] JAXA. *JEM Small Satellite Orbital Deployer (J-SSOD).* URL: `https://humans-in-space.jaxa.jp/en/biz-lab/experiment/facility/ef/jssod/` (cit. on p. 1).

[6] Jerry Wright. *NanoRacks CubeSats Deployed from Station.* URL: `https://www.nasa.gov/content/nanoracks-cubesats-deployed-from-station` (cit. on p. 2).

[7] Mark Garcia. *The Nanoracks Bishop Airlock in the grips of the Canadarm2 robotic arm.* URL: `https://www.nasa.gov/image-feature/the-nanoracks-bishop-airlock-in-the-grips-of-the-canadarm2-robotic-arm` (cit. on p. 3).

[8] Nanoracks. *ISS Satellite Launch Services.* URL: `https://nanoracks.com/products/iss-launch/` (cit. on p. 3).

[9] Kristine Rainey. *Deploying Small Satellites From ISS.* URL: `https://www.nasa.gov/mission_pages/station/research/benefits/cubesat` (cit. on p. 3).

[10] *Launch Schedule ISS Cargo.* URL: `https://www.rocketlaunch.live/?includePast=1&tag=iss-cargo` (cit. on p. 3).

[11] J.Cerzosimo. «Influence factors on CubeSat lifetime deployed from ISS». In: (2023) (cit. on pp. 4, 67, 68).

[12] Roger R Bate, Donald D Mueller, Jerry E White, and William W Saylor. *Fundamentals of astrodynamics.* Courier Dover Publications, 2020 (cit. on pp. 6, 7, 11, 48).

[13] David A Vallado. *Fundamentals of astrodynamics and applications.* Vol. 12. Springer Science & Business Media, 2001 (cit. on pp. 7, 11, 15, 17, 19, 22, 27, 30, 31, 35, 38, 42, 48, 50, 51).

[14] Daniel L Oltrogge and Chia-Chun Chao. «Standardized approaches for estimating orbit lifetime after end-of-life». In: *Paper AAS 07-261 presented at the AAS/AIAA Space Flight Mechanics Conference. Mackinac, MI.* 2007 (cit. on pp. 22, 28, 40).

[15] United States. National Oceanic, Atmospheric Administration, and United States. Air Force. *US Standard Atmosphere, 1976.* Vol. 76. 1562. National Oceanic and Atmospheric Administration, 1976 (cit. on p. 25).

[16] L. G. Jacchia. «Revised Static Models of the Thermosphere and Exosphere with Empirical Temperature Profiles». In: *SAO Special Report* 332 (May 1971) (cit. on p. 26).

[17] JO Cappellari, CE Vélez, and Arthur J Fuchs. *Mathematical theory of the Goddard trajectory determination system.* Vol. 71106. Goddard Space Flight Center, 1976 (cit. on p. 26).

[18] RE Shanklin Jr, T Lee, MK Mallick, RA Kuseski, and JO Cappellari Jr. «A comparative study of the Harris-Priester, Jacchia-Roberts, and MSIS atmospheric density models in the context of satellite orbit determination». In: *NASA. Goddard Space Flight Center Sixth Ann. Flight Mech.(Estimation Theory Symp.* 1981 (cit. on p. 26).

[19] JM Picone, DP Drob, RR Meier, and AE Hedin. *NRLMSISE-00: A new empirical model of the atmosphere.* 2003 (cit. on p. 26).

[20] Lorenzo Casalino and Andrea Forestieri. «Approximate optimal LEO transfers with J2 perturbation and dragsail». In: *Acta Astronautica* 192 (2022), pp. 379–389 (cit. on pp. 36, 51).

[21] Leon Blitzer. «Handbook of orbital perturbations». In: *University of Arizona* (1970) (cit. on p. 36).

[22] Kenneth Moe and Mildred M Moe. «Gas–surface interactions and satellite drag coefficients». In: *Planetary and Space Science* 53.8 (2005), pp. 793–801 (cit. on p. 36).

[23] D.Brodowski. *NRLMSISE-00.* URL: https://www.brodo.de/space/nrlmsise/ (cit. on p. 37).

[24] JM Picone, AE Hedin, D Pj Drob, and AC Aikin. «NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues». In: *Journal of Geophysical Research: Space Physics* 107.A12 (2002), SIA–15 (cit. on p. 38).

[25] T.S. Kelso. *EOP and Space Weather Data.* URL: `https://celestrak.org/SpaceData/` (cit. on p. 38).

[26] T.S. Kelso. *TLE.* URL: `https://celestrak.org/NORAD/documentation/tle-fmt.php` (cit. on p. 47).

[27] T.S. Kelso. *ISS TLE.* URL: `https://celestrak.org/NORAD/archives/request.php?FORMAT=2le` (cit. on p. 47).

