



AEROSPACE ENGINEERING  
BACHELOR'S DEGREE THESIS  
2020 - 2021



---

Geocentric Orbit Calculation and State Vector  
Determination Introducing Perturbing  
Accelerations

---

*Author:*

María Alventosa Corts  
maalcor2@etsid.upv.es

*Supervisors:*

Santiago Emmanuel Moll López  
sanmollp@etsid.upv.es

Daniel Villalibre Vilariño  
daviivi@upv.edu.es

Valencia - July 15, 2021



## Acknowledgements

*To begin with, I would like to offer all the hours, effort invested and commitment to my family, who have accompanied me throughout this journey, helping me overcome the difficulties and hard times, encouraging me to keep going when it all seemed to fall apart, and celebrating with me each step achieved along the way. I could not be more grateful.*

*I would also like to thank all of my friends for always being by my side, for their encouragement and for backing me up in every decision I have made. For their trust in me and for having the ability to make me see things from another perspective. Without them, this journey would have certainly been much harder.*

*Last but not least, I am thankful for having had the chance to have Santi Moll López as my final thesis supervisor, for giving me tips and helping me solve any issues I struggled with throughout the process. Also, for having been such a great Maths teacher along the degree and for making his subjects easier to understand and enjoyable. Special thanks to Dani as well, for being willing to help me whenever I needed, particularly with software issues.*

*I would like to finally thank the department of Applied Mathematics for their course in 'Órbitas, Satélites y Relatividad', thanks to which I developed a deep interest in the space sector and found out that I would love to devote myself to it.*



### Abstract

The aim of this project is to solve the fundamental equation of relative motion between two bodies in space, introducing an increased level of accuracy, which is attained by considering the most relevant perturbing accelerations when performing the calculations. Making use of a mathematical robot, namely MATLAB<sup>™</sup>, it is possible to numerically integrate the differential equation by means of solver ode45. The result obtained is the state vector of the secondary body, which is found orbiting around the main one.

For the purpose of this paper, the case under study for the two-body problem is that of a satellite in orbit around the Earth, offering an approach to two different real cases. The analysis is carried out for both the ISS, an example of Low Earth Orbit, and for LAGEOS-1, which is found between LEO and Geostationary orbits. The perturbations which are to be implemented along this document comprise the gravitational accelerations due to the Earth's zonal harmonics, the drag forces caused by the atmosphere, or the gravity of several bodies in the solar system which influence the orbit determination as well. It is worth mentioning that relativistic effects are also accounted for in the calculations. As perturbations are introduced, the position and velocity errors with respect to the real state vectors are plotted as a function of time, so that the evolution and improvement of the results can be appreciated, and some relevant conclusions can be extracted.

Furthermore, a graphical, user-friendly interface is designed in order to dynamically select the satellite type as well as the perturbations to be added to the problem, and clearly visualize the orbit, its parameters and the state vector error. In addition, it allows to provide a  $\Delta v$  impulse so as to perform an orbital maneuver in order to make a change either in the apogee or perigee location, or in the orbit plane.

**Keywords:** orbit, satellite, two-body problem, relative motion,  $\Delta v$  impulse, state vector, perturbation, gravity, atmospheric drag, relativistic effects, solar radiation pressure

## Resumen

La finalidad principal del presente proyecto es resolver la ecuación fundamental del movimiento relativo de dos cuerpos en el espacio, introduciendo un alto nivel de precisión, que se alcanza tomando en consideración las aceleraciones perturbadoras más relevantes al efectuar los cálculos. Haciendo uso de un robot matemático, en concreto MATLAB<sup>™</sup>, es posible integrar numéricamente la ecuación diferencial mediante el *solver* ode45. El resultado obtenido es el vector de estado del cuerpo secundario, que se encuentra orbitando alrededor del cuerpo principal.

A efectos de este trabajo, el caso de estudio relacionado con el problema de los dos cuerpos es el de un satélite en órbita alrededor de la Tierra, ofreciendo aproximaciones a dos casos reales. El análisis se lleva a cabo tanto para la ISS, un ejemplo de órbita LEO, como para el satélite LAGEOS-1, que se encuentra entre las órbitas LEO y Geoestacionarias. Las perturbaciones que van a ser implementadas a lo largo de este documento comprenden las aceleraciones gravitatorias debidas a los armónicos zonales, las fuerzas de rozamiento causadas por la atmósfera, o la gravedad de algunos cuerpos del sistema solar que influyen a la hora de determinar la órbita. Cabe mencionar que los efectos relativistas también se incluyen en los cálculos a modo de perturbaciones. A medida que éstas van siendo introducidas, se representan los errores de posición y velocidad con respecto a los vectores de estado reales en función del tiempo, de forma que se puede apreciar la evolución y mejora de los resultados con objeto de extraer conclusiones relevantes.

Por otra parte, se propone el diseño de una interfaz gráfica intuitiva que permita al usuario seleccionar de forma dinámica el tipo de satélite y el número de perturbaciones que se desee añadir al problema, para posteriormente poder visualizar la órbita calculada, así como sus parámetros orbitales y los gráficos del error en función del tiempo. Además, es posible introducir un impulso  $\Delta v$  al satélite con el fin de efectuar una maniobra orbital de cambio del apogeo o perigeo, o de cambio de plano de la órbita.

**Palabras clave:** órbita, satélite, problema de los dos cuerpos, movimiento relativo, impulso  $\Delta v$ , vector de estado, perturbación, gravedad, *drag* atmosférico, efectos relativistas, presión por radiación solar

## Resum

La principal finalitat del present projecte és resoldre l'equació fonamental del moviment relatiu de dos cossos en l'espai, introduint un alt nivell de precisió, que es pot aconseguir si es consideren les acceleracions pertorbadores més rellevants a l'hora d'efectuar els càlculs. Fent ús d'un robot matemàtic, en concret MATLAB<sup>™</sup>, és possible integrar numèricament l'equació diferencial mitjançant el *solver* ode45. El resultat obtingut és el vector d'estat del cos secundari, que es troba en òrbita al voltant del cos principal.

A efectes d'aquest treball, el cas d'estudi relacionat amb el problema dels dos cossos és el d'un satèl·lit que orbita al voltant de la Terra, oferint aproximacions a dos casos reals. L'anàlisi es duu a terme tant per a la ISS, un exemple d'òrbita LEO, com per al satèl·lit LAGEOS-1, que es troba entre les òrbites LEO i Geoestacionàries. Les pertorbacions que seran implementades al llarg d'aquest document comprenen les acceleracions gravitatòries degudes als harmònics zonals, les forces de fricció causades per l'atmosfera, o la gravetat d'alguns cossos del sistema solar que influeixen a l'hora de determinar l'òrbita. Cal esmentar que els efectes relativistes també s'inclouen en els càlculs a mode de pertorbacions. A mesura que aquestes van sent introduïdes, es representen els errors de posició i velocitat respecte als vectors d'estat reals en funció del temps, de manera que es pot apreciar l'evolució i millora dels resultats per tal d'extraure conclusions rellevants.

D'altra banda, es proposa el disseny d'una interfície gràfica intuïtiva que permeti a l'usuari seleccionar de manera dinàmica el tipus de satèl·lit i el nombre de pertorbacions que es desitja afegir al problema, per a posteriorment poder visualitzar l'òrbita calculada, així com els seus paràmetres orbitals i els gràfics de l'error en funció del temps. A més, és possible introduir un impuls  $\Delta v$  al satèl·lit amb la finalitat d'efectuar una maniobra orbital de canvi de l'apogeu o perigeu, o de canvi de pla de l'òrbita.

**Paraules clau:** òrbita, satèl·lit, problema dels dos cossos, moviment relatiu, impuls  $\Delta v$ , vector d'estat, pertorbació, gravetat, *drag* atmosfèric, efectes relativistes, pressió per radiació solar

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>The Two-Body Problem</b>	<b>2</b>
2.1	Theoretical Background . . . . .	2
2.1.1	Inertial Reference Frame . . . . .	2
2.1.2	Body-Centered Reference Frame . . . . .	5
2.2	Approach to real cases . . . . .	6
2.2.1	LAGEOS-1 orbit . . . . .	6
2.2.2	International Space Station orbit . . . . .	11
<b>3</b>	<b>Introduction to Orbital Perturbations</b>	<b>14</b>
3.1	Theoretical Background . . . . .	15
3.1.1	Gravitational Perturbations . . . . .	15
3.1.2	Atmospheric Drag Perturbation . . . . .	20
3.1.3	Relativistic Effects . . . . .	21
3.1.4	Lunar Gravity Perturbation . . . . .	22
3.1.5	Solar Gravity Perturbation . . . . .	24
3.1.6	Perturbation due to Solar System Planets . . . . .	25
3.1.7	Solar Radiation Pressure . . . . .	30
3.2	Approach to real cases . . . . .	31
3.2.1	LAGEOS-1 orbit . . . . .	32
3.2.2	International Space Station orbit . . . . .	46
<b>4</b>	<b>Orbital Parameters</b>	<b>60</b>
4.1	LAGEOS-1 . . . . .	62
4.2	International Space Station . . . . .	64
<b>5</b>	<b>Graphical Interface: Orbit Determination</b>	<b>66</b>
<b>6</b>	<b>Conclusions</b>	<b>70</b>
<b>7</b>	<b>Future Work</b>	<b>72</b>
<b>8</b>	<b>Technical Specifications</b>	<b>73</b>
8.1	Standard Regulations . . . . .	73
8.2	Work Conditions . . . . .	76
<b>9</b>	<b>Project Budget</b>	<b>77</b>
9.1	Personnel Costs . . . . .	77
9.2	Hardware Costs . . . . .	78
9.3	Software Costs . . . . .	78
9.4	Total Budget . . . . .	78
	<b>References</b>	<b>81</b>
	<b>Appendices</b>	<b>I</b>
<b>A</b>	<b>MATLAB code</b>	<b>I</b>
A.1	General code . . . . .	I
A.2	Interpolating polynomials . . . . .	IV
A.3	Code for LAGEOS-1 . . . . .	XVI
A.4	Code for ISS . . . . .	XXIV



## List of Figures

1	Motion of $m_1$ , $m_2$ and the center of mass, $G$ , from an inertial reference frame . . . . .	3
2	Motion of $m_2$ and the center of mass, $G$ , as seen from $m_1$ . . . . .	4
3	Motion of $m_1$ and $m_2$ as seen from the center of mass, $G$ . . . . .	4
4	LAGEOS-1 satellite with corner-cube reflectors [1] . . . . .	6
5	Horizons interface configuration [2] . . . . .	8
6	LAGEOS-1 trajectory during May 2021 without introducing perturbations . . . . .	9
7	LAGEOS-1 absolute position error during May 2021 without introducing perturbations . . . . .	10
8	LAGEOS-1 absolute velocity error during May 2021 without introducing perturbations . . . . .	10
9	View of several superimposed ISS orbits [3] . . . . .	11
10	ISS trajectory during May 2021 without introducing perturbations . . . . .	12
11	ISS absolute position error during May 2021 without introducing perturbations . . . . .	13
12	ISS absolute velocity error during May 2021 without introducing perturbations . . . . .	13
13	Importance of perturbing accelerations as a function of altitude [4] . . . . .	14
14	Spherical reference frame [5] . . . . .	15
15	Spherical harmonics [6] . . . . .	16
16	Legendre polynomials . . . . .	18
17	Evolution of density with altitude [5] . . . . .	20
18	Earth-centered reference frame including lunar gravity perturbation [5] . . . . .	22
19	Plot of the resulting interpolating polynomials for the position of the Moon . . . . .	23
20	Earth-centered reference frame including solar gravity perturbation [5] . . . . .	24
21	Plot of the resulting interpolating polynomials for the position of the Sun . . . . .	25
22	Plot of the resulting interpolating polynomials for the position of Mercury . . . . .	26
23	Plot of the resulting interpolating polynomials for the position of Venus . . . . .	26
24	Plot of the resulting interpolating polynomials for the position of Mars . . . . .	27
25	Plot of the resulting interpolating polynomials for the position of Jupiter . . . . .	27
26	Plot of the resulting interpolating polynomials for the position of Saturn . . . . .	28
27	Plot of the resulting interpolating polynomials for the position of Uranus . . . . .	28
28	Plot of the resulting interpolating polynomials for the position of Neptune . . . . .	29
29	Plot of the resulting interpolating polynomials for the position of Pluto . . . . .	29
30	Scheme of the situation for obtaining the shadow function [5] . . . . .	31
31	LAGEOS-1 trajectory during May 2021 introducing gravitational perturbations . . . . .	32
32	LAGEOS-1 absolute position error during May 2021 introducing gravitational perturbations . . . . .	33
33	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational perturbations . . . . .	33
34	LAGEOS-1 trajectory during May 2021 introducing gravitational and atmospheric drag perturbations . . . . .	34
35	LAGEOS-1 absolute position error during May 2021 introducing gravitational and atmospheric drag perturbations . . . . .	35
36	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational and atmospheric drag perturbations . . . . .	35
37	LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	36
38	LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	37
39	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	37
40	LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	38
41	LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	39
42	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	39
43	LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	40

44	LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	41
45	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	41
46	LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations . . . . .	42
47	LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations	43
48	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations	43
49	LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations .	44
50	LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations . . . . .	45
51	LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations . . . . .	45
52	ISS trajectory during May 2021 introducing gravitational perturbations . . . . .	46
53	ISS absolute position error during May 2021 introducing gravitational perturbations . . .	47
54	ISS absolute velocity error during May 2021 introducing gravitational perturbations . . .	47
55	ISS trajectory during May 2021 introducing gravitational and atmospheric drag perturbations	48
56	ISS absolute position error during May 2021 introducing gravitational and atmospheric drag perturbations . . . . .	49
57	ISS absolute velocity error during May 2021 introducing gravitational and atmospheric drag perturbations . . . . .	49
58	ISS trajectory during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	50
59	ISS absolute position error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	51
60	ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations . . . . .	51
61	ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	52
62	ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	53
63	ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations . . . . .	53
64	ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	54
65	ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	55
66	ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations . . . . .	55
67	ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations . . . . .	56
68	ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations . .	57
69	ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations . .	57
70	ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations . . . . .	58
71	ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations	59

72	ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations	59
73	Orbit Determination: Orbit tab . . . . .	67
74	Orbit Determination: Position Error tab . . . . .	67
75	Orbit Determination: Velocity Error tab . . . . .	68
76	Orbit Determination: Orbital Parameters I tab . . . . .	68
77	Orbit Determination: Orbital Parameters II tab . . . . .	69
78	Orbit Determination: Orbital Parameters III tab . . . . .	69
79	Gantt Chart of the final degree project . . . . .	77

## List of Tables

1	Evolution of LAGEOS-1 orbital parameters by adding perturbations I . . . . .	62
2	Evolution of LAGEOS-1 orbital parameters by adding perturbations II . . . . .	63
3	Evolution of ISS orbital parameters by adding perturbations I . . . . .	64
4	Evolution of ISS orbital parameters by adding perturbations II . . . . .	65
5	Breakdown of graduate engineer expenses . . . . .	77
6	Breakdown of supervisor expenses . . . . .	78
7	Breakdown of hardware costs . . . . .	78
8	Breakdown of software costs . . . . .	78
9	Total costs including taxes . . . . .	79

## Listings

1	Atmosphere function for drag perturbation [5] . . . . .	I
2	Function to draw an earth-sized sphere [7] . . . . .	I
3	Script to obtain the interpolating polynomial of the position of the Moon as seen from the Earth . . . . .	IV
4	Script to obtain the interpolating polynomial of the position of the Sun as seen from the Earth . . . . .	V
5	Script to obtain the interpolating polynomial of the position of Mercury as seen from the Earth . . . . .	VI
6	Script to obtain the interpolating polynomial of the position of Venus as seen from the Earth	VII
7	Script to obtain the interpolating polynomial of the position of Mars as seen from the Earth	IX
8	Script to obtain the interpolating polynomial of the position of Jupiter as seen from the Earth . . . . .	X
9	Script to obtain the interpolating polynomial of the position of Saturn as seen from the Earth . . . . .	XI
10	Script to obtain the interpolating polynomial of the position of Uranus as seen from the Earth . . . . .	XII
11	Script to obtain the interpolating polynomial of the position of Neptune as seen from the Earth . . . . .	XIV
12	Script to obtain the interpolating polynomial of the position of Pluto as seen from the Earth	XV
13	Main function for LAGEOS-1 [5] . . . . .	XVI
14	Rates function containing the differential equation for LAGEOS-1 [5] . . . . .	XVII
15	Output function for LAGEOS-1 [5] . . . . .	XXI
16	Main function for ISS [5] . . . . .	XXIV
17	Rates function containing the differential equation for ISS [5] . . . . .	XXV
18	Output function for ISS [5] . . . . .	XXIX

## Nomenclature

$a$	Semimajor axis
$\vec{a}_G$	Acceleration vector of the center of mass of the set of two bodies
$\vec{a}_{2/1}$	Acceleration vector of the secondary body with respect to the main one
$\vec{a}_2$	Acceleration vector of the secondary body as seen from an inertial frame
$\vec{a}_1$	Acceleration vector of the main body as seen from an inertial frame
$A$	Cross-sectional surface of the secondary body
$A_S$	Cross-sectional absorbing area of the secondary body
$\alpha$	Asteroid surface's absorptivity
$B$	Ballistic coefficient
$c$	Speed of light
$C$	Thermal capacity
$C_R$	Radiation pressure coefficient
$C_D$	Drag coefficient
$\vec{C}$	Laplace vector
$\vec{D}$	Drag force vector
$\Delta m$	Propellant mass needed for executing an impulsive maneuver
$\Delta v$	Speed impulse needed for executing an impulsive maneuver
$e_z$	$z$ coordinate of vector $\vec{e}$
$\vec{e}$	Eccentricity vector
$E$	Eccentric anomaly
$\epsilon$	Specific energy // Emissivity
$\vec{F}_{12}$	Force exerted by the secondary body on the main body
$\vec{F}_{21}$	Force exerted by the main body on the secondary body
$g_0$	Gravity acceleration at sea level
$G$	Universal gravitational constant
$\gamma$	Flight path angle
$h_z$	$z$ coordinate of vector $\vec{h}$
$\vec{h}$	Specific angular momentum vector
$H_i$	Scale heights
$i$	Inclination
$I_{sp}$	Specific impulse
$J_k$	Zonal harmonics
$k$	Counter variable
$\vec{k}$	$z$ axis direction vector
$K$	Thermal conductivity
$m_i$	Mass of the satellite

$m_1$	Mass of the main body
$m_2$	Mass of the secondary body
$m_3$	Mass of the third body
$m_E$	Mass of the Earth
$m_{ISS}$	Mass of the ISS
$m_{LAGEOS-1}$	Mass of LAGEOS-1
$M$	Mean anomaly
$\mu$	Gravitational parameter of the Earth
$n_x$	x coordinate of vector $\vec{n}$
$n_y$	y coordinate of vector $\vec{n}$
$n_{av}$	Mean motion
$\vec{n}$	Direction of the line of nodes
$\nu$	Shadow function
$\omega$	Argument of perigee // Rotation frequency
$\vec{\omega}_E$	Angular velocity vector of the Earth
$\Omega$	Right Ascension of the Ascending Node
$\vec{\Omega}$	Absolute angular velocity
$\dot{\vec{\Omega}}$	Absolute angular acceleration
$p$	Orbital parameter
$\vec{p}$	General perturbing acceleration
$P_k$	Legendre polynomials
$P_{SR}$	Solar Radiation Pressure coefficient
$\phi$	Polar angle
$\Phi$	Gravitational potential perturbation
$r_a$	Apogee radius
$r_p$	Perigee radius
$\vec{r}$	Position vector of the secondary body with respect to the main body
$\vec{r}_x$	Position vector of body $x$ with respect to the Earth
$\vec{r}_{x/s}$	Position vector of body $x$ with respect to the satellite
$\vec{r}_0$	Initial position vector
$\dot{\vec{r}}$	Velocity vector of the secondary body with respect to the main body
$\ddot{\vec{r}}$	Acceleration vector of the secondary body with respect to the main body
$\vec{r}_{rel}$	Relative position vector of the secondary body with respect to the main body
$\dot{\vec{r}}_{rel}$	Relative velocity vector of the secondary body with respect to the main body
$\ddot{\vec{r}}_{rel}$	Relative acceleration vector of the secondary body with respect to the main body
$R$	Earth radius at the equator
$R_0$	Radius of the Sun

$R_{E-S}$	Mean distance from Sun to Earth
$\vec{R}_1$	Position vector of the center of mass of the main body
$\vec{R}_2$	Position vector of the center of mass of the secondary body
$\dot{\vec{R}}_1$	Velocity vector of the center of mass of the main body
$\dot{\vec{R}}_2$	Velocity vector of the center of mass of the secondary body
$\ddot{\vec{R}}_1$	Acceleration vector of the center of mass of the main body
$\ddot{\vec{R}}_2$	Acceleration vector of the center of mass of the secondary body
$\vec{R}_G$	Position vector of the center of mass of the set of two bodies
$\vec{R}_{G_0}$	Initial position vector of the center of mass of the set of two bodies
$\dot{\vec{R}}_G$	Velocity vector of the center of mass of the set of two bodies
$\ddot{\vec{R}}_G$	Acceleration vector of the center of mass of the set of two bodies
$\rho$	Density
$\rho_s$	Surface density
$S_0$	Radiated power intensity at the Surface of the Sun
$S_{E-S}$	Radiated power intensity at the orbit of the Earth
$\sigma$	Stefan-Boltzmann constant
$t$	Time
$t - t_p$	Time from perigee
$T$	Period
$T(a)$	Local subsolar temperature
$T_p$	Effective temperature of the photosphere
$\vec{t}$	Time vector
$\vec{t}_{span}$	Time span vector
$\theta$	True anomaly // Angle between vectors $\vec{r}$ and $\vec{r}_{sun}$
$\theta_1$	Angle between vector $\vec{r}$ and a line drawn from the satellite tangent to the Earth
$\theta_2$	Angle between vector $\vec{r}_{sun}$ and a tangent drawn from the Sun to the Earth
$\vec{u}_r$	Unit vector corresponding to $\vec{r}$
$\vec{u}_{r_{sun}}$	Unit vector corresponding to $\vec{r}_{sun}$
$\vec{u}_{v_{rel}}$	Unit vector corresponding to $\vec{v}_{rel}$
$v_p$	Tangential velocity
$v_r$	Radial velocity
$\vec{v}$	Velocity vector of the secondary body with respect to the main body
$\vec{v}_0$	Initial velocity vector
$\dot{\vec{v}}$	Acceleration vector of the secondary body with respect to the main body
$\vec{v}_G$	Velocity vector of the center of mass of the set of two bodies
$\vec{v}_{atm}$	Atmosphere velocity



$\vec{v}_{rel}$	Relative velocity vector of the secondary body with respect to the main body
$V$	Gravitational potential
$x$	x coordinate of position vector
$\ddot{x}$	x coordinate of acceleration vector
$x_1$	x coordinate of vector $\vec{R}_1$
$x_2$	x coordinate of vector $\vec{R}_2$
$y$	y coordinate of position vector
$\ddot{y}$	y coordinate of acceleration vector
$y_1$	y coordinate of vector $\vec{R}_1$
$y_2$	y coordinate of vector $\vec{R}_2$
$\vec{y}$	State vector of secondary body
$\vec{y}_0$	Initial state vector
$z$	z coordinate of position vector
$\ddot{z}$	z coordinate of acceleration vector
$z_1$	z coordinate of vector $\vec{R}_1$
$z_2$	z coordinate of vector $\vec{R}_2$

# 1 Introduction

Over the course of the years, the evolution and level of development achieved in the Orbital Mechanics field has been mainly driven by the human being's interest and need to explore beyond what is known so as to find an answer to most of the questions regarding the way the Universe works, and the Physics behind it. Advances in such field, as well as in technology and telecommunications, have opened the doors to a whole world of possibilities, allowing for space missions to take place or for satellites to be put into orbit with very different purposes.

