UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Telecomunicació
i Aeroespacial de Castelldefels

TRABAJO DE FIN DE GRADO

TFG TITLE: Determination of the along-orbit average density of the thermosphere

DEGREE: Grado en Ingeniería de Sistemas Aeroespaciales

AUTHOR: Adrián Martínez Calzado

ADVISORS: Jordi Gutiérrez Cabello
Pilar Gil Pons

DATE: September 8, 2020

Título: Determinación de la densidad promedio de la termosfera a lo largo de la órbita
Autor: Adrián Martínez Calzado

Directores: Jordi Gutiérrez Cabello
Pilar Gil Pons

Fecha: 8 de septiembre de 2020

Resumen

Hoy en día, el estudio de la dinámica de la atmósfera superior es muy importante a la hora estudiar con precisión la trayectoria de satélites en órbita baja. La región que nos interesa en este proyecto es la parte baja de la termosfera (entre 100 y 300 km) donde la densidad de la atmósfera comienza a ser considerablemente importante. Hay varios modelos de atmósfera que pronostican las condiciones físicas a esas altitudes. Sin embargo, estos modelos tienen un error considerable cuando calculan la densidad (generalmente entre el 15-20 %). Debido a esto, un grupo de investigadores de la UPC tiene como objetivo reducir estos errores entre la previsión y los valores reales. Dentro de esta investigación, este proyecto tiene como objetivos estudiar la densidad promedio a lo largo de la órbita que los satélites pueden encontrar en altitudes ionosféricas, utilizando dos modelos atmosféricos diferentes (*NRLMSISE-00* y *JB2008*), y verificar si podemos obtener la densidad con relativa precisión en caso de que a hubiera algún problema con los satélites.
Para lograrlo, hemos desarrollado un código en MATLAB basado en el propagador de órbitas *SGP4* (Simplified General Perturbation 4) en el cual hemos calculado la perturbación de drag por nuestra cuenta usando los modelos atmosféricos. Esto, sin embargo, introduce un factor importante a considerar y estudiar, que es la determinación del coeficiente balístico.
Una vez que desarrollamos y testeamos el código, elegimos dos satélites diferentes para simular sus trayectorias y hacer el estudio: un CubeSat 1U (*ArduSat-X*) y un satélite esférico (*STARSHINE-2*). Ambos satélites se encontraban en diferentes situaciones en términos de actividad solar lo que nos permitió estudiar el problema en diferentes condiciones. Aparte de eso, como *STARSHINE-2* es esférico, nos permitió determinar cómo de importante es considerar el coeficiente balístico con precisión.
Los resultados han sido contrarios a lo que esperábamos y fueron un poco sorprendentes. La principal conclusión que obtuvimos es que los errores de los modelos atmosféricos son mayores de lo que leímos en cualquier bibliografía. Obtenemos valores en torno al 40-50 % de error con *STARSHINE-2*, lo que nos muestra que la termosfera inferior se conoce menos de lo que esperábamos, poniendo en valor la necesidad de seguir investigando. Además, nos dimos cuenta de la importancia de ser precisos con el coeficiente balístico porque puede introducir más error al propagar la trayectoria, dificultando la derivación de la densidad para conocer su valor real. Lamentablemente, hoy en día no podemos ser precisos con el coeficiente balístico con satélites que no son esféricos debido a las incertidumbres a la hora de determinar el área. Eso hace que los satélites esféricos sean perfectos para estudiar esta parte de la atmósfera.

Title : Determination of the along-orbit average density of the thermosphere
Author: Adrián Martínez Calzado

Advisors: Jordi Gutiérrez Cabello
Pilar Gil Pons

Date: September 8, 2020

Overview

Nowadays, the study of the dynamics of the upper atmosphere has become a must to study satellite trajectories at low orbit with accuracy. The region we are interested in this project is the lower part of the thermosphere (between 100 and 300 km) where the atmosphere density begins to be considerably important. There are several atmosphere models which forecast the physical conditions at that altitudes. However, these models have a considerable error when they compute the density (usually between 15-20%). Due to that, a group of researchers at UPC aims to reduce these errors between the forecast and real values. Within this research, this project aims to study the average density along-orbit that satellites may encounter at ionospheric altitudes using two different atmospheric models (*NRLMSISE-00* and *JB2008*) and check if we can obtain the density with relative accuracy in case of a problem with the satellites.
To achieve that, we have developed a code in MATLAB based on the Simplified General Perturbation 4 (*SGP4*) orbit propagator in which we compute the drag perturbation on our own using an atmospheric model. This, nevertheless, introduces an important factor to consider and study, which is the determination of the ballistic coefficient.
Once we developed and test the code, we chose two different satellites to simulate their trajectories and make the study: a 1U CubeSat (*ArduSat-X*) and a spherical satellite (*STARSHINE-2*). Both satellites were in different situations in terms of solar activity which allowed us to study the problem in different conditions. Apart from that, as *STARSHINE-2* is spherical, it allowed us to determine how important is to consider the ballistic coefficient with accuracy.
The results have been against what we expected to get and were a bit surprising for us. The main conclusion we got is that the errors from the atmospheric models are bigger than what we read in any bibliography. We get values around 40-50% of error with *STARSHINE-2*, which shows us that the lower thermosphere is less known that we expected, putting in value the necessity of doing further research. Furthermore, we realised the importance of being precise with the ballistic coefficient because could introduce more error when we propagate the trajectory, making difficult deriving the density and so know its real value. Unluckily, nowadays we cannot be precise with the ballistic coefficient with satellites that are not spherical due to the uncertainties when we determine the area. That makes spherical satellites perfect to study this part of the atmosphere.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Jordi and Pilar to give me the chance of doing this amazing and challenging project with them, and also for guidance and supporting me during all these months.

I would like to thank also to my family and friends for supporting me, especially to my brother David for advising on how to make the report better.

ACRONYMS

A_p Planetary Equivalent Amplitude

C_D Drag Coefficient

CIRA72 COSPAR International Reference Atmosphere 1972

COE Classical Orbital Elements

DSTDTC Temperature change from Disturbance Storm Time index

ECI Earth-Centered Inertial

EUV Extreme Ultraviolet

$F_{10.7}$ 10.7 cm wavelength solar radio flux

GAST Greenwich Apparent Sidereal Time

GMST Greenwich Mean Sidereal Time

GNSS Global Navigation Satellite System

HASDM High Accuracy Satellite Drag Model

IRI-2012 International Reference Ionosphere model - 2012

ISS International Space Station

JAXA Japan Aerospace Exploration Agency

JB2008 Jacchia-Bowman 2008 atmospheric model

JPL Jet Propulsion Laboratory

J-SSOD JEM Small Satellites Orbital Deployer

LEO Low Earth Orbit

LLA Latitude Longitude and Altitude

LST Local Apparent Sidereal Time

M_{10} 279.56-280.27 nm wavelength solar radio flux

MSISE Mass Spectrometer Incoherent Scatter Radar

NASA National Aeronautics and Space Administration

NORAD North American Aerospace Defence Command

NRLMSISE-00 Naval Research Laboratory MSISE class atmospheric model from 2000

S_{10} 26-34 nm wavelength solar radio flux

SGP4 Simplified General Perturbation 4

TEME True Equator Mean Equinox

TLE Two-Line Elements set

UT1 Universal Time 1

UTC Coordinate Universal Time

VOP Variation of Parameters

WGS-72 World Geodetic System 1972

$\mathbf{Y}_{10}$ 0.1-0.8 nm wavelength solar radio flux

CHAPTER 1. INTRODUCTION

1.1. Background and motivation

The Earth's atmosphere extends up to an altitude of about 10000 km, where the exosphere merges into the solar wind. Most of the satellites orbiting the Earth are within the atmospheric range, and will eventually fall towards the Earth's surface due to the atmospheric drag. As a consequence, it is critical to understand the nature and behaviour of the atmosphere (especially in its lower layers, where the density is higher) in order to achieve an accurate understanding of satellite trajectories.

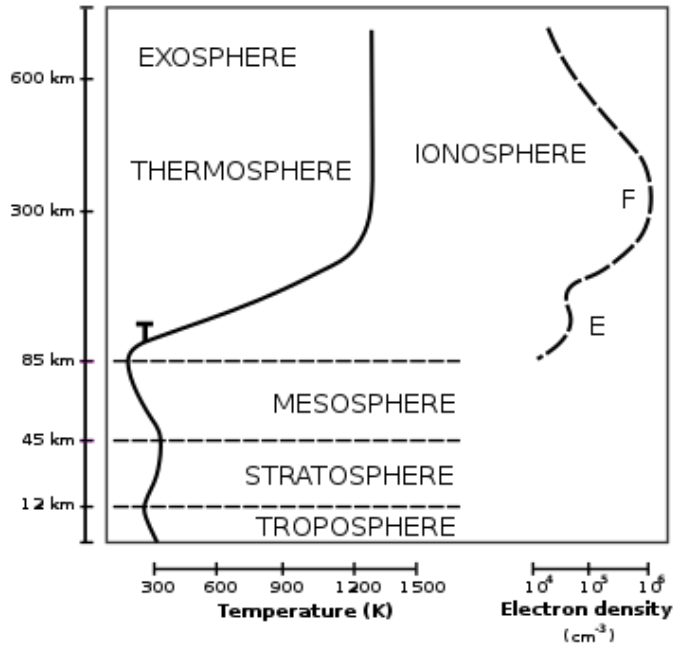

Figure 1.1: Atmosphere temperature profile including ionosphere. Credits:[8]

Nowadays, several empirical models forecast the density of the atmosphere. Usually, their applicability range reaches the exobase, (see, for instance, *MSISE-90*, [2]). In addition, these models yield substantial errors in the computation of density, which may be of the order of 15%-20% [1].

A group of research in the UPC is currently working on a project aimed to reduce the difference between the density forecasts provided by models, and actual density values. The general idea consists of putting into orbit a swarm of femtosatellites[1] equipped with accelerometers and GNSS receivers, with the purpose of measuring the atmospheric drag and then derive the density. The region to be studied is the lower thermosphere (100-300 km). This region is interesting for two reasons which are closely related. On the one hand, for scientific reasons, as this part of the atmosphere is very poorly known. On the other hand, because the satellites orbiting in this region are in their re-entry phase.

[1] The term "femtosatellite" or "femtosat" is usually applied to artificial satellites with a wet mass below 100 g [29]

Therefore, the better the characterization of the thermosphere, the more accurate re-entry trajectory predictions will be. Despite the fact that most satellites disintegrate during re-entry, some, as for instance *Tiangong*-1, may reach the Earth's surface and thus potentially cause serious damage.

Over the last decades, our technology has improved a lot and this also affects how satellites are designed. Despite in satellites "old technology" is used (because it has been tested and used in space, and is quite resistant to this environment), this is good enough to make them smaller, specially when we drop the requirement of it being *space-qualified*. Nowadays, we can send satellites into space with a size of $10\times10\times10$ cm (*CubeSats*) or even smaller. This fact opens wide options and opportunities in the space sector because it allows to reduce the cost of the missions, benefiting not only the business part but also in scientific research.

1.2. Goals and methodology

We aim to study and simulate the average density along-orbit that satellites may encounter at thermospheric altitudes. We will use MATLAB in order to build a code which will implement:

- the orbit propagator *SGP4*, to simulate the orbital perturbations which the satellite will experience;
- two different atmosphere models, *NRLMSISE-00* and *JB2008*, which allow to compute the density;
- a drag perturbation model.

The code will be used to analyse the effect of the input physics, namely, the drag coefficient and the specific atmosphere model, on the satellite trajectories and to compare with real cases. Moreover, we will see whether in case of a problem with the satellites we can recover some information with our current models.

1.3. Structure of the project

The present project is structured as follows. In Chapter 2 we briefly describe the thermosphere and the models which we will use to compute the density. We will also describe the concept of atmospheric drag and its effect on satellite trajectories. In Chapter 3 we describe the structure and details of our code and our simulations. Chapter 4 summarises the validity tests we performed and the main results we obtain. Finally, in Chapter 5 we derive the main conclusions and outline future developments based on our work.

CHAPTER 2. THE THERMOSPHERE

The Earth's atmosphere is divided into different layers. The most common criteria that is used to divide it is through the temperature profile. As we see in Figure 2.1, the atmosphere is divided into the following layers: troposphere, stratosphere, mesosphere, thermosphere and exosphere.

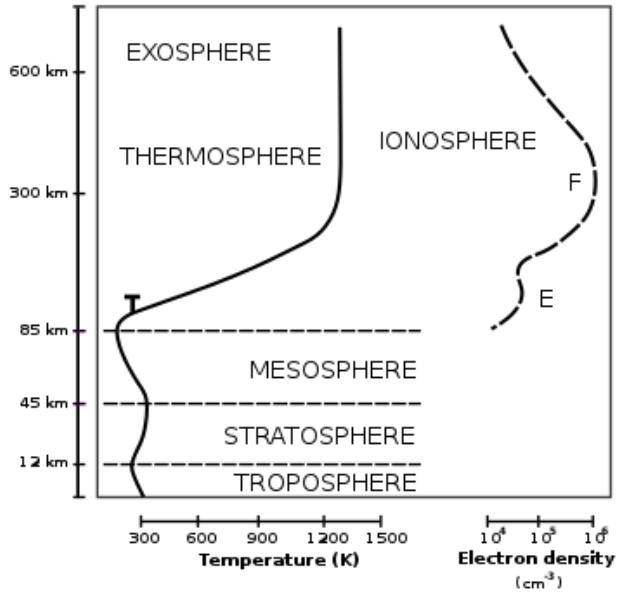

Figure 2.1: Atmosphere temperature profile including ionosphere. Credits:[8]

The thermosphere is the part of the atmosphere which ranges between altitudes $\sim$ 100 km (close, but unrelated, to the Kárman line or legal limit of space), and up to $\sim$ 500-600 km, although due to the variability of physical conditions, it can rise up to 1000 km.

As we can see in Figure 2.2, the temperature in the lower part of the thermosphere (between $\sim$ 100 and 200-250 km) increases very quickly, whereas it remains almost constant above this upper threshold. The thermospheric temperature varies during the day, being about 200 K higher during illuminated time intervals than during night conditions. In addition, solar activity has a deep impact on the temperatures, which may vary between $\sim$ 600 K and up to $\sim$ 2000 K near peaks of activity [11].

Atmospheric heating is a consequence of several effects. First, there is the absorption of solar EUV (Extreme Ultraviolet) and X-ray radiation. Second, at thermospheric altitudes, there are few triatomic molecules such as CO_2, H_2O and O_3 that behave as efficient infrared radiators, and thus cause an additional increase in the temperature. The third important source of heat is the geomagnetic activity, as energetic particle precipitation from the magnetosphere and Joule heating may provoke an increase in temperature. Altogether cause altitude variations in the thermopause (uppermost limit of the thermosphere) between 500 and 1000 km.

EUV and X-rays which heat the atmosphere also ionize atoms and molecules, creating an ionized atmospheric shell between $\sim$ 80 km and $\sim$ 600 km which is named *ionosphere*. The ionosphere actually overlaps the mesosphere, the thermosphere and the exosphere

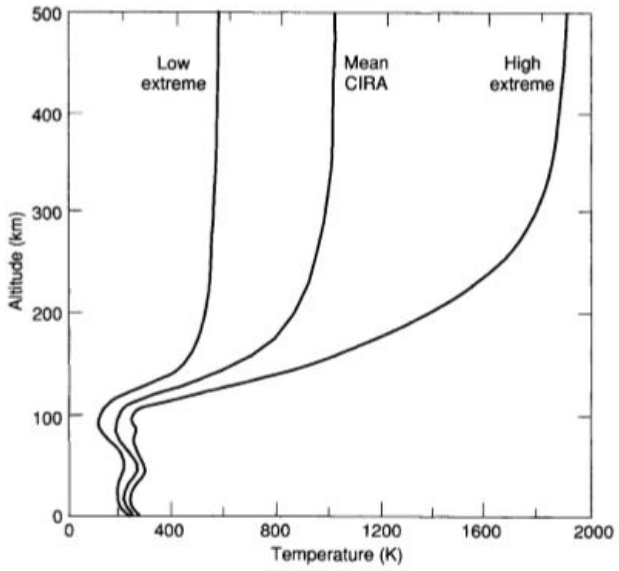

Figure 2.2: Mean CIRA temperatures and low and high extremes temperatures. Credits: [7].

(see Figure 2.1). Electron density increases at the lower parts of the ionosphere (up to $\sim$ 300 km) and decreases in its uppermost layers until it merges with the magnetosphere plasma. Note that ionization in the atmosphere affects wave propagation and thus the performance of satellite communications. Atmospheric layers below height $\sim$ 80 km are not ionized because the overlaying atmosphere has already absorbed most of the solar EUV and soft X-ray fluxes; furthermore, its relatively high density allows efficient recombination of ions and electrons when produced by the remaining ionizing radiation.

The physical conditions of the air in the thermosphere and above is different from that of the lower atmospheric layers. The atmospheric mixing in these high layers is not very effective because the density is so low that gas-particles collide very infrequently. This leads to the occurrence of very simple molecules whose altitude distribution depends on their masses (see Figure 2.3). Thus, at the upper thermosphere O, H and He are the main components of the air. Actually, atomic oxygen is the main constituent of the thermosphere due to the dissociation of ozone (O_3) and other oxygen-carrying molecules.

At present, there are several numerical models which simulate the physical conditions and the chemistry of the thermosphere. In this project, we use *NRLMSISE-00* and *JB2008*, which will be described in the next paragraph. As any numerical models, these two suffer from limitations due to the necessity of imposing simplifying assumptions which make their goal feasible. Specifically, *NRLMSISE-00* does not model the semi-annual density variation as function of the solar activity which, as we have seen, is critical for the physics of the thermosphere. *JB2008* is generally more accurate than *NRLMSISE-00* because it better models EUV data and takes into account semiannual density variation. However, for low solar activity, *JB2008* has some problems, causing the differences between *NRLMSISE-00* and *JB2008* to decrease.

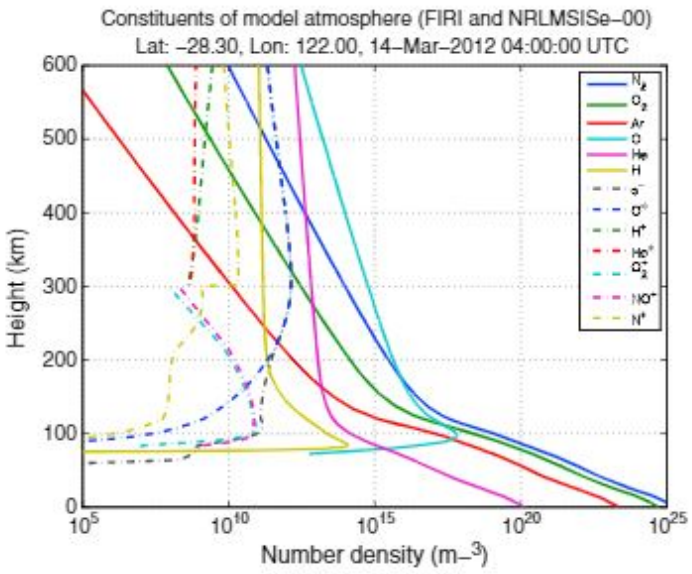

Figure 2.3: Neutral and ion densities generated by the International Reference Ionosphere model - 2012 (IRI-2012) and NRLMSISE-00. IRI-2012 can be found at [33]. Credits:[4]

2.1. Atmospheric models

Before describing both models (*NRLMSISE-00* and *JB2008*) in more detail, first, we are going to talk about the atmospheric models in general an what they take into account.

There are two types of atmospheric models: *static* and *time-varying*. The static ones are the simplest because assume that all the atmospheric parameters remain constant. However, some factors affect them [19]:

1) **Latitudinal variation:** It is easier to visualize if we imagine a satellite with a circular, inclined orbit. When the satellite passes over Earth's equator, the altitude will not be the same that when the satellite is in its maximum latitude because of the Earth's oblateness. So, if the altitude change so does the density.

2) **Longitudinal variations:** Usually, these variations are related to the time-varying models because of the diurnal effect but, they can also affect to static models. The atmosphere is far from being uniform because of the Earth's surface heterogeneity: it is not the same to consider the mountains of Himalaya than the expanses of the open Pacific or Atlantic ocean, even if they are at the same latitude.

When we take into account the time variance, the problem turns more difficult. We now must model [19]:

1) **Diurnal variations:** These variations occur because the Sun heats approximately 50% of the Earth at a time. As the atmosphere is a fluid, when the Sun heats it, the atmosphere expands creating a bulge (here we have the maximum value of the density). This bulge is centered on the meridians where the local time is 2-2:30 p.m. Its center is over the equator at the equinoxes but moves to higher –or lower–

latitudes depending on the Sun's declination, which varies mainly due to the obliquity of the ecliptic. Thus, the atmospheric density depends on latitude, local time, and time of the year. The minimum value of the density occurs approximately opposite the bulge at local time 4:00 a.m. or so each day.

2) **27-day solar-rotation cycle:** This effect is due to the Sun's rotational period. This rotation causes a fluctuation which is correlated with the solar decimetric-wavelength radio flux (1m - 10 cm). The irregular changes that appear in this cycle are related to the growth and decay of the active solar regions. These regions are very difficult to study because they have may different patterns of growth, stability, and decay. Moreover, they have an uncertain cyclical pattern because rotate with the Sun over its 27-day period which is also dependent on the solar latitude (the poles rotate in approximately 30 days). Therefore, these patterns are difficult to predict and it is also difficult to determine how they heat the Earth's atmosphere. These uncertainties make problematic to predict with accuracy the solar flux in EUV and soft X-rays.

3) **Solar 11-year cycle of sunspots:** High energy solar photons depend very markedly on the state of solar activity. This cycle is in fact a 22-year cycle if the reversal in the solar magnetic field is taken into account. At the height of the cycle, depending on the satellite's altitude, this effect can create a larger disturbance through solar-radiation pressure than through drag.

4) **Semi-annual variations:** These variations last about six months as its name conveys. Are related to the varying distance of the Earth to the Sun and the Sun's.

5) **Cyclical variations:** An 11-year cycle approximately parallels but lags a few years behind the sunspot cycle. The minimum of this cycle it is about 6-7 years after a maximum so, it is not in the midway. The magnitude of the peak of the 11-year cycle varies from cycle to cycle. The exact reason is not known but is likely related to sunspot activity.

6) **Rotating atmosphere:** The atmosphere rotates with the Earth, causing additional time-varying changes in the density. The velocity usually is larger in the Earth's surface because of the friction.

7) **Winds:** Predicting and knowing the thermospheric weather pattern and the effect of thermospheric winds on the density is very difficult and extremely complex. Winds cause temperature variations and therefore changes in density and we do not have enough high-quality observations to predict them. Nowadays, there remains a lot of work to be done in order to properly understand the dynamics of the upper atmosphere.

8) **Magnetic-storm variations:** Usually, the magnetic field does not affect much to the atmosphere. However, when the geomagnetic activity increase, its effect on the atmosphere so does.

9) **Irregular short-periodic variations:** These effects are small and often are related to transient geomagnetic disturbances. Here we have solar flares and other effects such as variation hydrogen currents within the atmosphere.

10) **Tides:** They cause a small variation in the atmospheric density and so do not introduce a significant error on the models.

2.1.1. NRLMSISE-00

NRLMSISE-00 is an empirical model that improves over *MSISE-90* model. Both were developed by the US Naval Research Laboratory. *NRLMSISE-00* computes the density, temperature and composition of the air (for a limited set of molecules and elements), from the Earth's surface and up to the exobase. The necessary data for the development of this model were obtained along decades by the Mass Spectrometer Incoherent Scatter Radar. Additional data were obtained by different satellite accelerometers in their re-entry phase, which allowed improvements in the predictions [2].

To compute all these variables, the model needs the following inputs:

- **Year**: This value does not affect the results but we must introduce it as an input. It only affects if we have to seek the conditions of one particular year in terms of solar and geomagnetic activity.
- **Day**: Day of the year.
- **Seconds**: Seconds of the day in UTC.
- **Altitude**: The geodetic altitude of the satellite in kilometers.
- **Longitude**: Geodetic longitude of the satellite in degrees.
- **Latitude**: Geodetic latitude of the position of the satellite in degrees.
- **Local apparent Sidereal Time (LST)**: Is the local time in hours at the longitude of the satellite.
- **Solar activity index $\mathbf{F}_{10.7}$**: 10.7 cm solar radio flux. Its units are in solar flux units (1 sfu = 10^{-22} W m^2 Hz^{-1}). Its value can vary from below 50 sfu to above 300 sfu.[12]
- **Averaged solar activity index $\mathbf{F}_{10.7A}$**: The 10.7 cm solar radio flux averaged in 81 days.
- **Geomagnetic index $\mathbf{A}_p$**: Vector of 8 positions of 3-hour geomagnetic index. Its value can vary from 0 to 400 nT.
- **Geomagnetic index A**: average of 8, 3-hour Ap over a day. Its value can vary from 0 to 400 nT.

2.1.2. Jacchia-Bowman 2008 (JB2008)

The *JB2008* model improved the *JB2006*, which is based on the *CIRA72* (COSPAR International Reference Atmosphere 1972) model. *CIRA72* integrates the diffusion equations, using the *Jacchia 71* temperature formulation in order to compute density [3]. *JB2008* model can be used to compute density and temperature at altitudes between 175 km and 1000 km. This model also takes into account the semiannual variation. It is considered one of the best models available nowadays, with an error of the order of 9-10 % at 400 km [1].

As in the *NRLMSISE-00*, *JB2008* needs some inputs to compute the temperature and density. Those inputs are:

- **Modified Julian day**.
- **Right Ascension of the Sun (radian)**.
- **Declination of the Sun (radian)**.
- **Right Ascension of the satellite (radian)**.
- **Geocentric latitude of the satellite (radian)**.
- **Height of the satellite (km)**.
- **Solar activity index $F_{10.7}$** (see Section 2.1.1).
- **Averaged solar activity index $F_{10.7B}$**: average over 81 days centered on the input time.
- **S_{10}**: measure of the 26-34 nm solar extreme ultraviolet (EUV) emission. The measurements are normalized and converted to sfu (solar flux unit).
- **S_{10B}**: S_{10} average over 81 days, centered on the input time.
- **Solar activity index M_{10}**: also a proxy for solar extreme ultraviolet radiation (279.56-280.27 nm), converted into sfu.
- **M_{10B}**: M_{10} average over 81 days, centered on the input time.
- **Y_{10}**: measure of 0.1-0.8 nm solar X-ray emission and Lyman-α, converted into sfu.
- **Y_{10B}**: Y_{10} average over 81 days, centered on the input time.
- **DSTDTC (K)**. Is the temperature change computed from the DST index. The DST (Disturbance Storm Time) is used to indicate the strength of the storm-time ring in the inner magnetosphere. It is used instead of the A_p index because it allows improving the accuracy.

2.2. Atmospheric drag

The atmospheric drag is not easy to compute, because it is based on poorly known atmospheric properties. However, without taking into account the exact properties of the atmosphere, we know that the drag affects mainly the semi-major axis and eccentricity. The orbit tends to circularize due to the mechanical energy loss (see Figure 2.4). However, for $e = 0$ orbits, the asymmetry between daylight and nocturnal thermosphere induces a slight increase in the eccentricity.

We can define the drag acceleration as follow:

$$\vec{a} = -\frac{1}{2}\rho v_{\text{rel}}^2 \frac{AC_D}{m}\hat{v}_{\text{rel}} \tag{2.1}$$

This equation seems inoffensive but is difficult to compute with accuracy. The reason is simple, determining its variables is complicated. We can understand this better if we analyze each variable in detail.

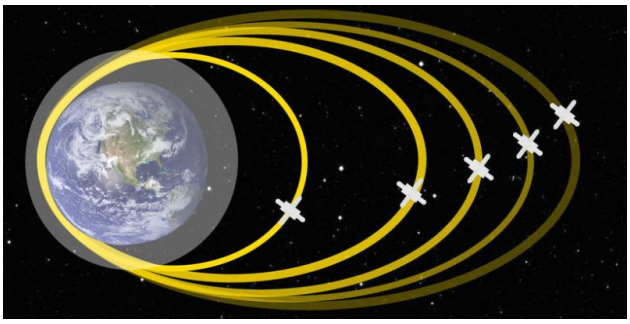

Figure 2.4: Orbit decay due to atmospheric drag. Credits:[5]

- $\mathbf{C}_D$. Our intuition tells us that the drag coefficient is constant, but in fact, it is not. There are several models that compute the values of C_D under different conditions. These changes are studied in reference [6], in particular for the case that we want to study. However, these models are estimations and thus subject to substantial errors.