[28] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis.* Cengage learning, 2015 (cit. on p. 48).

[29] Meysam Mahooti. *SGP4 - MathWorks.* URL: `https://www.mathworks.com/matlabcentral/fileexchange/62013-sgp4` (cit. on p. 55).

[30] Felix R Hoots, Ronald L Roehrich, and TS Kelso. «Spacetrack report no. 3». In: *Project Spacetrack Reports, Office of Astrodynamics, Aerospace Defense Center, ADC/DO6, Peterson AFB, CO* 80914 (1980), p. 14 (cit. on pp. 55, 57).

[31] MH Lane, PM Fitzpatrick, and JJ Murphy. *On the representation of air density in satellite deceleration equations by power functions with integral exponents.* Tech. rep. AIR PROVING GROUND CENTER EGLIN AFB FL, 1962 (cit. on p. 57).

[32] *History of deployed CubeSats.* URL: `https://humans-in-space.jaxa.jp/en/biz-lab/experiment/strategy/j-ssod/history/` (cit. on p. 71).

[33] ESA. *eoPortal: Earth Observation Mission.* URL: `https://www.eoportal.org/` (cit. on pp. 72–75).

[34] Erik Kulu. *Airtable- Database.* URL: `https://airtable.com/shrafcwXODMMKeRgU/tbldJoOBP5wlNOJQY` (cit. on pp. 72–75).

[35] Emre Baceski et al. «HAVELSAT: A software defined radio experimentation CubeSat». In: *2015 7th International Conference on Recent Advances in Space Technologies (RAST).* IEEE. 2015, pp. 831–834 (cit. on p. 72).

[36] B. Karabulut1 A.R Aslan1. *QB50 Project, TR01/TR02, BeEagleSAT and HavelSat.* URL: `http://www.unisec-global.org/pdf/uniglo4/day2/16_uludag.pdf` (cit. on pp. 72, 75).

[37] NASA. *NSSDCA Master Catalog.* URL: `https://nssdc.gsfc.nasa.gov/nmc/SpacecraftQuery.jsp` (cit. on pp. 73, 75).

[38] Gunter Krebs. *Gunter's Space Page - Information on Launch vehicles, Satellites, Space Shuttle and Astronautics.* URL: `https://space.skyrocket.de/index.html` (cit. on pp. 73, 75).

[39] J Mwangi Mbuthia and Heywood Ouma. «1KUNS-PF: 1 st Kenyan University NanoSatellite-Precursor Flight». In: *Cell* 254 (2016) (cit. on p. 73).

[40] Chris Peat. *Heavens Above*. URL: `https://www.heavens-above.com/?lat=0&lng=0&loc=Unspecified&alt=0&tz=UCT` (cit. on p. 73).

[41] MH Azami, G Maeda, P Faure, T Yamauchi, S Kim, H Masui, Mengu Cho, et al. «Birds-2: A constellation of joint global multi-nation 1u cubesats». In: *Journal of Physics: Conference Series*. Vol. 1152. 1. IOP Publishing. 2019 (cit. on p. 73).

[42] *Live Real Time Satellite Tracking and Predictions*. URL: `https://www.n2yo.com/` (cit. on p. 73).

[43] *Sputnix*. URL: `https://sputnix.ru/en/satellites-sputnix/in-orbit/siriussat-1-en` (cit. on p. 73).

[44] *In-The-Sky.org*. URL: `https://in-the-sky.org/index.php` (cit. on p. 73).

[45] *Flock 1/2 – Planet Labs Earth Observation Satellites*. URL: `https://spaceflight101.com/flock/` (cit. on p. 74).

[46] *S3 – Shootingstar Sensing Satellite*. URL: `https://spaceflight101.com/spacecraft/$%5E3$-shootingstar-sensing-satellite/` (cit. on p. 74).

[47] Glen Kissel, Ryan Loehrlein, Nathan Kalsch, Wyatt Helms, Zack Snyder, and Sujan Kaphle. «UNITE CubeSat: From Inception to Early Orbital Operations». In: (2019) (cit. on p. 74).

[48] Huai Ying Lim et al. «Thermo-mechanical design for a miniaturized quantum light source on board the SpooQy-1 CubeSat». In: *arXiv preprint arXiv:2006.14442* (2020) (cit. on p. 74).

[49] Y Prevereaud, F Sourgen, D Mimoun, A Gaboriaud, JL Verant, and JM Moschetta. «Predicting the Atmospheric Re-entry of Space Debris Through the QB50 EntrySat Mission». In: *6th European Conference on Space Debris, Darmstadt, Germany*. 2013, pp. 22–25 (cit. on p. 74).

[50] *SOAR (SATELLITE FOR ORBITAL AERODYNAMICS RESEARCH)*. URL: `https://discoverer.space/soar-satellite-for-orbital-aerodynamics-research/` (cit. on p. 74).