In order for this to be successfully carried out, it is crucial, not only to perform the required mathematical calculations which determine and define the exact path which is going to be followed by the space vehicle or satellite, but also to do it with the maximum level of accuracy achievable by the mathematical robots and powerful computers available. That is essentially the main goal of the present document, providing a precise solution to the two-body problem in space, which consists of determining the state vector of a satellite which is found orbiting around the Earth, in this case.

The forces that take part in the problem of determining the orbit are certainly more than just the pure gravitational attraction between both bodies. There exist several perturbing forces in space which influence and modify the trajectory that must be followed by the satellite. These comprise, for instance, the main body's zonal harmonics, the friction forces caused due to the atmosphere, which specially affect the orbits of satellites which are closer to the planet, the solar radiation pressure or the effects of relativity. The purpose is to account for as many of these perturbing forces as possible, in addition to the mutual gravitational attraction forces, thereby increasing the level of accuracy of the results and reducing the error in the calculations with respect to the proper real values of the state vector of a given satellite for a certain time slot.

For that purpose, this paper has been divided into two main sections: the first one devoted to explain carefully the problem of two bodies in space with its two different approaches, and the second one regarding the addition of the aforementioned perturbing accelerations. Throughout the development of the project, an approach to two different real satellites is provided: LAGEOS-1 and the International Space Station. In the first place, the two-body problem is solved for both satellites without introducing perturbations. Then, the perturbing accelerations start being added one by one in such a way that the reader can appreciate how the error keeps diminishing as more perturbations are included. Moreover, the orbital parameters are obtained for each case under study so that they can be readily compared to each other and the evolution and reduction of the error can be quantified. The mathematical robot that will be used for performing the numerical integration and the required calculations is MATLAB<sup>™</sup>.

Furthermore, it is possible to deepen the understanding of the problem and clearly visualize the improvement of the results by executing an application which shows an interactive interface by means of which the user can configure the perturbations and the kind of satellite in order to obtain its trajectory, the orbital parameters and the graphs showing the state vector error. Additionally, this user-friendly interface allows to introduce a  $\Delta v$  impulse in any of the axes by setting a value for the specific impulse and specifying the time instant at which the impulse is desired to take place. This impulsive maneuver aims to perform an apogee or perigee change, or even a change in the orbit plane.

Once the core of the project has been exposed and the results have been depicted and compared to each other, the main and most relevant conclusions are extracted, and the possibility of performing some future research and analysis is proposed.

To finish off, at the end of this paper, the budgeting and costs associated to carrying out this final thesis project appear broken down into software costs, hardware costs and personnel working hours costs.

## 2 The Two-Body Problem

This section focuses on determining the equation of motion between two bodies which are only subject to their mutual gravitational attraction forces, both from an inertial reference frame and from the point of view of one of the bodies, which are in constant movement. The aim of this procedure is mainly to obtain the state vector of a given body, which provides the numerical values of the position and velocity components for a certain time instant. This will be carried out without taking into consideration, in the first place, any kind of perturbation that may affect the orbit calculation and the determination of the state vector of the secondary body, which is found orbiting around the main one.

### 2.1 Theoretical Background

#### 2.1.1 Inertial Reference Frame

This frame of reference, XYZ, allows to obtain the equation of motion for the classical two-body problem as seen from a point of view which is external to both bodies, and which can be moving with a constant speed with respect to the fixed stars, but does not change its orientation towards them.

$\vec{R}_1$  and  $\vec{R}_2$  are the position vectors of the centers of mass of the main and secondary bodies, respectively, as seen from the inertial reference frame, being  $m_1$  and  $m_2$  their respective masses. These bodies are only submitted to the attraction forces between them, being  $\vec{F}_{12}$  the force exerted by body 2 on body 1, and  $\vec{F}_{21}$  the opposite.

If now the position vector of body 2 with respect to body 1, that is, the vector pointing from the center of mass of  $m_1$  to that of  $m_2$ , is obtained, the result is vector  $\vec{r}$ .

$$\vec{r} = \vec{R}_2 - \vec{R}_1 \quad (1)$$

Then, the unit vector corresponding to  $\vec{r}$  is given by  $\vec{u}_r = \frac{\vec{r}}{r}$ . From this vector, the force exerted by one of the bodies on the other one is given by Newton's Law of Universal Gravitation, as it can be seen in Equation (2).

$$\vec{F}_{12} = \frac{Gm_1m_2}{r^2}\vec{u}_r \quad (2)$$

In this expression, the parameter G is defined as the universal gravitational constant, and has a value of  $6.6742 \cdot 10^{-20} \text{ km}^3/(\text{kg} \cdot \text{s}^2)$ . In addition,  $r$  represents the modulus of the position vector, which is the absolute distance between both masses,  $m_1$  and  $m_2$ , and is given by Equation (3).

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3)$$

By applying Newton's Second Law, it is known that  $\vec{F}_{12} = m_1 \cdot \ddot{\vec{R}}_1$ , where  $\ddot{\vec{R}}_1$  is the absolute acceleration of body 1. Hence, Equation (4) is achieved.

$$m_1 \cdot \ddot{\vec{R}}_1 = \frac{Gm_1m_2}{r^2}\vec{u}_r \quad (4)$$

According to Newton's Third Law or Principle of Action and Reaction, the force exerted by body 1 on body 2 has the same magnitude as  $\vec{F}_{12}$ , but opposite direction, in such a way that  $\vec{F}_{21} = -\vec{F}_{12}$ . Since  $\vec{F}_{21} = m_2 \cdot \ddot{\vec{R}}_2$ , a similar expression is obtained.

$$m_2 \cdot \ddot{\vec{R}}_2 = -\frac{Gm_1m_2}{r^2}\vec{u}_r \quad (5)$$

If both Equation (4) and (5) are divided by  $m_1$  and  $m_2$  respectively, what results then is the system of equations which govern the motion of two bodies due to their own gravitational attraction force, as seen from an inertial reference frame.

$$\ddot{\vec{R}}_1 = \frac{Gm_2}{r^2} \vec{u}_r \longrightarrow \ddot{\vec{R}}_1 = G \frac{m_2}{r^3} \vec{r} \quad (6)$$

$$\ddot{\vec{R}}_2 = -\frac{Gm_1}{r^2} \vec{u}_r \longrightarrow \ddot{\vec{R}}_2 = -G \frac{m_1}{r^3} \vec{r} \quad (7)$$

Moving on to the motion of the center of mass of the set of two bodies, its expression is given by Equation (8). In addition, the expressions for the absolute velocity and acceleration of the center of mass, measured from an inertial reference system, are shown in Equation (9) and (10). It is worth noticing that the sub-index  $G$ , which refers to the center of gravity in this context, must not be confused with the universal gravitational constant.

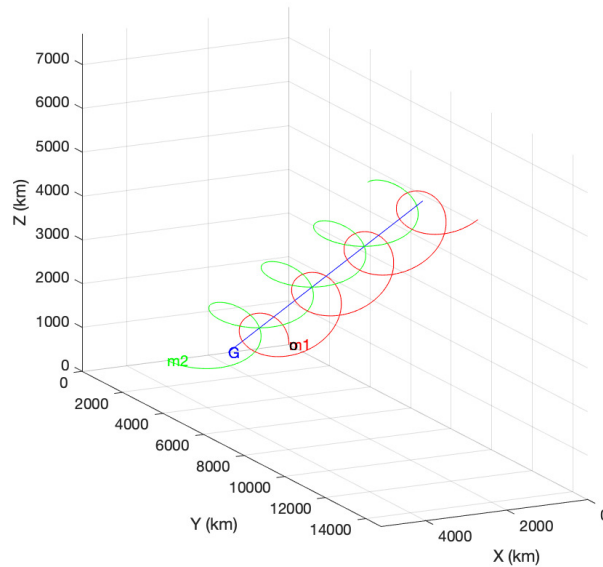
$$\vec{R}_G = \frac{m_1 \vec{R}_1 + m_2 \vec{R}_2}{m_1 + m_2} \quad (8)$$

$$\vec{v}_G = \dot{\vec{R}}_G = \frac{m_1 \dot{\vec{R}}_1 + m_2 \dot{\vec{R}}_2}{m_1 + m_2} \quad (9)$$

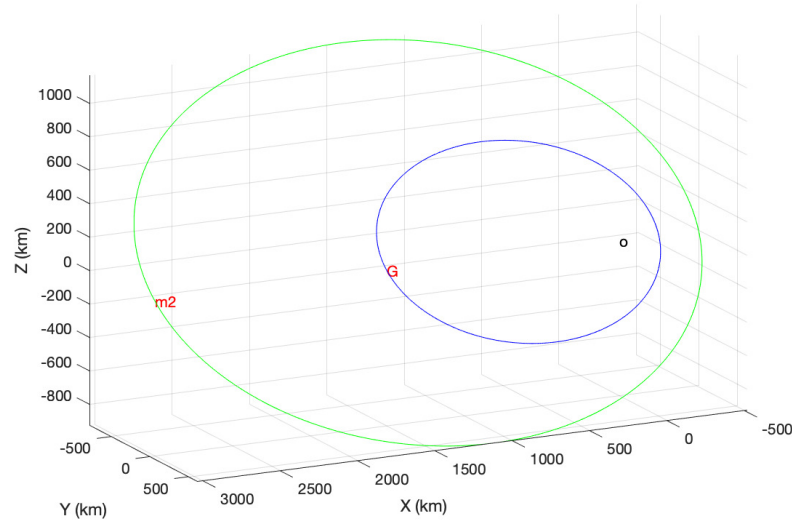
$$\vec{a}_G = \ddot{\vec{R}}_G = \frac{m_1 \ddot{\vec{R}}_1 + m_2 \ddot{\vec{R}}_2}{m_1 + m_2} \quad (10)$$

By adding now both Equation (6) and (7) multiplied by the masses, the result is  $m_1 \ddot{\vec{R}}_1 + m_2 \ddot{\vec{R}}_2 = \vec{0}$ . Hence, the numerator in Equation (10) is zero, which means the acceleration vector becomes null, meaning that the center of mass of the system follows a Uniformly Rectilinear Motion, moving along a straight line with a constant velocity, according to  $\vec{R}_G = \vec{R}_{G_0} + \vec{v}_G t$ , where  $\vec{R}_{G_0}$  is the initial position vector.

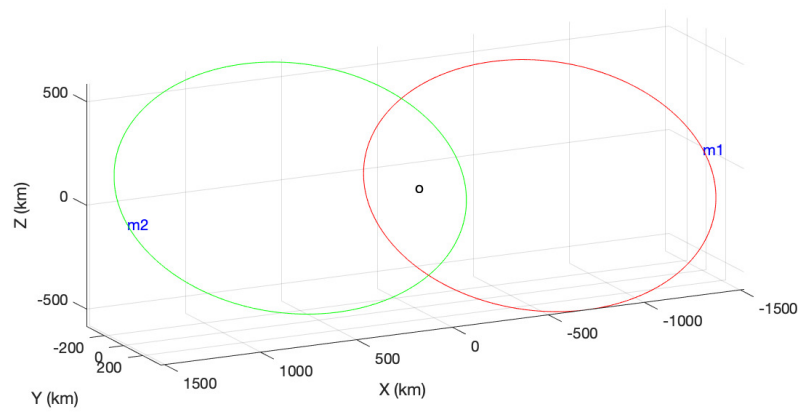
To finish off with the present section, it is worth representing graphically the motion of two bodies of equal masses, with respect to an inertial reference frame, as well as from the point of view of one of both bodies and the center of mass of the system. These three cases are plotted in Figures 1, 2 and 3.



**Figure 1:** Motion of  $m_1$ ,  $m_2$  and the center of mass,  $G$ , from an inertial reference frame



**Figure 2:** Motion of  $m_2$  and the center of mass,  $G$ , as seen from  $m_1$



**Figure 3:** Motion of  $m_1$  and  $m_2$  as seen from the center of mass,  $G$

### 2.1.2 Body-Centered Reference Frame

The purpose of this section is to derive an expression for the equation of motion of the two-body problem in the same way as it was done in the previous case, but instead of setting as reference an inertial frame, the reference point this time is one of both bodies, which is usually the one with a larger mass.

Departing from the position vector,  $\vec{r}$ , obtained in Equation (1), by just differentiating it with respect to time twice and then substituting Equation (6) and (7), the resulting expression is that of the relative motion between two bodies which are only submitted to the attraction forces between them.

$$\ddot{\vec{r}} = \ddot{\vec{R}}_2 - \ddot{\vec{R}}_1 \quad (11)$$

$$\ddot{\vec{r}} = -G \frac{m_1}{r^3} \vec{r} - G \frac{m_2}{r^3} \vec{r} \quad (12)$$

$$\ddot{\vec{r}} = -G \frac{(m_1 + m_2)}{r^3} \vec{r} \quad (13)$$

If  $\mu = G(m_1 + m_2)$ , with units  $km^3/s^2$ , is called the *gravitational parameter*, then the equation of relative motion results in Equation (14).

$$\ddot{\vec{r}} = -\frac{\mu}{r^3} \vec{r} \quad (14)$$

This expression, known as the fundamental equation of relative motion between two bodies, is a non-linear second order differential equation which rules the motion of body 2 with respect to body 1, which is indeed the same as the motion of body 1 as seen from body 2, since the only thing that changes is the minus sign in the equation of relative motion. Since the position vector was initially defined with respect to an inertial frame, it must now be adapted to a reference system attached to body 1, which is regarded as the origin of such reference frame. This frame is constantly moving with  $m_1$ .

If  $\vec{\Omega}$  and  $\dot{\vec{\Omega}}$  are the absolute angular velocity and angular acceleration, respectively, then the absolute acceleration of a body, as seen from an inertial point of view,  $\ddot{\vec{r}}$ , is related to its relative acceleration, as seen from a moving reference frame,  $\ddot{\vec{r}}_{rel}$ , as shown in Equation (15). The last term in the expression is the Coriolis acceleration.

$$\ddot{\vec{r}} = \ddot{\vec{r}}_{rel} + \dot{\vec{\Omega}} \times \vec{r}_{rel} + \vec{\Omega} \times (\vec{\Omega} \times \vec{r}_{rel}) + 2\vec{\Omega} \times \dot{\vec{r}}_{rel} \quad (15)$$

According to this expression, shall  $\vec{\Omega}$  and  $\dot{\vec{\Omega}}$  be equal to 0, that is, if body 1 is not rotating, then the absolute acceleration of body 2 will be equal to the relative one.

The last step is to separate Equation (14) into the three different expressions which refer to each of the coordinates of vector  $\vec{r}$ , in order to subsequently integrate in a numerical way so as to find the absolute position vector of body 2.

$$\ddot{x} = -\frac{\mu}{r^3} x \quad \ddot{y} = -\frac{\mu}{r^3} y \quad \ddot{z} = -\frac{\mu}{r^3} z \quad (16)$$

In case the relative position of mass 1 or mass 2 with respect to the center of mass should be obtained, the resulting expression is quite similar to that of the relative motion between both bodies, and it can be derived from the fact that the position vector of  $G$  with respect to itself is null, in such a way that  $m_1 \vec{R}_1 + m_2 \vec{R}_2 = 0$ . From that expression, it can be stated that  $\vec{R}_1 = -m_2/m_1 \cdot \vec{R}_2$ . By substituting the latter expression in Equation (1), it follows that  $\vec{r} = (m_1 + m_2)/m_1 \cdot \vec{R}_2$ . Finally, this expression is in turn substituted into the equation of relative motion of body 2 with respect to the center of mass, given by (17).

$$-G \frac{m_1 m_2}{r^2} \vec{u}_r = m_2 \ddot{\vec{R}}_2 \quad (17)$$

The resulting expressions of the relative motion of bodies 1 and 2 with respect to  $G$  are shown in Equation (18).

$$\ddot{\vec{R}}_1 = -\left(\frac{m_2}{m_1 + m_2}\right)^3 \frac{\mu}{R_1^3} \vec{R}_1 \quad \ddot{\vec{R}}_2 = -\left(\frac{m_1}{m_1 + m_2}\right)^3 \frac{\mu}{R_2^3} \vec{R}_2 \quad (18)$$

## 2.2 Approach to real cases

For the purpose of this project, the chosen reference system is the geocentric one, in which the Earth is regarded as the main body about which a satellite is found orbiting, it being the secondary body. Henceforth, the fundamental equation of relative motion between two bodies is to be applied to each of the cases of study throughout this paper, following the mathematical development of the body-centered reference frame.

The final goal is, as it has already been mentioned, to find the state vector of a satellite orbiting around the Earth, and the first approach given in this section consists of performing the required calculations without accounting for any perturbing accelerations, by just considering the mutual gravitational attraction between both bodies.

In order to be able to compare and contrast the results as they are obtained, it is interesting to consider at least two cases of study. In other words, it is worth finding the solution for two different satellites travelling in geocentric orbits, as an approach to two different real cases, for which the state vector is known and can be extracted from a database provided by NASA. These two satellites are going to be LAGEOS-1 and the International Space Station, which orbits are, respectively, MEO (Medium Earth Orbit) and LEO (Low Earth Orbit).

### 2.2.1 LAGEOS-1 orbit

Launched on 4 May, 1976 from California, LAGEOS-1, short for *Laser Geodynamic Satellite*, is the first ever NASA satellite devoted to taking precise measurements by means of a technique called *laser ranging*, with the aim of providing relevant information about the Earth's dynamics, its gravitational field, its shape and the migration of its axis of rotation [1]. It is located in a Medium Earth Orbit, right between LEO and Geostationary orbits.



**Figure 4:** LAGEOS-1 satellite with corner-cube reflectors [1]

Regarding the satellite itself, its shape is spherical, as seen in Figure 1, approaching that of a cannonball, and it is not equipped with any sensors, nor any moving parts, electronics nor attitude control, hence it is a passive research device, which does not allow communication. Its mass is 406.965 kg, and its diameter is 60 cm long. It contains a brass core, and the external shell is made of aluminium. These materials were deliberately chosen so that the Earth's magnetic field had the lowest possible impact over LAGEOS-1 orbit. The surface shell is covered with 426 of the so-called *corner-cube reflectors*, which consist of reflecting prisms, some of which are made of germanium, whereas the vast majority of them are made of fused silica glass. These retro-reflectors reflect laser pulses coming from ground stations located on Earth, and part of this signal travels back to the station. The exact distance between the satellite and the station is then measured by previously determining with precision the time it takes the laser pulse to travel to LAGEOS-1 and back [8].

The satellite is supposed to stay in orbit for approximately 8.4 million years, when it is expected to fall back to Earth. With the aim of sending a message to future civilisations living by that time, LAGEOS-1 carries a plaque, designed by Carl Sagan, which shows the Earth orbiting around the Sun, the numbers from 1 to 10 written in binary code, and three world maps, one corresponding to Pangea, another one showing the Earth as it is today, and a final one which shows how the Earth is expected to be by the time the satellite falls back to the planet.