- $\mathbf{v}_{\text{rel}}$. This is the velocity of the spacecraft relative to the atmosphere. Oversimplifying, we may think that the atmosphere rotates with the Earth without other phenomena that alter it. But, in fact, in the atmosphere, there are waves and tides. Circulation and winds in the thermosphere are driven by these waves and tides and thus they may not be neglected.

- ρ. We have already seen that due to the problem complexity it is really difficult to compute the density with accuracy.

- **A**. It is the cross-sectional area, which is normal to the velocity vector. We must recall that this velocity vector is given with respect to the atmosphere. If we want to know the area with precision, we need a precise attitude subsystem to compute it properly. But, even if we had a precise attitude system, the satellite could re-enter changing its area randomly due to operational problems and friction with the air, making it impossible to predict the cross-sectional area at a certain epoch. On the other hand, if we want to study a decaying satellite which is not operational, the area will be uncertain too. This issue could be easily dealt with if spherical satellites were considered.

CHAPTER 3. STRUCTURE AND DESCRIPTION OF THE SIMULATION

In this chapter, we describe the simulation of satellite trajectories. Figure 3.1 shows a summary sketch of the entire process, and the details of each block are presented below.

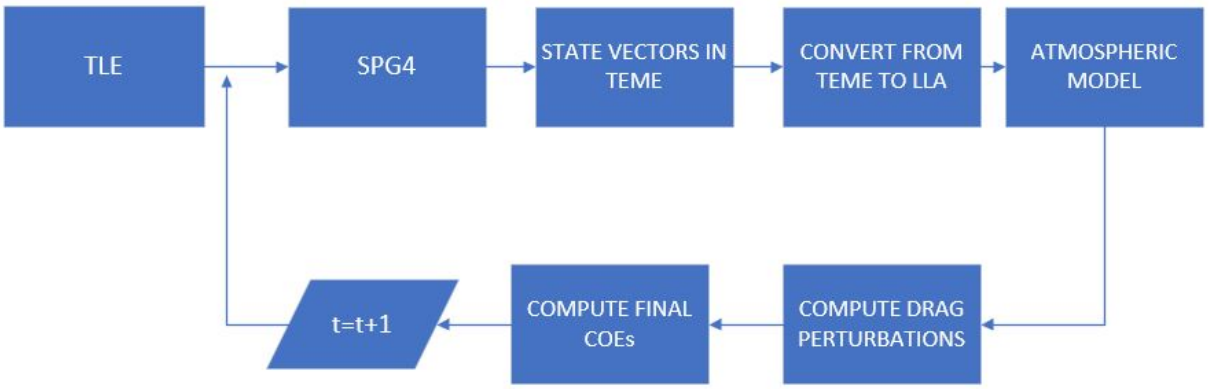

Figure 3.1: Structural organization of the code

Before beginning to explain the different calculations we have done, we have to describe which model of Earth we have taken into account. This may seem trivial, but it definitely is not. Furthermore, misinterpretations related to the models used may lead to significant errors which are, of course, avoidable.

To begin with, it is well known that the Earth is not a perfect sphere, but rather similar to an ellipsoid. There exist different ellipsoid approaches to fit the Earth's shape, and each one has different constants. Therefore, it is important to use always the same model, lest different constants might be mixed. The model we are going to use is WGS-72. The reason we have chosen this one is that the orbital simulator *SGP4* (see below) was developed using this model. Thus, note that all the following calculations and simulations have been done with WGS-72 constants [10].

3.1. SGP4

During the 1960s, the NORAD (North American Aerospace Defense Command) began the development of a fully-analytical orbital model aimed to allow the tracking of satellites. This model was developed to substitute the numerical integration model they were using, which demanded a lot of computing power. We know this analytical model as *SGP* (Simplified General Perturbation) and it allows us to calculate the satellite's position and velocity at a particular time. But, of course, this model has several limitations. The first one is that the mass of the satellite relative to the Earth is assumed to be negligible. This is actually not a problem, as satellite masses are negligible compared to the mass of the Earth. The other limitations are that the satellite is assumed to have low eccentricity, near-Earth orbit, and that they are not in a rapidly-decaying orbit. All these conditions were fulfilled for most of

the satellites that were in orbit when the *SGP* was developed. *SGP* takes into account the main perturbations for LEO orbits, which are the Earth's nonuniform mass distribution and atmospheric drag.

SGP was extended to include semi-analytic treatment of orbits with periods greater than 225 minutes. These 225 minutes (6000 km of altitude) are a "natural gap", a result of the historical choices of orbits for different satellite missions. Above 225 minutes, main perturbations are no longer the atmospheric drag and the Earth's nonuniform density, but instead, orbital resonances with the Earth's nonuniform gravitational field, solar and lunar gravitational forces, and solar radiation pressure [9]. This extension of *SGP* is known as *SGP4*. We have made a schematics (Figure **??**) to understand how the MATLAB code works [10].

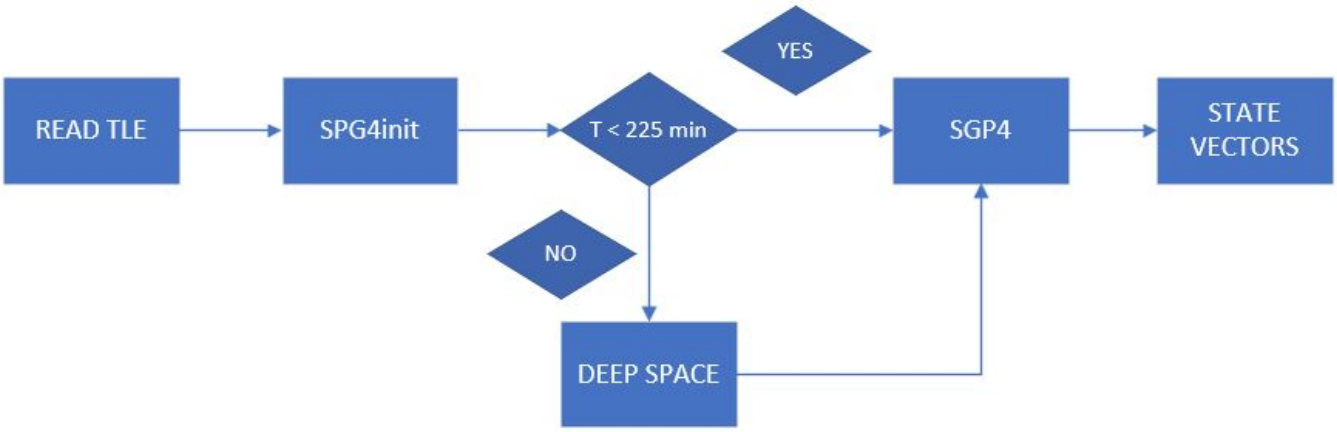

Figure 3.2: SGP4 structural organization

SGP4 uses as inputs the NORAD Two-Line orbital Elements or TLEs (see below.) First, the propagator reads these initial conditions (provided by TLE data) and stores all this data in a structure. This is sent as an input to the *SGP4init*, that initializes many variables which later on will be used by *SGP4*. This is done to save CPU, because if we want to make a loop and plot how to evolve the state vectors, we can do it only including the *SGP4* function, and thus we avoid the necessity to calculate all the variables again. Then, we have to classify the orbits to apply the correct perturbation to each type of orbit. So, as we said before, if the orbital period is greater than 225 minutes, it applies the "Deep Space" perturbations before the main *SGP4* function. If its period is lower than 225 minutes, it goes directly to the main *SGP4* function. To finish, *SGP4* uses all the perturbations that were previously calculated to propagate the orbit properly. The version of *SGP4* code we use is available in the *Celestrak* web site [13].

3.1.1. Two-Line Elements sets

Two-Line elements sets are similar to classical orbital elements (COEs), but they are not using exactly the same format; besides that, TLEs provide further information, as the ballistic coefficient. For the correct use of the *SGP4* and, hence, for the development of the project, it is extremely important to understand how TLE sets are obtained, and the precise meaning of each value. It is also important because misinterpretations of their meanings can lead to huge errors when computing the state vectors.

First, let us consider how they are computed. All the satellites are tracked by different space surveillance networks which measure their positions (e.g. in the case of the US, the *US Space Surveillance Network*). Usually, the reference frame used is the topocentric coordinate system (SEZ) (see Appendix A), because it is convenient both for ground-based and space-based station. As we are referring to NORAD TLEs, we can assume that the network they use is the US Space Surveillance Network. Using this network, the position of the satellite in different places can be measured and stored. Once the measurements are taken, the TLE that describes the satellite's orbit can be computed. We are not going to explain all the mathematical procedure of how they are computed because it is not the scope of the project, but let us just say that with these measurements, and using the least square method and *SGP4*, it is possible to compute the "mean" values of the orbital elements. This is what TLEs are, mean values for the orbit properties obtained from the measurements taken. Thus, if we compute the Keplerian elements of any satellite's orbit at a certain epoch ($t = 0$), the results will not generally be same as the values available in the TLE. Indeed, it is difficult that we obtain the same values because they do not define a particular satellite epoch but, as mentioned above,"averages". We can consider, for a better understanding of the TLEs, that are somewhat analogous to the COEs for the two-body problem, but now we have the perturbations and it becomes necessary to define them in another way.

It is important to realize that when we do the simulations, the drag term (Bstar) of the TLE must be set to a zero value because we will calculate our own drag disturbances and add them to the results so obtained by means of *SGP4*. Something important to mention is that neither the precise *SGP4* version nor the code to calculate the TLEs which NORAD uses are available. This could lead to some discrepancies in our results, but it is not possible to check if this is the case.

For a detailed explanation and examples on how to obtain TLE's, please refer to [14][15]. In addition, the reference [13] provides a code aimed to compute TLEs.

In Table 3.1 we present an example of a real TLE as well as the meaning of each of the values in this TLE.

Field	Content	Example
1	Line number	1
2	Satellite number	25544
3	Classification (U=Unclassified, C=Classified, S=Secret)	U
4	International Designator (last two digits of launch year)	98
5	International Designator (launch number of the year)	067
6	International Designator (piece of the launch)	A
7	Epoch Year (last two digits of year)	20
8	Epoch (day of the year and fraction of the day)	126.53937142
9	Mean motion first derivative	.00000838
10	Mean motion second derivative	00000-0
11	Bstar	23085-4
12	Ephemerides type	0
13	Element set number	999
14	Checksum	6

Field	Content	Example
1	Line number	2
2	Satellite number	25544
3	Inclination (deg)	51.6442
4	Right Ascension of the Ascending Node (deg)	199.5485
5	Eccentricity (with decimal point)	0001122
6	Argument of the perigee	241.4902
7	Mean anomaly	251.9068
8	Mean motion	15.49349074
9	Revolution number of epoch	22540
10	Checksum	3

Table 3.1: Fields in line 1 (upper table) and line 2 (lower table) for TLE:
1 25544U 98067A 20126.53937142 .00000838 00000-0 23085-4 0 9996
2 25544 51.6442 199.5485 0001122 241.4902 251.9068 15.49349074225403

3.2. Conversion from TEME to LLA

Once we have computed with *SGP4* the position and velocity of the satellite we want to study, we have to compute the atmospheric density. A problem then becomes apparent: the outputs of the *SGP4* and the inputs of density models are not the same. We have to convert the state vectors (satellite's position and velocity vectors) into latitude, longitude and altitude. This is equivalent to transform from the True Equator Mean Equinox Coordinate System (TEME) to the geographic reference of Longitude, Latitude and Altitude (LLA).

The frame of reference used in TLEs and *SGP4* is TEME coordinate system. Lets explain what TEME is and the steps that we have done to convert the state vectors into longitude, latitude and altitude.

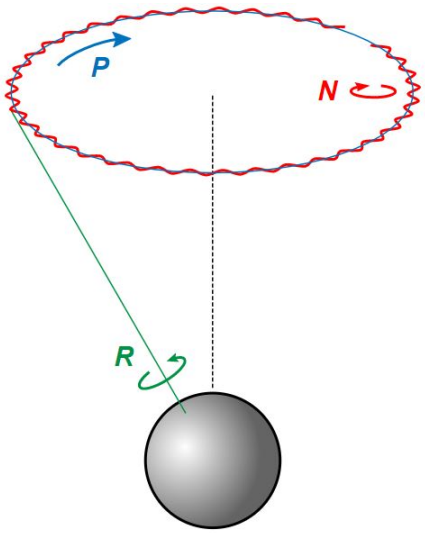

Figure 3.3: Nutation and precession representation. Credits:[16]

3.2.1. TEME (True Equator Mean Equinox) Coordinate System

TEME coordinate system is a particular Earth-Centered Inertial (ECI) frame. The direction of the Earth's true rotation axis, or what we know as the North Pole, changes over time. The same happens with the true direction of the vernal equinox. Since observations are made from a point fixed to the Earth's surface, the results will be relative to the true equator. However, the direction of the vernal equinox depends on the Earth's orientation in space and, hence, it is not a fixed point on the Earth's surface nor in the celestial sphere. Thus, an approximation of its true direction must be made. In the approximation taken, the precession of the vernal equinox is considered, but its nutation is neglected [17].

3.2.2. Conversion to LLA

Once we understand TEME, the next step is to convert into LLA. We can compute geocentric LLA easily from TEME. Let us comment that for *JB2008* the required inputs are the geocentric latitude and longitude (thus, referred to the angles between the position vector of the point and an Earth-centred reference system). On the other hand, *NRLMSISE-00* needs geodetic values (referred to the angles between the local normal to the ellipsoid and an Earth-centred reference system). However, this is not a serious problem, as we can compute both geodetic and geocentric values simultaneously.

It is important to realise that both geodetic and geocentric longitudes are the same because the Earth's oblateness does not affect their values. On the other hand, for both models, we use geodetic altitude because it is the true distance between the satellite and the Earth's surface. Unfortunately, the bibliography does not specify which altitude they are referring to, so we choose the most logical value. The complete mathematical procedure that we have used can be found in appendix C. The method we made is a mix of two other methods, intending to get the quickest subroutine.

To be able to compute this conversion, first, we need to compute the Modified Julian Date in UT1, and the Julian centuries of terrestrial time, because we need both to compute

Greenwich Mean Sidereal Time (GMST).

3.3. Atmosphere models

In the previous chapter, we have succinctly described the two main-stream atmospheric models (*NRLMSISE-00* and *JB2008*) and some of their inputs, but we did not explain how we do to obtain them. So, let see in both cases where we find and how we computed the input data.

3.3.1. NRLMSISE-00

As we will see, this case is simpler than that of the *JB2008* model. The only parameter to be computed is the LST (Local apparent Sidereal Time). LST is the sum of the longitude of the position of the satellite and the Greenwich Apparent Sidereal Time (GAST), and then convert it into hours.

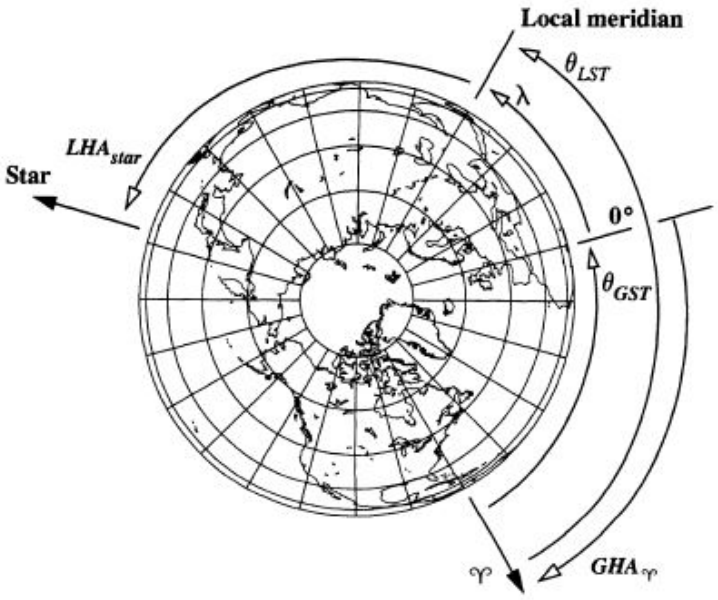

Figure 3.4: LST angle representation. Credits:[19]

The $F_{10.7}$ and A_p parameters are available in the *Celestrak* web [18], among other sources.

Regarding the atmospheric model, we do not have to code anything here because the code of the NRLMSISE-00 is fully available in [20].

3.3.2. JB2008

In the case of *JB2008*, the first parameter that we need to compute is the Modified Julian Date. As we know the epoch, it is an easy and quick task. The next thing we need is the Right Ascension of the satellite. If we check the definition of the Right Ascension, we see

that it is the same as the LST in radians. So, we only need to compute the GAST and add it to the longitude.

Other important parameters we have to compute are the Right Ascension (RA) and the Declination (δ) of the Sun. To compute both, we need the Modified Julian Date for barycentric dynamical time. Once we have computed it, we use the MatLab function (JPL Ephemerides DE430) to compute the position of the Sun with respect to the Earth's centre (in Cartesian coordinates x, y and z). Knowing its position, computing the Right Ascension and the declination is as simple as:

$$RA_{\text{Sun}} = \arctan\left(\frac{y}{x}\right) \tag{3.1}$$

$$\delta_{\text{Sun}} = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \tag{3.2}$$

Obtaining $F_{10.7}$ to Y_{10B} (Solar X-Ray and Lyman alpha 81-day average centered index) does not require any additional calculation, just knowing the epoch. We can find all these parameters in [21]. The same happens with DSTDTC, as knowing the epoch we can get its value looking at the same reference. The files that we have used are *DTCFILE.txt* and *SOLFSMY.txt*, and the code and an example of how to use *JB2008* is available in reference [22]. All this software is already coded in the same reference.

3.4. Computation of Drag perturbation

Until now, we have computed the state vector and the atmospheric density at the position of the satellite at a particular epoch, but we have not taken into account how this density affects the satellite's trajectory. Apart from that, some of the computations we described so far were performed using existing codes, and our main task was to assure our interpretation of the meaning of the variables and the physics implemented was correct. Once the drag perturbations are taken into account, it is important to figure out how to implement them in the existing codes. This process is described in this section.

First of all, let us recall that orbital elements change over time due to the different perturbations that satellites suffer in orbit. To know how drag affects COEs, we have to use the Gaussian form of the Lagrange planetary equations, also known as VOP (Variation Of Parameters) equations. They are expressed as:

$$\frac{da}{dt} = \frac{2}{n\sqrt{1-e^2}}\left[e\sin(\nu)F_r + \frac{p}{r}F_s\right] \tag{3.3}$$

$$\frac{de}{dt} = \frac{\sqrt{1-e^2}}{na}\left[\sin(\nu)F_r + \left(\cos(\nu) + \frac{e+\cos(\nu)}{1+e\cos(\nu)}\right)F_s\right] \tag{3.4}$$

$$\frac{di}{dt} = \frac{r\cos(u)}{na^2\sqrt{1-e^2}}F_w \tag{3.5}$$

$$\frac{d\Omega}{dt} = \frac{r\sin u}{na^2\sqrt{1-e^2}\sin(i)}F_w \tag{3.6}$$

$$\frac{d\omega}{dt} = \frac{\sqrt{1-e^2}}{nae}\left[-\cos(\nu)F_r + \sin(\nu)\left(1+\frac{r}{p}\right)F_s\right] - \frac{r\cot i \sin u}{h}F_w \tag{3.7}$$

$$\frac{dM_o}{dt} = \frac{1}{na^2e}\left[(p\cos(\nu) - 2er)F_r - (p-r)\sin(\nu)F_s\right] \tag{3.8}$$

where a is the semi-major axis, e the eccentricity, i the inclination, Ω the Right Ascension of the Ascending Node, ω the argument of the perigee, M_o the mean anomaly, ν the true anomaly, r the position in polar form, p the semilatus rectum, h the specific angular momentum, n the mean motion and u the argument of latitude. We have to be careful with these equations because they are in RSW coordinate system (see appendix B) and F_r, F_s and F_w are not forces but specific forces (or forces per unit mass of the satellite, that is, accelerations).

Once these equations are introduced, let us think how to apply them to our case. We want to study the orbits of satellites at low altitude. Usually, at low altitude, we have satellites with very low eccentricity, nearly circular. So, we can assume that our orbit is circular. If we do not consider that, we will not get good results with the simulation. The reason is because in the equations 3.7 and 3.8, the eccentricity e appears in the denominator. This leads to unphysical divergences in some computed quantities when values of e close (but not equal) to zero are considered. Thus, assuming (near-)circular orbits we can not use equations 3.4, 3.7 and 3.8 because they are no longer valid.

Drag only affects the R and S axes, therefore i and Ω will not be affected. So, we can only apply the first equation 3.3. Our derivation of the temporal variation of the semi-major axis was as follows:

$$p = a(1-e^2) \tag{3.9}$$

$$r = \frac{a(1-e^2)}{1+e\cos(\nu)} \tag{3.10}$$

$$F_r = a_{\text{drag}}\sin(\phi_{\text{flight path}}) \tag{3.11}$$

$$F_s = a_{\text{drag}}\cos(\phi_{\text{flight path}}) \tag{3.12}$$

$$V_{\text{rel}}^2 = n^2a^2\left[\frac{1+e^2+2e\cos(\nu)}{1-e^2}\right] \tag{3.13}$$

From [19]:

$$\sin(\phi_{\text{flight path}}) = \frac{e\sin(\nu)}{\sqrt{1+e^2+2e\cos(\nu)}} \tag{3.14}$$

$$\cos(\phi_{\text{flight path}}) = \frac{1+e\cos(\nu)}{\sqrt{1+e^2+2e\cos(\nu)}} \tag{3.15}$$

Now if we substitute equations 3.9, 3.10, 3.11, 3.12 and 3.13 in equation 3.3 we get:

$$\frac{da}{dt} = -\frac{\rho C_D A n a^2}{m}\left[\frac{1+e^2+2e\cos(\nu)}{1-e^2}\right]^{3/2} \tag{3.16}$$

For circular orbits ($e = 0$):

$$\frac{da}{dt} = -\frac{\rho C_D A n a^2}{m} \tag{3.17}$$

Once we have the final equation, we only have to solve it. Of course, we are not going to solve this equation analytically but numerically. Note that an analytical solution is not feasible, as the density ρ varies with time. The numerical method used is the Runge-Kutta of 4^{th} order. Solving this equation, we get the semi-major axis, while taking into account the density at each epoch.

Note also that we have solved this equation taking into account that the difference between time steps is very small. Thus, the only parameter that changes significantly between time steps is the semi-major axis a. The other parameters can be considered constant, that is, their value are the same as in a prior step. Let us emphasize that we can make this assumption because of the small difference between time steps which is important for the numerical method to work properly, otherwise this approximation could not be done. Doing several simulations we realised that time steps of 1 minute are small enough to have good results, without having a too high CPU demand.

On the other hand, if we take a look to equation 3.3 we see we cannot use the state vector directly. Therefore, we have to convert the state vectors we have obtained from *SGP4* into COEs before solving the equation (the subroutine to do this conversion is available in [13] and it is explained step by step in [19]). Once we have solved the equation and corrected a, we store the results in COE form to use them in the following step.

The results obtained using only equation 3.17 (which explicitly contains the drag coefficient C_D) are equivalent to those obtained using *SGP4* with its drag term (Bstar) set to zero. In addition, a further issue must be considered, because we are considering a satellite with an almost circular orbit, whose trajectory is affected by different perturbations. We know that the gravitational field is conservative but, drag is a non-conservative force. Then, the mechanical energy of the satellite will become smaller with time. The specific mechanical energy of a circular orbit is can be found as (μ is the gravitational parameter):

$$E = -\frac{\mu}{2a} \tag{3.18}$$

Therefore, we clearly see that everything makes sense, since the losses in the semi-major axis are caused by the losses of mechanical energy.

In addition, this reduction in the semi-major axis caused a reduction in the orbital period and, if the period is smaller, the mean motion is larger. The relation between semi-major axis and mean motion (Equation 3.19) allows us to calculate how drag affects the mean anomaly.

As we have a circular orbit and the i and Ω will not be affected, the parameters that changes over time are the semi-major axis (a) and mean anomaly (M_o). This is something, as we have seen, we cannot compute with VOP equations because they are no longer valid for small eccentricities.

$$n = \sqrt{\frac{\mu}{a^3}} \tag{3.19}$$

Thus we must proceed as follows: first we have to differentiate equations 3.18 and 3.19.

$$\frac{dE}{da} = \frac{\mu}{2a^2} \tag{3.20}$$

$$\frac{dn}{da} = -\frac{3}{2}\sqrt{\frac{\mu}{a^5}} \tag{3.21}$$

Now we can develop a bit more equation 3.21:

$$\frac{dn}{da} = \frac{-3}{2a^2}\frac{\mu}{\mu}\frac{a}{a}\sqrt{\frac{\mu}{a}} \tag{3.22}$$

$$\frac{dn}{da} = -\left(\frac{3an}{\mu}\right)\frac{dE}{da} \tag{3.23}$$

As a result, taking into account the mean anomaly equation, we finally get how the mean anomaly vary over time, having the problem ”completely” modeled and ready to do the simulations.

$$\Delta E = \frac{\mu}{2\Delta a^2} \tag{3.24}$$

$$\Delta n = -\left(\frac{3an}{\mu}\right)\Delta E \tag{3.25}$$

$$\Delta M = \Delta n \Delta t \tag{3.26}$$

CHAPTER 4. VALIDITY TESTS AND RESULTS

After providing the theoretical background and our code description, we present the simulations we have performed in order to test it and to obtain the desired results. Our calculations will consider two types of satellites: a 1-unit *CubeSat* and a spherical satellite. The former is relevant because it is a very common type of satellite (see, e.g. Chapter 1), and its cubic shape should allow a reasonable estimation of the ballistic coefficient. The latter is useful because, given its geometry, its drag coefficient does not depend on its instantaneous orientation.

More than 1200 *CubeSats* have been launched until early 2020 so, in order to reduce our list (taking also into consideration the range of altitudes we want to study) we only consider the ones which were launched from the *ISS* (International Space Station). These satellites are launched from the Japanese module *Kibo* using the *J-SSOD* (JEM Small Satellites Orbital Deployer, see Figure 4.1).

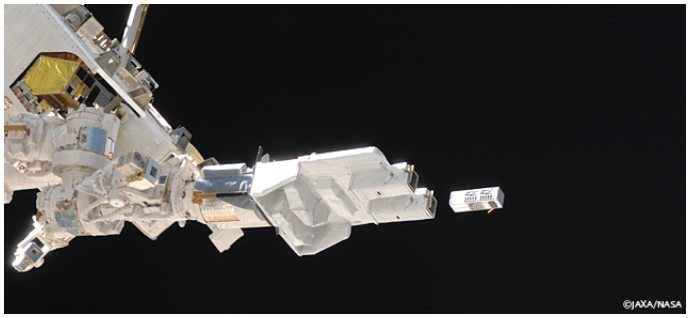

Figure 4.1: JEM Small Satellites Orbital Deployer (*J-SSOD*). Credits:[25]

From the list that we can find in *JAXA* web (see the URL [26]), we have chosen to perform our simulations with the *ArduSat-X CubeSat* (see Figure 4.2), from *NASA*, which was deployed on November 19th 2013, and re-entered into the atmosphere on April 15th 2014.

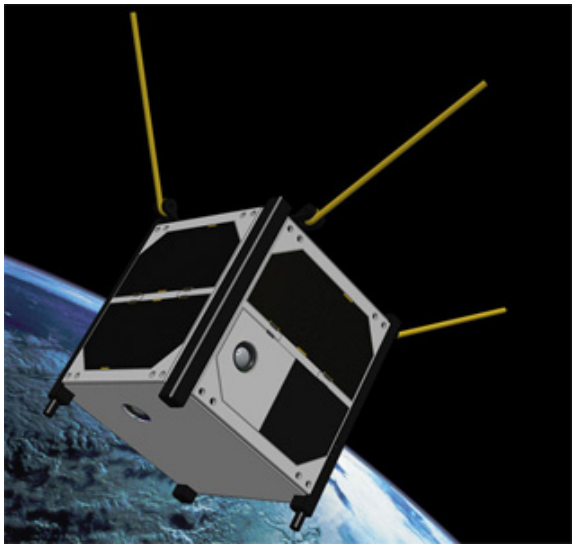

Figure 4.2: *ArduSat*-X *CubeSat*. Credits:[27]

The other satellite we are going to consider is the *STARSHINE-2* (see Figure 4.3). It was a spherical satellite deployed from the *Shuttle Endeavour* on December 16$^{\text{th}}$ 2001, and re-entered the atmosphere on April 16$^{\text{th}}$ 2002. Its mission aimed to measure the density of the lower atmosphere so, we could get some interesting results from this simulation.