Once LAGEOS-1 is introduced and defined, it is worth moving on to the resolution of the problem itself, which in this case is the two-body problem, taking the Earth as the center of the reference frame. For that purpose, the MATLAB™ code shown in Appendix A is implemented and executed. Some relevant data that appear in the code are defined hereafter. It is interesting to comment that the unit used when it comes to measuring distances in space is the kilometer instead of the meter.

- $G = 6.6742 \cdot 10^{-20} \text{ km}^3 / (\text{kg} \cdot \text{s}^2)$
- $R = 6378 \text{ km}$
- $m_E = 5.972 \cdot 10^{24} \text{ kg}$
- $m_{LAGEOS-1} = 406.965 \text{ kg}$
- $\mu_{LAGEOS-1} = 398583.224 \text{ km}^3 / \text{s}^2$

The aim is to numerically integrate the fundamental equation of relative motion between two bodies by making use of the already implemented in MATLAB™ solver `ode45`. Its generic expression is defined in Equation (19).

$$[\vec{t}, \vec{y}] = \text{ode45}(\text{odefun}, \vec{t}_{span}, \vec{y}_0, \text{options}) \quad (19)$$

According to the previous expression, `ode45` requires 4 inputs, though the *options* input is actually dispensable. The first input required refers to the function which contains the differential equation to be solved for. Henceforth, such function will be called *rates.m*. The second input to be introduced to the solver is the time interval desired for the integration. For all the cases under study throughout this paper, the time span introduced will be 1 hour, meaning that the solution is going to be obtained for every hour inside a closed interval, starting from 1 May, 2021 at 00:00:00, and finishing on 31 May, 2021, at 23:00:00. The last input required is  $y_0$ , which is the initial conditions state vector. This vector is directly obtained from a NASA database named Horizons [2], which provides relevant data and allows to generate ephemerides for several bodies in the solar system. For the case of LAGEOS-1 satellite, the configuration of the web interface must be performed as shown in Figure 5. By pushing the Generate Ephemeris button, the text file containing the state vector at every hour during May, 2021 for LAGEOS-1 is automatically downloaded. From such file, the initial conditions vector can be extracted.

- $\vec{r}_{0LAGEOS-1} = (8.433979054738516 \cdot 10^2, 1.139252816099018 \cdot 10^4, 4.625360099492756 \cdot 10^3) \text{ km}$
- $\vec{v}_{0LAGEOS-1} = (2.207532373984499, 1.831438744379734, -4.896873467934014) \text{ km/s}$
- $\vec{y}_{0LAGEOS-1} = (\vec{r}_{0LAGEOS-1}, \vec{v}_{0LAGEOS-1})$



**HORIZONS Web-Interface**

This tool provides a web-based *limited* interface to JPL's **HORIZONS system** which can be used to generate ephemerides for solar-system bodies. Full access to **HORIZONS** features is available via the primary [telnet interface](#). **HORIZONS system news** shows recent changes and improvements. A [web-interface tutorial](#) is available to assist new users.

**Current Settings**

Ephemaris Type [\[change\]](#) : **VECTORS**  
 Target Body [\[change\]](#) : **LAGEOS-1 (spacecraft) [1976-039A]** [-108820]  
 Coordinate Origin [\[change\]](#) : **Geocentric** [500]  
 Time Span [\[change\]](#) : Start=**2021-05-1**, Stop=**2021-06-1**, Step=**1 h**  
 Table Settings [\[change\]](#) : quantities code=**2**; output units=**KM-S**; reference plane=**FRAME**; CSV format=**YES**  
 Display/Output [\[change\]](#) : **download/save** (plain text file)

**Special Options:**

- [set default ephemeris settings](#) (preserves only the selected target body and ephemeris type)
- [reset all settings to their defaults](#) (caution: all previously stored/selected settings will be lost)
- [show "batch-file" data](#) (for use by the [E-mail interface](#))

**ABOUT SSD**   **CREDITS/AWARDS**   **PRIVACY/COPYRIGHT**   **GLOSSARY**   **LINKS**

**FIRSTGOV**  
Your First Click to the U.S. Government

2021-Jul-06 08:03 UT  
(server date/time)

**NASA**

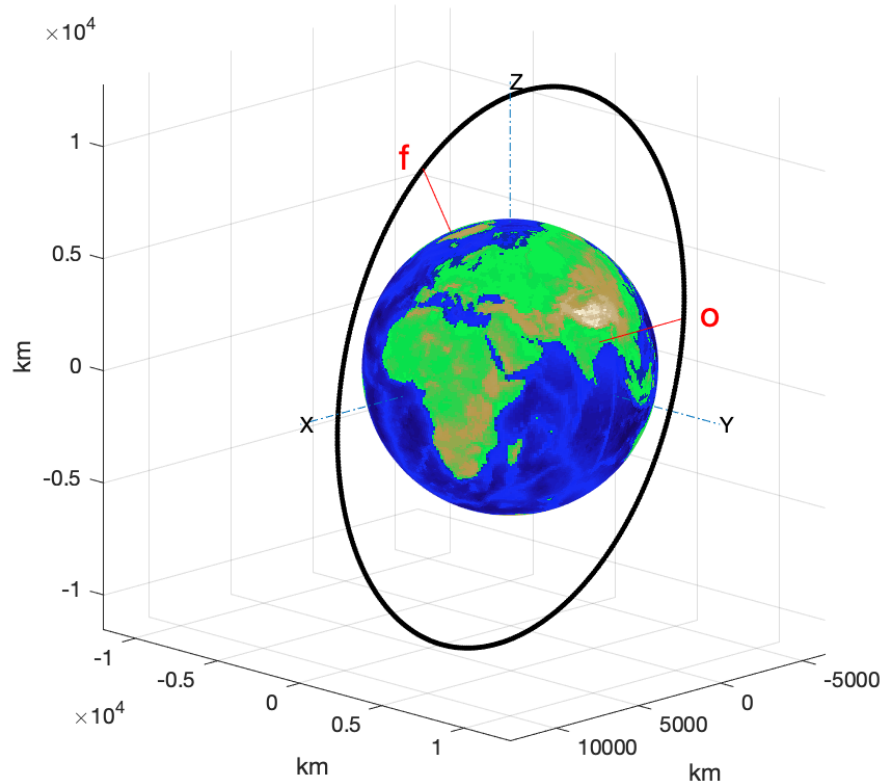
Site Manager: Ryan S. Park  
Webmaster: Alan B. Chamberlin

**Figure 5:** Horizons interface configuration [2]

The last input of the solver `ode45` to be introduced is not required, but it is interesting to include it in order to reduce the error tolerance of the differential equation as much as possible. To that end, the absolute and relative error tolerances introduced are  $10^{-13}$ , as shown in Equation (20).

$$options = odeset('RelTol', 10^{-13}, 'AbsTol', 10^{-13}) \quad (20)$$

The vectors resulting from the integration are the final time vector,  $\vec{t}$ , which is actually equal to the time span input vector, and the final state matrix of the satellite,  $\vec{y}$ , in which each row corresponds to the state vector for each time instant. By plotting the resulting position vectors in 3D around the Earth, the points corresponding to each time instant are so closely placed that the shape of the orbit is defined and the trajectory followed by the satellite during May, 2021 can be appreciated in Figure 6. Moreover, the initial and final points in the orbit are marked with an 'o' and an 'f' symbol, respectively, for greater clarity.



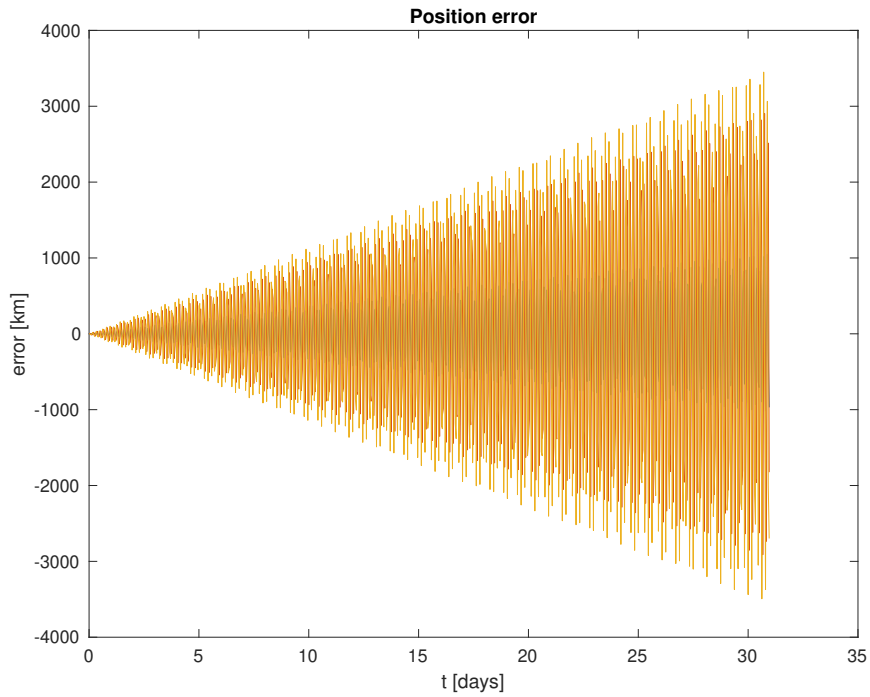
**Figure 6:** LAGEOS-1 trajectory during May 2021 without introducing perturbations

The final state matrix,  $\vec{y}$ , is then compared to the matrix of real values extracted from the Horizons generated ephemeris, and the absolute error vector is obtained and plotted with respect to time, differentiating between the position and velocity components of the state vector. These plots are depicted in Figures 7 and 8.

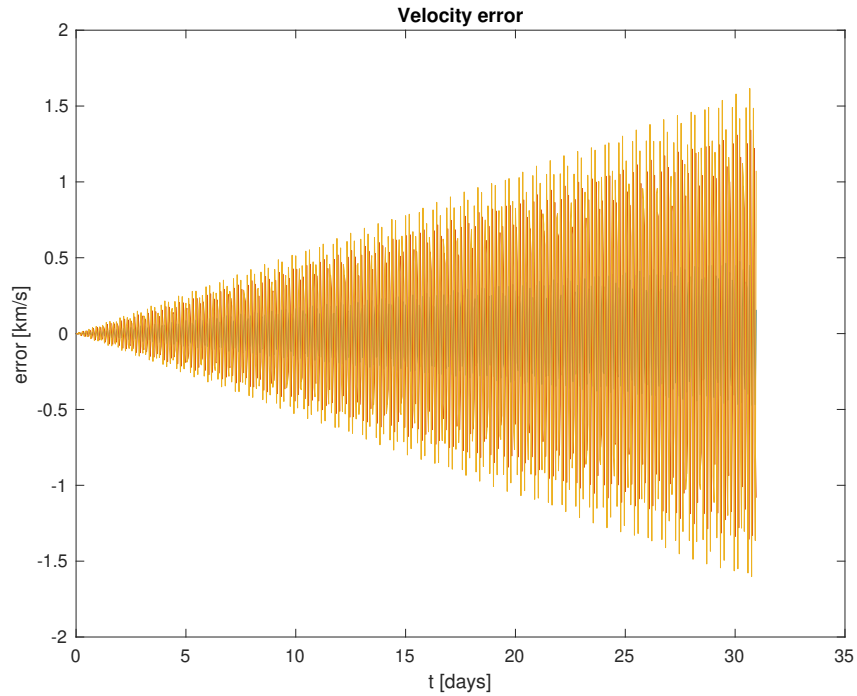
For each of the error plots, each of the components for both the position and the velocity vectors correspond to a different tonality in the graph. It can be appreciated how both graphs present a similar triangular shape, meaning that the error keeps increasing linearly as time advances, due to the fact that the error keeps accumulating. At the beginning, for about the first five days from the launch of the satellite, the position obtained from the numerical integration approaches the real position according to the Horizons database, since the position error is lower than 1000 km and the velocity error does not overcome half a kilometer per second, as it can be observed in the plots. Notwithstanding, as time reaches day 31, the error obtained presents relatively high values, which are non-negligible.

Therefore, this solution cannot be considered valid at all, and even less so when it comes to the trajectory of a satellite in a space mission, where precision and accuracy are a crucial key to success, and where the time devoted and financial investment are factors to be taken into account.

That is the main point of adding perturbing accelerations to the differential equation of relative motion: to increase the accuracy of the results and thus, be able to appreciate how both the position and velocity errors get reduced, reaching a valid solution for the state vector of the satellite to the extent possible.



**Figure 7:** LAGEOS-1 absolute position error during May 2021 without introducing perturbations



**Figure 8:** LAGEOS-1 absolute velocity error during May 2021 without introducing perturbations

### 2.2.2 International Space Station orbit

The International Space Station, most commonly known as ISS, is the biggest artificial satellite in space, consisting of several inhabitable pressurized modules which were launched by Russian rockets such as Soyuz and Proton, and by the American Space Shuttle. Its main purpose is to carry out microgravity studies and research about physics, astronomy, astrobiology and many other disciplines. To that end, some manned missions are performed occasionally, in which several astronauts travel to the space station so as to conduct some activities and analyses.

The station is located in a LEO orbit, at around 400 km above the Earth surface, hence the atmospheric drag, even though density is quite low at such altitudes, does have an influence over the trajectory of the satellite. For that reason, several reboosts are frequently needed in order to keep the ISS in its corresponding orbit, which are carried out by supply vehicles sent to the station [5].

The inclination of the ISS, as well as its period, are constant. Nevertheless, due to the rotational motion of the Earth, it covers most of its surface, as it can be seen in Figure 9, where several orbits drawn along time appear superimposed, seeing the Earth in a fixed location [3].



**Figure 9:** View of several superimposed ISS orbits [3]

Moving on to the resolution of the two-body problem for the case of the International Space Station, it is worth defining some parameters, as it was done for the case of LAGEOS-1. The general ones were already defined in the previous case, in particular the universal gravitational constant, the Earth radius and its mass. The mass and gravitational parameter of the ISS appear defined below.

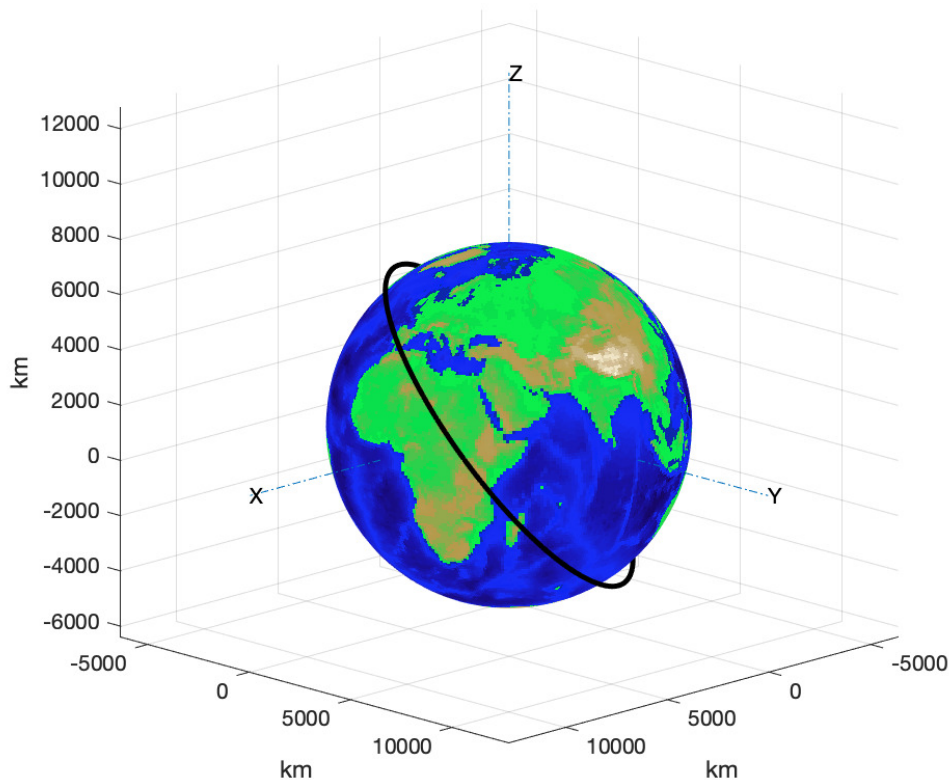
- $m_{ISS} = 419725 \text{ kg}$
- $\mu_{ISS} = 398583.224 \text{ km}^3/\text{s}^2$

The differential equation of relative motion between two bodies is contained in function *rates.m*, as in the case of LAGEOS-1. This function is integrated by means of solver *ode45*, which generic expression is given by Equation (19). The time span vector is the same as before, starting on 1 May, 2021 at 00:00:00 and performing the calculations for every hour until 31 May, 2021 at 23:00:00. Then, the initial conditions vector for the ISS is extracted from the Horizons web interface and shown hereafter.

- $\vec{r}_{0ISS} = (-6.120379106014391 \cdot 10^2, -4.951627922275376 \cdot 10^3, 4.609784569533521 \cdot 10^3) \text{ km}$
- $\vec{v}_{0ISS} = (6.775251885353307, 1.955736826657800, 2.995327764020777) \text{ km/s}$
- $\vec{y}_{0ISS} = (\vec{r}_{0ISS}, \vec{v}_{0ISS})$

The *options* input sets the absolute and relative error tolerances to  $10^{-13}$ , as in the case of LAGEOS-1, and it is configured in MATLAB™ as shown in Equation (20).

By executing the code from Appendix A corresponding to the ISS without accounting for any perturbing accelerations, the results are the ISS trajectory in Figure 10, and the position and velocity errors in Figures 11 and 12, respectively.

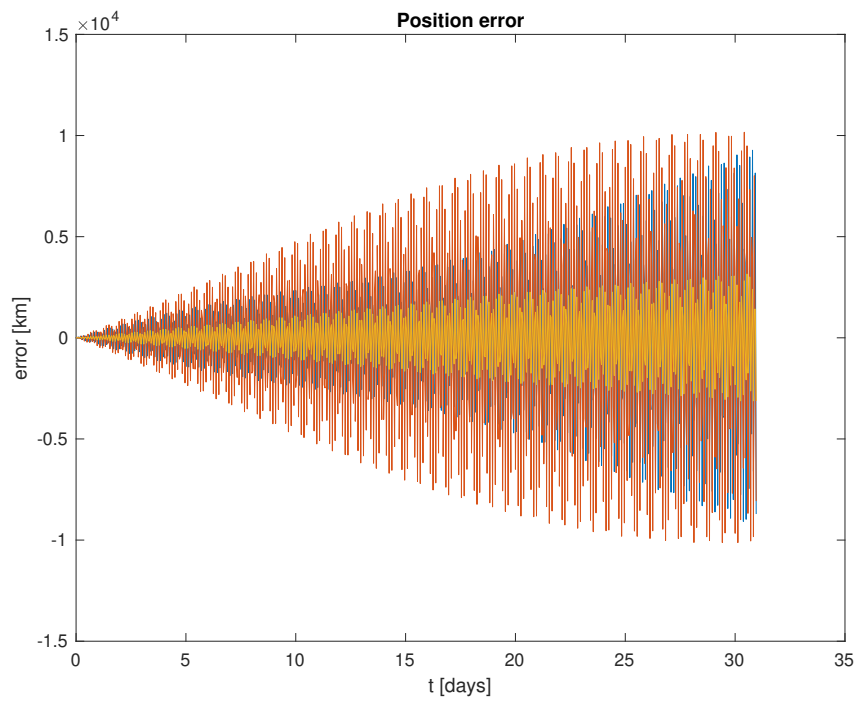


**Figure 10:** ISS trajectory during May 2021 without introducing perturbations

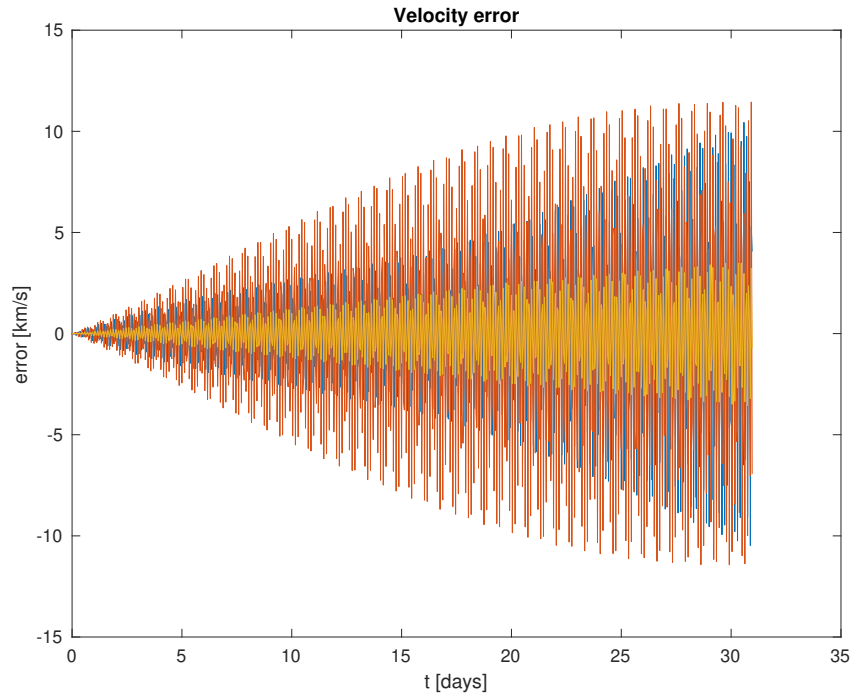
Regarding the 3D plot of the trajectory followed by the ISS along one month according to the results of the integration, it can be stated that all of the different path points fall inside a plane which defines the orbit. Nonetheless, the plot of the orbit calculated for a considerable amount of time should look like the one shown in Figure 9, considering the Earth is in a fixed position and it is the orbit that 'rotates' around it. Since that is clearly not the case, the results obtained are likely to differ to a great extent from the real values provided by Horizons. Hence, the state vector absolute error should be quite high, even higher than in the case of LAGEOS-1 satellite.

By observing the position and velocity errors depicted in Figures 11 and 12, it can be appreciated how these errors are actually really large, being the absolute position error one order of magnitude higher than in the case of LAGEOS-1, reaching differences of around 10000 km at the end of the month. The velocity also differs largely from the real values, obtaining errors up to around 12 km/s. The triangular shape of the ISS state vector error is quite similar to that of the previous case, achieved due to the error accumulation with time. It is worth commenting that each color appreciated in the graphs corresponds to the absolute error magnitude of each of the different coordinates of the position and velocity vectors.

It is then necessary to include some perturbations to the integration so that the results obtained are more accurate, the position and velocity errors get reduced, and the solution can be considered acceptable. That is the purpose of the following section.



**Figure 11:** ISS absolute position error during May 2021 without introducing perturbations



**Figure 12:** ISS absolute velocity error during May 2021 without introducing perturbations

### 3 Introduction to Orbital Perturbations

The main point of the current chapter lies in the fact that, in order that the calculation of the state vector for certain discrete values of time is performed as accurately as possible, the two-body problem cannot be limited to the fundamental equation of relative motion in which the only force acting is the mutual gravitational attraction exerted between both bodies. When approaching reality and introducing a certain degree of accuracy in the calculations, several considerations must be taken into account in the shape of phenomena which perturb the ideal, or Keplerian, orbit, obtained in the previous section.

Some of these perturbations include the different *zonal harmonics* of the planet acting as the main body, the atmospheric drag which changes with density and thus, with altitude over the surface of the planet, the effect of lunar gravity or the relativistic effects which start being noticed when the speed achieved by the body which is in orbit, in this case, the satellite, reaches considerably high values.

The perturbations are taken into consideration and included in the calculation of the state vector by means of the addition of extra terms to the fundamental equation of relative motion, in the shape of  $\vec{p}$  terms, which are actually acceleration terms. Each one of the extra terms accounts for a different perturbation. This method is called Cowell's method and it consists of performing a step-by-step straightforward integration [9]. Equation (21) shows the shape of the expression of relative motion which considers perturbations and will be integrated according to Cowell's method.

$$\ddot{\vec{r}} = -\mu \frac{\vec{r}}{r^3} + \vec{p} \quad (21)$$

This second order differential equation is translated into a system of first order differential equations such as the one given by Equation (22) so as to carry out the numerical integration.

$$\begin{cases} \dot{\vec{r}} = \vec{v} \\ \dot{\vec{v}} = -\mu \frac{\vec{r}}{r^3} + \vec{p} \end{cases} \quad (22)$$

It is worth mentioning the fact that the order of magnitude of the different accelerations coming from perturbations is generally lower than that of the gravitational attraction forces between both bodies. Atmospheric drag, though, does affect the orbit calculation when the satellite is orbiting inside the range 100 - 1000 km above the Earth surface. Figure 13 shows a comparison amongst orders of magnitude of the different accelerations affecting the two-body motion. Clearly, both  $J_2$  and atmospheric drag at low altitudes are the most relevant ones after the Earth's gravitational attraction. In the following subsections, each of the different perturbations affecting the calculation of the orbit is carefully exposed and explained, and an application to a real case is shown as well.

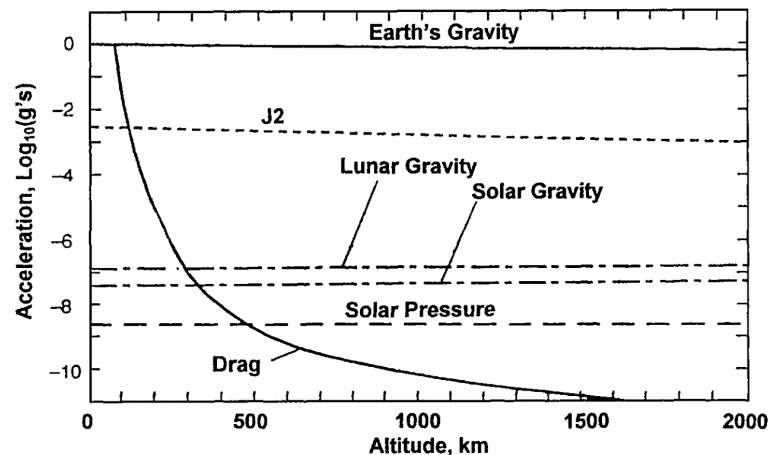


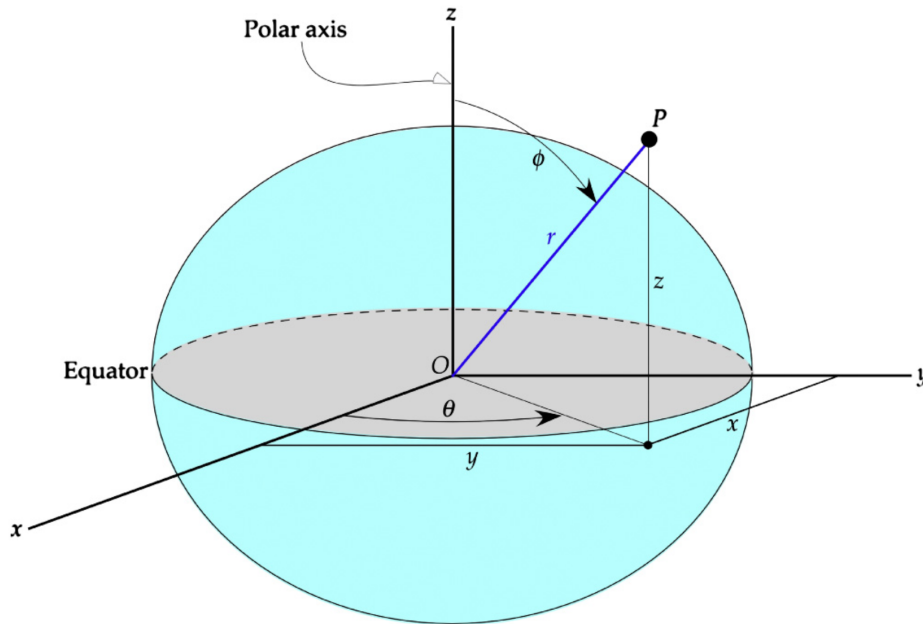
Figure 13: Importance of perturbing accelerations as a function of altitude [4]

### 3.1 Theoretical Background

#### 3.1.1 Gravitational Perturbations

The Earth, as well as any other object in the universe, is not a perfectly shaped sphere with a symmetrical mass distribution. It rather looks like an oblate spheroid and, indeed, the gravitational field is different depending on the latitude and the radius, due to the bump located in the equator which is a consequence of the centrifugal forces which appear due to the spinning motion of the planet.

It is interesting to define a reference frame, with its center being the center of mass of the spheroid, and the  $z$  axis being the rotational symmetry axis of the Earth, around which the gravitational field is approximately symmetrical. The  $x$  and  $y$  axis can be located anywhere within the equator plane, but for the purpose of this section, the  $x$  axis will be pointing to the vernal point, and the  $y$  axis will be forming a Cartesian right-handed frame. These Cartesian axes can be seen in Figure 14, where in addition, two different angles are shown. Angle  $\phi$  is defined by Equation (23) and it is called *polar angle*. On the other hand, angle  $\theta$  is named *azimuth angle*. They act as latitude and longitude angles, respectively.



**Figure 14:** Spherical reference frame [5]

$$\phi = \tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \quad (23)$$

The gravitational potential with no perturbations due to the Earth oblateness is defined by Equation (24), where  $r$  is the absolute value of the position vector,  $\vec{r}$ , as shown in Figure 14.

$$V = -\frac{\mu}{r} \quad (24)$$

Notwithstanding, as it has been stated already, the shape of the Earth approaches an oblate spheroid. An important hypothesis to take into consideration at this point is that, in spite of the irregularities of the Earth surface and its mass distribution, it will be assumed that the gravitational field is symmetrical with respect to the azimuth angle. The only variations will be considered with respect to  $r$  and  $\phi$ . Hence, the previous expression must account for another term which depends on the radius as well as on the polar angle. The final expression is given by Equation (25).



$$V(r, \phi) = -\frac{\mu}{r} + \Phi(r, \phi) \quad (25)$$

The term  $\Phi(r, \phi)$  in the previous expression stands for the gravitational potential perturbation, and is defined according to an infinite series [10], as shown in Equation (26).

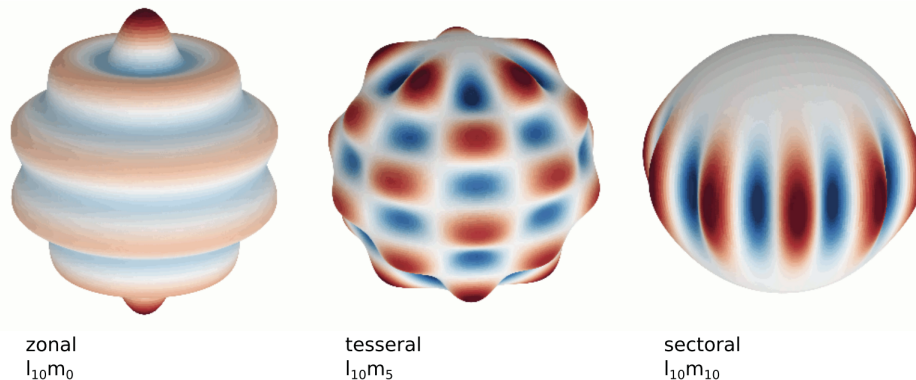
$$\Phi(r, \phi) = \frac{\mu}{r} \sum_{k=2}^{\infty} J_k \left(\frac{R}{r}\right)^k P_k(\cos \phi) \quad (26)$$

The parameter  $R$  is the Earth radius in the equator, and its value can be approximated to 6378 km. Then,  $J_k$  are known as the *zonal harmonics* of the main body, in this case, the planet. They consist of a series of dimensionless coefficients which have been empirically derived from experiments and observations of orbiting bodies. Each planet has its own zonal harmonics. In the case of the Earth, the first and most relevant ones are shown below [11]. It is interesting to comment that the first zonal harmonic,  $J_1$ , has been omitted due to the fact that the center of the established reference frame coincides with the center of mass of the main body, and thus,  $J_1$  is null.

- $J_2 = 0.00108263$
- $J_3 = -2.33936 \cdot 10^{-3} J_2$
- $J_4 = -1.49601 \cdot 10^{-3} J_2$
- $J_5 = -0.20995 \cdot 10^{-3} J_2$
- $J_6 = 0.49941 \cdot 10^{-3} J_2$
- $J_7 = 0.32547 \cdot 10^{-3} J_2$

As it can be appreciated,  $J_2$  is clearly more important than the rest of zonal harmonics, which are three orders of magnitude lower. Moreover, from  $J_7$  on, the values of subsequent harmonics become much smaller than that of  $J_2$ , in such a way that, when it comes to the summation in Equation (26), the only relevant coefficient will definitely be  $J_2$ . The more precision is required, the more terms should be added to the summation. However, for the purpose of the current analysis, henceforth it will be sufficient to keep on working with the zonal harmonics from  $J_2$  to  $J_6$ .

It is interesting to comment that, apart from this kind of spherical harmonics, there exist two more types inside this category: sectoral and tesseral harmonics, which can be considered negligible for the current study, but are shown in Figure 15 for greater clarity. The sectoral harmonics mainly consider the differences in shape longitudinally, due to the contrast between the densities of water and terrain, and the tesseral harmonics account for more specific regions in which the shape of the surface changes sharply [6].



**Figure 15:** Spherical harmonics [6]

The second zonal harmonic of the Earth accounts for its oblateness or its spheroid shape due to its equatorial bulge. The difference between the equatorial radius and the polar one allows to define a new parameter, named *oblateness*, which is given by Equation (27).

$$\text{oblateness} = \frac{\text{equatorial radius} - \text{polar radius}}{\text{equatorial radius}} = \frac{6378 - 6357}{6378} = 0.00330293 \quad (27)$$

This effect derives in variations over time in two different orbital elements: the Right Ascension of the Ascending Node,  $\Omega$ , and the Argument of the Perigee,  $\omega$ . These time variations are expressed in Equation (28) and (29), and their consequence is the regression of nodes and the change in the location of the perigee, which are not a subject of the present document.

$$\frac{d\Omega}{dt} = -\left(\frac{3J_2\sqrt{\mu}R^2}{2a^{7/2}(1-e^2)^2}\right)\cos i \quad (\text{rad/s}) \quad (28)$$

$$\frac{d\omega}{dt} = -\left(\frac{3J_2\sqrt{\mu}R^2}{2a^{7/2}(1-e^2)^2}\right)\left(\frac{5}{2}\sin^2 i - 2\right) = \frac{d\Omega}{dt} \frac{\frac{5}{2}\sin^2 i - 2}{\cos i} \quad (\text{rad/s}) \quad (29)$$

Regarding the zonal harmonics from the third to the sixth one, it is worth commenting what they represent. Starting from  $J_3$ , this harmonic accounts for long periodic effects. Moving on to the fourth one, this harmonic takes into consideration secular variations in orbital parameters due to the equatorial bulge of the planet [12]. The fifth and sixth harmonics have very little influence on the orbit determination, but they are also related to the Earth's oblateness.

Coming back to Equation (26), the last term to be defined is  $P_k$ , known as the *Legendre polynomials*. These can be calculated by means of an expression called *Rodrigues' formula*, given by Equation (30).

$$P_k(x) = \frac{1}{2^k k!} \frac{d}{dx^k} (x^2 - 1)^k \quad (30)$$

Since only the terms corresponding to  $k = 2, 3, 4, 5, 6$  will be kept for the development of the perturbing acceleration expression due to the Earth oblateness, it is worth obtaining the Legendre polynomials and representing them graphically in order to get a first idea of what they look like.

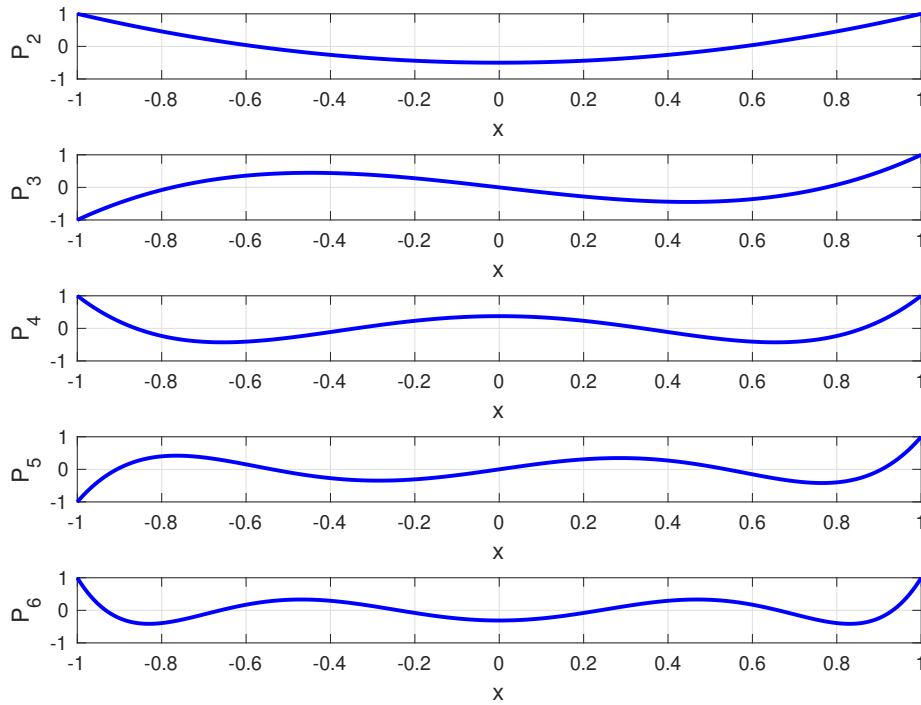
$$P_2(x) = \frac{1}{2}(3x^2 - 1) \quad (31)$$

$$P_3(x) = \frac{1}{2}(5x^3 - 3x) \quad (32)$$

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3) \quad (33)$$

$$P_5(x) = \frac{1}{8}(63x^5 - 70x^3 + 15x) \quad (34)$$

$$P_6(x) = \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5) \quad (35)$$



**Figure 16:** Legendre polynomials

By just considering  $J_2$  zonal harmonic and  $P_2$  Legendre polynomial, Equation (26) becomes Equation (36).

$$\Phi(r, \phi) = \frac{J_2 \mu}{2} \left(\frac{R}{r}\right)^2 (3\cos^2\phi - 1) \quad (36)$$

Once the expression for the gravitational potential perturbation is achieved, the next step is to look for the expression of the perturbing acceleration due to  $J_2$  zonal harmonic, which is obtained by developing Equation (37).

$$\vec{p} = -\nabla\Phi(r, \phi) \quad (37)$$

$$\frac{\partial\Phi(r, \phi)}{\partial x} = \frac{\partial\Phi}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial\Phi}{\partial\phi} \frac{\partial\phi}{\partial x} \quad \frac{\partial\Phi(r, \phi)}{\partial y} = \frac{\partial\Phi}{\partial r} \frac{\partial r}{\partial y} + \frac{\partial\Phi}{\partial\phi} \frac{\partial\phi}{\partial y} \quad \frac{\partial\Phi(r, \phi)}{\partial z} = \frac{\partial\Phi}{\partial r} \frac{\partial r}{\partial z} + \frac{\partial\Phi}{\partial\phi} \frac{\partial\phi}{\partial z} \quad (38)$$

$$\frac{\partial\Phi(r, \phi)}{\partial r} = -\frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^2 (3\cos^2\phi - 1) \quad \frac{\partial\Phi(r, \phi)}{\partial\phi} = -\frac{3}{2} J_2 \frac{\mu}{r} \left(\frac{R}{r}\right)^2 \sin\phi \cdot \cos\phi \quad (39)$$

By differentiating Equation (23), the partial derivatives of  $\phi$  with respect to each of the components can be obtained.

$$\frac{\partial\phi}{\partial x} = \frac{xz}{r^3 \sin\phi} \quad \frac{\partial\phi}{\partial y} = \frac{yz}{r^3 \sin\phi} \quad \frac{\partial\phi}{\partial z} = -\frac{\sin\phi}{r} \quad (40)$$

According to Equation (37) and making use of the expression  $\cos \phi = z/r$ , the components of the perturbing acceleration vector due to the Earth's zonal harmonic  $J_2$  are obtained in Equation (41), (42) and (43). These terms, added to the fundamental equation of relative motion, constitute the expression for the acceleration vector that must be numerically integrated in order to find the state vector of the orbiting body, introducing a higher degree of accuracy, which is the main goal to be achieved with this study.

$$p_{J_2,x} = -\frac{\partial \Phi(r, \phi)}{\partial x} = \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^2 \frac{x}{r} \left(5\left(\frac{z}{r}\right)^2 - 1\right) \quad (41)$$

$$p_{J_2,y} = -\frac{\partial \Phi(r, \phi)}{\partial y} = \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^2 \frac{y}{r} \left(5\left(\frac{z}{r}\right)^2 - 1\right) \quad (42)$$

$$p_{J_2,z} = -\frac{\partial \Phi(r, \phi)}{\partial z} = \frac{3}{2} J_2 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^2 \frac{z}{r} \left(5\left(\frac{z}{r}\right)^2 - 3\right) \quad (43)$$

By following the exact same procedure, but this time taking the expressions for the perturbing accelerations due to the Earth zonal harmonics from  $J_3$  to  $J_6$ , even though they are virtually negligible, as well as their corresponding Legendre polynomials, the three components of each of the perturbing accelerations due to the first zonal harmonics are shown [13] from Equation (44) to (55).

$$p_{J_3,x} = \frac{1}{2} J_3 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^3 \frac{x}{r} \left(7\left(\frac{z}{r}\right)^3 - 3\left(\frac{z}{r}\right)\right) \quad (44)$$

$$p_{J_3,y} = \frac{1}{2} J_3 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^3 \frac{y}{r} \left(7\left(\frac{z}{r}\right)^3 - 3\left(\frac{z}{r}\right)\right) \quad (45)$$

$$p_{J_3,z} = \frac{1}{2} J_3 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^3 \frac{z}{r} \left(1 - 10\left(\frac{z}{r}\right)^2 + \frac{35}{3}\left(\frac{z}{r}\right)^4\right) \quad (46)$$

$$p_{J_4,x} = \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^4 \frac{x}{r} \left(3 - 42\left(\frac{z}{r}\right)^2 + 63\left(\frac{z}{r}\right)^4\right) \quad (47)$$

$$p_{J_4,y} = \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^4 \frac{y}{r} \left(3 - 42\left(\frac{z}{r}\right)^2 + 63\left(\frac{z}{r}\right)^4\right) \quad (48)$$

$$p_{J_4,z} = \frac{5}{8} J_4 \frac{\mu}{r^2} \left(\frac{R}{r}\right)^4 \frac{z}{r} \left(15 - 70\left(\frac{z}{r}\right)^2 + 63\left(\frac{z}{r}\right)^4\right) \quad (49)$$

$$p_{J_5,x} = \frac{J_5}{8} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^5 \frac{x}{r} \left(35\left(\frac{z}{r}\right) - 210\left(\frac{z}{r}\right)^3 + 231\left(\frac{z}{r}\right)^5\right) \quad (50)$$

$$p_{J_5,y} = \frac{J_5}{8} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^5 \frac{y}{r} \left(35\left(\frac{z}{r}\right) - 210\left(\frac{z}{r}\right)^3 + 231\left(\frac{z}{r}\right)^5\right) \quad (51)$$

$$p_{J_5,z} = \frac{J_5}{8} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^5 \frac{z}{r} \left(693\left(\frac{z}{r}\right)^6 - 945\left(\frac{z}{r}\right)^4 + 315\left(\frac{z}{r}\right)^2 - 15\right) \quad (52)$$

$$p_{J_6,x} = -\frac{J_6}{16} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^6 \frac{x}{r} \left(35 - 945\left(\frac{z}{r}\right)^2 + 3465\left(\frac{z}{r}\right)^4 - 3003\left(\frac{z}{r}\right)^6\right) \quad (53)$$

$$p_{J_6,y} = -\frac{J_6}{16} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^6 \frac{y}{r} \left(35 - 945\left(\frac{z}{r}\right)^2 + 3465\left(\frac{z}{r}\right)^4 - 3003\left(\frac{z}{r}\right)^6\right) \quad (54)$$

$$p_{J_6,z} = -\frac{J_6}{16} \frac{\mu}{r^2} \left(\frac{R}{r}\right)^6 \frac{z}{r} \left(245 - 2205\left(\frac{z}{r}\right)^2 + 4851\left(\frac{z}{r}\right)^4 - 3003\left(\frac{z}{r}\right)^6\right) \quad (55)$$

If all these perturbing accelerations are added to the differential equation to be integrated, the resulting values for the state vector of the secondary body will definitely be more precise and accurate, as it will be seen later on for LAGEOS-1 and the ISS.