Figure 4.3: *STARSHINE-2* deployment from *Endeavour* during STS-108. Credits:[30]

4.1. Initial Conditions

Firstly, we are going to explain where we found the TLEs. To find the complete TLEs record, we use the SpaceTrack service web site (see URL [28]), where we can find all the satellites with their corresponding TLE data. So, we searched and downloaded all the TLEs for *ArduSat-X* and *STARSHINE-2*.

Once we have the complete set of TLEs, it is important to define the satellites characteristics and, specifically, their Ballistic Coefficient, because it affects directly the drag equation. Let us recall that the acceleration due to the drag may be written as:

$$\vec{a} = -\frac{1}{2}\rho v_{\text{rel}}^2 \frac{A\,C_D}{m}\hat{v}_{\text{rel}} \tag{4.1}$$

and the Ballistic Coefficient BC is:

$$BC = \frac{m}{A\,C_D} \tag{4.2}$$

The parameters that we have to define are the mass, the cross section and the C_D. The mass is the simplest one to define because it does not change over time. In our case the mass values are 1 kg for *ArduSat-X* and 39 kg for *STARSHINE-2*. Defining the effective area is not so easy. As we discussed in Chapter 2, if we want to determine the cross-section with precision, we need a good attitude subsystem. In *ArduSat-X* case, we do not have this information. Thus, we must make an estimation: we have assumed that the satellite always has a cross-sectional area of 10×10 cm^2. We know that this will introduce

error in the simulation, but it seems a reasonable option given that there is no alternative. In the case of *STARSHINE-2*, the issue of the orientation is of no consequence, since it is a spherical satellite. We know its diameter (48 cm, see URL [31]) so we can compute its cross section easily.

Determining the C_D is difficult, as we discussed in Chapter 2. But, we know that its value typically ranges between 2 and 2.6 (see Figure 4.4). Therefore, we are going to simulate the trajectory for $C_D = 2$, $C_D = 2.3$ and $C_D = 2.6$ and see which one fits better the real case.

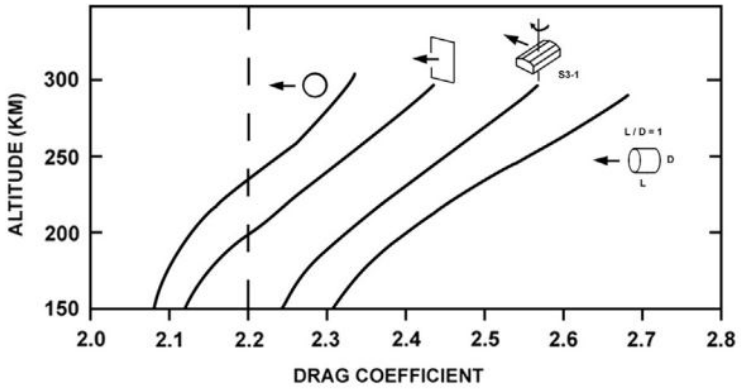

Figure 4.4: Drag coefficient values. Credits:[32]

We also know that the density is another important parameter. As we said in the Introduction, the typical error of these models is about 15-20%. It is true that according to the authors (see [1]), *JB2008* has an error of 10% at 400 km but, in general, this error could well be higher due the variability of the conditions so, its assumed to be 10-15%. Therefore, we are going to perform tests with the density values which we directly obtain from the numerical models, and also multiplying these values by factors of 0.8 and 1.2 in order to account for the uncertainties.

As a way to show more clearly why we have chosen these factors, let us see both the solar and the geomagnetic activity during the time interval for which we will do our simulations. For *ArduSat-X* we have the first TLE on November 22$^{\text{nd}}$ 2013, and the last on April 15$^{\text{th}}$ 2014 so, we are going to plot the corresponding values during this period. We proceed analogously with *STARSHINE-2*, which has its first TLE on December 16$^{\text{th}}$ 2001, and the last one on April 25$^{\text{th}}$ 2002.

We see from Figures 4.5 and 4.6 that we have a solar activity index F10.7B of approximately 150 sfu, and low geomagnetic activity for *ArduSat-X*. On the other hand, with *STARSHINE-2* we see from Figure 4.8 that we have solar activity indices ranging between 200 and 220 sfu, which means that it orbited during high solar activity. However, we see from Figure 4.7 that we have mild geomagnetic activity. With these data, we can obtain from figures 4.9 and 4.10 the error that we can expect.

Prior to the explanation of our simulation results, it is important to understand expected model errors as a function of the geomagnetic and solar activity. Figure 4.9 shows different models' standard deviation (in $\%$) in terms of the 3-hour averaged geomagnetic index A_p. *J70(JB2006)* and its updated version *JB2008*, together with *MSIS*, were described

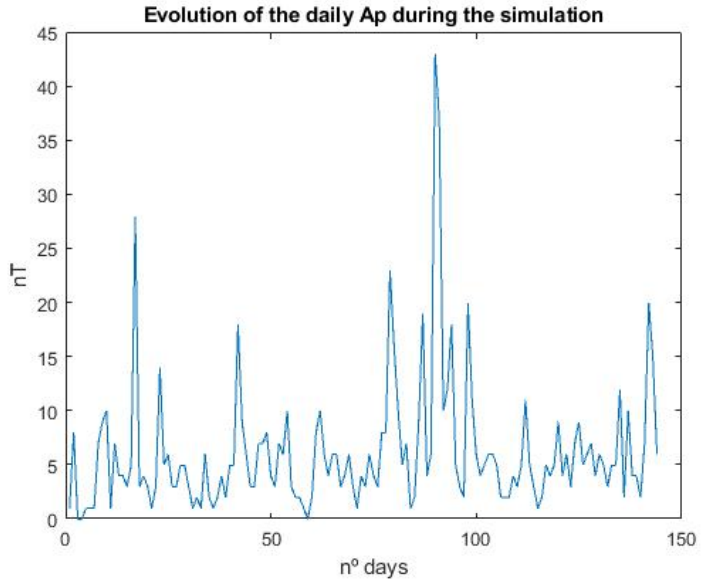

Figure 4.5: Evolution of the daily Ap during the simulation period of *ArduSat-X*.

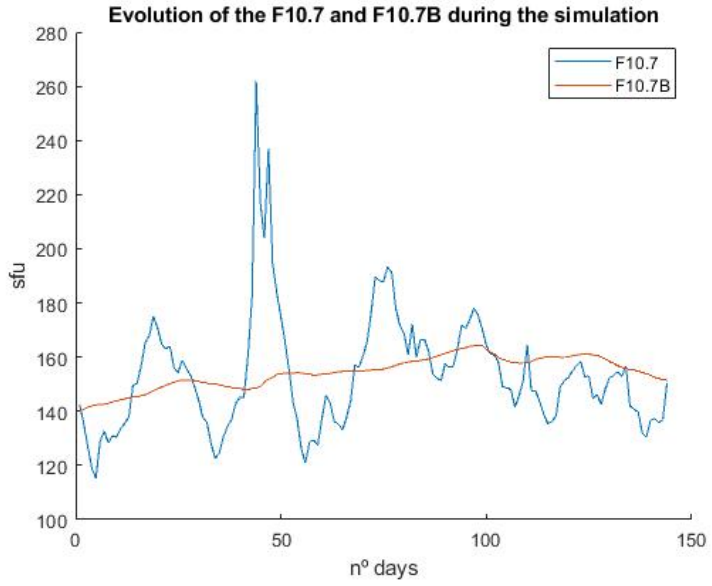

Figure 4.6: Evolution of the solar activity indices F10.7 (blue) and F10.7B (red) during the simulation period of *ArduSat-X*.

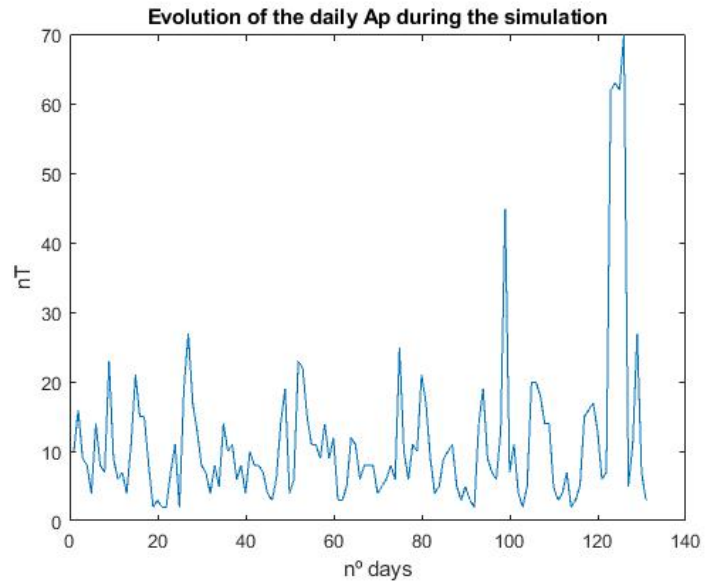

Figure 4.7: Evolution of the daily Ap during the simulation period of *STARSHINE-2*.

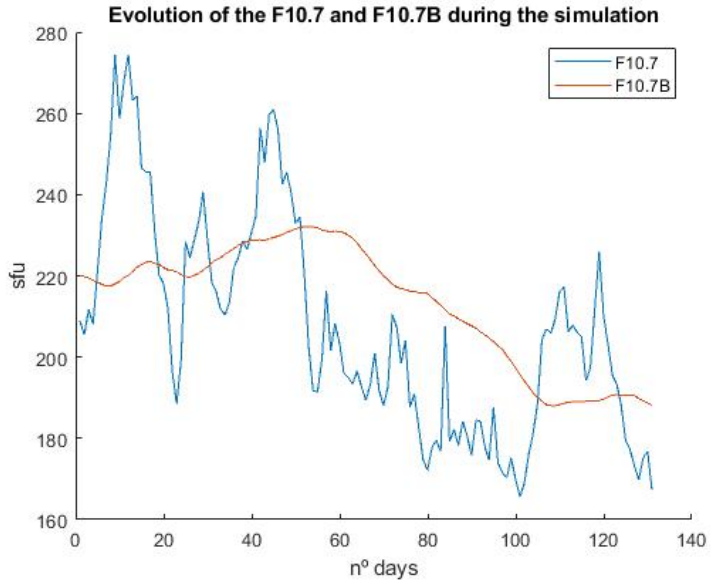

Figure 4.8: Evolution of the solar activity indices F10.7 (blue) and F10.7B (red) during the simulation period of *STARSHINE-2*.

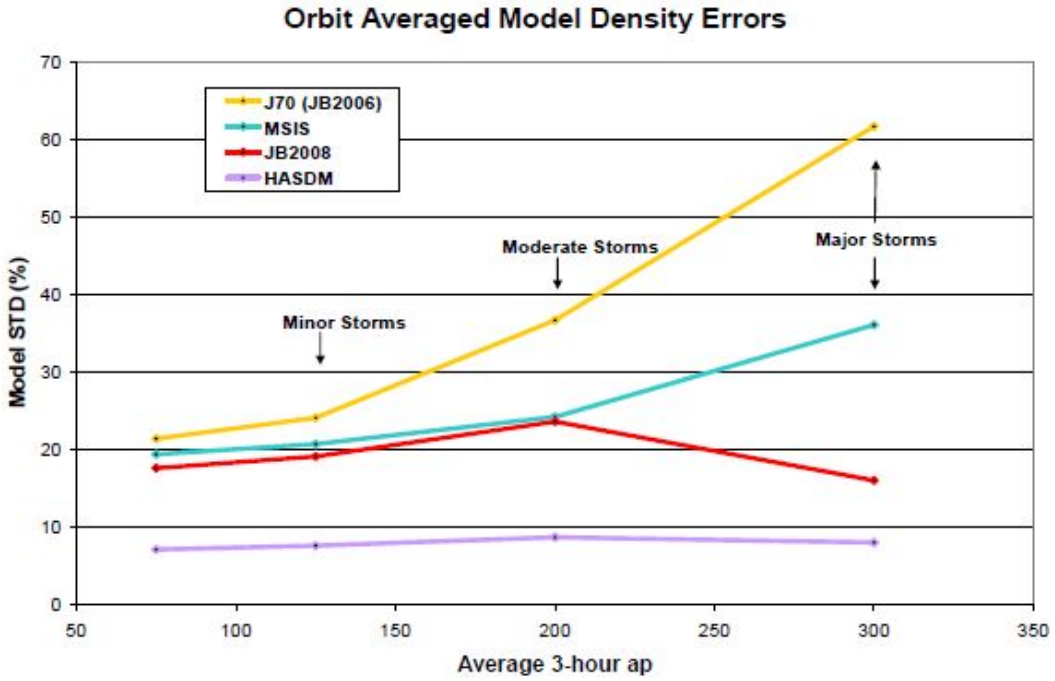

Figure 4.9: 1-standard deviation errors as function of Ap for models J70/JB2006 (yellow), MSIS (cyan), JB2008 (red) and HASDM (purple). Credits:[1]

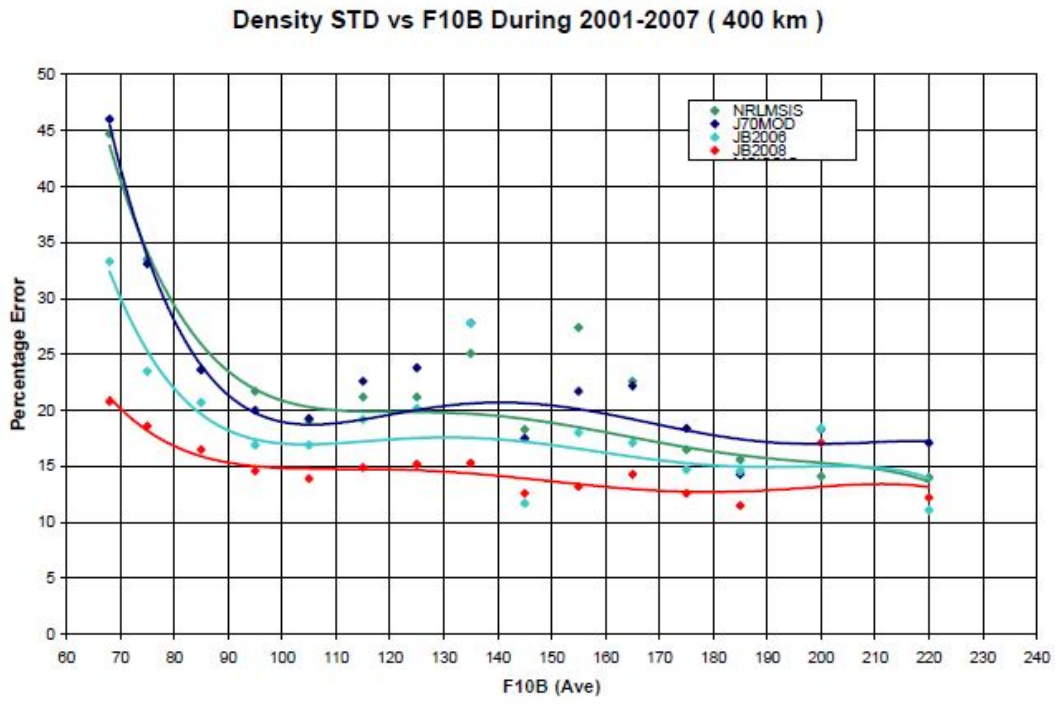

Figure 4.10: 1-standard deviation from different density models at 400 km altitude compared to HASDM density values. Credits:[1]

in Chapter 2. *HASDM* is the Air Force Space Command's High Accuracy Satellite Drag Model. This model processes drag information from trajectories of 75 to 80 satellites. Using this information, it computes the atmospheric density between 200 and 800 km. As *HASDM* measures data in real-time, it can compute the density with much lower errors than the rest of models (about $\sim$ 5%). Thus, it appears a good reference in Figures 4.9 and 4.10.

Figure 4.9 shows that the update of *JB2008* over *J70* considerably reduce errors, specially for averaged geomagnetic activity Ap values above $\sim$ 150 nT. The standard deviation even decreases with average Ap for values above 200 nT when *JB2008* is considered. *MSIS* and *JB2008* present a similar behaviour up to average Ap of 200 nT, giving standard deviations between $\sim$ 18 and 25%. For higher values the performance of *JB2008* is considerably better than that of *MSIS*. The results of Figure 4.10, which represents model behaviour with respect to solar activity, also yield error values near 20% for most average F10.7B values. As a summary, note that expecting average model errors of 20% is, as mentioned above, reasonable. Furthermore, peak errors around 30% cannot be discarded either.

We now present the results of our simulations of the trajectory of the *ArduSat-X* and *STARSHINE-2* with two atmospheric models (*NRLMSISE-00* and *JB2008*), three different C_D values (2, 2.3, 2.6) and three different densities (0.8ρ, ρ and 1.2ρ). Thus we did a total of 18 simulations with each density model.

4.2. Results

We aim to check whether our models improve or, at least, yield similar results to those of SGP4. In order to get this information, we will plot the errors resulting from the simulation of the x, y and z components of the position. In order to compute the associated errors, we compare the results we obtain propagating one TLE until the epoch of the next TLE ($x_{\text{propagated time}_{\text{TLE,i}} \rightarrow \text{time}_{\text{TLE,i+1}}}$, in the case of coordinate x), and the result we get at the epoch (t=0) with the next TLE ($x_{\text{time}_{\text{TLE,i+1}}}$, in the case of coordinate x). For instance, we calculate the error in coordinate x as:

$$\varepsilon_X = \left| x_{\text{propagated time}_{\text{TLE,i}} \rightarrow \text{time}_{\text{TLE,i+1}}} - x_{time_{TLE,i+1}} \right| \tag{4.3}$$

For the errors in coordinates y and z we would proceed analogously (in Figure 4.11 we present the schematics of how errors are integrated in the structure of the problem) . To present our results, we plot results with all density options (nominal density: ρ_{nom}, $0.8\,\rho_{nom}$, and $1.2\,\rho_{nom}$), but with a fixed value of C_D for each position component. In the main text we are showing only the best case. Nevertheless, in appendixes D and E we show the results for all the cases we have studied.

It is important to mention that the errors we compute are not the real errors. TLEs have errors of the order of kilometres at its epoch [10] but it is difficult to exactly assess them; in fact, they may be rather small. TLE accuracy depends on several factors such as the conditions of the space environment, type of orbit, amount of data collected by the surveillance network or the kind of sensors used. All these factors change with each TLE and, hence, so does their accuracy. Indeed, they introduce a systematical error into the problem which

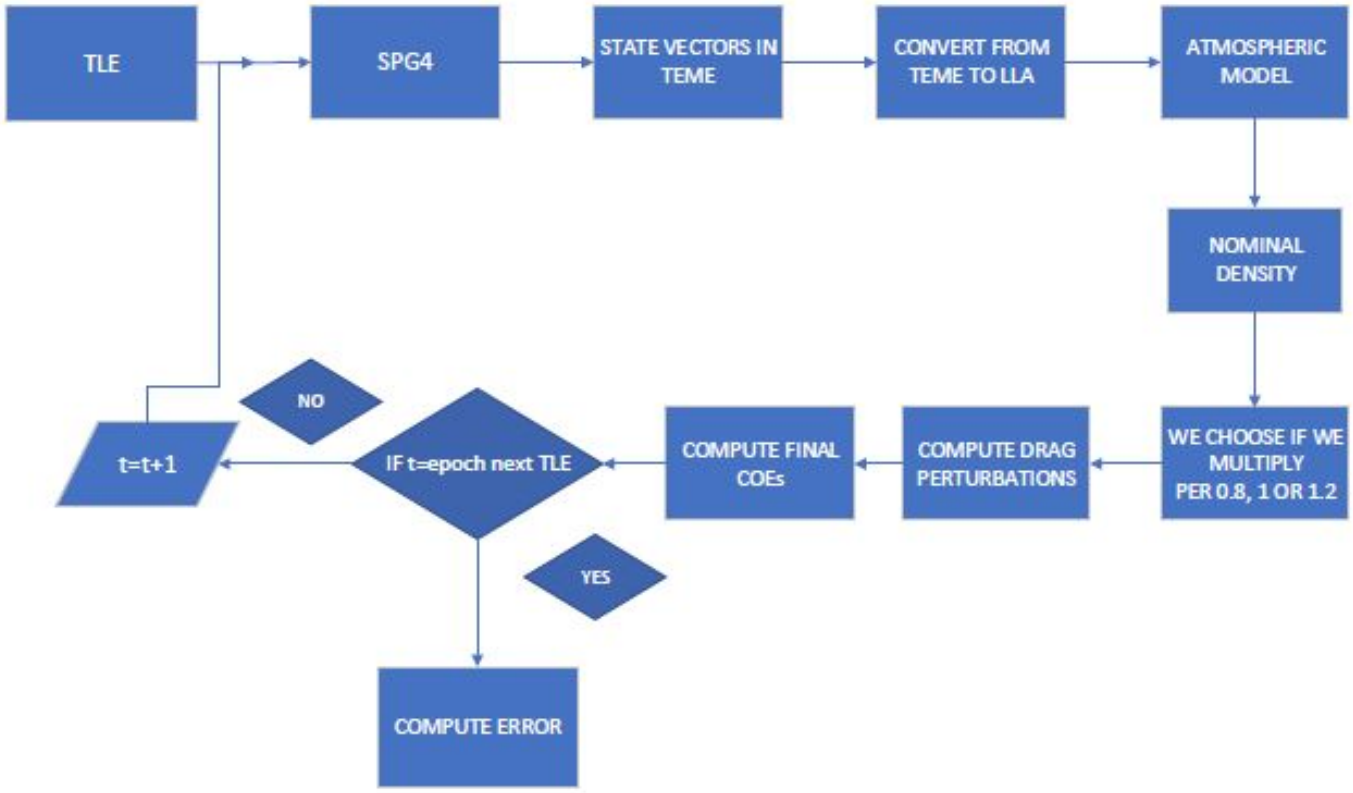

Figure 4.11: Structural organization of the simulation

we cannot determine.

4.2.1. ArduSat-X

The best simulation we have obtained, in this case, corresponds to C_D=2. However, that does not necessarily mean that the real value of C_D is 2 but just that this is the simulation with the best results. For the x axis, we can find the results shown in Figure 4.12. As we see, drawing conclusions from it is difficult. With such an amount of noise, we are losing information because we do not see clearly which model works better. Therefore, we decided to calculate and plot a moving mean of 15 values in order to reduce the noise and thus visualize error trends.

From the results, we can see clearly in all the plots that the problem can be split into three different parts: from $\sim$415 km to $\sim$360 km, from $\sim$360 km to $\sim$250 km and from $\sim$250 km to $\sim$180 km. From now on, we are going to refer to them as high, intermediate and low(-altitude) segments.

For the high segment, we can see that at $\sim$400 km the errors in the three components are the same (approximately 5 km) but then, from $\sim$370 km to $\sim$360 km we can see a peak in error, above all in the x and y components. The reason we have this peak is that there are three consecutive days without TLE data available. Plotting altitude versus the number of TLEs, we can see this anomaly (see 4.16). However, we obtained an extraordinarily good result in the three components with *NRLMSISE-00* when its nominal value was considered (red solid line). We have less than 10 km of mean error in the three components. Even

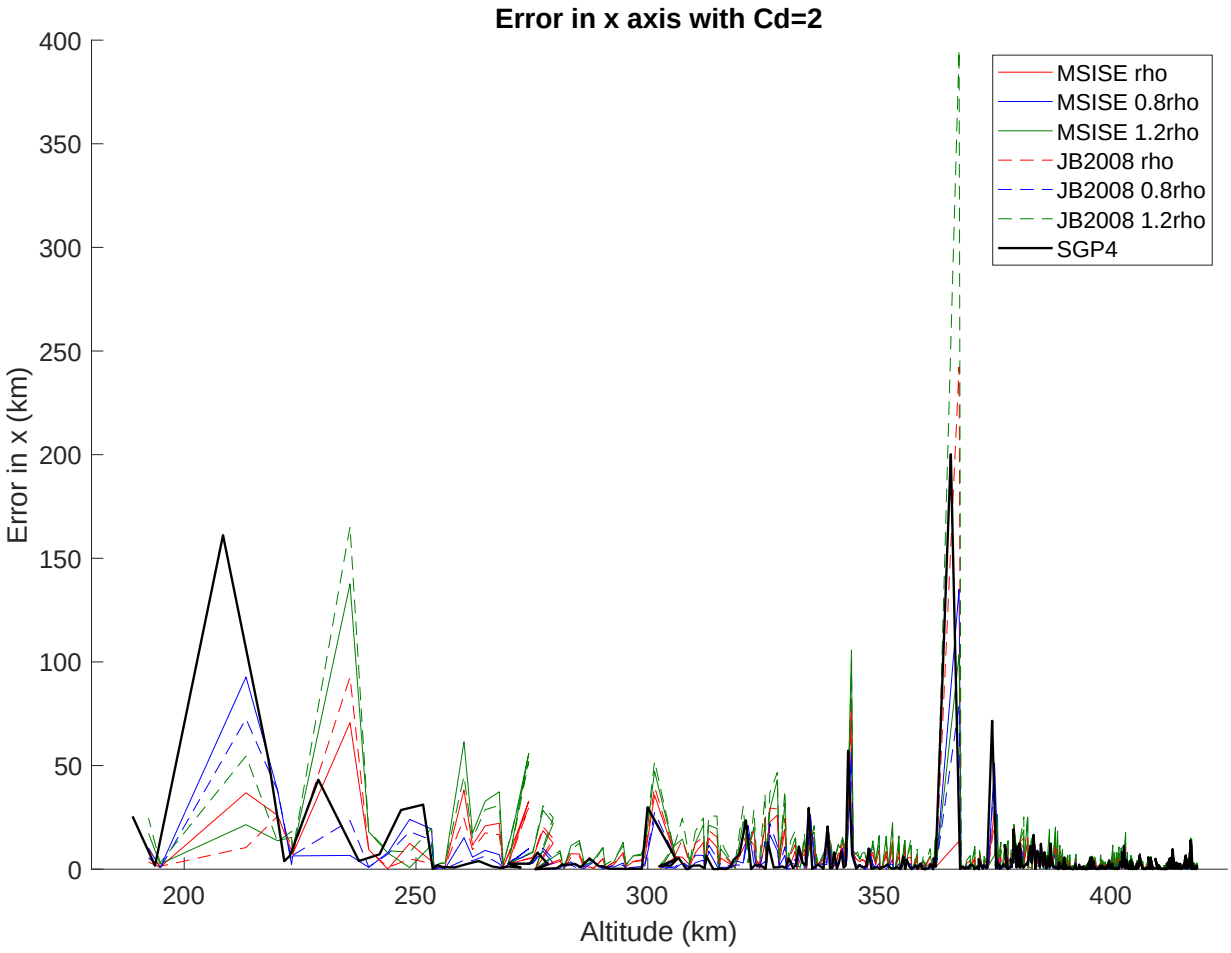

Figure 4.12: Errors in the x axis versus altitude, for calculations performed using MSISE model (solid line) and JB2008 model (dashed line), with variable densities: calculated values (red), 0.8 times the calculated values (blue) and 1.2 times the calculated values (green). All the data presented are for C_D=2.

with 0.8 times the value of the nominal density we still have better results than with SGP4. Nevertheless, we cannot discard the –highly likely– possibility that the ballistic coefficient we have chosen improves the result compensating the errors in density and in SGP4. On the other hand, with *JB2008* we have a poor result if we compare it with *MSISE*. Only with 0.8 times the nominal value of the density, we can get a better result than with SGP4. This result can be explained because when using *JB2008* the error is not compensated, differently to what happens with *MSISE*. The error in this case could be smaller than with *MSISE* and for that reason is not compensated. Apart from that peak, what we see is that for both models with 0.8 times the value of the density we have yields approximately the same results as SGP4.