### 3.1.2 Atmospheric Drag Perturbation

The kind of perturbing acceleration presented in this section refers to the deviation of the satellite from its orbit, as well as the aerodynamic heating caused due to the drag forces exerted by the atmosphere at relatively low altitudes from the Earth surface.

The limit altitude above the surface at which the terrestrial atmosphere starts being regarded as space is around 60 miles or 100 kilometers. Even though most of the Earth atmosphere is located within the first 100 km, from that point up to approximately 1000 kilometers above the surface, the air density is still enough to deorbit the body. Therefore, for satellites located in LEO orbits, which are usually placed between 150 and 1000 km, the atmospheric drag perturbation is considerably noticeable.

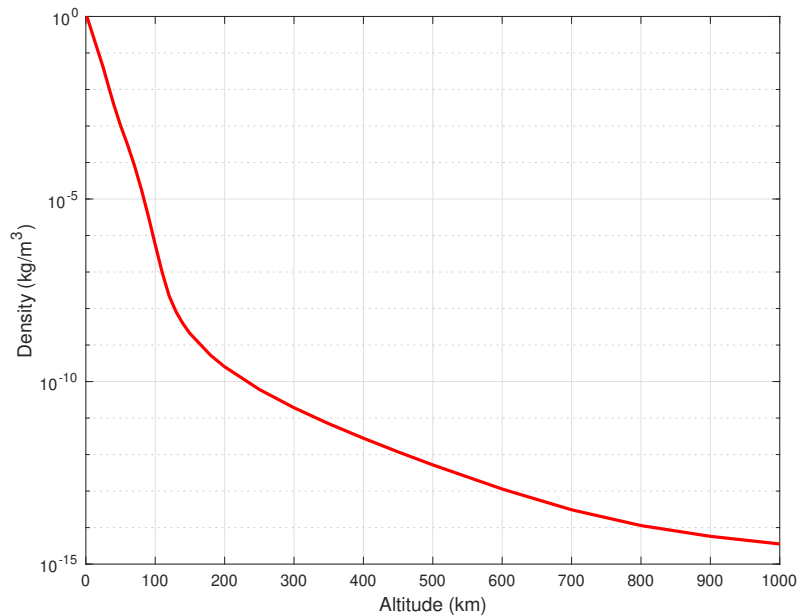
Inside the range of altitudes where this perturbing acceleration appears, the effect on the orbiting body results in an increase of aerodynamic heating due to drag, which is added to the decrease in speed of the satellite and its consequent loss of altitude. The density of the air, and thus the effect of aerodynamic drag, keeps diminishing with height up to an altitude of around 1000 km, above which the magnitude of the perturbation becomes negligible, since it is  $10^{-10}$  times the acceleration of the satellite.

There exist several standard atmosphere models that allow to depict the variation of density with altitude over the Earth surface, but according to [5], the one that is mostly used in the field of orbital mechanics is usually the *US Standard Atmosphere 1976*, or most commonly named *USSA76*, defined in function *atmosphere.m*, shown in Appendix A. This model represents the density variation by means of an exponential function of  $\rho(z)$ , which is given by Equation (56). Moreover, the parameter  $H_i$  is defined in Equation (57).

$$\rho(z) = \rho_i e^{-\frac{z-z_i}{H_i}} \quad (56)$$

$$H_i = -\frac{z_{i+1} - z_i}{\ln\left(\frac{\rho_{i+1}}{\rho_i}\right)} \quad (57)$$

In Figure 17, it can be appreciated how the value of density decreases as altitude increases, for each of the values of height included in the range affected by the atmospheric drag perturbation.



**Figure 17:** Evolution of density with altitude [5]

It is worth noticing the fact that those satellites or stations located in LEO orbits, such as the ISS, which orbits at around 400 km of altitude above the Earth surface, are still affected by atmospheric drag, since density at such altitudes presents non-negligible values. In the example of the International Space Station, as it has already been commented, several supply vehicles are required every now and then in order to provide thrust and compensate for the reduction in speed and altitude experienced due to drag.

Regarding the way to find an expression for the perturbing acceleration caused by atmospheric drag, it is interesting to notice that the  $\vec{p}$  term is actually the drag force divided by the mass of the orbiting body. The drag magnitude is given by Equation (58), where  $\rho(z)$  is the density, being its expression the aforementioned exponential function of the altitude. Moreover,  $\vec{v}_{rel}$  is the velocity of the satellite relative to the atmosphere.  $A$  is the cross-sectional surface of the satellite, perpendicular to the direction of the relative velocity vector, and finally,  $C_D$  is the drag coefficient, which has no dimensions.

$$D = \frac{1}{2}\rho(z)v_{rel}^2AC_D \quad (58)$$

The direction of the drag vector is opposite to that of the relative velocity of the body, since the drag force acts as a brake for the satellite in orbit, in such a way that  $\vec{D} = -D\vec{u}_{v_{rel}}$ . Since  $\vec{p}_{Drag} = \vec{D}/m_2$ , then Equation (59) is achieved.

$$\vec{p}_{Drag} = -\frac{1}{2m_2}\rho(z)v_{rel}^2AC_D\frac{\vec{v}_{rel}}{v_{rel}} = -\frac{1}{2}\rho(z)v_{rel}\frac{AC_D}{m_2}\vec{v}_{rel} \quad (59)$$

This last expression is the one to be added to the term in the right-hand side of the fundamental equation of relative motion in order to numerically integrate it and find the state vector of the satellite for certain discrete values of time. The MATLAB<sup>™</sup> function in which this is implemented is the aforementioned function *rates.m*.

It is worth commenting that the parameter given by  $\frac{AC_D}{m_2}$  in Equation (59) is usually represented as  $B$ , and named *ballistic coefficient*. In addition, the relative velocity of the satellite is given by Equation (60), in which  $\vec{v}$  is the inertial velocity of the body and  $\vec{v}_{atm}$  is the atmosphere velocity. At the same time, this atmospheric speed is calculated as shown, by means of the cross product of the angular speed of the atmosphere, assuming its circular motion is in concordance with the Earth ( $\vec{\omega}_E = 7.2921159 \cdot 10^{-5}\vec{k}$ ), times the inertial or absolute position vector of the satellite.

$$\vec{v}_{rel} = \vec{v} - \vec{v}_{atm} = \vec{v} - \vec{\omega}_E \times \vec{r} \quad (60)$$

### 3.1.3 Relativistic Effects

When it comes to determining the orbital parameters and state vector of a satellite travelling around the Earth, it is worth accounting for the fact that these satellites are usually equipped with caesium atomic clocks as well as frequency sources, thanks to which they are able to take precise time measurements in order to ensure accurate navigation.

Nonetheless, time is not absolute, but it depends on the frame of reference of the observer. The theory of relativity states that two spatially separated simultaneous events, as seen from different reference systems, are not absolutely simultaneous: the simultaneity is relative. Hence, the clock of a satellite placed hundreds of kilometers above the Earth surface will not perform the same time measurement as a clock located on the surface, since the altitude difference and the gravity field difference preclude time synchronization. Furthermore, when talking about the Earth, and according to relativity principles, time cannot be synchronized in a rotational frame. An acceptable solution is the synchronization of the clocks in a non-rotating reference frame [14].

That being said, the fact that neither the Earth nor the satellite are inertial reference frames, but they are in constant movement relative to each other, is the main cause of the appearance of relativistic effects, which should be accounted for in the calculations so that the results obtained are as accurate as possible. These relativistic effects, among others, are the time differences caused by gravity, the

time dilation and the acceleration of the satellite [14]. This acceleration can indeed be regarded as a perturbation, in the same way as it has been done for gravitational and atmospheric drag perturbations previously. According to Beutler (1991) [15], the components for the perturbing acceleration vector due to relativistic effects are given by Equation (61) to (63), where  $c$  is the speed of light, and its value is 299792458 m/s.

$$p_{Rel,x} = -3 \frac{\mu^2}{c^2 r^3} x \quad (61)$$

$$p_{Rel,y} = -3 \frac{\mu^2}{c^2 r^3} y \quad (62)$$

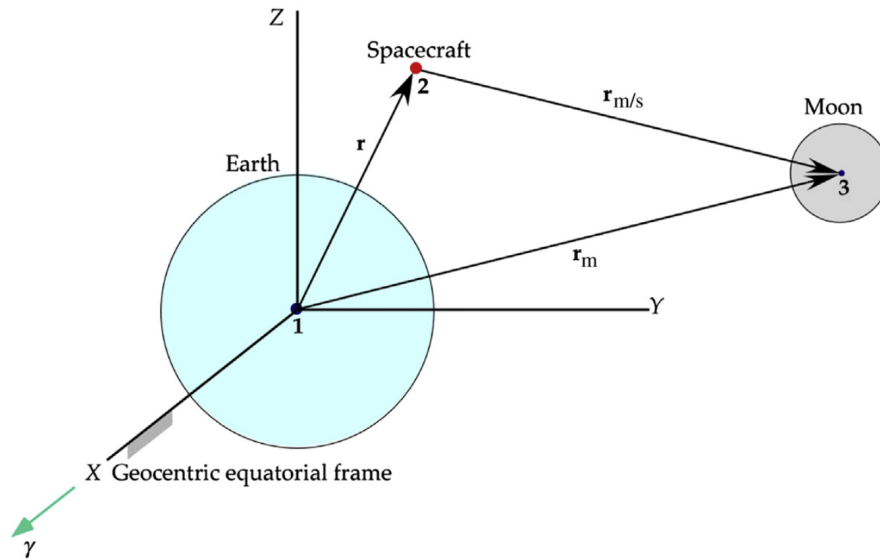
$$p_{Rel,z} = -3 \frac{\mu^2}{c^2 r^3} z \quad (63)$$

It is worth mentioning that the order of magnitude of this perturbing acceleration is around  $3 \cdot 10^{-10} \text{ m/s}^2$ , and therefore the effect of introducing this perturbation over the results obtained may be hardly noticeable [15].

### 3.1.4 Lunar Gravity Perturbation

This section focuses on obtaining an expression for the perturbing acceleration which results from including a third body in the problem, in this case the Moon, whose gravitational field influences the calculation of the orbit of a satellite around the Earth.

The scheme of the situation is represented in Figure 18, where vector  $\vec{r}$  represents the position of the satellite with respect to the Earth,  $\vec{r}_{moon}$  refers to the position of the Moon as seen from the Earth, and finally,  $\vec{r}_{moon/s}$  corresponds to the position of the Moon as seen from the satellite.



**Figure 18:** Earth-centered reference frame including lunar gravity perturbation [5]

The mathematical development begins by considering the addition of this third body in the acceleration of the second one with respect to the main body, as seen in Equation (64).

$$\ddot{\vec{r}} = \vec{a}_{2/1} = \vec{a}_2 - \vec{a}_1 = \left( -Gm_1 \frac{\vec{r}}{r^3} + Gm_3 \frac{\vec{r}_{moon/s}}{r_{moon/s}^3} \right) - \left( Gm_2 \frac{\vec{r}}{r^3} + Gm_3 \frac{\vec{r}_{moon}}{r_{moon}^3} \right) \quad (64)$$

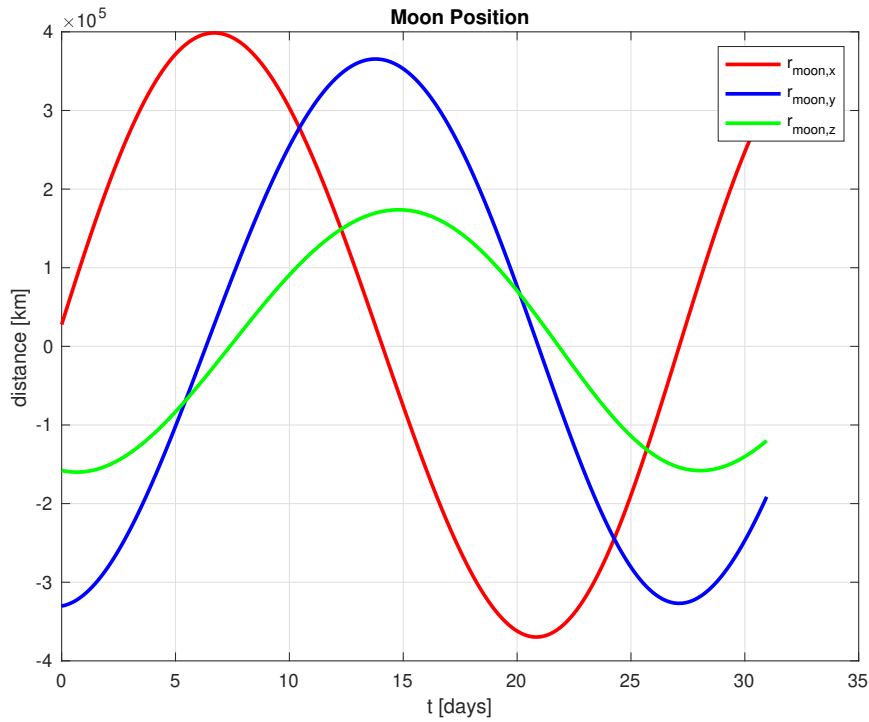
Knowing that  $\mu = G(m_1 + m_2)$  and considering  $\mu_{moon} = Gm_3$ , Equation (65) is obtained by reorganizing the previous expression. It is worth noticing that the second term is actually the perturbing

acceleration due to the effect of the Moon, which is added to the right-hand side of the fundamental equation of relative motion.

$$\ddot{\vec{r}} = -\mu \frac{\vec{r}}{r^3} + \mu_{moon} \left( \frac{\vec{r}_{moon/s}}{r_{moon/s}^3} - \frac{\vec{r}_{moon}}{r_{moon}^3} \right) = -\mu \frac{\vec{r}}{r^3} + \vec{p}_{moon} \quad (65)$$

In order to introduce the expression of  $\vec{p}_{moon}$  into the mathematical robot for integrating, every vector and coefficient must be previously defined. To begin with,  $\mu_{moon}$  is the gravitational parameter of the Moon, which has a value of  $4902.800066 \text{ km}^3/\text{s}^2$ . Then, only  $\vec{r}_{moon}$  and  $\vec{r}_{moon/s}$  are left to be defined.

Concerning vector  $\vec{r}_{moon}$ , the position of the Moon has been obtained from the Horizons web interface for every hour inside the month of May, 2021. Then, in order to find an expression as a function of time which can be inserted in the differential equation to be integrated, an interpolating polynomial of degree 20 has been created for each of the components, x, y and z, of vector  $\vec{r}_{moon}$ , as a function of vector  $\vec{t}_{span}$ . The degree of the polynomial has been selected as a trade-off between accuracy of results and computational time. The shape of these polynomials can be observed in Figure 19.



**Figure 19:** Plot of the resulting interpolating polynomials for the position of the Moon

Once the coordinates of  $\vec{r}_{moon}$  are obtained as a function of time, the position of the Moon as seen from the satellite,  $\vec{r}_{moon/s}$ , is trivial to obtain. It can be readily computed by calculating the difference between vector  $\vec{r}_{moon}$  and vector  $\vec{r}$ , as seen in Equation (66).

$$r_{moon/s,x} = r_{moon,x} - x \quad r_{moon/s,y} = r_{moon,y} - y \quad r_{moon/s,z} = r_{moon,z} - z \quad (66)$$

Finally, the resulting expressions regarding the lunar gravity perturbation are shown explicitly from Equation (67) to (69).

$$p_{Moon,x} = \mu_{moon} \left( \frac{r_{moon/s,x}}{r_{moon/s}^3} - \frac{r_{moon,x}}{r_{moon}^3} \right) \quad (67)$$



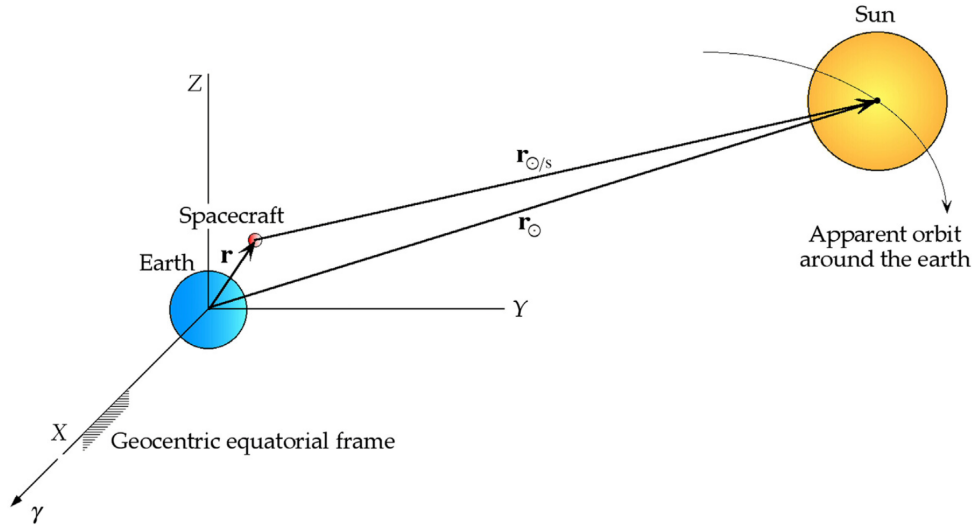
$$p_{Moon,y} = \mu_{moon} \left( \frac{r_{moon/s,y}}{r_{moon/s}^3} - \frac{r_{moon,y}}{r_{moon}^3} \right) \quad (68)$$

$$p_{Moon,z} = \mu_{moon} \left( \frac{r_{moon/s,z}}{r_{moon/s}^3} - \frac{r_{moon,z}}{r_{moon}^3} \right) \quad (69)$$

It is worth adding that, if  $a_0$  is the satellite acceleration due to the Earth's gravitational attraction, then the perturbing acceleration due to lunar gravity is around  $10^{-7} \cdot a_0$  [5], and thus the solution obtained when introducing it will not change much with respect to the results obtained for the case without perturbations.

### 3.1.5 Solar Gravity Perturbation

With the aim of accounting for the gravity of the Sun in the determination of a satellite orbit around the Earth, the procedure is exactly the same as the one followed when introducing lunar gravity. The scheme of the situation now, considering the apparent trajectory of the Sun around the Earth, is shown in Figure 20.



**Figure 20:** Earth-centered reference frame including solar gravity perturbation [5]

The gravitational parameter of the Sun,  $\mu_{sun}$ , has a value of  $132712440041.93938 \text{ km}^3/\text{s}^2$ . In this case, vectors  $\vec{r}_{moon}$  and  $\vec{r}_{moon/s}$  are replaced by  $\vec{r}_{sun}$  and  $\vec{r}_{sun/s}$ . The former is obtained by interpolating the data extracted from Horizons into a 20<sup>th</sup> degree polynomial for each coordinate, whereas the latter follows Equation (66). The shape of these polynomials are given by Figure 21.

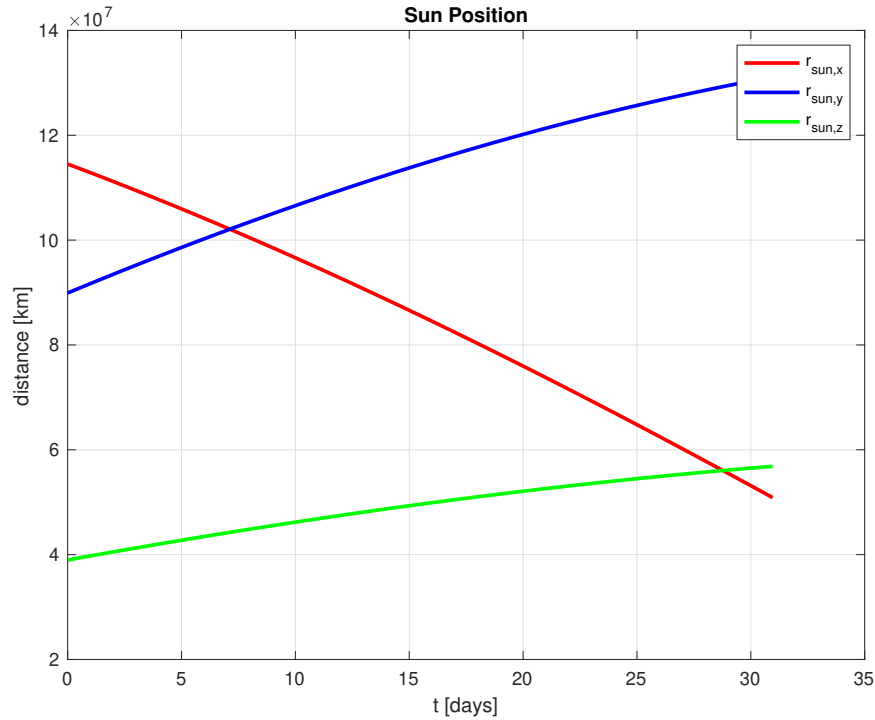
In the end, the expressions resulting from carrying out the same procedure as in the case of the Moon, but with the data of the Sun, are shown from Equation (70) to (72).

$$p_{Sun,x} = \mu_{sun} \left( \frac{r_{sun/s,x}}{r_{sun/s}^3} - \frac{r_{sun,x}}{r_{sun}^3} \right) \quad (70)$$

$$p_{Sun,y} = \mu_{sun} \left( \frac{r_{sun/s,y}}{r_{sun/s}^3} - \frac{r_{sun,y}}{r_{sun}^3} \right) \quad (71)$$

$$p_{Sun,z} = \mu_{sun} \left( \frac{r_{sun/s,z}}{r_{sun/s}^3} - \frac{r_{sun,z}}{r_{sun}^3} \right) \quad (72)$$

Furthermore, this perturbing acceleration is also seven orders of magnitude lower than the acceleration obtained for the two-body problem with no perturbations included [5].



**Figure 21:** Plot of the resulting interpolating polynomials for the position of the Sun

### 3.1.6 Perturbation due to Solar System Planets

The aim of this section is to increase, if possible, the accuracy of the results of the numerical integration in order to reduce the state vector error to the extent possible. For this purpose, the gravity of all the other planets in the solar system is also accounted for in the shape of perturbing accelerations.

The procedure is, again, exactly the same as the one for the Moon and the Sun, but substituting the third body by one of the planets each time. The only coefficients needed are the gravitational parameter of each planet, and the expression of the perturbing acceleration to be introduced in each case is similar to Equation (65). It is worth mentioning that Pluto, even though it is not considered to be a planet anymore, has been taken into account as well for more precision.

- $\mu_{mercury} = 22031.86855 \text{ km}^3/\text{s}^2$
- $\mu_{venus} = 324858.592 \text{ km}^3/\text{s}^2$
- $\mu_{mars} = 42828.375214 \text{ km}^3/\text{s}^2$
- $\mu_{jupiter} = 126686531.900 \text{ km}^3/\text{s}^2$
- $\mu_{saturn} = 37931206.159 \text{ km}^3/\text{s}^2$
- $\mu_{uranus} = 5793951.322 \text{ km}^3/\text{s}^2$
- $\mu_{neptune} = 6835099.97 \text{ km}^3/\text{s}^2$
- $\mu_{pluto} = 869.96 \text{ km}^3/\text{s}^2$

The data concerning the position of each planet with respect to the Earth for each time instant during the month of May, 2021 are extracted from Horizons interface, and the interpolating polynomials of degree 20 are constructed for each coordinate, x, y and z, of the position vector of each planet in the solar system. The evolution of each coordinate for each planet is shown from Figure 22 to 29.

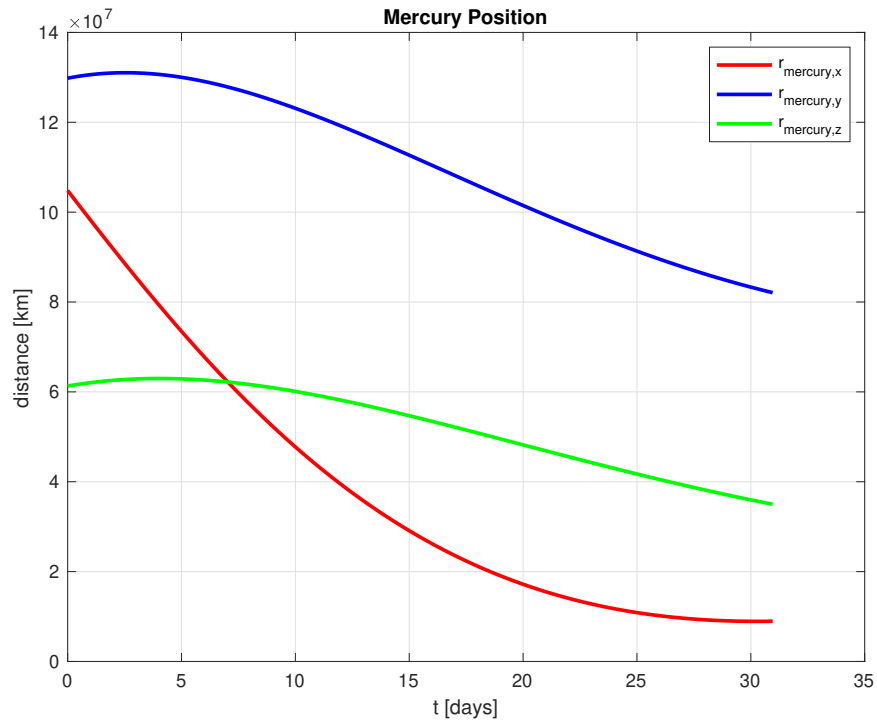


Figure 22: Plot of the resulting interpolating polynomials for the position of Mercury

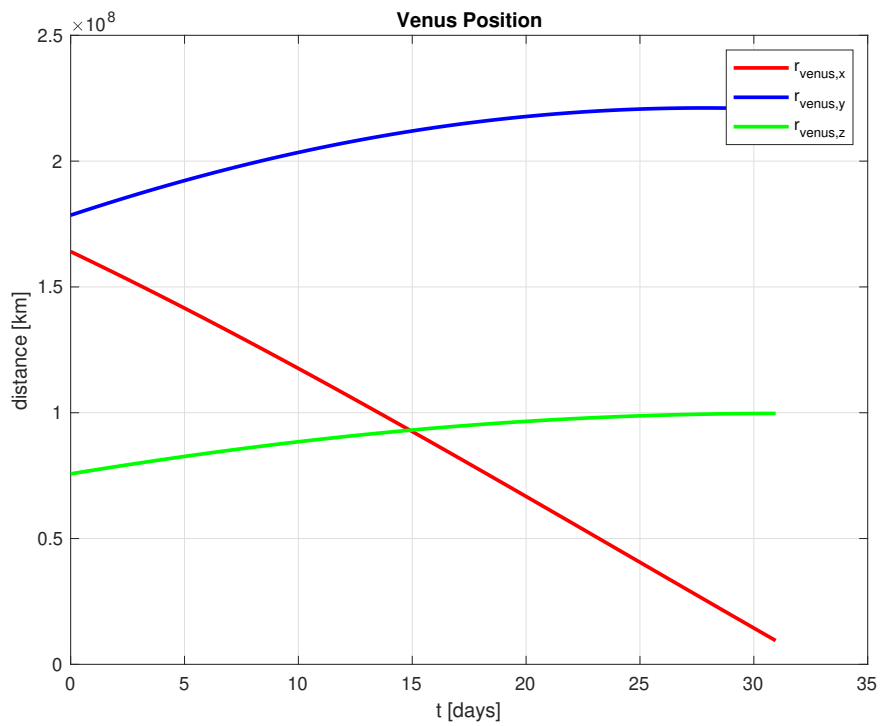


Figure 23: Plot of the resulting interpolating polynomials for the position of Venus

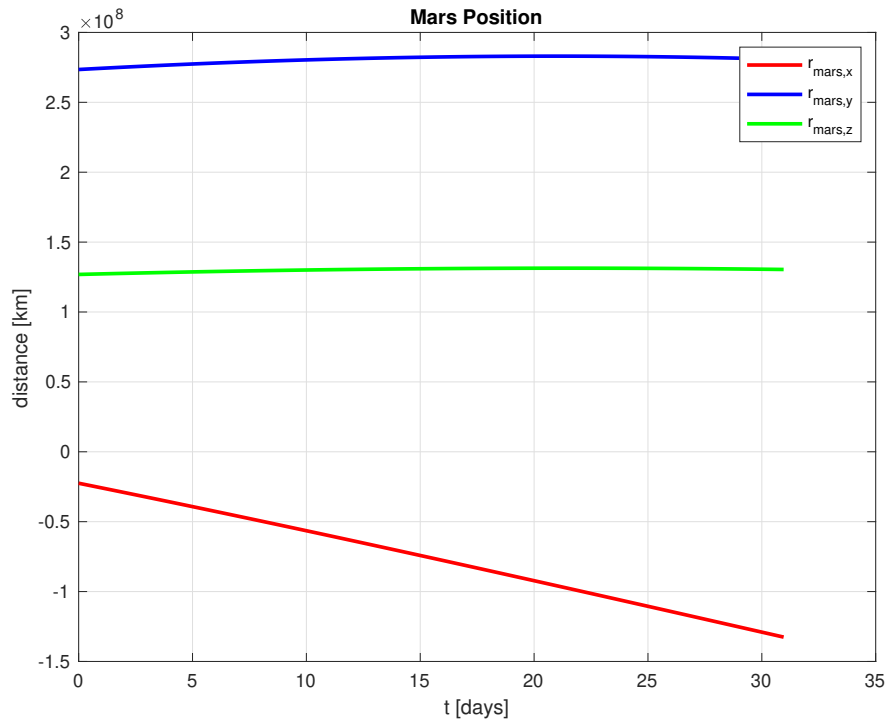


Figure 24: Plot of the resulting interpolating polynomials for the position of Mars

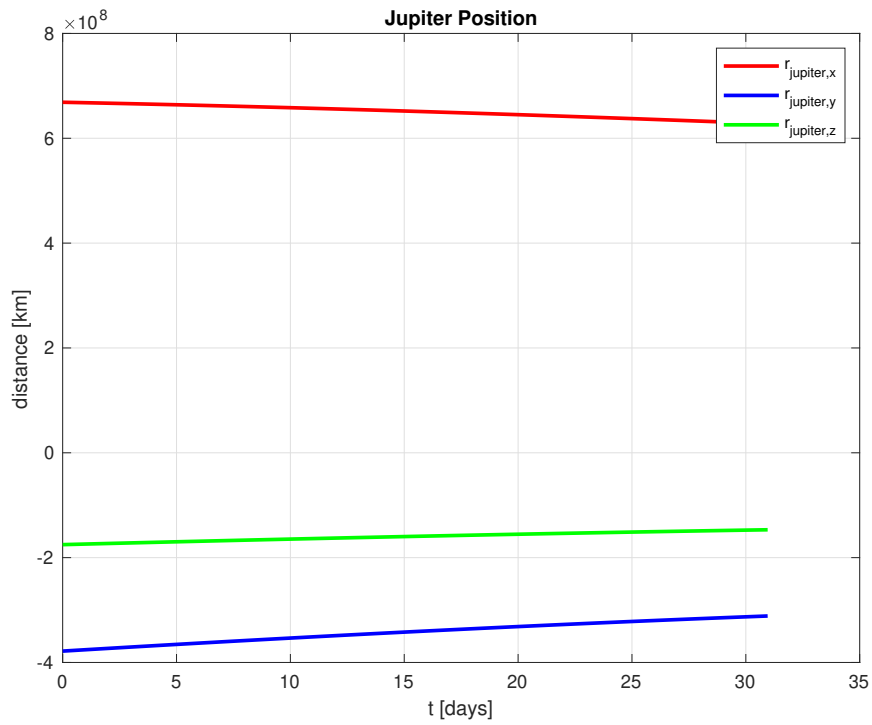


Figure 25: Plot of the resulting interpolating polynomials for the position of Jupiter

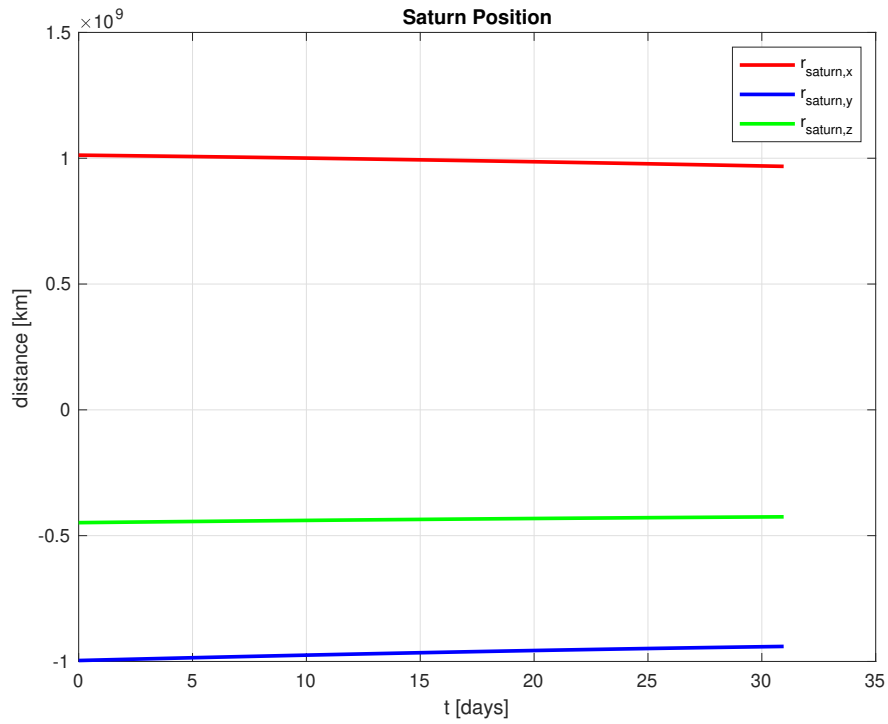


Figure 26: Plot of the resulting interpolating polynomials for the position of Saturn

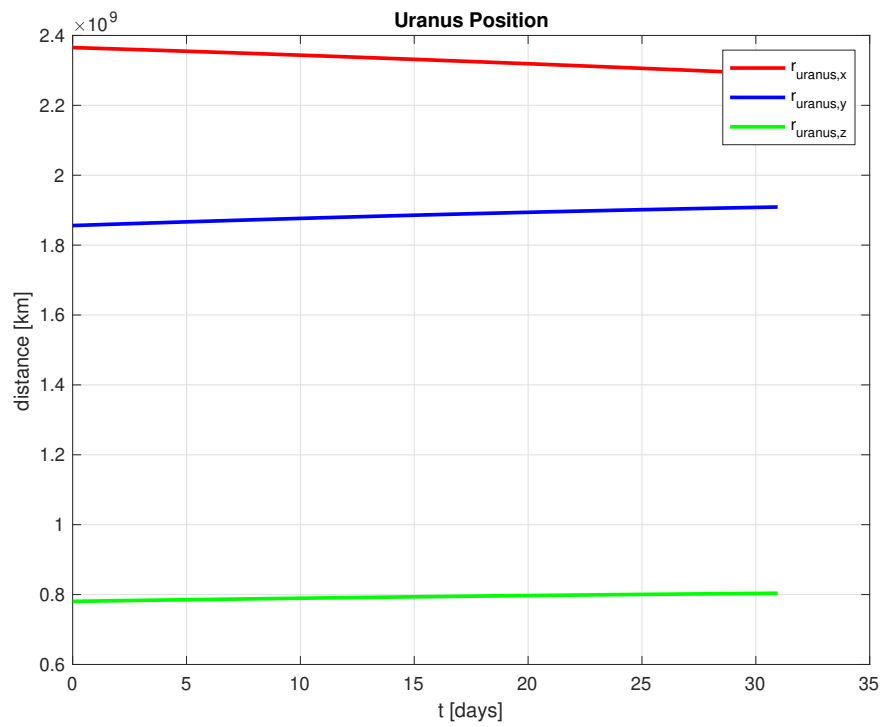
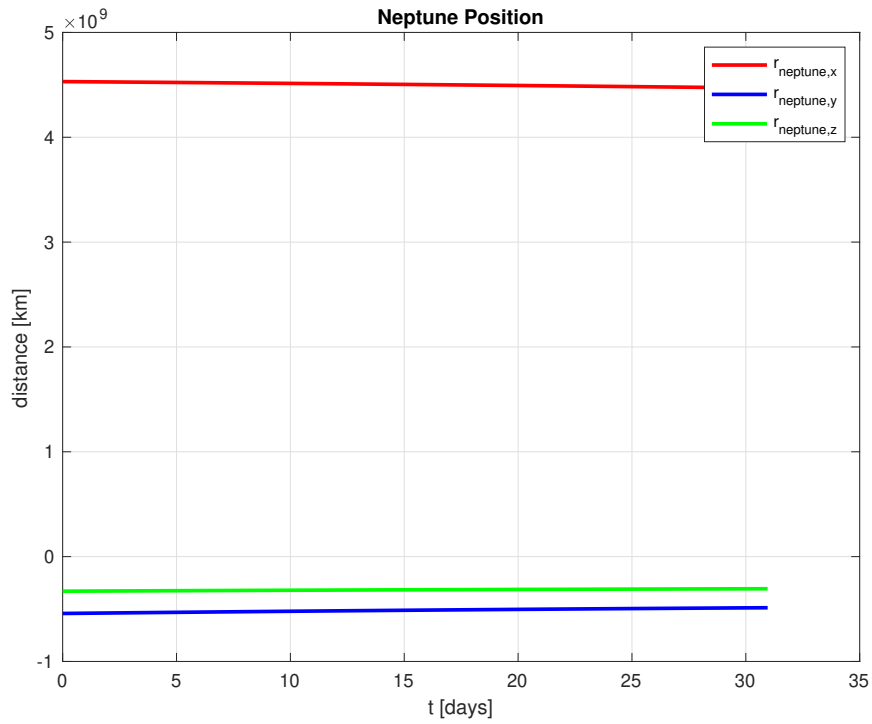
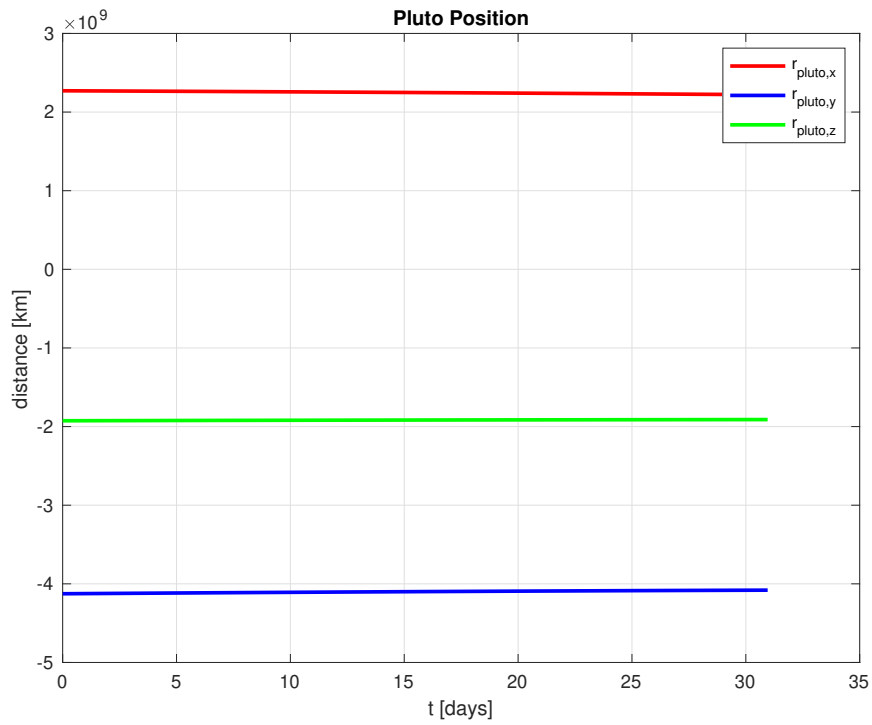


Figure 27: Plot of the resulting interpolating polynomials for the position of Uranus



**Figure 28:** Plot of the resulting interpolating polynomials for the position of Neptune



**Figure 29:** Plot of the resulting interpolating polynomials for the position of Pluto

### 3.1.7 Solar Radiation Pressure

In this section, one last perturbing acceleration is defined and implemented so that the solution to the two-body problem can be even more accurate, if possible.