Secondly, in the intermediate segment, we have an increase in error between 350 km and 300 km, that then decreases until $\sim$270 altitude is reached. The high peaks in 350 km are caused by the lack of TLE data. We have two TLEs within a time interval of 1500

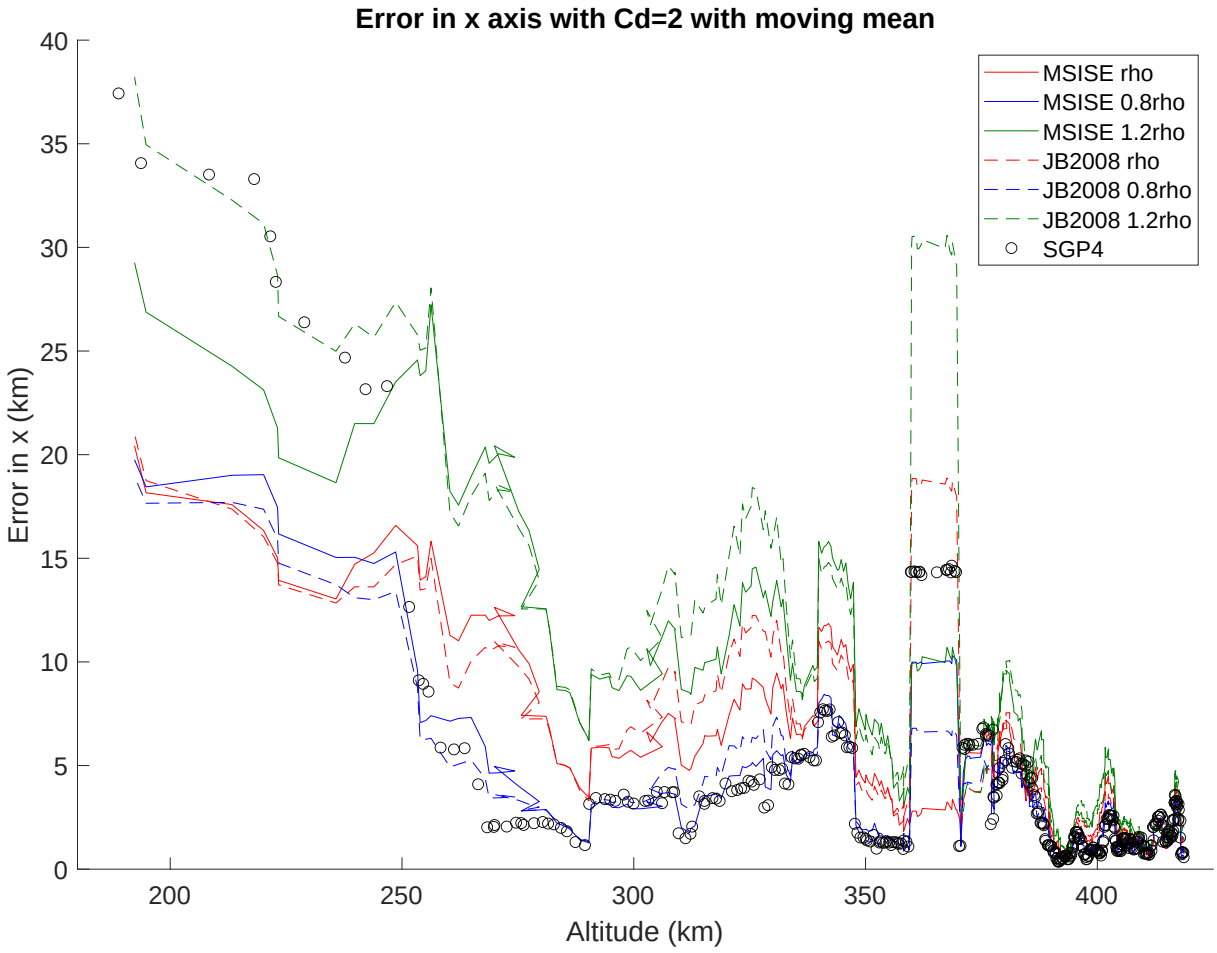

Figure 4.13: Errors in the x axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

minutes. At 300 km, we have also a peak because of a slight increase in the time step size. Here, we start to see that the best option is the blue one (0.8 times the nominal density obtained from *JB2008* and *NRLMSISE-00*). We note that the results are slightly better for *MSISE* but, in general, with both models, we obtain a similar accuracy in the determination. Besides, we do not have a better result than with *SGP4*.

Last but not least, in the low part we already see how the density becomes more important and the errors increase. However, we can get better results than with *SGP4*. Here, we see how using the atmosphere models improves the results. This is important because it tells us that what we have made is correct, and allow us to compute better the satellite trajectory. If we take a look at Figure 4.12 we see that what occurs in figures 4.13, 4.14 and 4.15 may be not quite real. We have two peaks and with the errors we have in both of them, and making a moving mean we get that 0.8 times the nominal value and the nominal value itself yield more or less equal results. However, the 0.8 times the nominal density values are always below the values of *SGP4*, and we get slightly better results. Thus, the best option to propagate the trajectory is the 0.8 case with both models since we do not

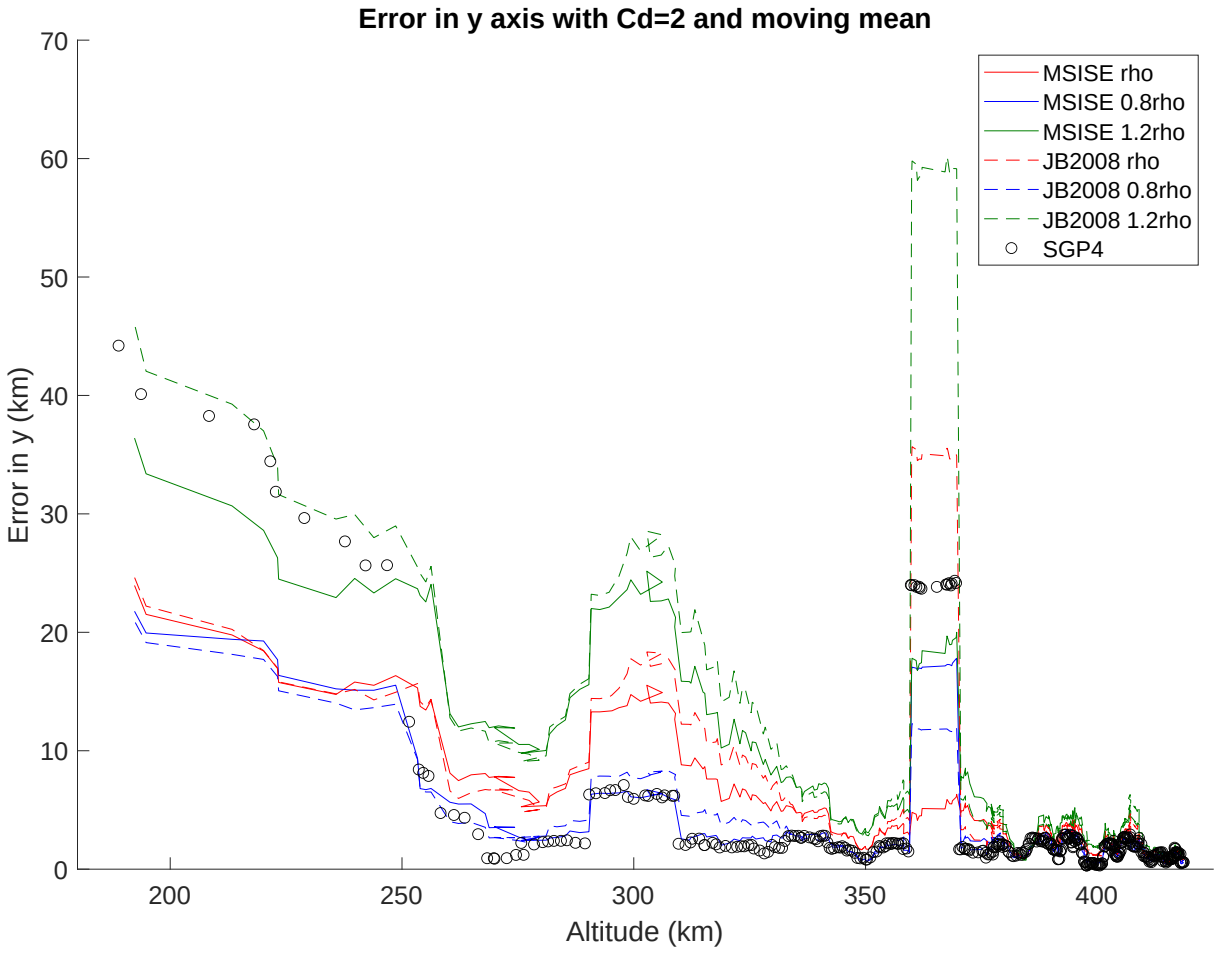

Figure 4.14: Errors in the y axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

find large differences between them.

So far, we have discussed the errors and their tendencies, but we have not explained in detail the sources of errors as well as our strategies to deal with them.

First of all, as we have said before, we have TLEs with a given time span (the time elapsed between consecutive TLEs). As explained in Chapter 3, TLEs tell us the main features of the perturbed orbit of a satellite. So, the accuracy of the trajectory depends on how we compute the different perturbations and how they affect the orbit over time. Since we make an approximation, the longer the time we propagate the orbit, the higher the error we accumulate. In the high and intermediate segments, we have seen that with 0.8 times the value of the nominal density, we obtain results close to those obtained with *SGP4*. Thus, the density is not the main source of error, which is something we can expect because the density in this region is not very high. *SGP4* is a simplified model of perturbations and what we see is that between 350 km and 300 km we have an important error when we have to propagate the orbit for more than $\sim$750 min without new TLEs (that is, without new measured orbital state vectors). Indeed, we can consider that the results are surprisingly

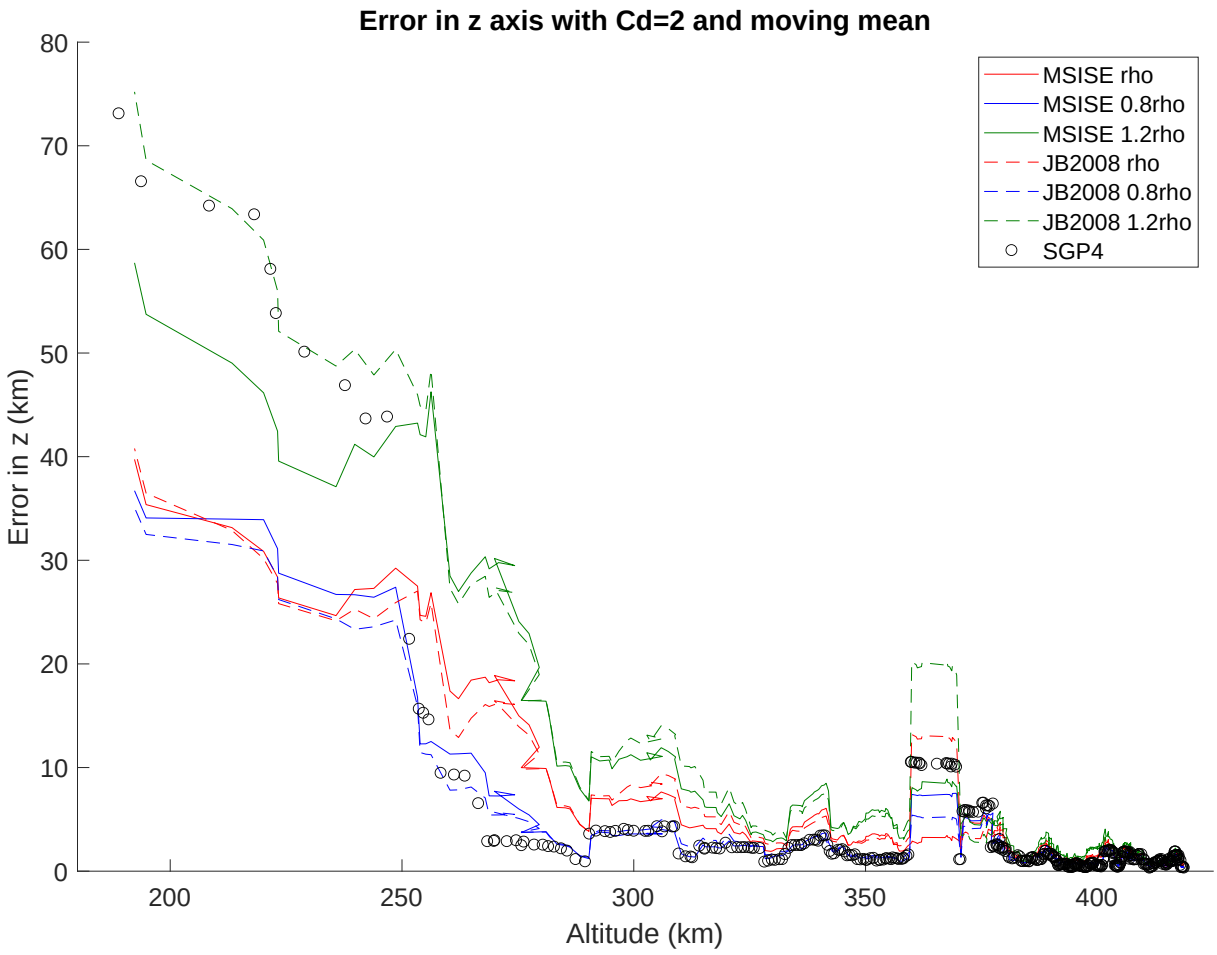

Figure 4.15: Errors in the z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

accurate considering that *SGP4* is a highly simplified model of perturbations, and that we are computing where will the satellite be in 12 hours –or even later– in a very low Earth orbit, where the perturbations are larger. The perturbations change quickly the trajectory, and neither *SGP4* nor our model's predictions can compute that with accuracy; so, if we want better accuracy, we must avoid these long periods without TLEs. Nevertheless, this does not depend on us, and so we have to deal with it.

Another important aspect we see from the results is that the way we model the drag perturbation makes the model better for long periods. This is something we also expected before we performed the simulations. The reason is that the *SGP4* drag term (bstar) is adjusted to the conditions found when the TLE is computed, and this value does not change over time. In our case, we have an atmospheric model that allows us to compute the density in each point, and so we are making a far better approximation. This effect can be seen very clearly in the low segment, and on the high segment as well since we have 3 days without TLE in that peak. The value used for the ballistic coefficient is also important for the low segment. As we discussed above, the cross section and the C_D are both sources

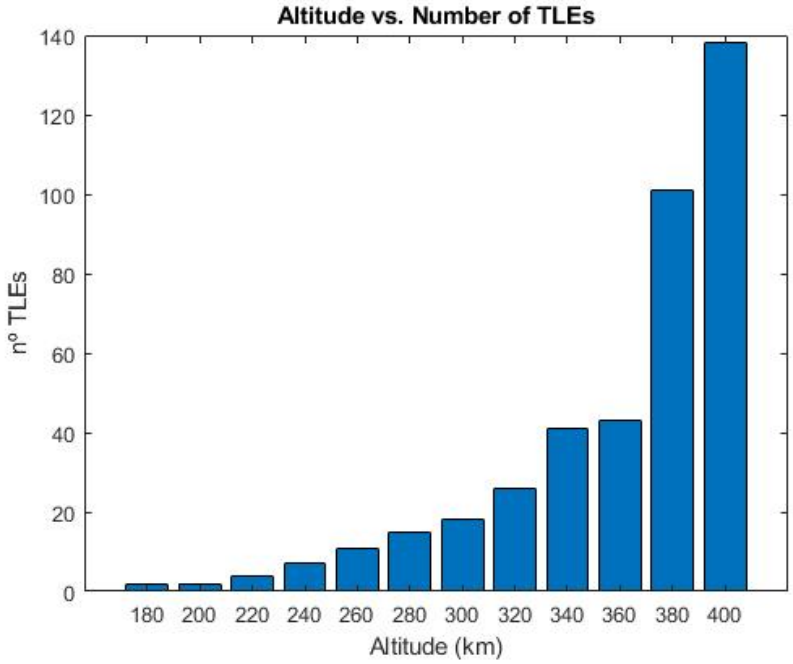

Figure 4.16: Altitude vs. number of TLEs

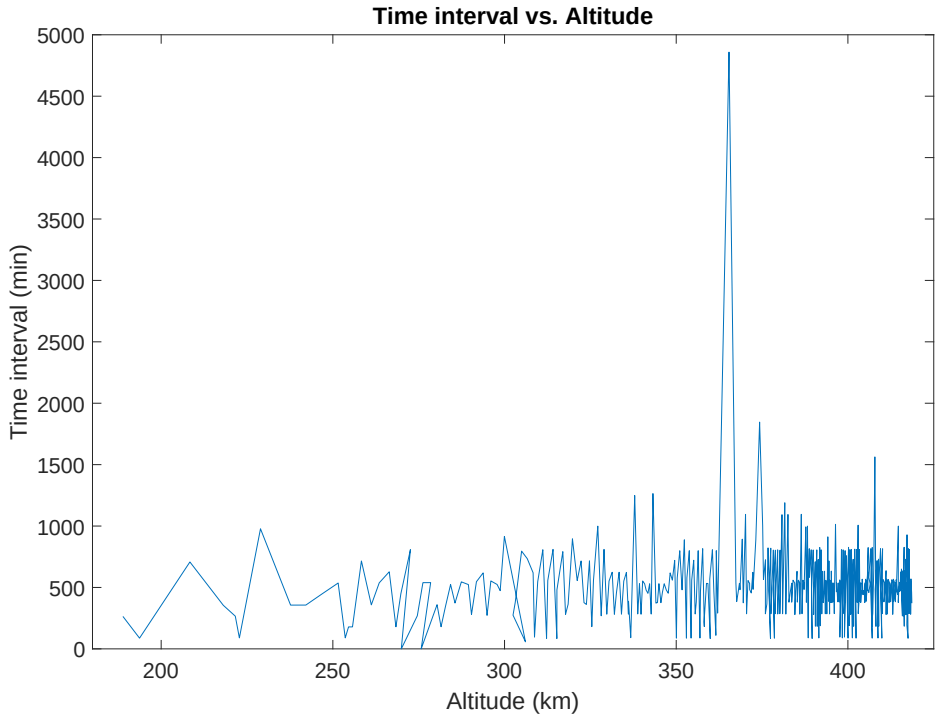

Figure 4.17: Altitude vs. time size

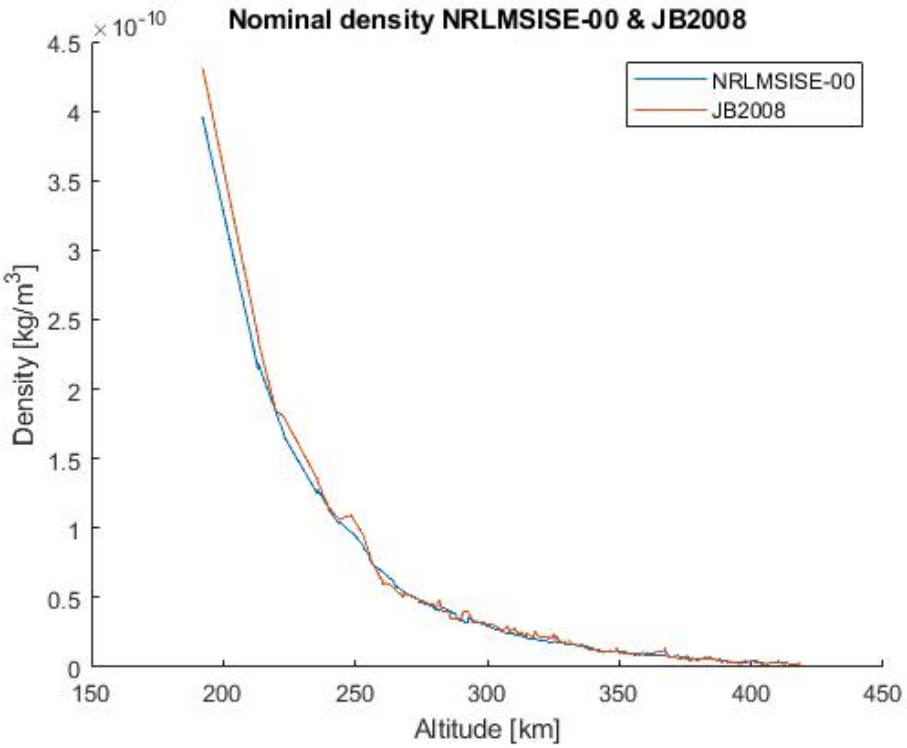

Figure 4.18: Altitude vs. along-orbit average density

of errors. The cross section, as we already know, is uncertain due to uncertainties in the attitude of the satellite. We have chosen a fixed value for this parameter, but the cross section value may change very fast. With C_D we have a lower error but, due to the large difference between the conditions of the illuminated and non-illuminated part of the orbit, its value also changes over time. However, we can say that this error is relatively small and is not the main problem. As we can not determine with precision the value of C_D we have computed models using the drag coefficient as a free parameter and changing it in such a way that we bracket the reasonable values. Besides, during the simulations, the C_D should also have changed its value because the conditions change when the satellite loses altitude. These errors provoke that the value of the ballistic coefficient may not be realistic, and introduce an error –very hard to estimate– when we model the drag perturbation. In a nutshell, we have no way to do it much better because we have to deal with difficult to appraise uncertainties. Despite all this, in the low segment, we have much better results than only with *SGP4*.

Regarding the atmospheric models, we have a small difference when we compute the density, as we can see in figure 4.18. Our simulations with *JB2008* are a bit faster, but this is maybe because *NRLMSISE-00* has to compute more variables. The truth is that with both models we have obtained good results, but *NRLMSISE-00* seems more suitable in these conditions.

One thing we cannot discard is that we are overestimating the ballistic coefficient because we were conservative with the cross section. The best results we got were obtained with C_D=2, and the real value should be more or less 2.2–2.4. If that is true, this means that the errors in the atmospheric models could be bigger than what the bibliography says. With the *STARSHINE-2* we will be able to determine if that it is true.

4.2.2. STARSHINE-2

We proceed now to analyse our results obtained from the study of *STARSHINE-2'* TLEs. In these simulations, we will be able to determine if the conclusions we got with the previous satellite were correct or not. If we recall, now we have a spherical satellite so, we can compute a better ballistic coefficient and then improve the drag calculation.

As in the previous section, we are going to show only the case with the best results; the rest of simulations will be found in Appendix E. The best case, as before, is for C_D=2.

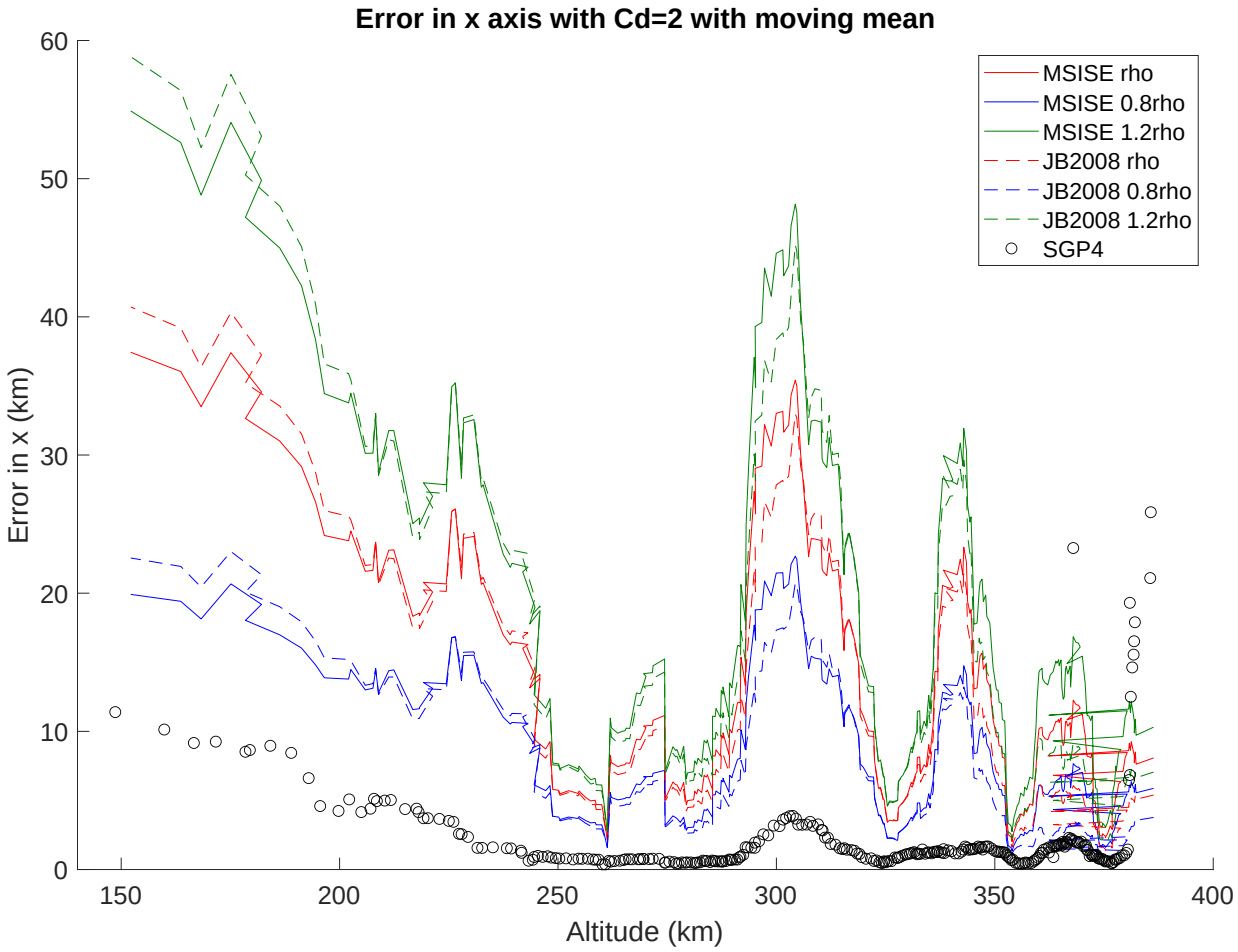

Figure 4.19: Errors in the x axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

As can be observed in Figures 4.19 to 4.21 the errors are significantly larger than expected. Actually, we did not expect to get these results because now the ballistic coefficient is far more reliable. We know there is still some uncertainty because of the C_D , but it does not seem enough to justify this amount of error. Thus, the only existing source of error in this case is the atmospheric model. Our results show that assuming an error of 20% is not enough in this case.

In 2.2 we saw that atmospheric heating is mostly produced between 100 km and 400 km ,

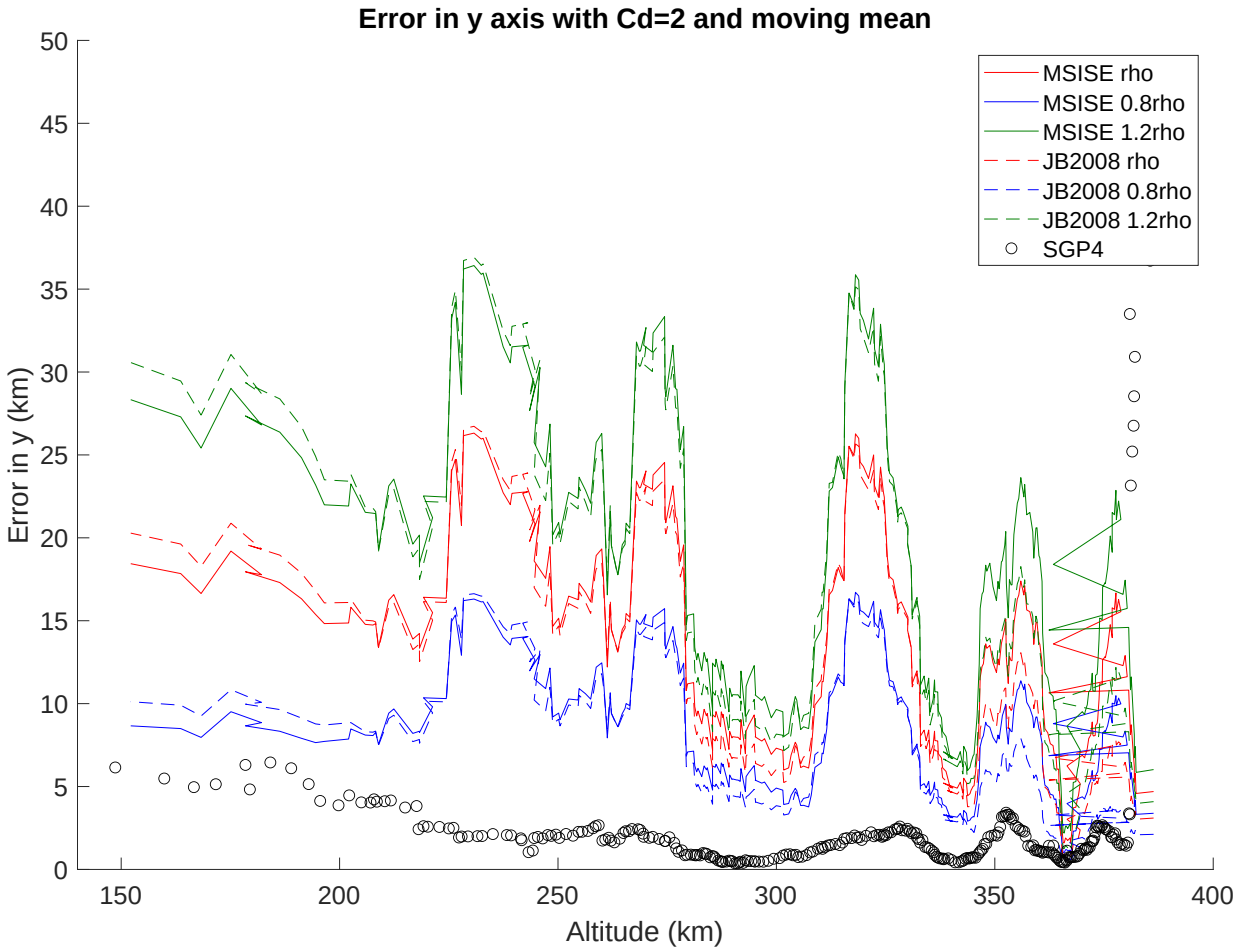

Figure 4.20: Errors in the y axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

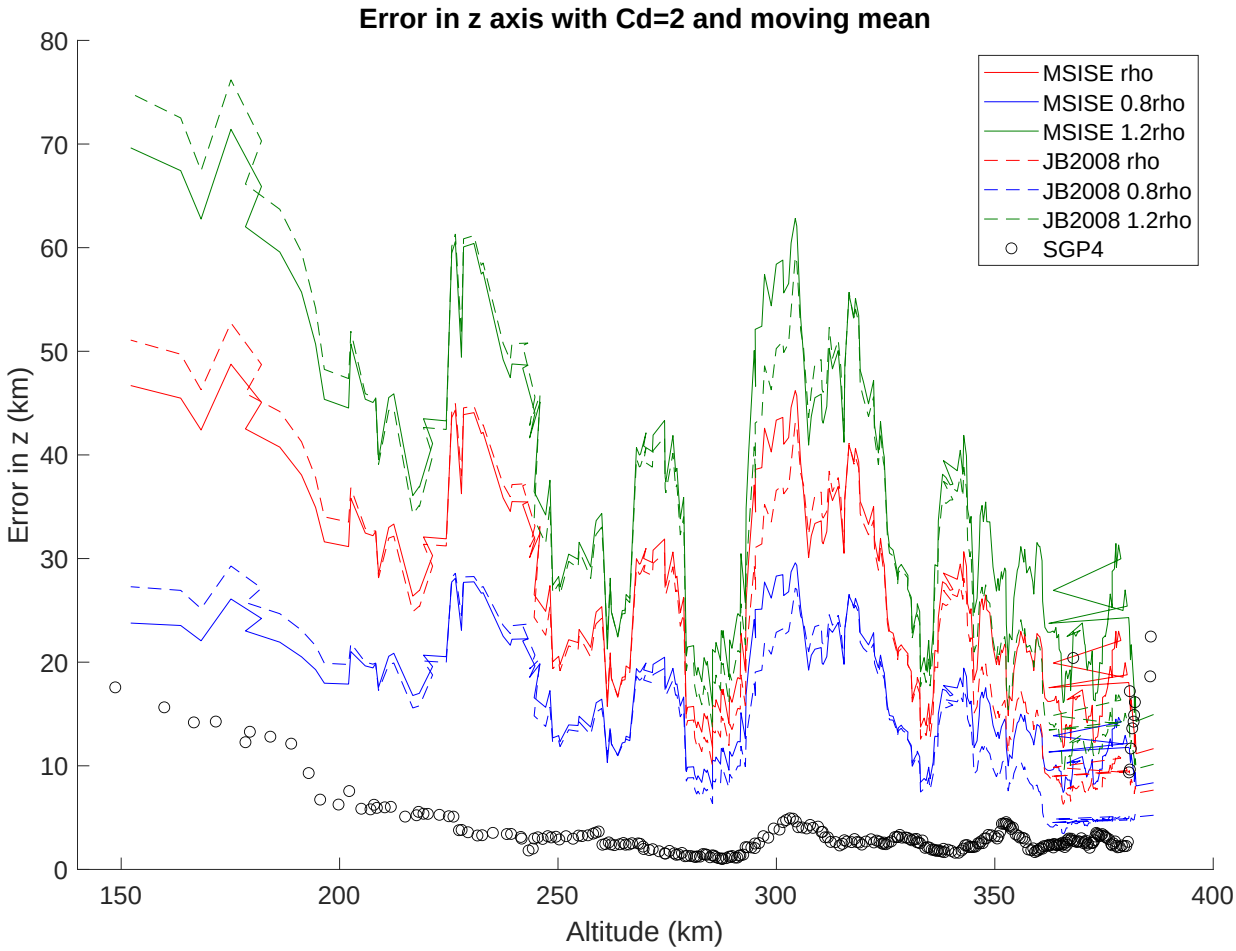

Figure 4.21: Errors in the z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.

and it is crucial to predict how this happens if we want to know the density. Both models, *JB2008* and *NRLMSISE-00*, are empirical models which adjust real data, and can predict some atmospheric characteristics if we know previously some environmental parameters. Therefore, getting large error values means that the models do not have enough data to properly predict the density in this situation.

Measuring data in very low Earth orbit is an extremely difficult task because of the rapidly decaying orbits satellites have at these altitudes. Moreover, we need satellites that are still alive and capable of measuring the drag with their accelerometers. In addition, we need to know as precisely as possible their C_D in order to derive the density with the minimum error. This results in a scarcity of available data to make a precise model. Apart from that fact, in the case of *STARSHINE-2*, there was high solar activity. Indeed, in 2002 there was a peak in solar activity, and this makes that knowing how the thermospheric heating behaves becomes even more crucial. That could explain the size of error we have found and show us that the low thermosphere is not yet well known.

In order to further explore the parameter space, we decided to perform a simulation with 0.5 times the nominal density value to see if we got better results (see figure 4.22). As we expected, results improved considerably, and similar trends to those identified in the case of *ArduSat-X* can be observed. We see that below $\sim$230 km we have less error than with SGP4 and, as before, when the altitude is "high", we get more or less the same results than with SGP4.

Nevertheless, we have here a combination of errors and we have to take them into account when we draw our conclusions. The reason is that we do not know exactly the error of the atmospheric models and also we have to consider that the ballistic coefficient is not completely accurate. However, we can see that the error of the atmospheric models here is around 40-50% and they overestimated the density. In fact, we can say that below 200 km the error is less than 50% since at low altitudes the C_D is near to 2 rather than 2.3 as we know from Figure 4.4.

Another thing we can see is that the error has less dependence on the time span between consecutive TLEs. That is something we expected as well, since we can compute a better ballistic coefficient. However, they still play an important role. We have lower errors, which means that with less TLEs we can get good results but, if we want to have the best accuracy, we need time span values lower than the ones we have now.

During the simulations, we also detected that *JB2008* is faster than *NRLMSISE-00*, as it was in the *ArduSat-X* case. However, what we do not see is that difference between the two extremes (0.8 and 1.2) with *JB2008*. As we note in Figure 4.24, both models have the same average density along-orbit and the two extremes with both models have more or less the same difference. To sum it up, what we can deduce is that both models overestimated the density, which results in huge errors when we propagate *STARSHINE*'s orbit.

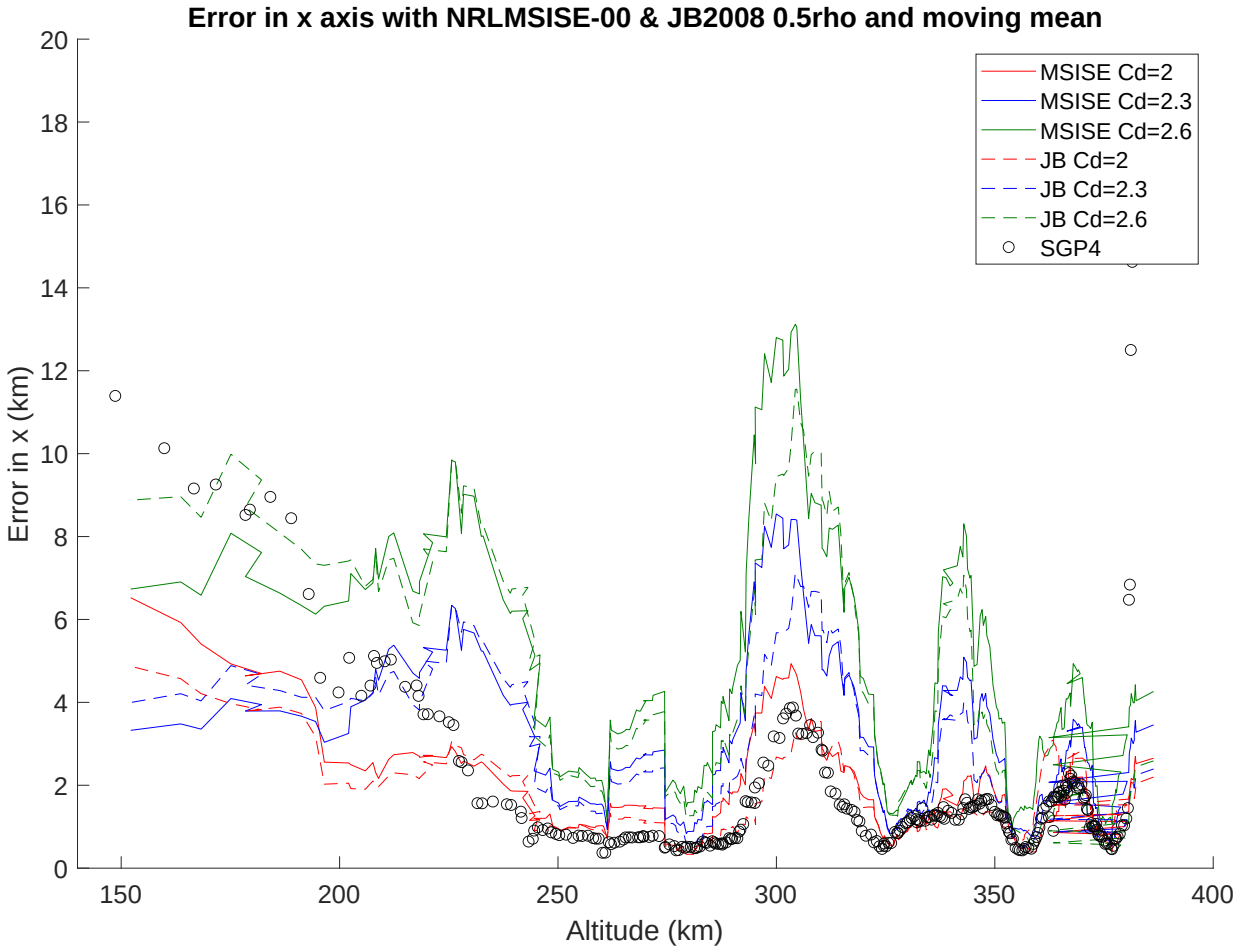

Figure 4.22: Errors in the x axis versus altitude using moving mean of 15 values and different C_D with 0.5 times the density value for both atmospheric models.

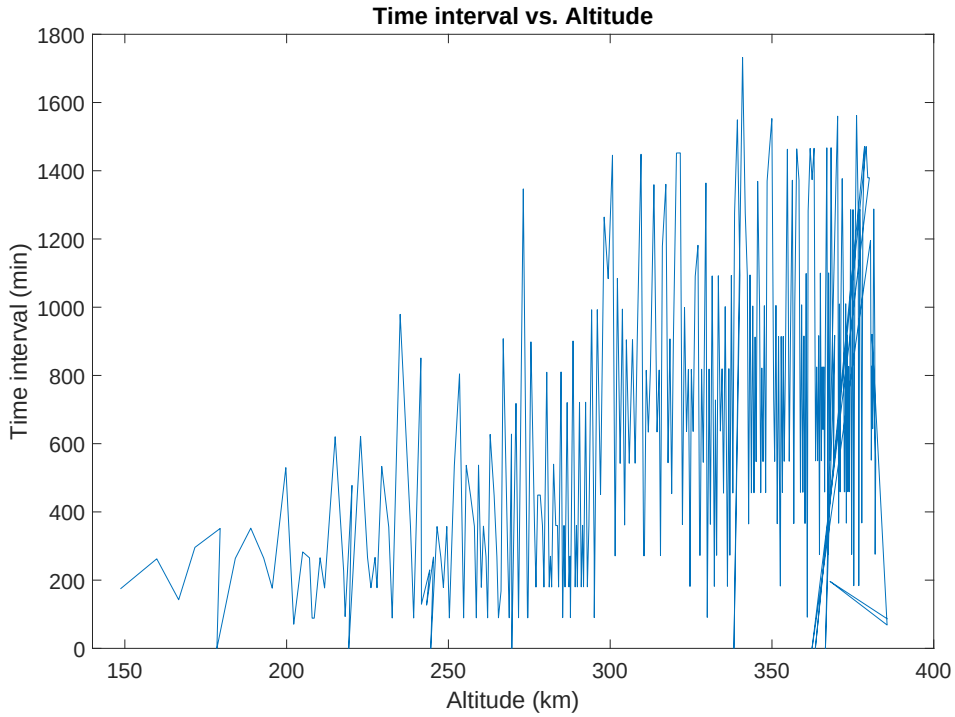

Figure 4.23: Time interval *STARSHINE-2* versus altitude.

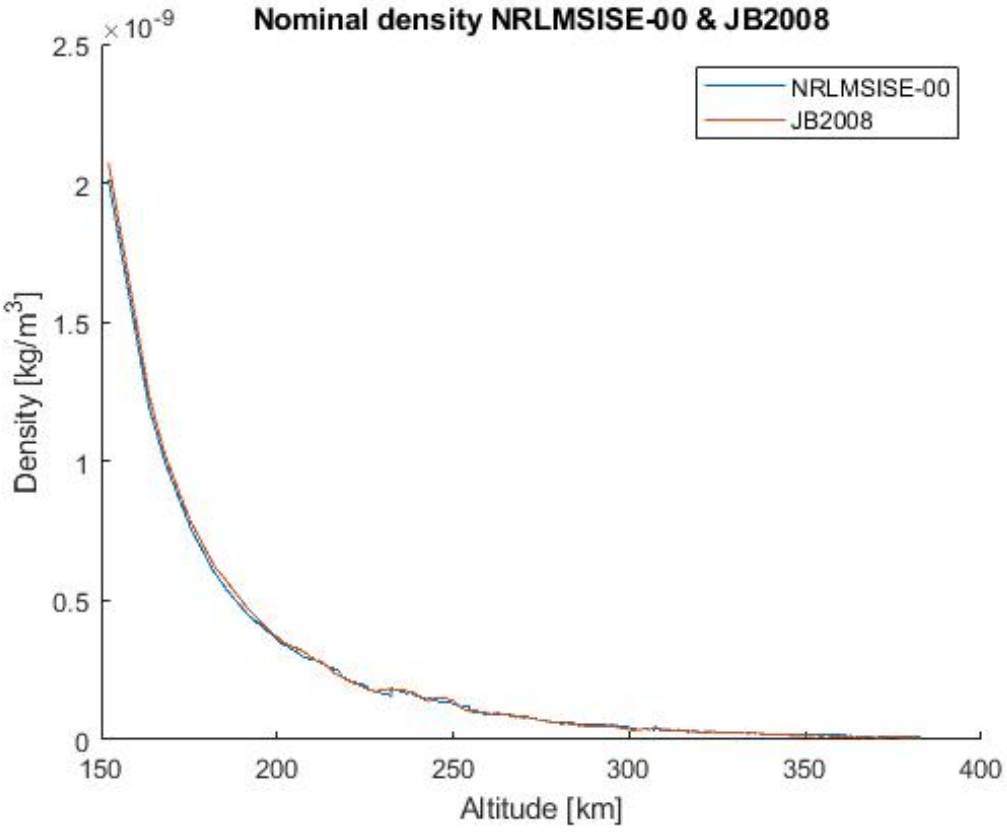

Figure 4.24: Altitude vs. along-orbit average density

CHAPTER 5. CONCLUSIONS

5.1. Summary and conclusions

The aim of this study was to check whether, using current atmospheric models, we can compute the density of the lower thermosphere with precision. To do so, we have performed a number of simulations with two different satellites (*ArduSat-X* and *STARSHINE-2*) and two different frequently used atmospheric models (*JB2008* and *NRLMSISE-00*). The various simulations we have done are summarized in Table 5.1.

Name	**SGP4**	C_D**=2**	C_D**=2.3**	C_D**=2.6**	ρ	**1.2**ρ	**0.8**ρ	**0.5**ρ
ArduSat-X	X	X	X	X	X	X	X	–
STARSHINE-2	X	X	X	X	X	X	X	X

Table 5.1: Summary of the simulations.

In order to perform our simulations we have developed a code in MATLAB. Our code is based on the Simplified General Perturbations orbit propagator *SGP4*, but we additionally compute the drag perturbation on our own using a convenient atmospheric model. Note that the codes mentioned above, their integration in our comprehensive program, and the interpretation of some input variables and the final results were far from straightforward, and took a considerable part of the dedication time of our project. As an example, the use of "mean" rather than specific time variables in *SGP4* led to confusion when comparing with TLE data. Also, the non-availability of the procedures to generate the TLEs from the observational data further hampered our work. Another interesting conceptual issue came up when converting *SGP4* output into COEs, which led to osculating elements. That is, the associated variables vary considerably, even in time scales of seconds. In addition, when low eccentricity orbits are considered, the argument of the perigee presented huge variations during a single orbit. The reason is that this variable is almost undetermined when eccentricities close to zero are considered. These important issues are described in the bibliography (e.g. [10]). The entire interpretation process allowed us to deeply understand the codes and the problems of orbit determination and atmospheric modelling.

With the atmospheric models, we had fewer problems since the bibliography appeared more consistent, and the required variables and their interpretation was clearer. However, we had a lot of work to convert *SGP4* outputs to atmospheric models inputs. On the other hand, we had several problems with the drag. VOP equations have limitations with very low eccentricities which forced us to find an appropriate alternative. We finally developed VOP equations and found that using a relation between the semi-major axis (which actually could be obtained with existing VOP equations), and the mean motion, we could derive the rate of change of the mean anomaly, which is affected by the drag. Once we had all the code written and carefully tested, we tried to optimize it so that we could speed up calculations. We found that we had some redundancies in the code, which were convenient for the testing process, but made it slow. Solving these small problems, we considerably reduced the time of the simulation from $\sim$ 24 hours to 10-15 minutes and once we did this, we started the relevant simulations.

Detailed analysis is delicate, as the involved parameters are both uncertain and entangled

with each other (see Section 2, where we summarized all the uncertainties in the models we are considering.)

Bibliographic data tend to underestimate density errors, and most quoted values are not higher than $\sim 20\%$. The reason is that, in most cases, they were determined under optimal or standard circumstances. Specifically, most density analyses were done at high altitudes, where the general zone of interest of active satellite dynamics is located. Our most relevant conclusion is that density errors at thermospheric altitudes are considerably higher than expected, and may reach a factor ~ 2 (higher than most reported values in the bibliography). This makes it worthwhile to do further research on this subject. In addition, we can also conclude that:

- as expected, density errors increase for lower altitudes in thermospheric regions (where drag effects become more relevant);
- it is important to better constrain C_D values in order to disentangle atmospheric density values;
- solar activity is critical for the determination of the density. We analysed the TLEs of *STARSHINE-2*, a spherical satellite whose C_D should be practically constant (and thus help constrain density values), which yielded results poorer than those from the CubeSat *ArduSat-X*. The reason is that solar activity was near maximum during the flight of *STARSHINE*. Nevertheless, with *ArduSat-X* we might have underestimated the ballistic coefficient, so we cannot discard that we have similar results in terms of error with both satellites. Further research is needed to clarify this issue.

5.2. Future work

Our work emphasizes the importance of doing further research on the determination of thermospheric densities. Specifically, we propose the following lines of future work:

- Further analyse model errors and determine how they propagate and, ultimately, affect position determination accuracy;
- analysis of the expected behaviour of a proposed spherical picosatellite with a GNSS receiver to test our results;
- perform systematic analysis of existing spherical satellites, which would allow to contemplate a number of situations with almost constant C_D;
- extend our analysis to the case of eccentric satellites. Such analysis would be easy to implement with a minor modification of the code we developed;
- the former two studies would allow us to consider situations of different solar activity and analyse its crucial effects on density error estimation;
- study possible applications in orbit determination;
- consider the viability of measuring the C_D of small satellites in the wind tunnel for thermospheric conditions *MARHy* (ICARe Lab in Orleans, CNRS, France);

- add a model of thermospheric winds.

BIBLIOGRAPHY

[1] Bowman, B.R., Tobiska, W.K, Marcos, F.A., Huang, C.Y., Lin, C.S., Burke, W.J. "A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices". *AIAA*. 6438, (2008). doi:10.2514/6.2008-6438 IX, IX, 5, 11, 27, 30

[2] Picone, J.M., Hedin, A.E., Drob, D.P., Aikin, A.C., "NRLMSISE-00 empirical model of the armosphere: Statical comparisons and scientific issues". *Journal of geophysical research*. **107**(1468), (2002). doi:10.1029/2002JA009430 5, 11

[3] CCMC Metadata Registry. *The Jacchia-Bowman 2008 Empirical Thermospheric Density Model*. [Query: March 11, 2020]. Available: https://kauai.ccmc.gsfc.nasa.gov/CMR/view/model/SimulationModel?resourceID=spase://CCMC/SimulationModel/JB2008/2008 11

[4] Pederick, L.H., Cervera, M.A., "Semiempirical Model for Ionospheric Absortion based on the NRLMSISE-00 atmospheric model". *Radio Science*. **49**(49), 81-93, (2014) IX, 9

[5] Frazier, Sarah. Revolutions in Understanding the Ionosphere, Earth's Interface to Space. At: *NASA* [online]. NASA, 2017. [Query: March 13, 2020]. Available: https://www.nasa.gov/feature/goddard/2016/revolutions-in-understanding-the-ionosphere-earth-s-interface-to-space IX, 13

[6] Minguell Montes, B. Determination of drag coefficients in a free molecular flow. Treball de fi de grau, UPC, Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, 2018 [Query: March 13, 2020]. Available: https://upcommons.upc.edu/handle/2117/121121 13

[7] Air Force Geophysics Laboratory. " Ionospheric physics". *Handbook of Geophysics and the Space Environment*. (Air Froce Geophysics Laboratory, Hanscom AFB, 1985):9-1 - 9-2. IX, 8

[8] Ionosphere. At: *Wikipedia* [online]. Wikimedia Foundation, 2007. [Query: March 14, 2020]. Available: https://en.wikipedia.org/wiki/Ionosphere#/media/File:Atmosphere_with_Ionosphere.svg IX, IX, 5, 7

[9] Kelso, T.S, Orbit Propagation: Part II. At: *CelesTrak* [online]. CelesTrak, 2019. [Query: March 22, 2020]. Available: https://celestrak.com/columns/v01n04/ 16

[10] Vallado, David A., Crawford, P., Hujsak, R., Kelso, T.S., "Revisiting Spacetrack Report ♯ 3". *AIAA*. 6753, 1-25. (2006). doi:10.2514/6.2006-6753 15, 16, 31, 45

[11] Thermosphere-overview. At: *UCAR* [online]. UCAR (University Corporation for Atmospheric Research), 2008. [Query: April 13, 2020]. Available: https://scied.ucar.edu/shortcontent/thermosphere-overview 7

[12] F10.7 cm Radio Emissions. At: *NOAA* [online]. NOAA. [Query: April 13, 2020]. Available: https://www.swpc.noaa.gov/phenomena/f107-cm-radio-emissions 11

[13] Astrodynamics Software. At: *Celestrak* [online]. Celestrak, 2019. [Query: February 10, 2020]. Available: https://celestrak.com/software/vallado-sw.php 16, 17, 23

[14] Vallado, D. A.. "Statical Orbit Determination". *Fundamentals of Astrodynamics and Aplications*. (First edition, McGraw-Hil, 1997): 649-740. ISBN: 0-07-0668340-5 17

[15] Vallado, D. A., Crawford, P.. "SPG4 orbit determination". *AIAA*. 6770. (2008). doi: 10.2514/6.2008-6770 17

[16] Nutation. At: *Wikipedia* [online]. Wikimedia Foundation, 2004. [Query: April 15, 2020]. Available: https://en.wikipedia.org/wiki/Nutation$\sharp$/media/File:Praezession.svg IX, 19

[17] Kelso, T.S, Orbital Coordinate System: Part I. At: *CelesTrak* [online]. CelesTrak, 2019. [Query: April 15, 2020]. Available: http://www.celestrak.com/columns/v02n01/ 19

[18] EOP and Space Weather Data. At: *Celestrack* [online]. Celestrak, 2020. [Query: April 18, 2020]. Available: https://celestrak.com/SpaceData/ 20

[19] Vallado, D. A. . *Fundamentals of Astrodynamics and Applications*. (First edition, McGraw-Hil, 1997). ISBN: 0-07-0668340-5 IX, X, X, 9, 20, 22, 23, 55, 57

[20] Mahooti, M.. NRLMSISE-00 Atmosphere Model. At: *Matlab*[online]. Matlab Central File Exchange, 2020. [Query: April 18, 2020]. Available: https://es.mathworks.com/matlabcentral/fileexchange/56253-nrlmsise-00-atmosphere-model 20

[21] The Jacchia-Bowman 2008 Empirical Thermospheric Density Model. At: *Space Enviroment*[online]. Space Environment, 2017. [Query: May 3,2020]. Available: http://sol.spacenvironment.net/jb2008/ 21

[22] Mahooti, M.. Jacchia-Bowman Atmospheric Density Model. At: *Matlab*[online]. Matlab Central File Exchange, 2020. [Query: April 18, 2020]. Available: https://es.mathworks.com/matlabcentral/fileexchange/56163-jacchia-bowman-atmospheric-density-model 21

[23] Kelso, T.S, Orbital Coordinate System: Part III. At: *CelesTrak* [online]. CelesTrak, 2019. [Query: May 10, 2020]. Available: https://celestrak.com/columns/v02n03/ 59

[24] Geodetic to Geocentric Latitude. At: *Matlab*[online]. Matlab, 2020. [Query: May 10, 2020]. Available: https://es.mathworks.com/help/aeroblks/geodetictogeocentriclatitude.html 59

[25] ISS deployment. At: *JAMSS*[online]. Japan Manned Space Systems Corporation, 2020. [Query: June 30, 2020]. Available: https://www.jamss.co.jp/en/business/deployment.html IX, 25

[26] List of deployed CubeSats using J-SSOD. At: *JAXA*[online]. Japan Aerospace Exploration Agency, 2019. [Query: May 10, 2020]. Available: https://iss.jaxa.jp/en/kiboexp/jssod/history/ 25

[27] ArduSat 1, X. At:*Gunter's space page*[online]. Gunter's space page, 2018. [Query: May 10, 2020]. Available: https://space.skyrocket.de/doc_sdat/ardusat-1.htm IX, 25

[28] Space-Track. t:*Space-Track*[online]. Space-Track. [Query: May 10, 2020]. Available: https://www.space-track.org/ 26

[29] Small satellites. At: *Wikipedia*[online]. Wikimedia Foundation, 2019. [Query: April 15, 2020]. Available: https://en.wikipedia.org/wiki/Small_satellite#Femtosatellites 5

[30] STARSHINE(satellite). At: *Wikipedia*[online]. Wikimedia Foundation, 2020. [Query: July 28, 2020]. Available: https://en.wikipedia.org/wiki/STARSHINE_(satellite) IX, 26

[31] Starshine 1,2 4. At:*Gunter's space page*[online]. Gunter's space page, 2020. [Query: July 19, 2020]. Available: https://space.skyrocket.de/doc_sdat/starshine-1.htm 27

[32] Vallado, D., Finkleman, D.. "A critical assessment of satellite drag and atmospheric density modeling." *Acta Astronautica*.**Volume**(95), 141-165. (2014) IX, 27

[33] International Reference Ionosphere-IRI-2012. At:*Community Coordinate Model Center*[online]. Community Coordinate Model Center, 2020. [Query: September 2, 2020]. Available: https://ccmc.gsfc.nasa.gov/modelweb/models/iri2012_vitmo.php IX, 9

APPENDICES

APPENDIX A. TOPOCENTRIC HORIZON COORDINATE SYSTEM (SEZ)

This reference frame rotates with the Earth and is widely used for satellite observation and sensor systems. The S axis points to the south from the site. The E axis points to the east from the site (undefined in the poles). Finally, the Z axis points radially outwards from the site following the same direction of the position vector of the site from the Earth's center.