The Solar Radiation Pressure is based on the fact that the Sun is a large body that is constantly emitting photons, which do not have mass, but they do carry some energy and momentum, instead. These photons are emitted from the surface of the Sun, which is considered to be a black body and is known as the *photosphere*. In order to calculate the radiated power intensity at this surface, the Stefan-Boltzmann law may be applied, according to which the intensity follows Equation (73), where  $\sigma$  is the Stefan-Boltzmann constant and  $T_p$  is the effective temperature of the photosphere, which has a value of 5777 K.

$$S_0 = \sigma T_p^4 = 5.67 \cdot 10^{-8} \cdot 5777^4 = 63.15 \cdot 10^6 \text{ W/m}^2 \quad (73)$$

Knowing that the intensity times the square of the radius is a constant parameter, then the intensity of solar radiation at the orbit of the Earth can be readily determined, since the radius of the Sun and the mean distance from Sun to Earth are known values. This parameter is indeed the energy flux carried away by photons, and it is represented by  $S_{E-S}$ . It is also known as *solar constant* [5].

$$S_{E-S} \cdot R_{E-S}^2 = S_0 \cdot R_0^2 \longrightarrow S_{E-S} = S_0 \left( \frac{R_0}{R_{E-S}} \right)^2 \quad (74)$$

$$S_{E-S} = 63.15 \cdot 10^6 \left( \frac{696000}{149.6 \cdot 10^6} \right)^2 = 1367 \text{ W/m}^2 \quad (75)$$

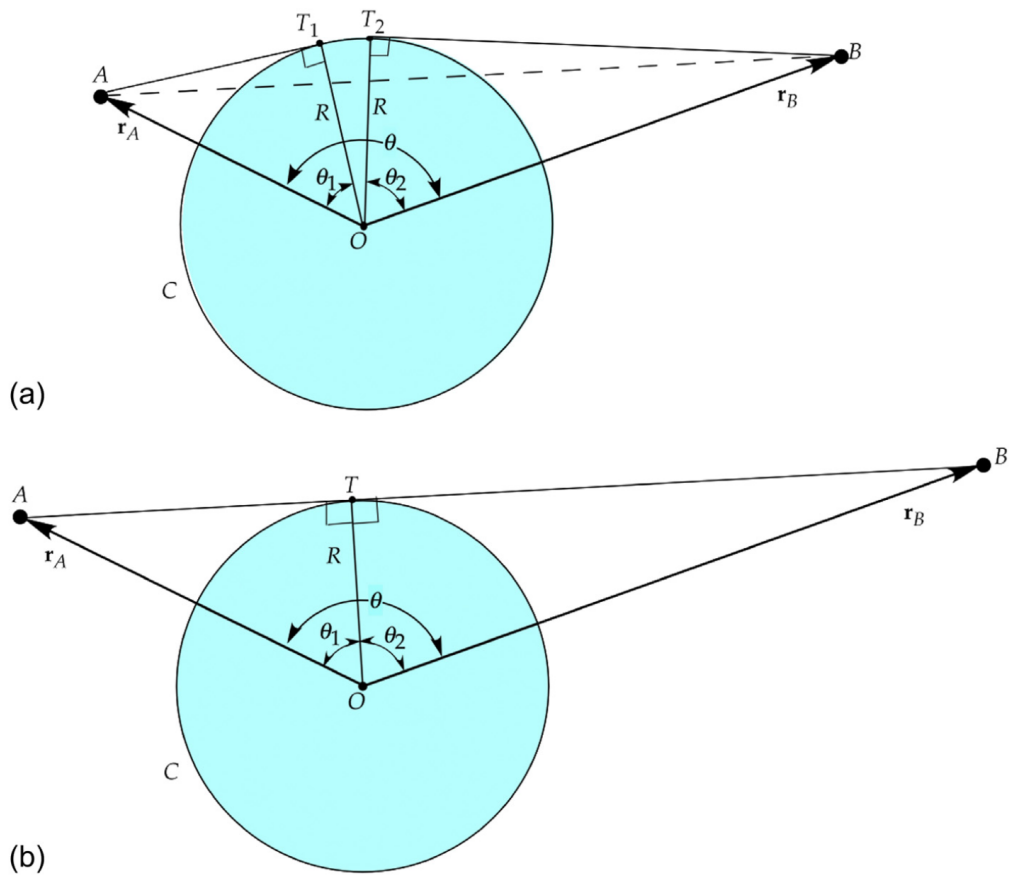
In order to obtain the solar radiation pressure constant, or momentum flux, the energy flux must be divided by the speed of light, as shown in Equation (76). It can be appreciated how negligible the radiation pressure value due to the Sun is, when compared to the pressure at sea level, 101325 Pa.

$$P_{SR} = \frac{S_{E-S}}{c} = 4.56 \cdot 10^{-6} \text{ N/m}^2 = 4.56 \text{ } \mu\text{Pa} \quad (76)$$

Moving on to the expression for the radiation pressure perturbing acceleration, it comes given by Equation (77), where  $\nu$  is the shadow function, which should return 0 if the satellite is in shadow, and 1 if it is reached by sunlight. Then,  $C_R$  is the radiation pressure coefficient, which has a value between 1 and 2: 1 for black bodies, which absorb radiation, and 2 for reflecting bodies, since they double the pressure exerted over the satellite. The coefficient  $A_S$  refers to the cross-sectional absorbing area of the satellite, which is reached by photons. Finally,  $m_2$  is the mass of the satellite under study, and  $\vec{u}_{r_{sun}}$  is the unit vector corresponding to the position of the Sun with respect to the Earth. This is considered as an approximation, since actually that vector should be the unit vector pointing from the satellite towards the Sun, but the distance Earth-satellite is negligible compared to the distance Earth-Sun, and the angle between both vectors is minuscule. Therefore, by introducing this hypothesis, the results obtained are still accurate. It is worth mentioning that the perturbing acceleration is negative because of the fact that the force exerted by photons is directed towards the satellite, moving away from the Sun [5].

$$\vec{p}_{SRP} = -\nu \frac{S_{E-S} C_R A_S}{c m_2} \vec{u}_{r_{sun}} \quad (77)$$

To finish off, it is interesting to include the procedure to obtain the shadow function for each time instant during the whole month for which the solution is calculated. Figure 30 depicts a draft of how the problem looks like, including the vectors and relevant angles. The method consists of obtaining, in the first place, the angle between vectors  $\vec{r}$  and  $\vec{r}_{sun}$ , which is angle  $\theta$  in Figure 30, by applying the dot product expression. Then,  $\theta_1$  corresponds to the angle between vector  $\vec{r}$  and a line drawn from the satellite tangent to the Earth. Similarly,  $\theta_2$  is the angle comprised between vector  $\vec{r}_{sun}$  and a tangent drawn from the Sun to the Earth.



**Figure 30:** Scheme of the situation for obtaining the shadow function [5]

The algorithm that allows to find the shadow function [11] consists of adding the values of angles  $\theta_1$  and  $\theta_2$ , and comparing the result to the value of angle  $\theta$ . If  $\theta$  is higher or equal, then the satellite is in shadow, and  $\nu$  has a value of 0, cancelling the effect of SRP in that particular instant. Otherwise,  $\nu$  equals 1 and the satellite is reached by sunlight, considering the effect of SRP is active in such cases.

This perturbing acceleration, though, is 9 orders of magnitude lower than the non-perturbed acceleration [5], the one that results from solving the two-body problem without accounting for perturbations. Hence, any change in the results will definitely be hardly noticeable.

### 3.2 Approach to real cases

Following the development of the cases of application of the two-body problem presented in the previous section, which were satellite LAGEOS-1 and the ISS, the purpose of the present section is to keep analyzing both cases by adding each one of the perturbations presented and studied throughout this paper, with the aim of appreciating the evolution and improvement of the results, by comparing the state vector error plots obtained for each case. In the end, it will be observed how the addition of each perturbation affects the results in a different way and magnitude, and some relevant conclusions will be extracted.

The required coefficients will be defined in each case, and the code in which the expressions, functions and differential equations are defined appears in Appendix A. It is worth commenting that the graphs obtained in each case correspond to the accumulation of all of the perturbations included up to such case, in such a way that the evolution of the results and the reduction of the error can be progressively appreciated.



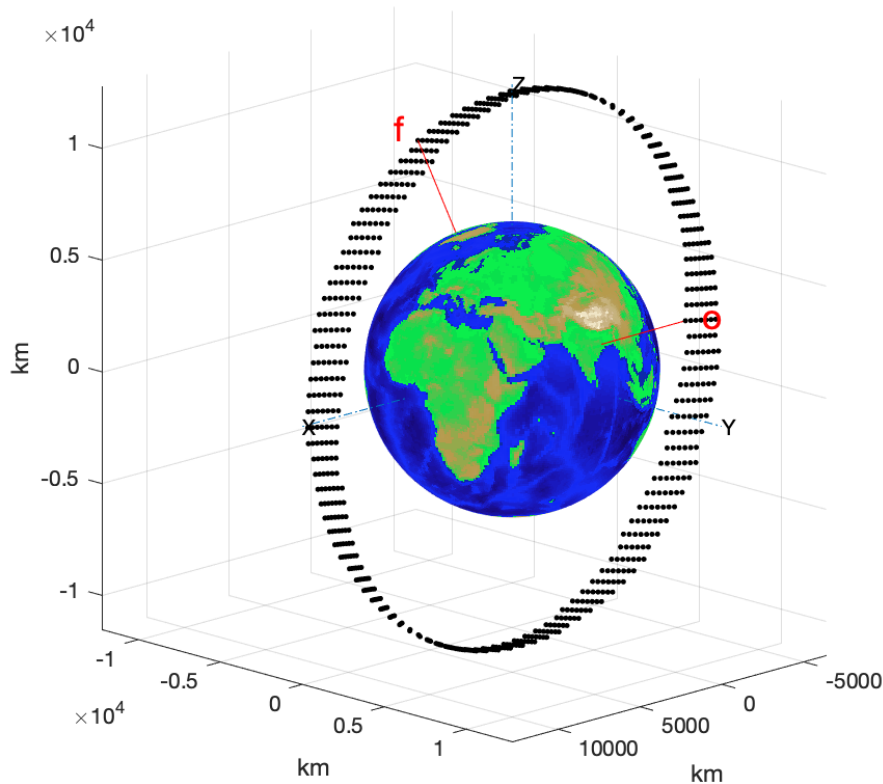
### 3.2.1 LAGEOS-1 orbit

#### 1. Gravitational Perturbations

In this subsection, the perturbing accelerations due to the Earth's zonal harmonics from  $J_2$  to  $J_6$  are added to the differential equation of relative motion, which appears in function *rates.m*, and the results obtained are shown in Figure 31 to 33.

To begin with, it is worth remembering that, according to Figure 13,  $J_2$  perturbation is the one that most influences the results after the Earth's gravity. Therefore, a great error reduction must be appreciated in this case.

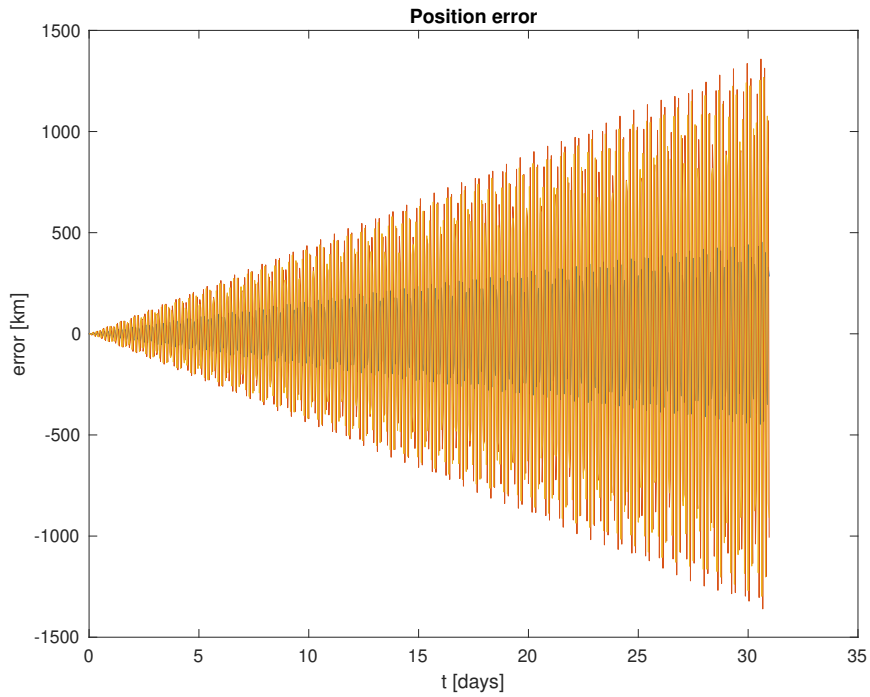
By comparing the trajectory of the satellite with the one obtained without perturbations, it is clearly observed how, in this case, the path along one month is not in the same orbital plane as it was before. The points plotted in Figure 31 seem to define orbits in quite different planes, approaching the real trajectory followed by LAGEOS-1 around the Earth, in contrast with Figure 10, in which all points within the trajectory define an orbit located in a single plane.



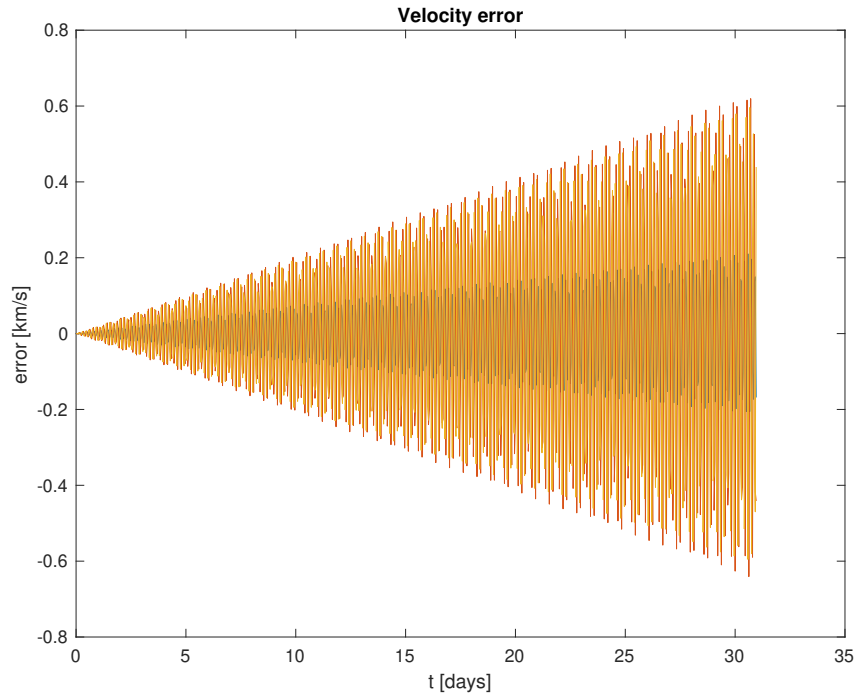
**Figure 31:** LAGEOS-1 trajectory during May 2021 introducing gravitational perturbations

Regarding the error of the satellite's position and velocity with respect to the real state vectors extracted from Horizons, the position error gets reduced from around 4000 km at the end of the month to around 1500 km. The same happens for speed, which goes from a maximum of 1500 m/s to a maximum of around 600 m/s. It can be observed how the triangular shape in the position and velocity error graphs is maintained, due to the error accumulation with time.

Summing up, it can be stated that gravitational perturbations are the ones which have a greater influence on the results, as expected, due to the fact that the error has been reduced to less than half the one obtained for the case without perturbations.



**Figure 32:** LAGEOS-1 absolute position error during May 2021 introducing gravitational perturbations



**Figure 33:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational perturbations

## 2. Atmospheric Drag Perturbation

When introducing atmospheric drag's perturbing acceleration, according to Equation (59), there are some parameters that need to be defined beforehand for each satellite under study, which are its cross-sectional surface, perpendicular to the direction of the relative velocity vector, and its drag coefficient.

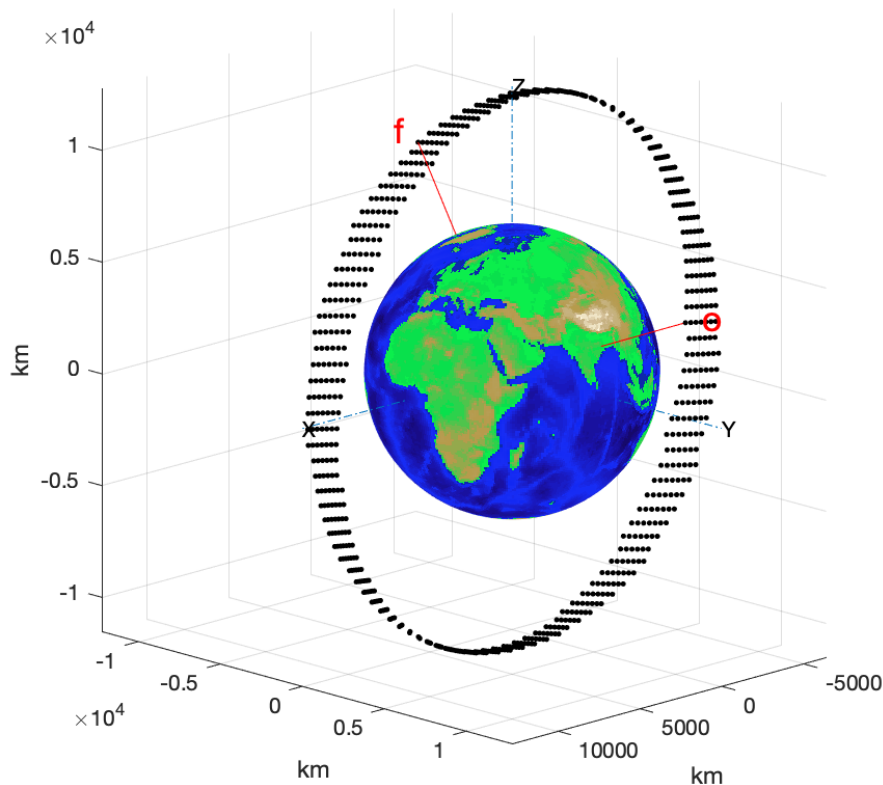
These parameters, for the case of LAGEOS-1, are specified hereunder.

- $C_D \approx 20$  [16]
- $A = \pi \cdot (0.6/2 \text{ m})^2 = 0.2827 \text{ m}^2$ , since the satellite has a diameter of 60 cm [8]

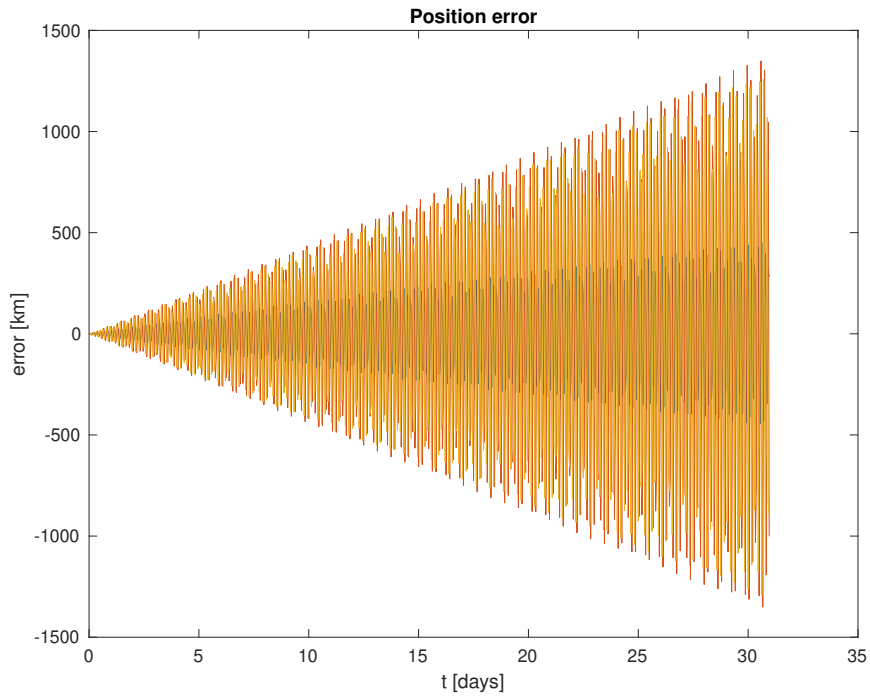
It is worth noticing that both Figure 31 and 34 really look identical, and the same happens with the error plots when comparing the ones including drag perturbation to the ones that do not account for it.

This result is not far from expected, since the orbit of LAGEOS-1 is actually around 5900 km above the Earth surface, in between LEO and Geostationary orbits, where the drag forces due to air density and friction with the atmosphere are practically negligible and hardly influence the results obtained.