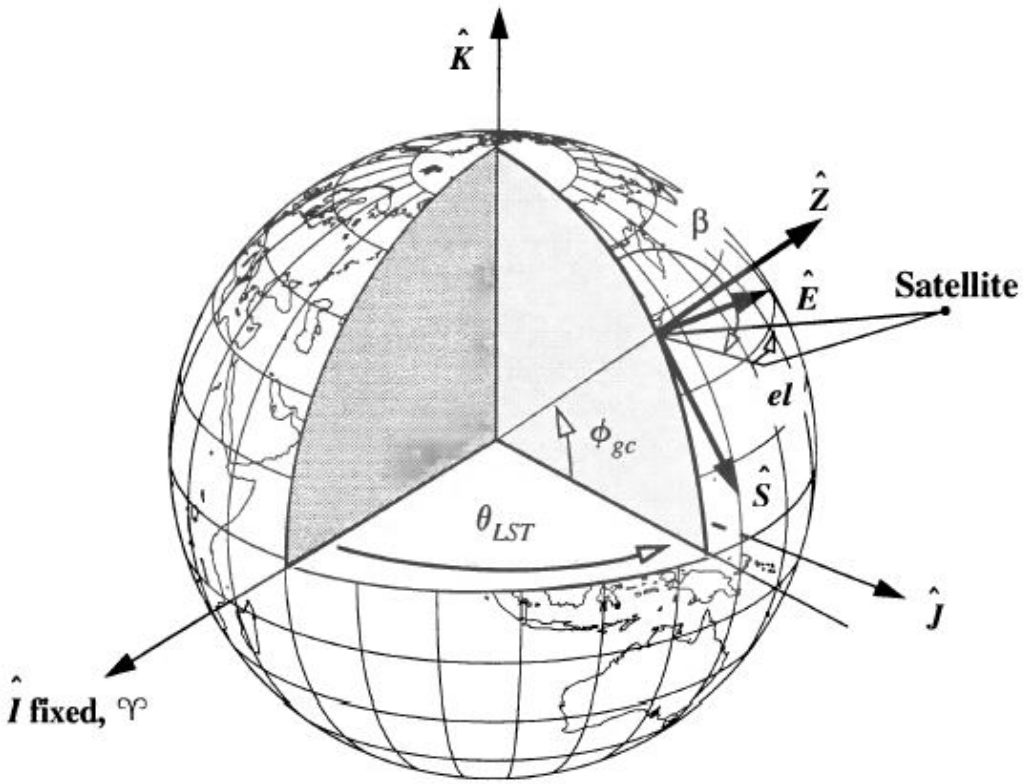

Figure A.1: Topocentric Horizon Coordinate System. Credits:[19]

APPENDIX B. SATELLITE COORDINATE SYSTEM (RSW)

This coordinate system is centered on the satellite and moves with it. R axis points radially outwards from the satellite following the same direction of the position vector of the satellite from the Earth's center. W axis is normal to the orbital plane. Finally, S axis is normal to the position vector.

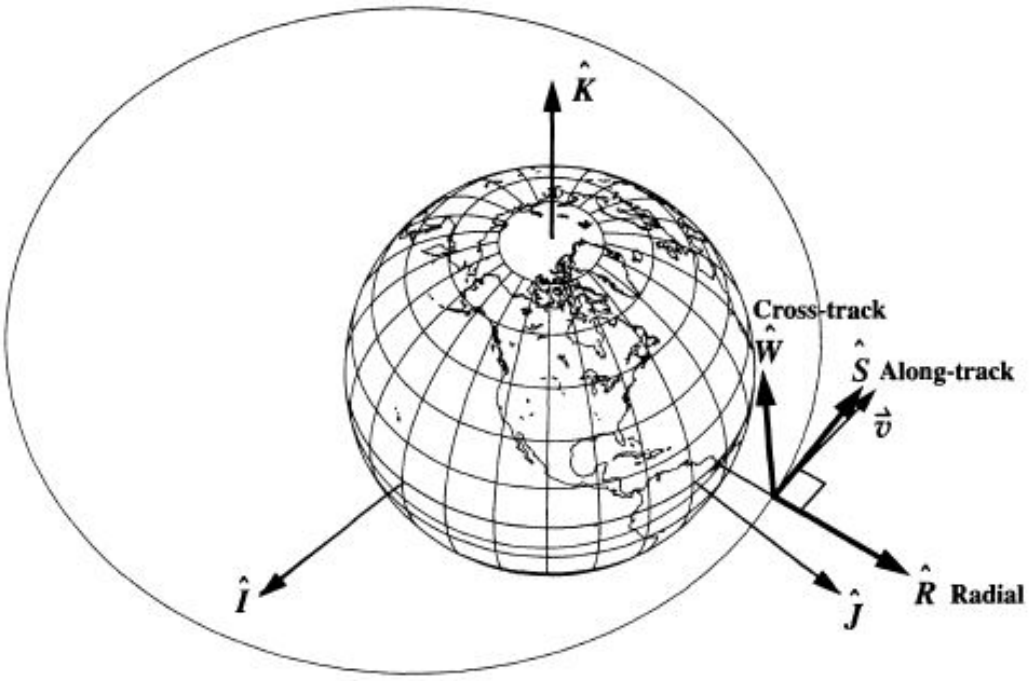

Figure B.1: Satellite coordinate system. Credits:[19]

APPENDIX C. CONVERSION FROM TEME TO LLA

First we compute geocentric latitude and longitude:

$$LAT_{geocentric} = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \tag{C.1}$$

$$LONG = \arctan\left(\frac{y}{x}\right) - \theta_{GMST} \tag{C.2}$$

Then we declare all the variables:

$$f = \frac{1}{298.26} \tag{C.3}$$

$$a = 6378135m \tag{C.4}$$

$$r = \sqrt{x^2+y^2+z^2} \tag{C.5}$$

$$e^2 = 2f - f^2 \tag{C.6}$$

$$e'^2 = \frac{e^2}{1-e^2} \tag{C.7}$$

$$b = a(1-f) \tag{C.8}$$

$$R = rcos(LAT_{geocentric}) \tag{C.9}$$

Now we can compute the geodetic latitude by iterating the geodetic latitude μ the following equations [24] and fixing a tolerance between μ_i and μ_{i-1} of 10^{-12}:

$$\mu_o = LAT_{geocentric} \tag{C.10}$$

$$\beta = \arctan\left(\frac{(1-f)sin(\mu)}{cos(\mu)}\right) \tag{C.11}$$

$$\mu = \arctan\left(\frac{z+be'^2sin(\beta)^3}{R-ae^2cos(\beta)^3}\right) \tag{C.12}$$

At last, we need to compute the altitude. To do it, we only have to solve this two equations [23]:

$$N = \frac{1}{\sqrt{1-e^2sin(\mu)^2}} \tag{C.13}$$

$$Altitude = \frac{R}{cos(\mu)} - aN \tag{C.14}$$

APPENDIX D. ARDUSATX RESULTS

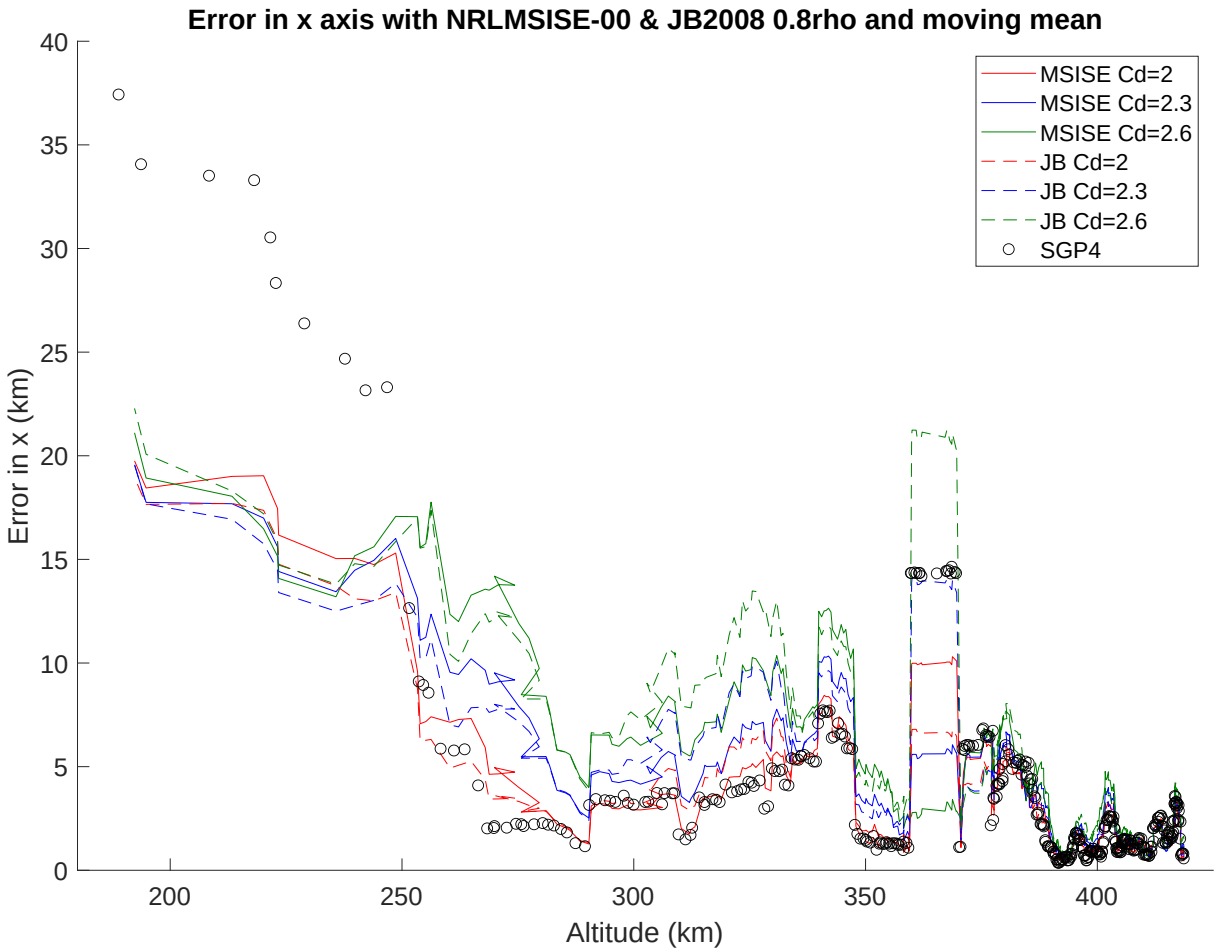

Figure D.1: Errors in the X axis versus altitude. All the data presented are for 0.8 times the density value

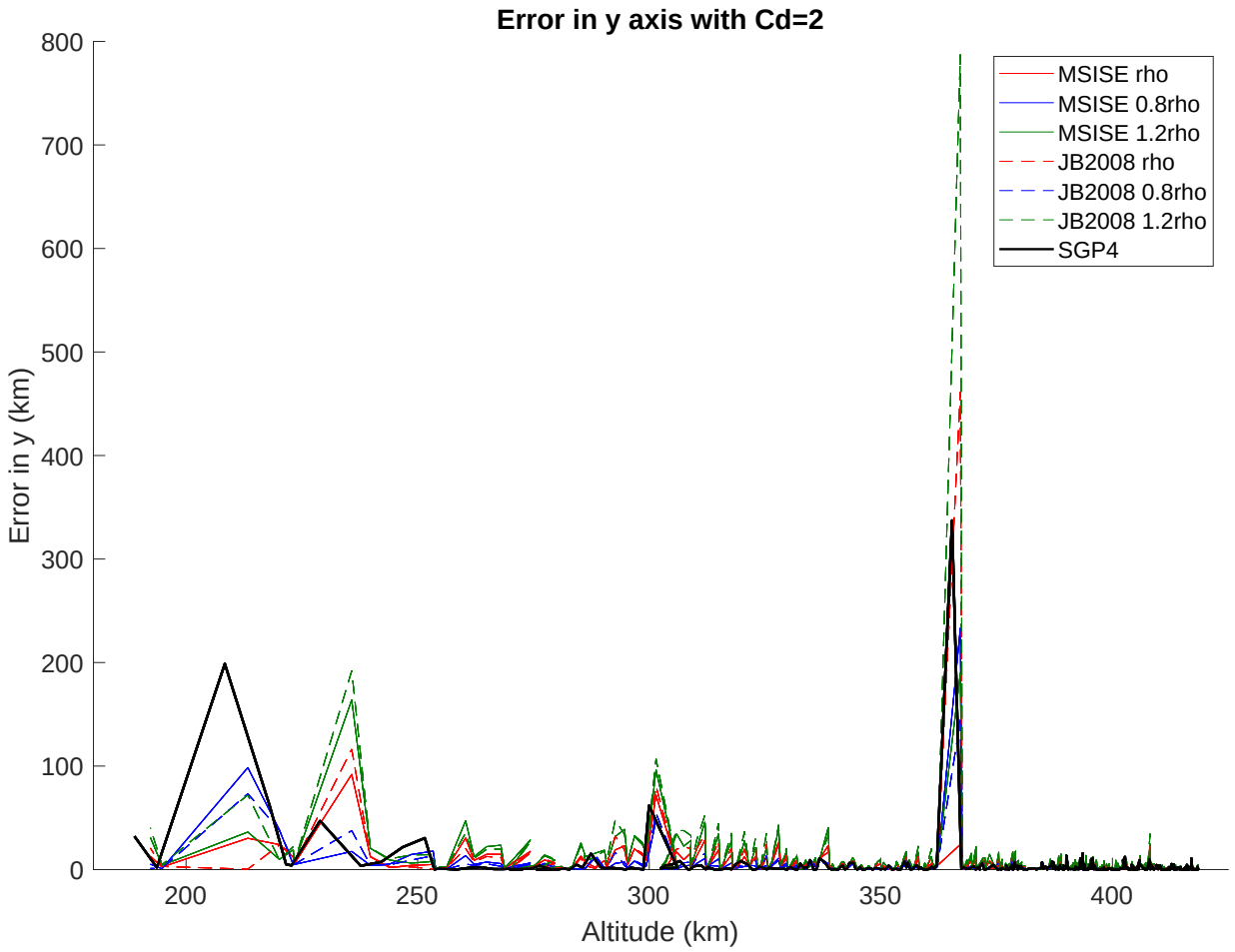

Figure D.2: Errors in the Y axis versus altitude. All the data presented are for C_D=2

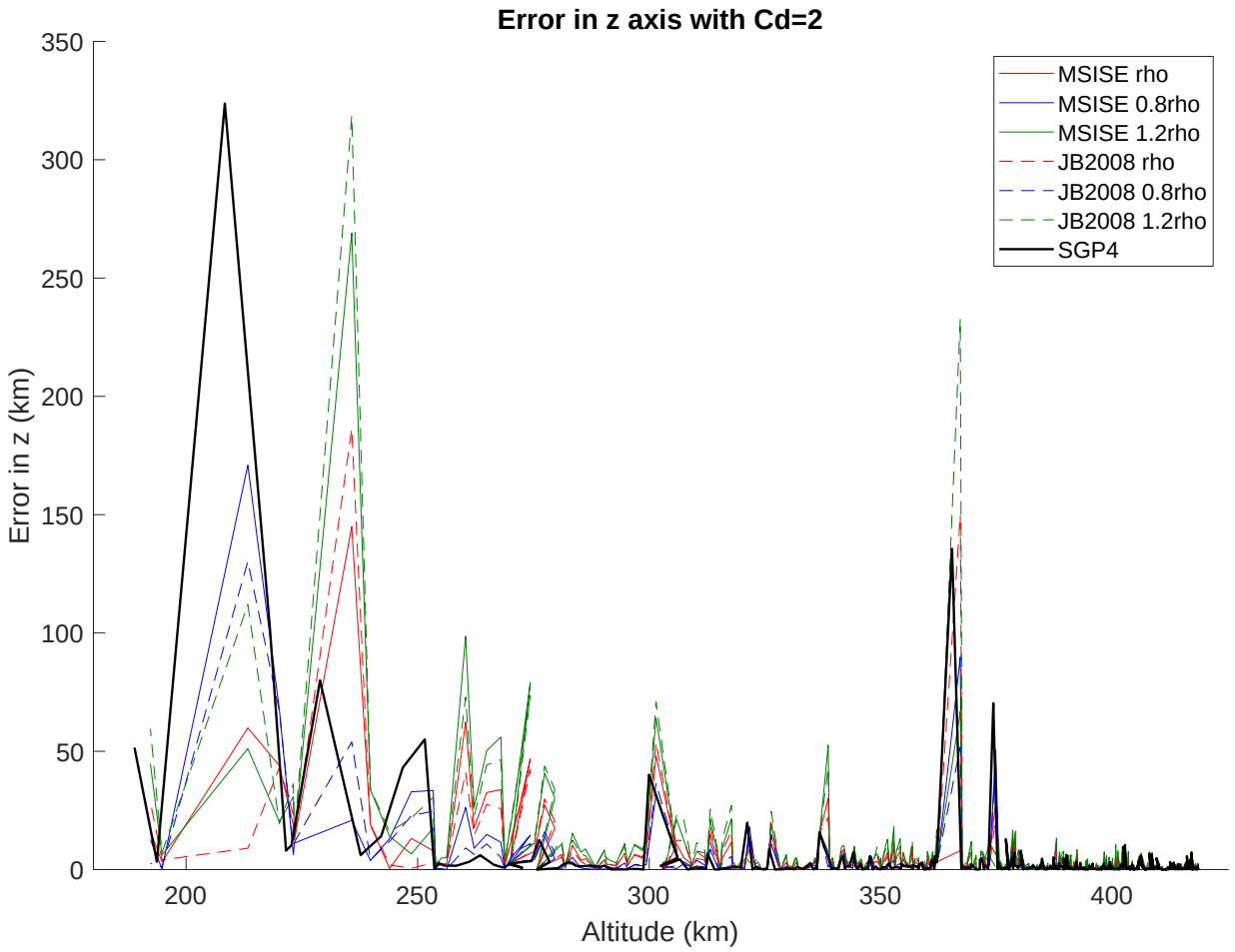

Figure D.3: Errors in the Z axis versus altitude. All the data presented are for C_D=2

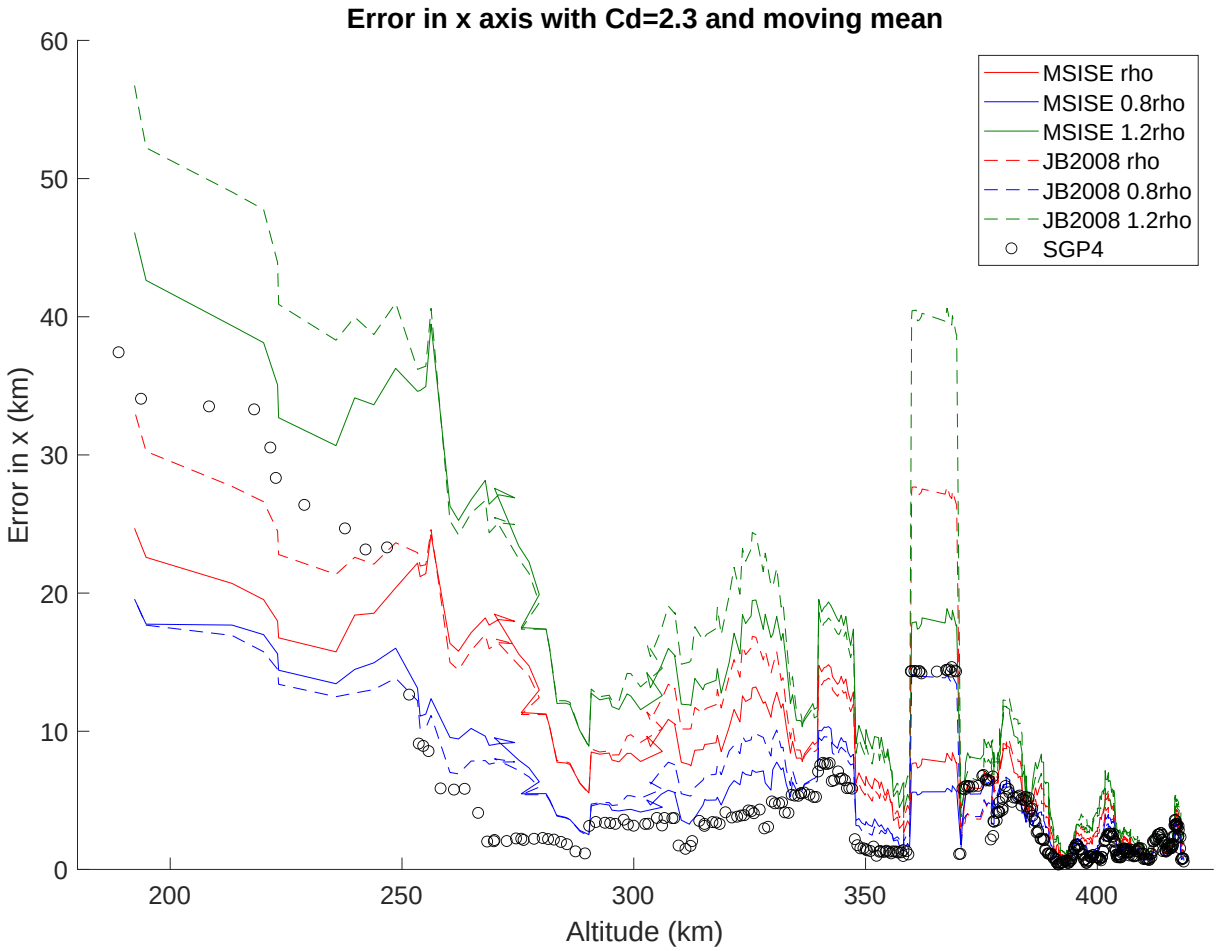

Figure D.4: Errors in the X axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

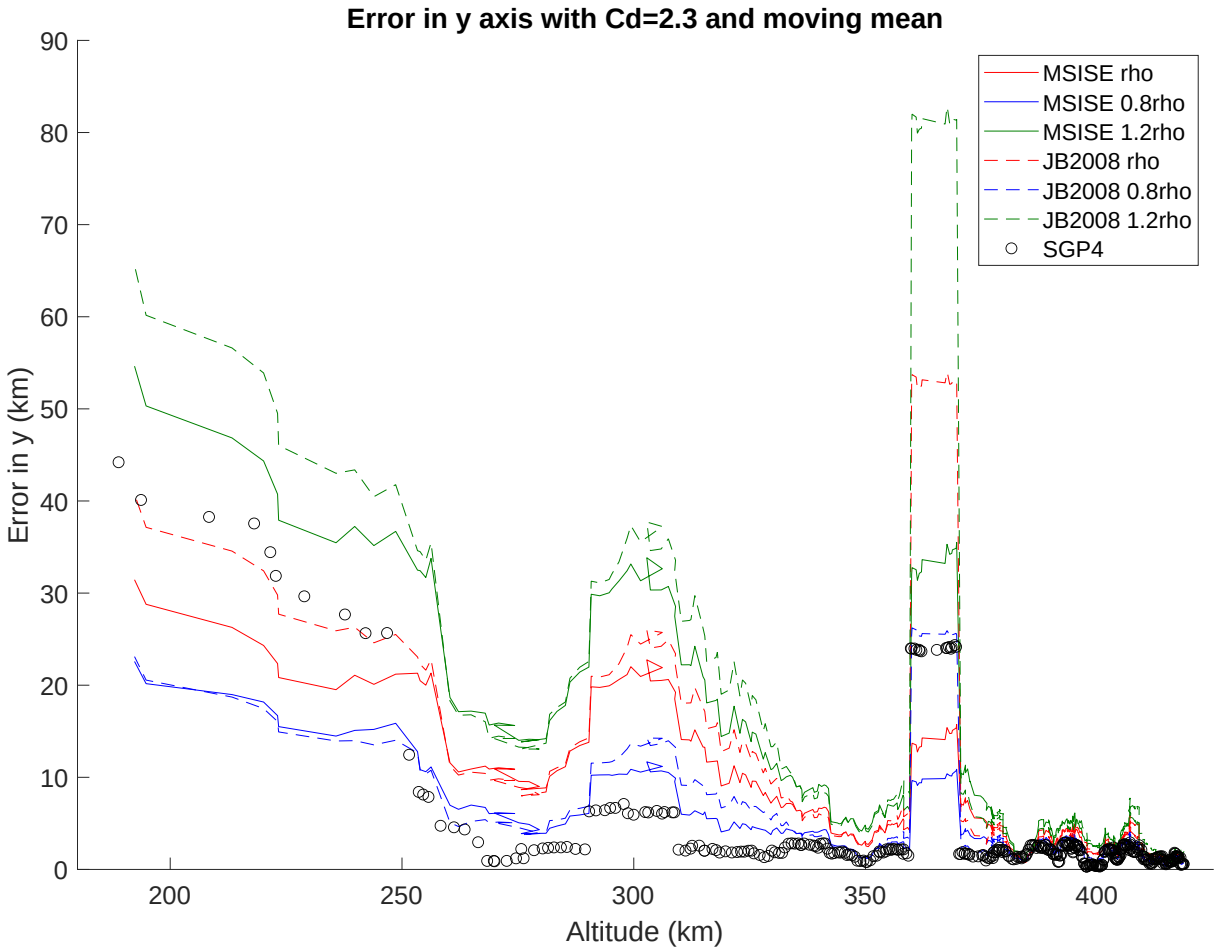

Figure D.5: Errors in the Y axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

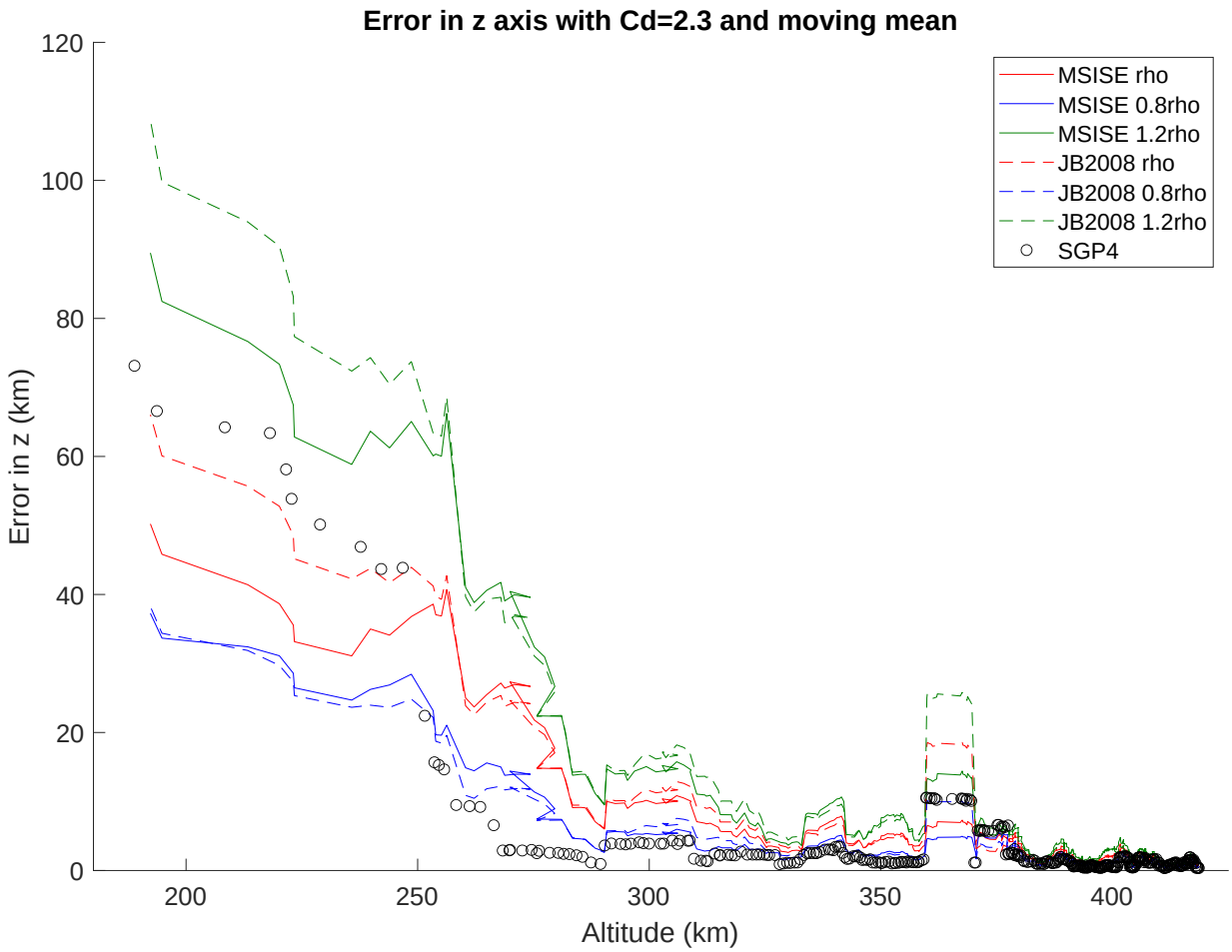

Figure D.6: Errors in the Z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

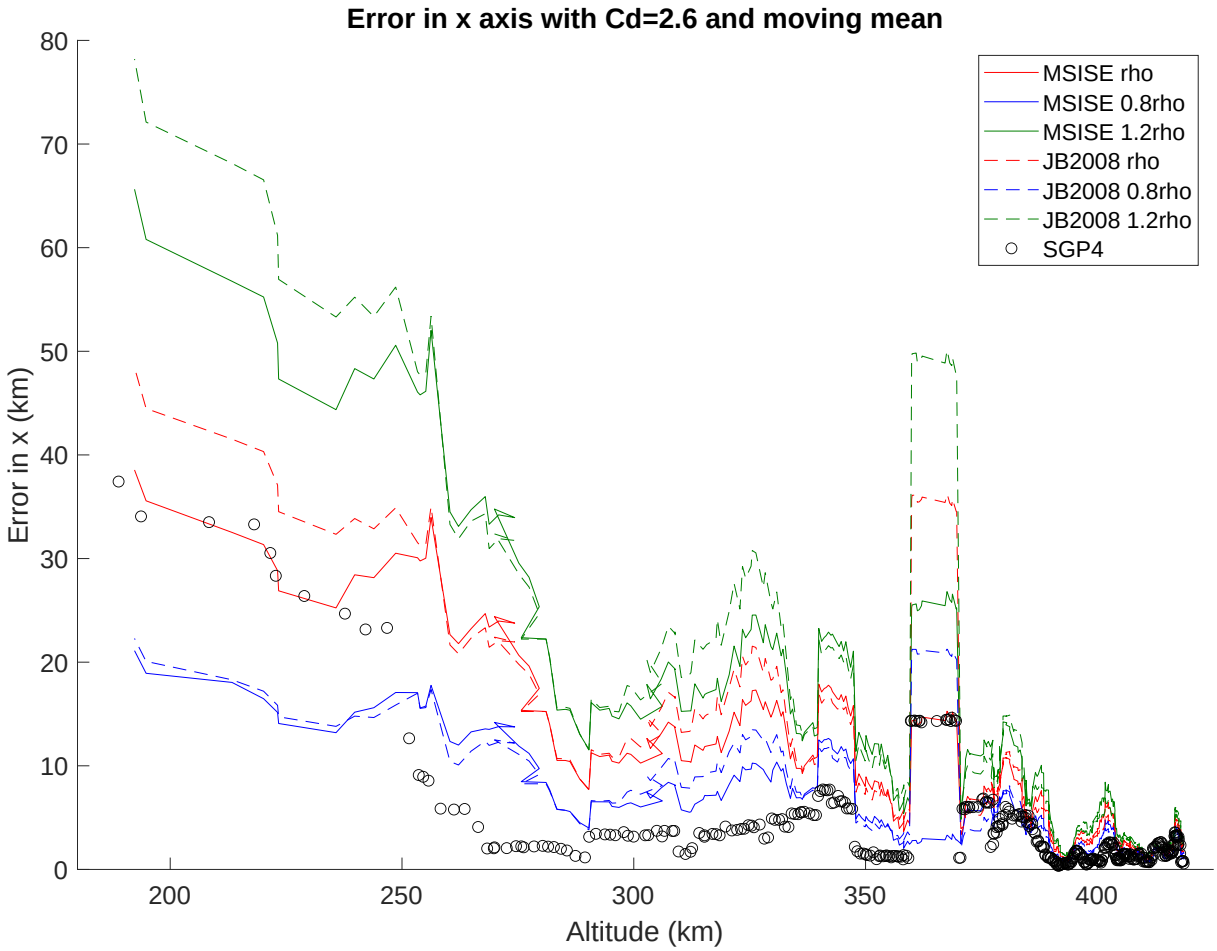

Figure D.7: Errors in the X axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

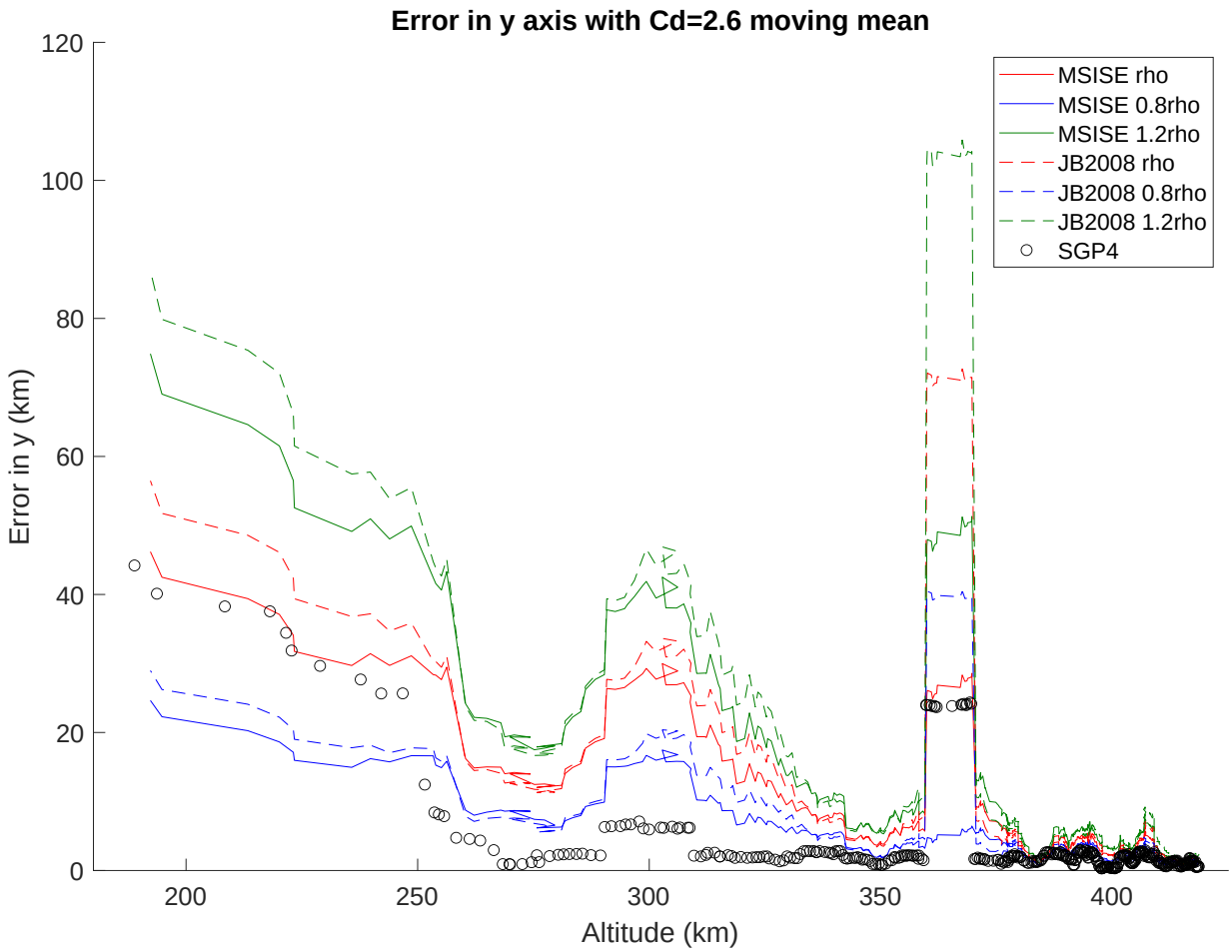

Figure D.8: Errors in the Y axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

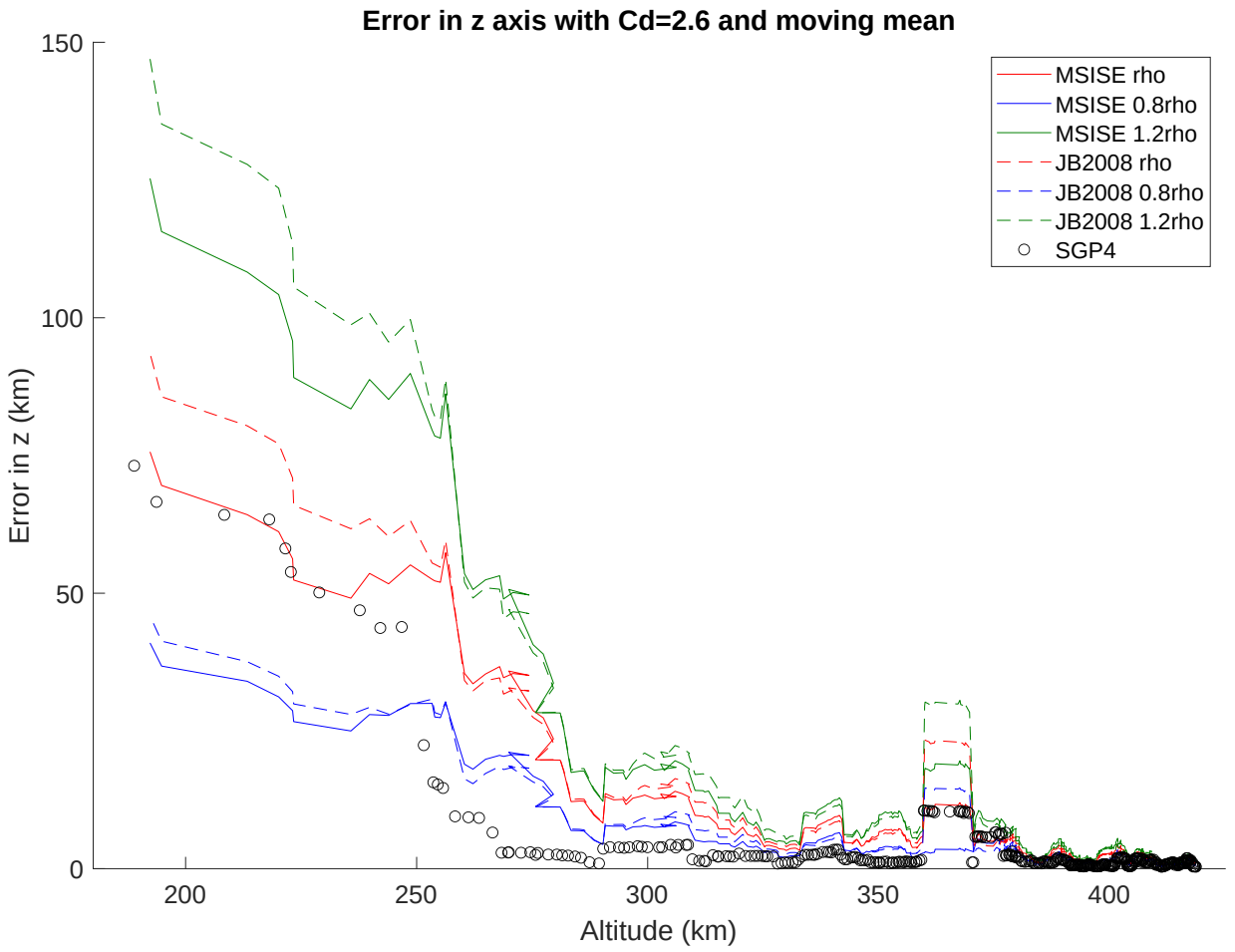

Figure D.9: Errors in the Z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

APPENDIX E. STARSHINE-2 RESULTS

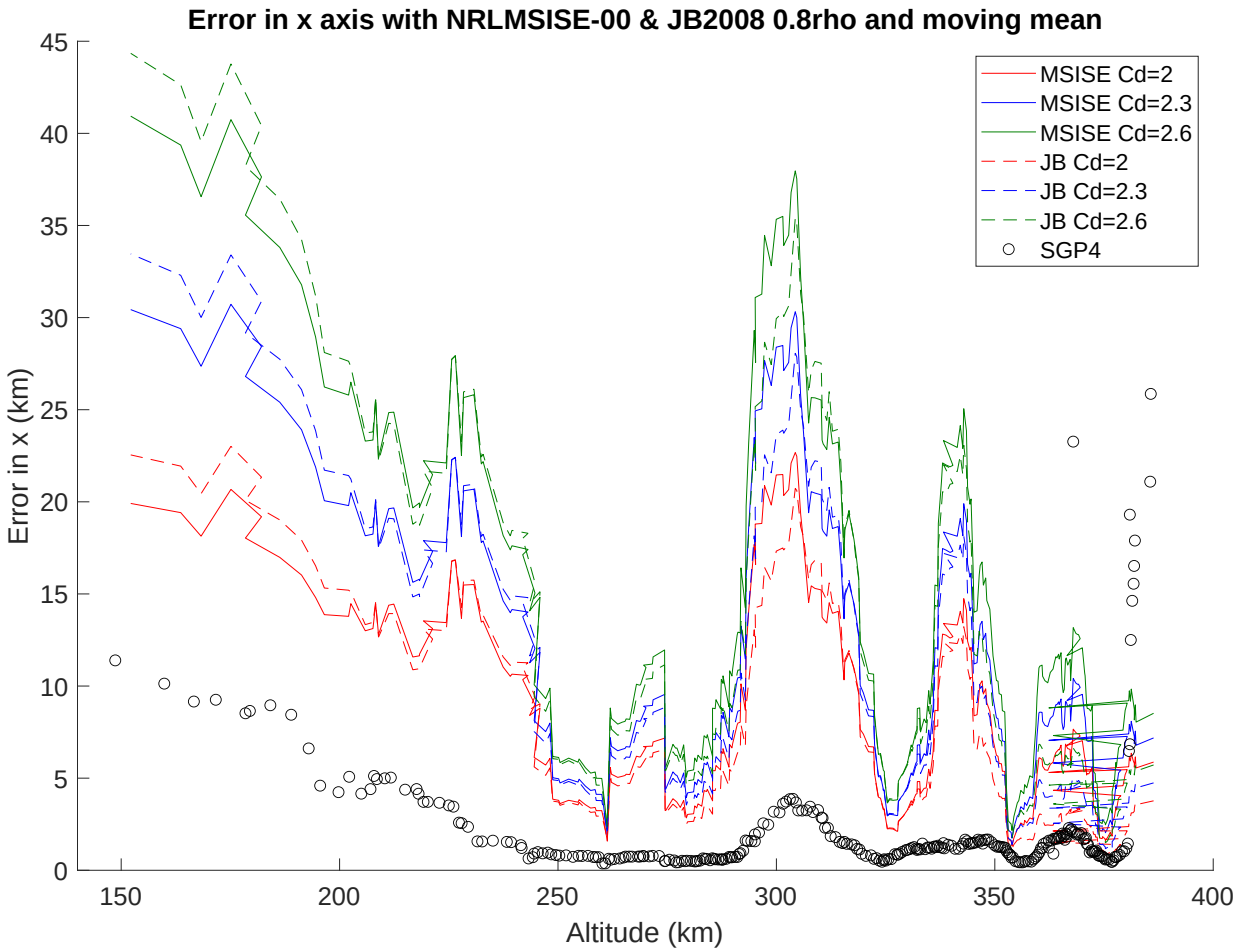

Figure E.1: Errors in the X axis versus altitude. All the data presented are for 0.8 times the density value

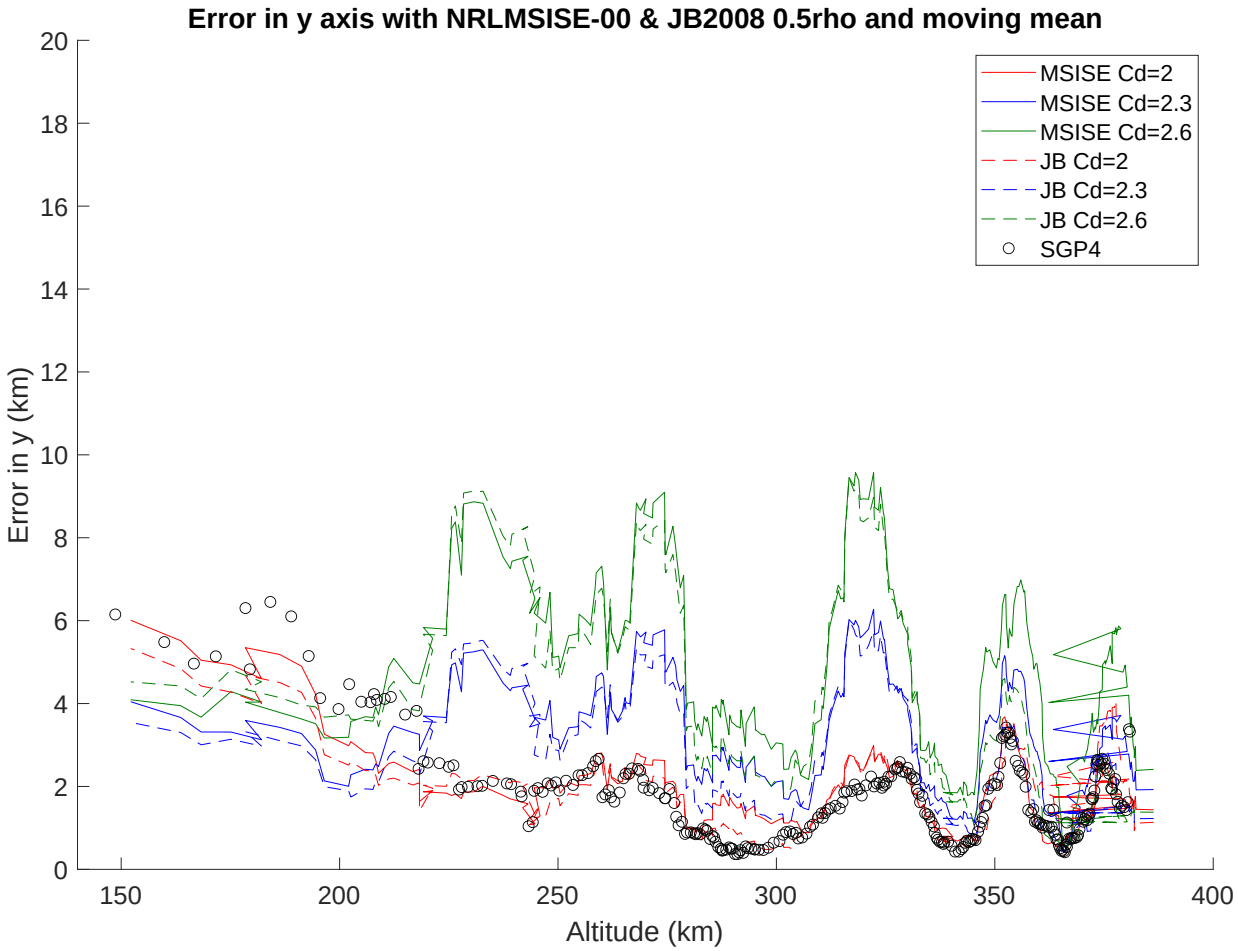

Figure E.2: Errors in the Y axis versus altitude. All the data presented are for 0.5 times the density value

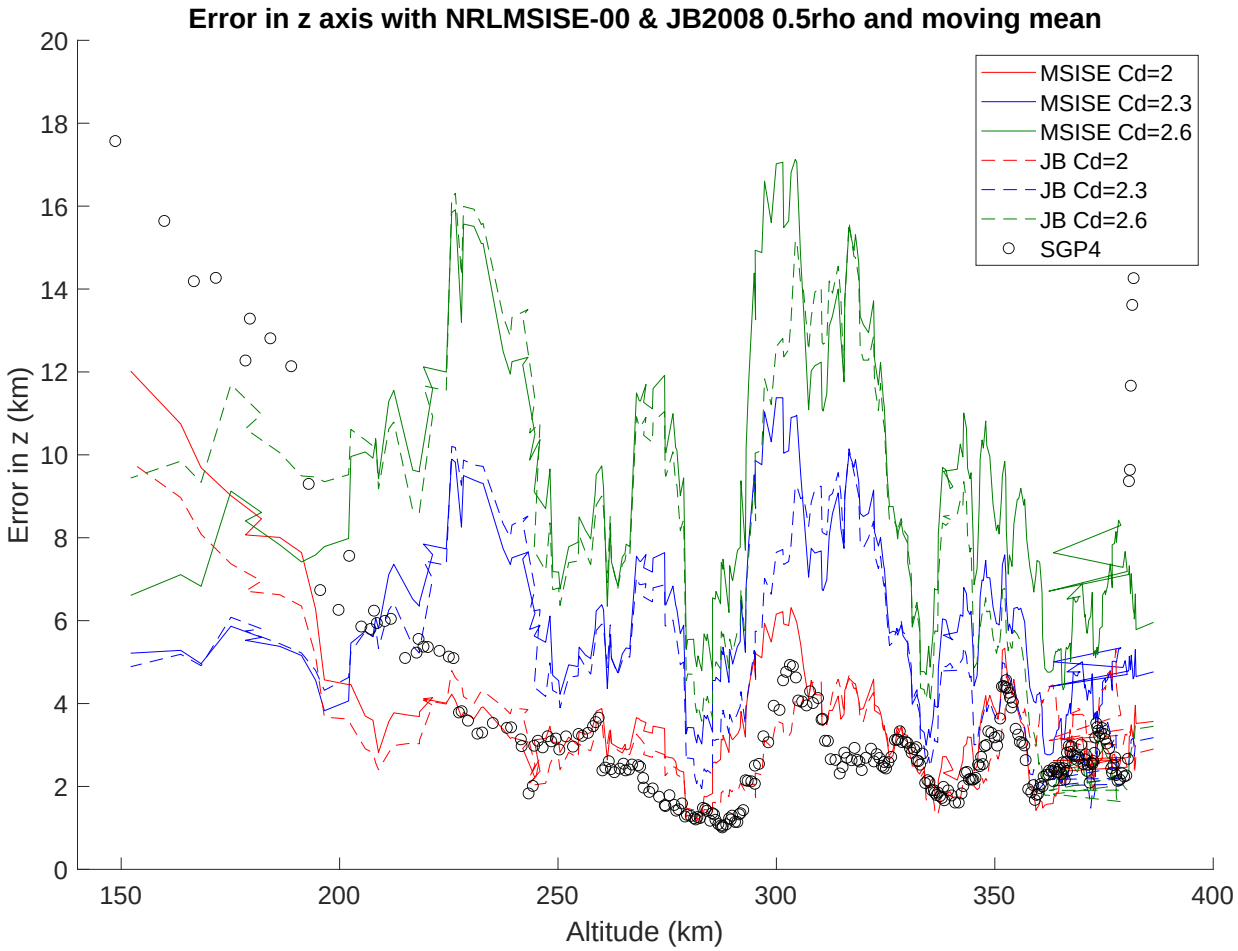

Figure E.3: Errors in the Z axis versus altitude. All the data presented are for 0.5 times the density value

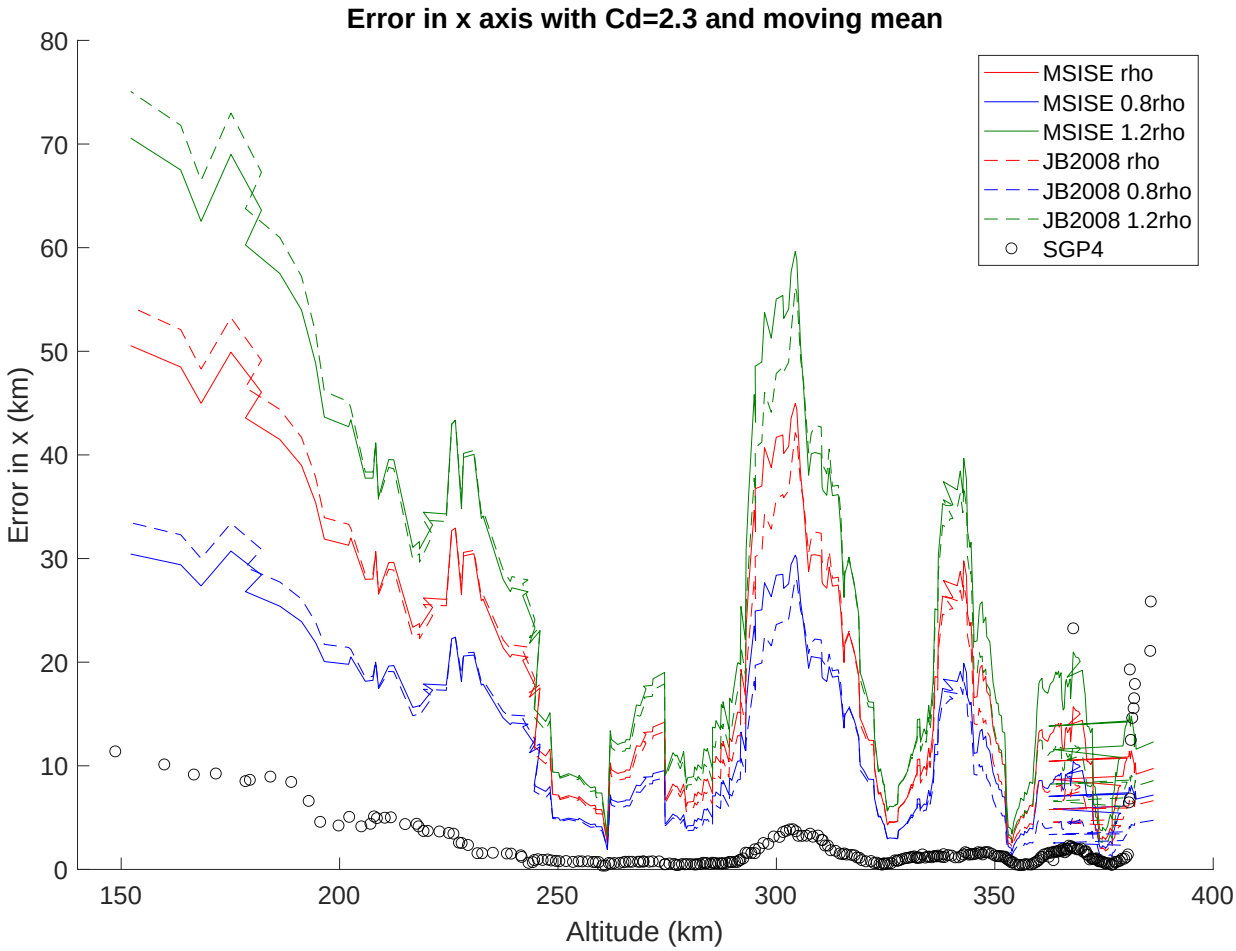

Figure E.4: Errors in the X axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

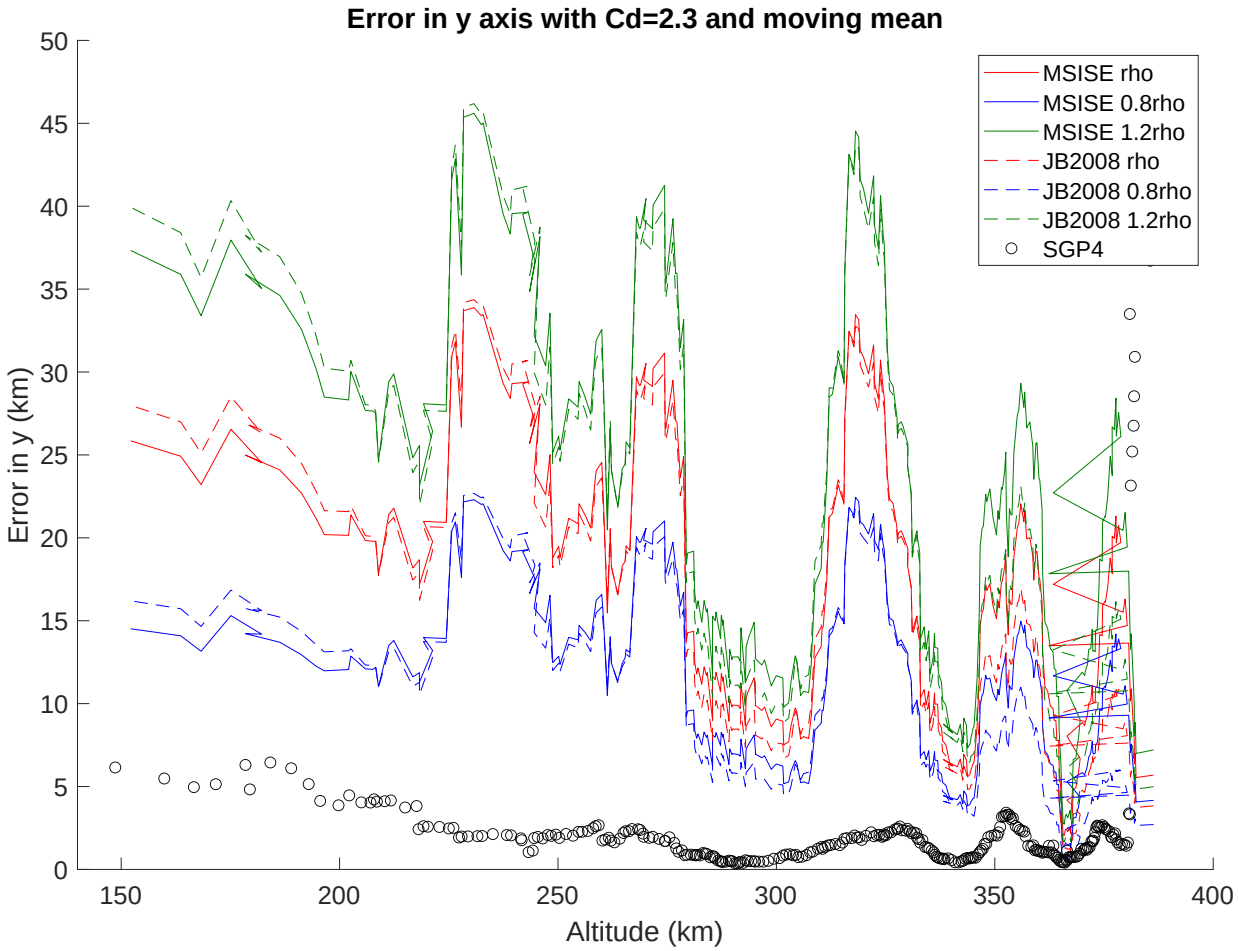

Figure E.5: Errors in the X axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

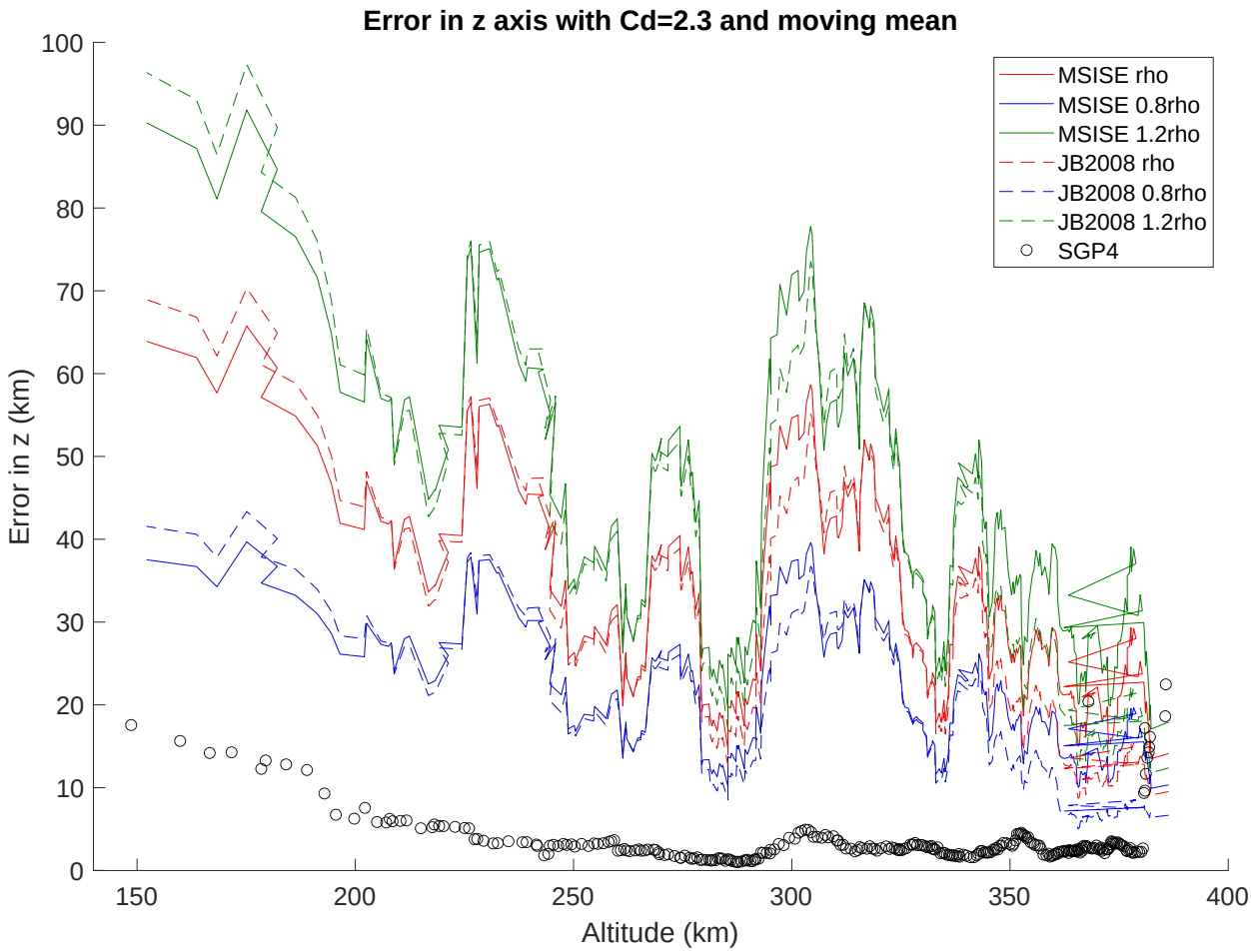

Figure E.6: Errors in the Z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.3

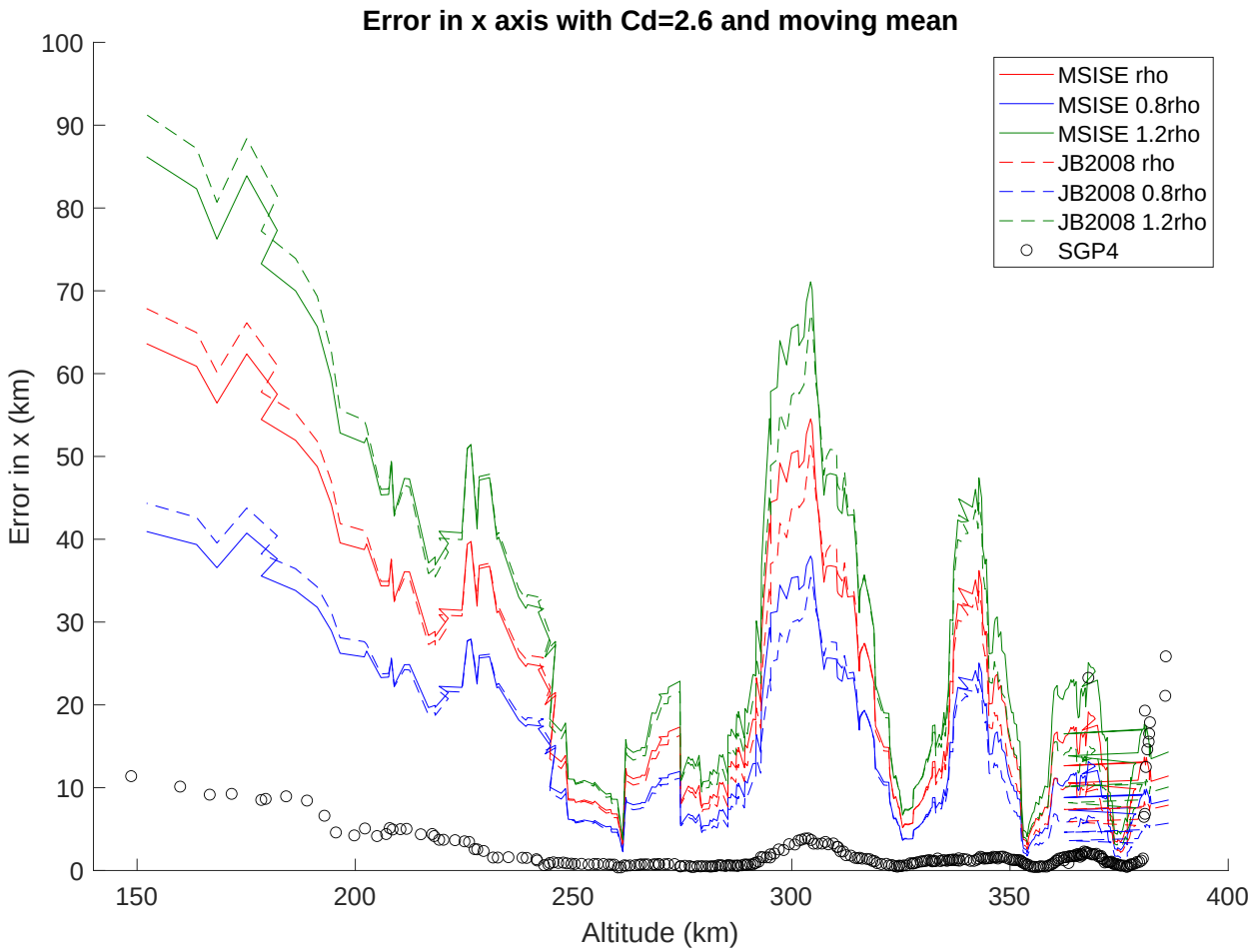

Figure E.7: Errors in the X axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

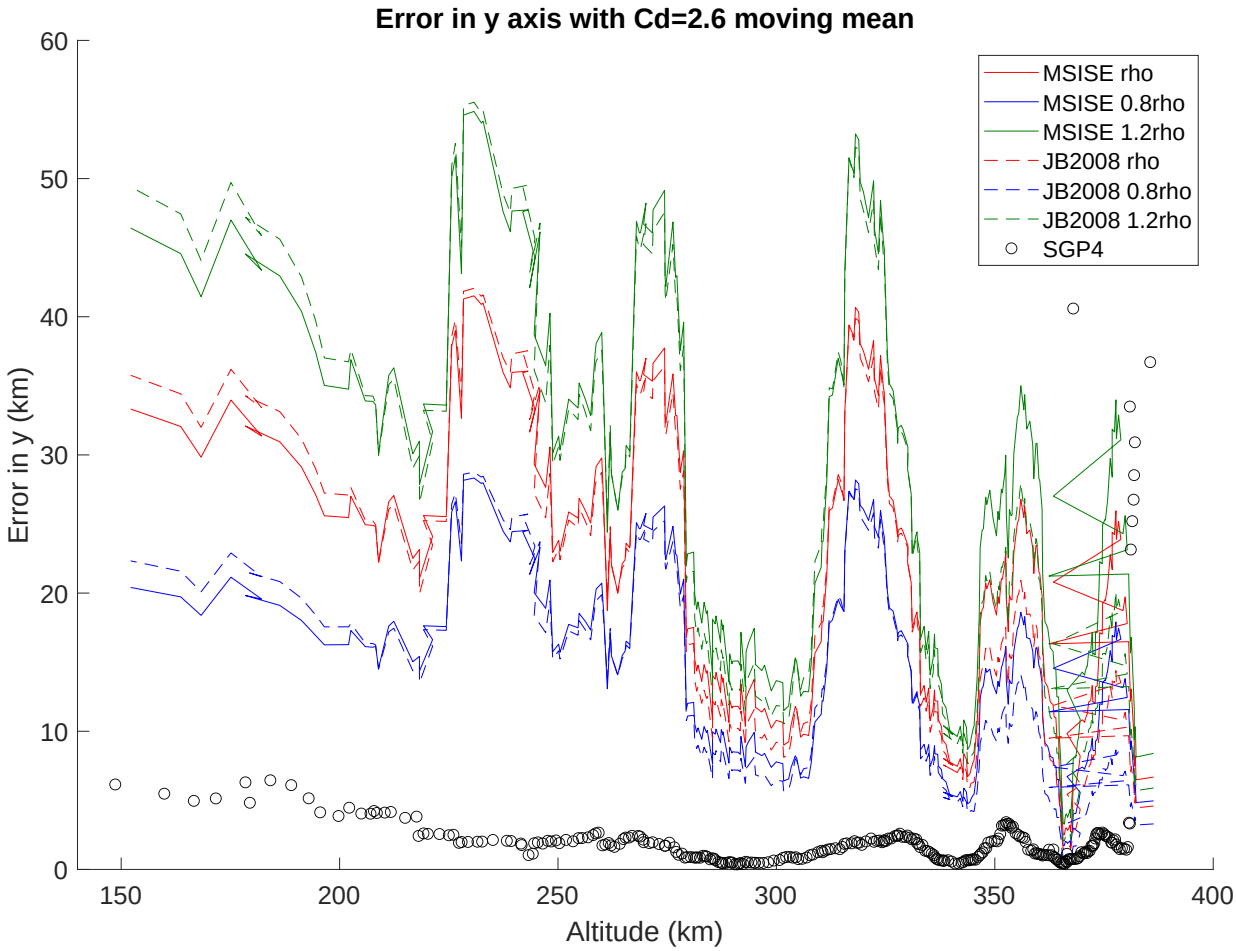

Figure E.8: Errors in the Y axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

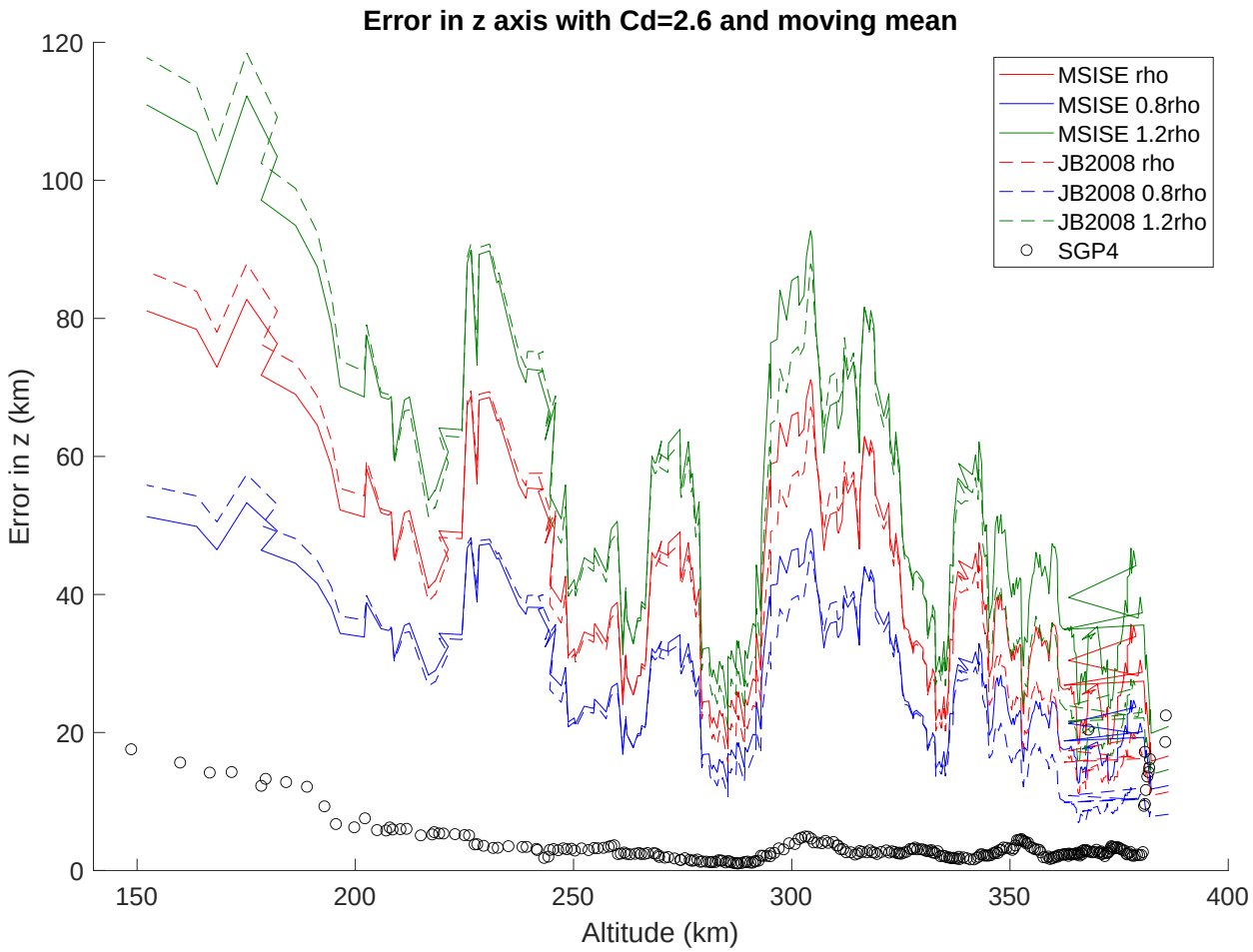

Figure E.9: Errors in the Z axis versus altitude using moving mean of 15 values. All the data presented are for C_D=2.6

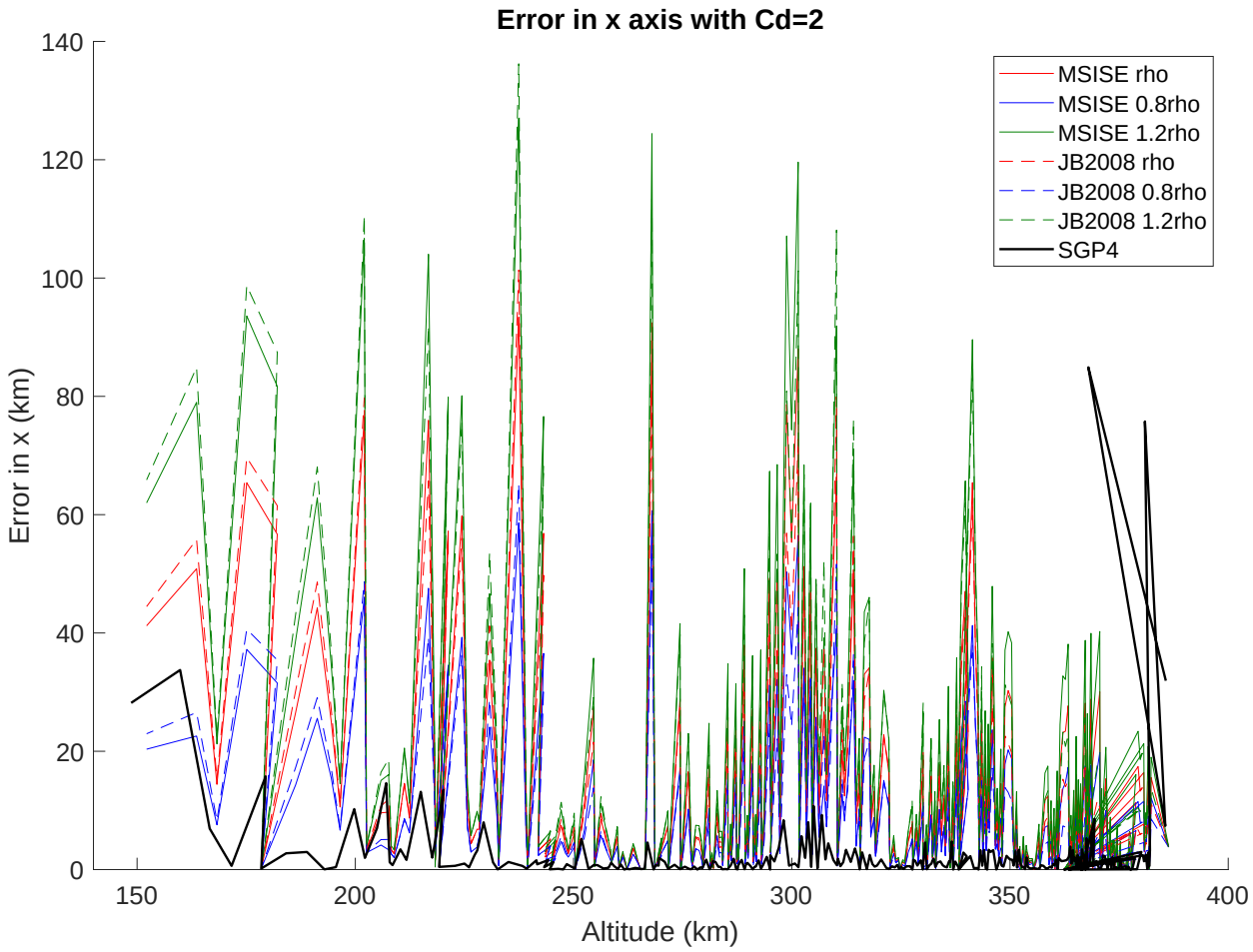

Figure E.10: Errors in the X axis versus altitude. All the data presented are for C_D=2

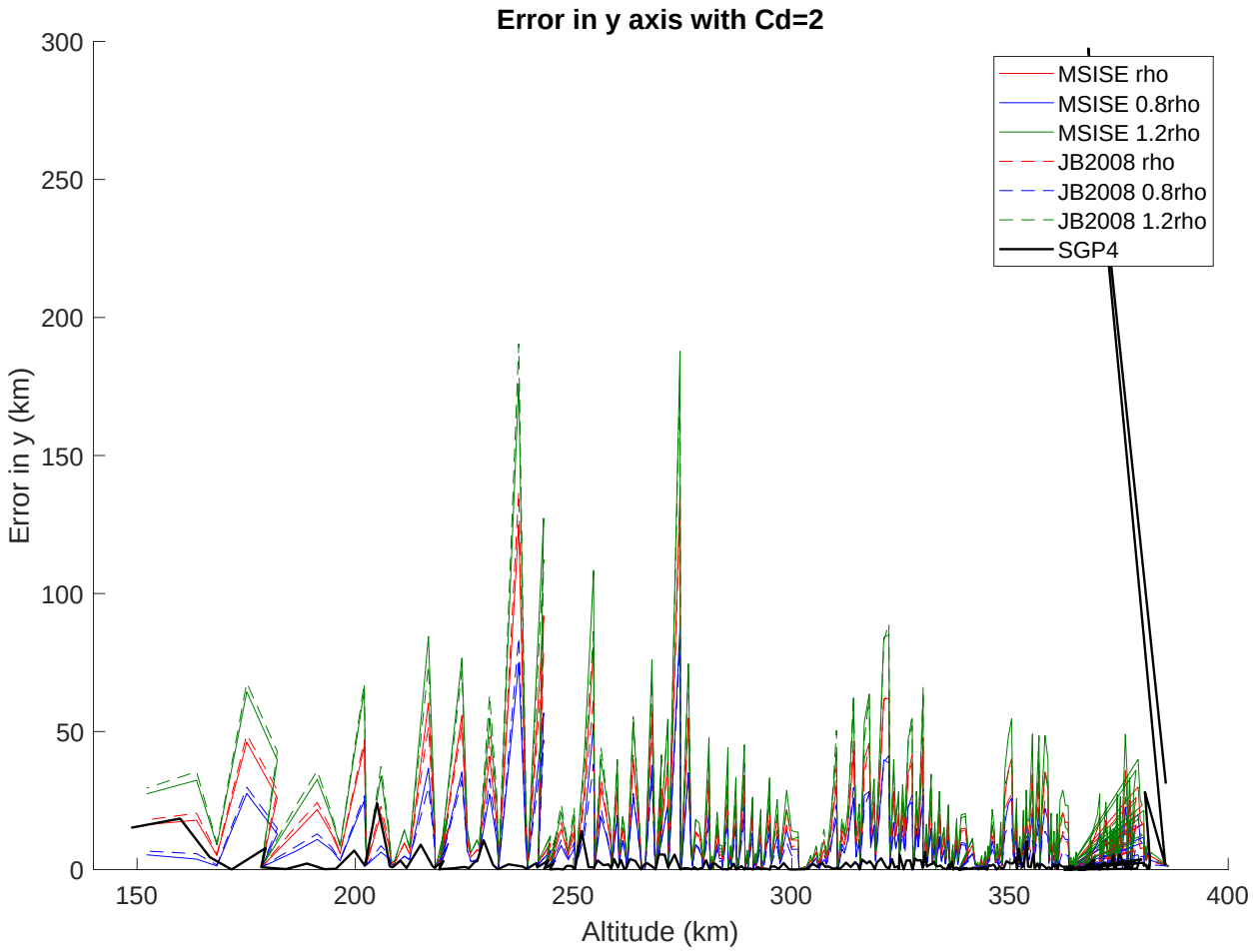

Figure E.11: Errors in the Y axis versus altitude. All the data presented are for C_D=2

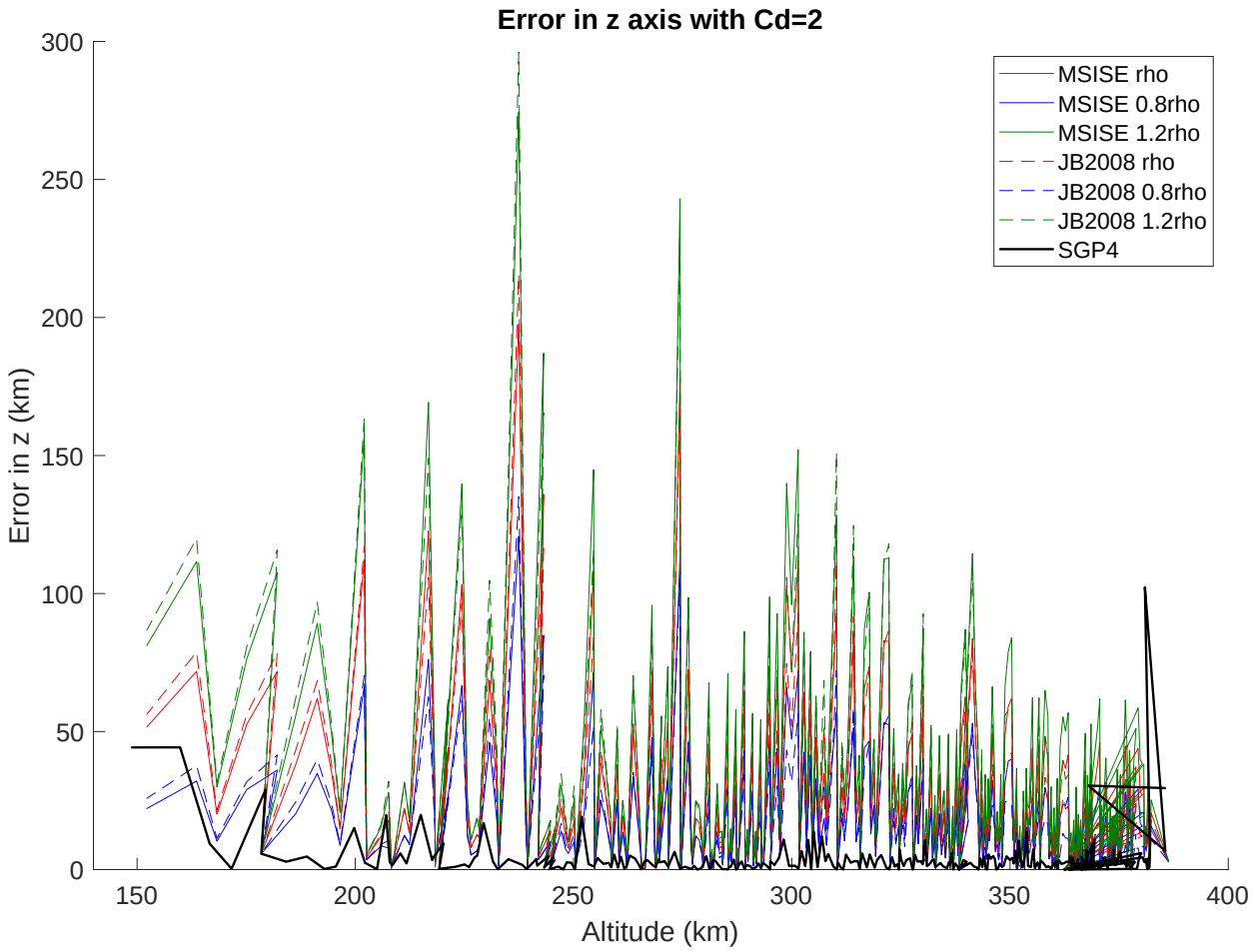

Figure E.12: Errors in the Z axis versus altitude. All the data presented are for C_D=2