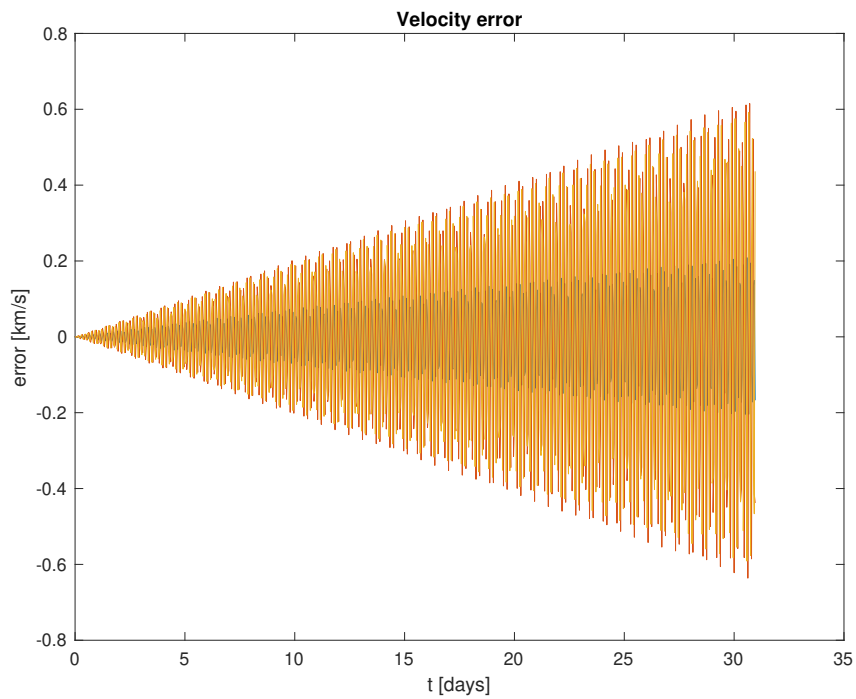
Nevertheless, later on in the section of orbital parameters, it will be appreciated how there actually is a slight change in the numerical value of most coefficients, corroborating in that way the fact that the addition of the drag perturbation affects the orbit calculation, even if minimally.



**Figure 34:** LAGEOS-1 trajectory during May 2021 introducing gravitational and atmospheric drag perturbations



**Figure 35:** LAGEOS-1 absolute position error during May 2021 introducing gravitational and atmospheric drag perturbations



**Figure 36:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational and atmospheric drag perturbations

### 3. Relativistic Effects

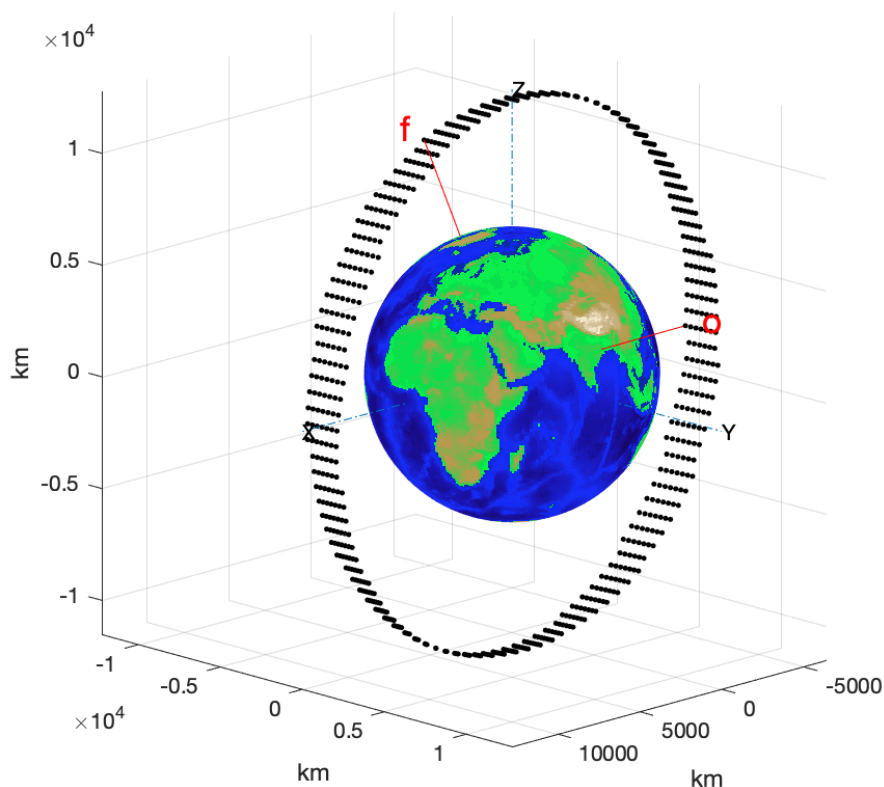
In order to consider relativistic effects in the calculations due to the relative movement between the Earth and the satellite, with the aim of taking precise time measurements related to the laser ranging technique used by LAGEOS-1, expressions from (61) to (63) must be added to the right-hand side of the differential equations in function  $rates.m$ .

For that purpose, no coefficient other than the Earth's gravitational parameter and the speed of light, which have already been defined, are needed.

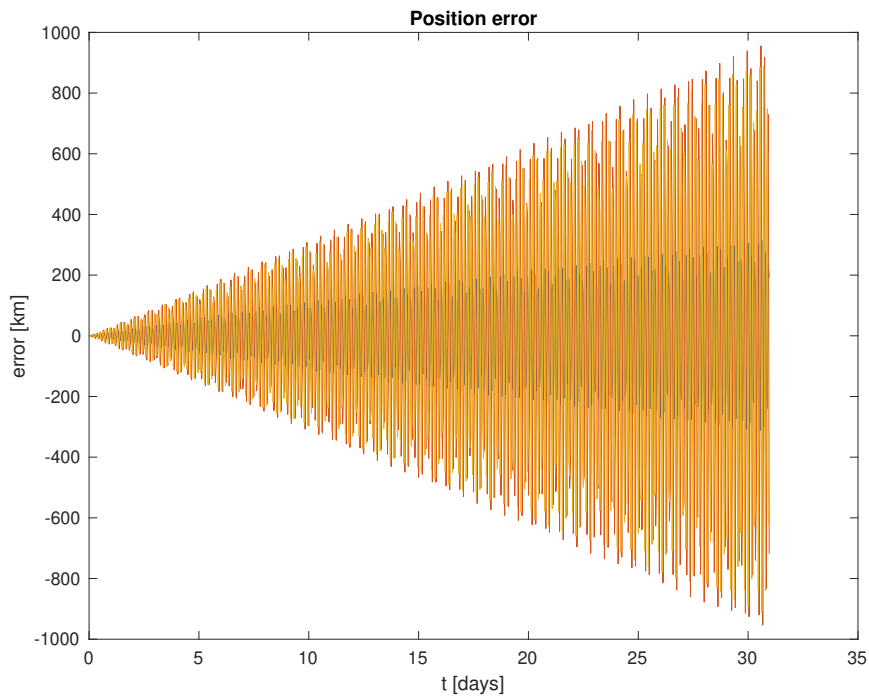
According to the trajectory 3D plot, the result is approximately the same as in the previous case. Apparently, the orbit calculated when accounting for relativity has not been modified at all.

Nonetheless, when comparing graphs 35 and 36 to the position and velocity error plots obtained in this case, it can be stated that relativistic effects really do influence the calculations and introduce a higher degree of accuracy in the results, since the position error diminishes from 1500 km to 1000 km in day 31, and the velocity error experiences a reduction of around 150 m/s.

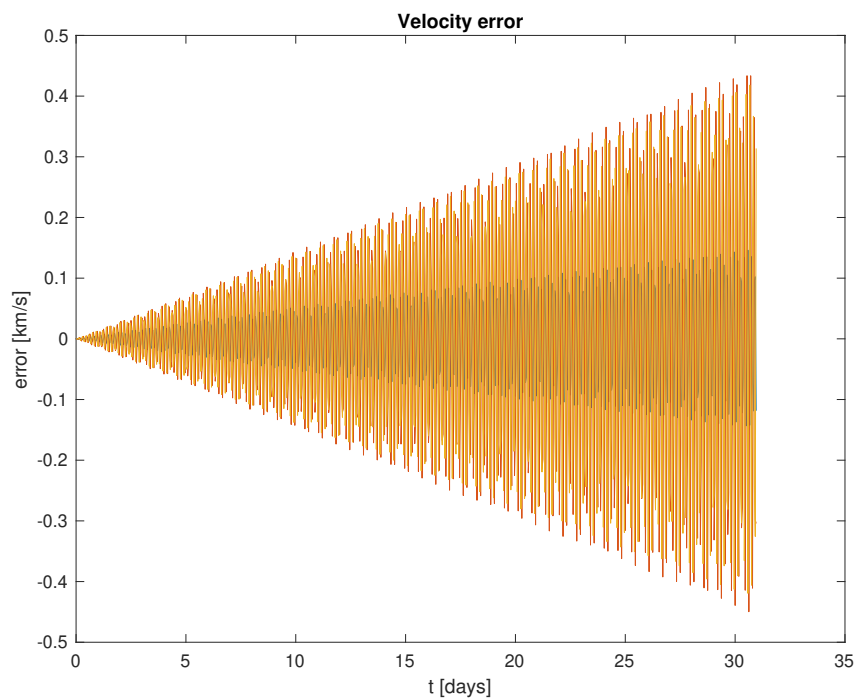
Therefore, it can be concluded that relativistic effects are crucial in order to account for the different time synchronization on the Earth surface and on space, and should definitely be included if a higher precision is required in the calculations.



**Figure 37:** LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations



**Figure 38:** LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations



**Figure 39:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations

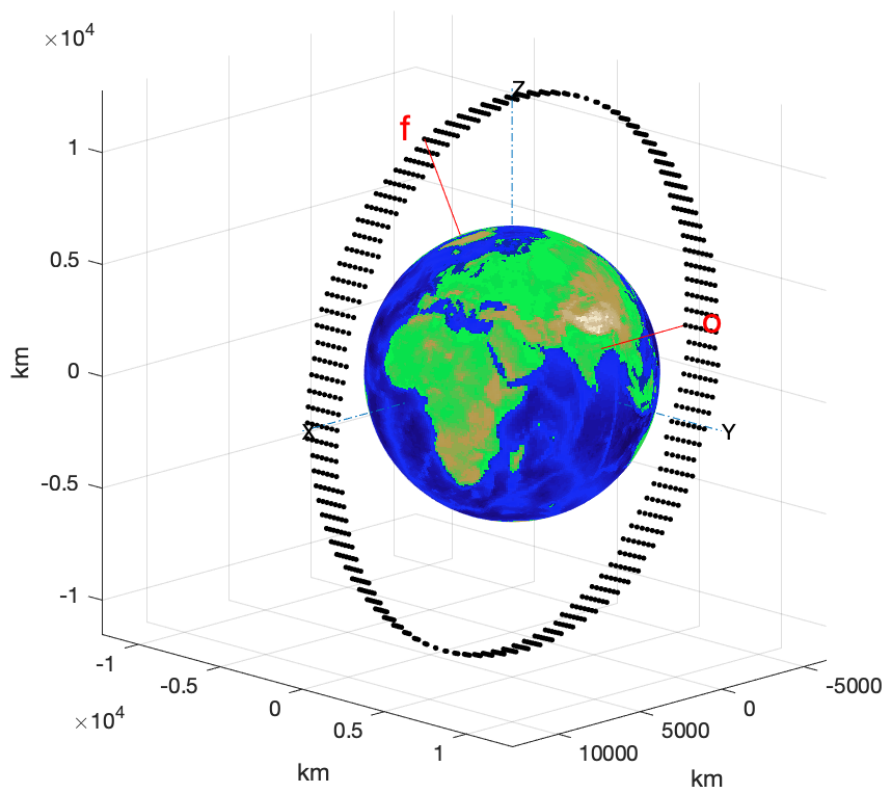
#### 4. Lunar Gravity Perturbation

When introducing the perturbation due to the gravity of the Moon over the orbit of LAGEOS-1 satellite, equations from (67) to (69) must be inserted, as in previous cases, in MATLAB™ function *rates.m*. Once the fundamental differential equation of relative motion is integrated by means of solver *ode45*, the results obtained are shown in Figure 40 to 42.

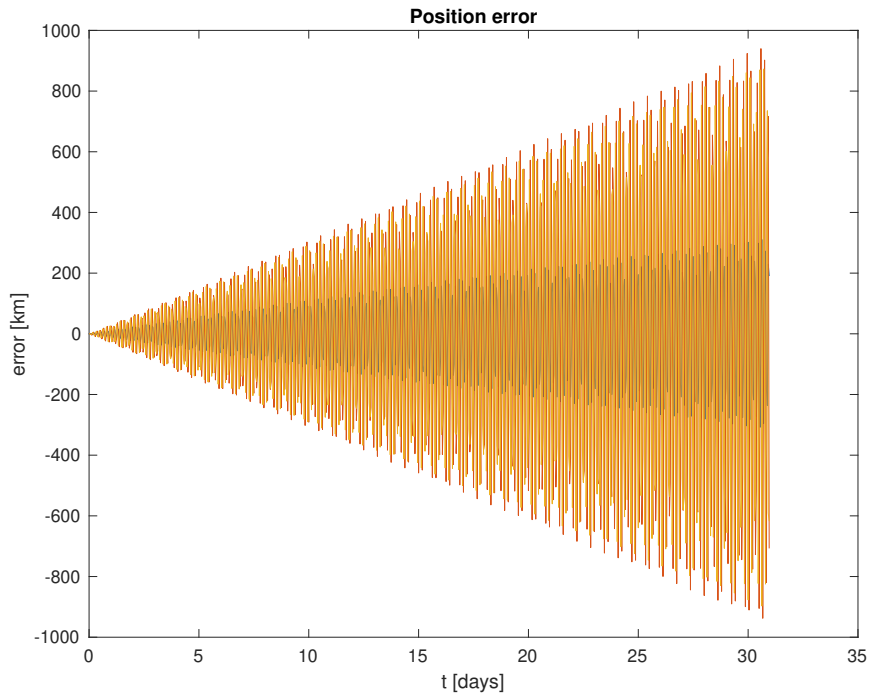
By comparing these graphs with the ones obtained in the previous case, no differences are observed. These results could be expected, since, according to Figure 13, lunar gravity is actually 7 orders of magnitude lower than the acceleration due to the mutual gravitational attraction forces between both bodies.

However, the effects of introducing this perturbation will be seen later on, in the orbital parameters section, in the same way as in the case of atmospheric drag. The difference will mostly be visible in decimal numbers, although for some parameters the values differ to a greater extent.

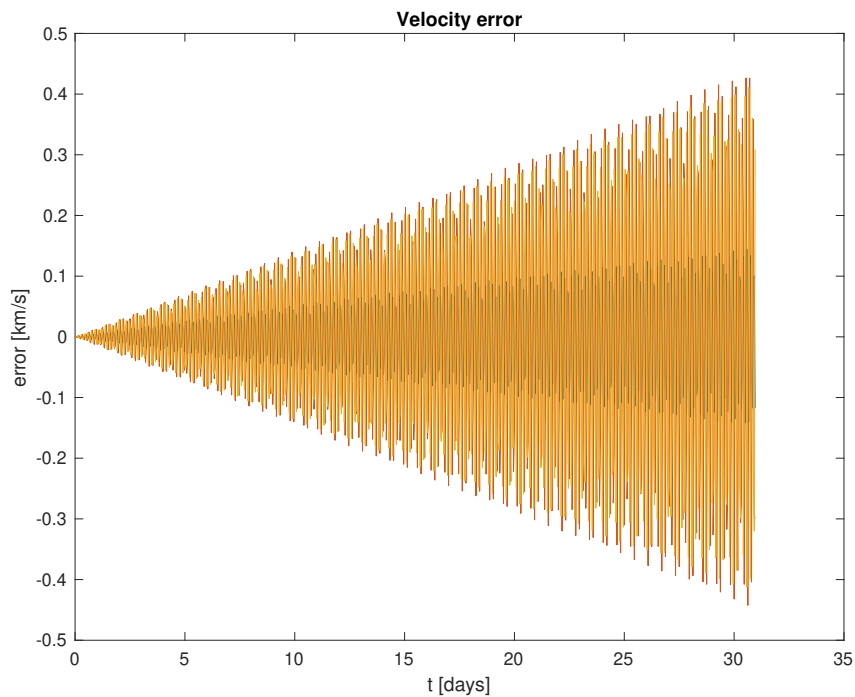
It is worth commenting the fact that the error plots keep presenting the characteristic triangular shape of the rest of cases, and the trajectory plot has not experienced a significant change since the introduction of gravitational perturbations.



**Figure 40:** LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations



**Figure 41:** LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations



**Figure 42:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations



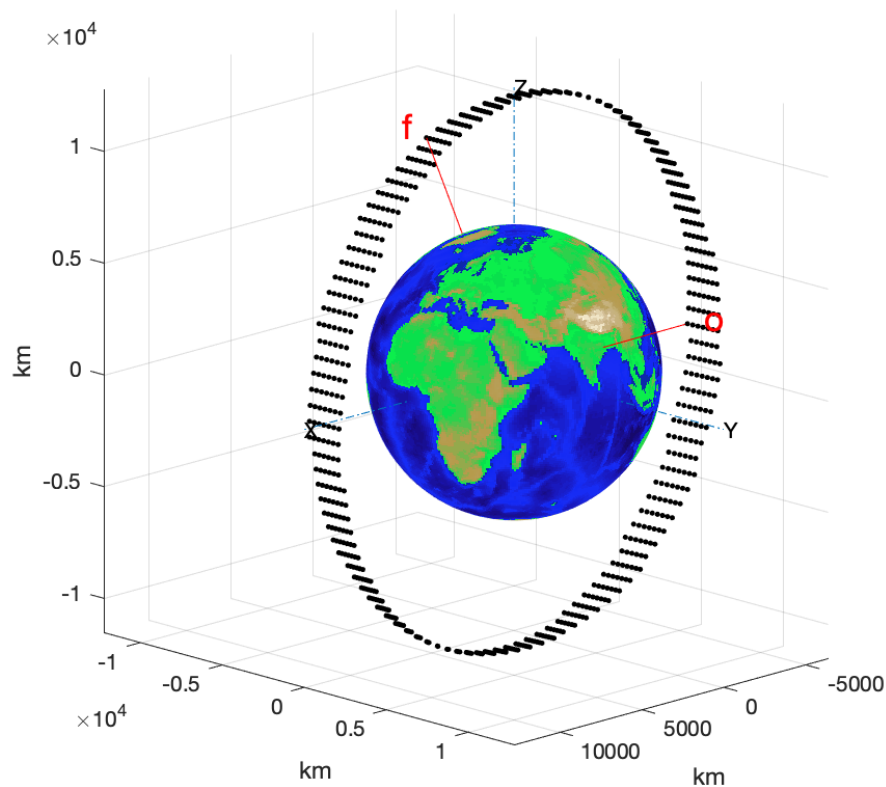
## 5. Solar Gravity Perturbation

For introducing this perturbing acceleration, expressions from (70) to (72) must be introduced, as usual, in the function to be numerically integrated.

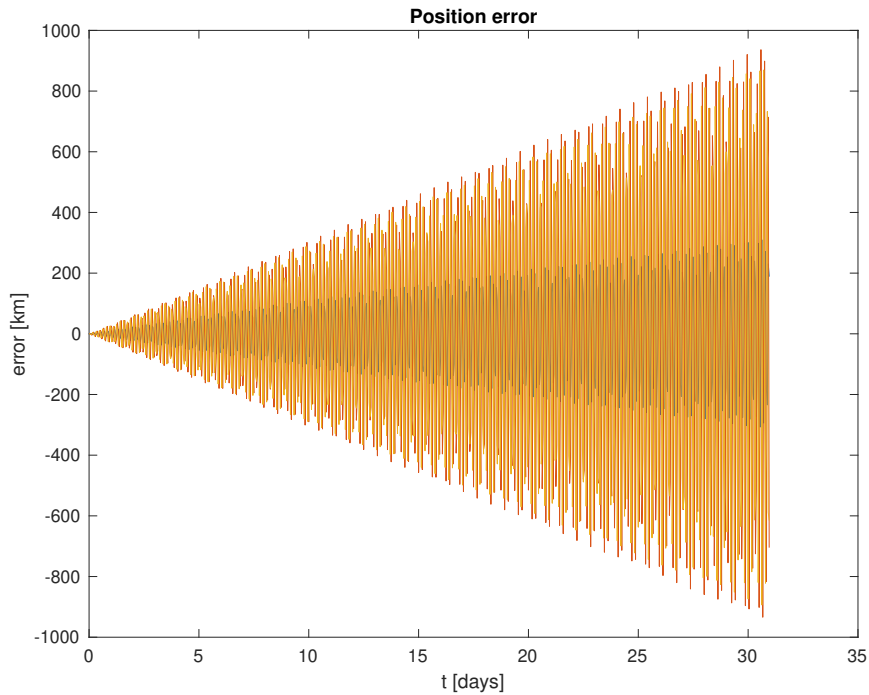
The gravity of the Sun also affects, but to a lesser extent, the calculation of the state vector of the satellite orbiting around the Earth. Actually, according to Figure 13, the magnitude of this perturbation represents  $10^{-7}$  times the acceleration of the satellite without perturbations. It is even less relevant than the perturbation due to the gravity of the Moon.

For such reason, the graphs obtained in this case are also identical to the ones of the previous case, with the same plot of the trajectory of the satellite and the same shape and numerical values in the state vector error plots.

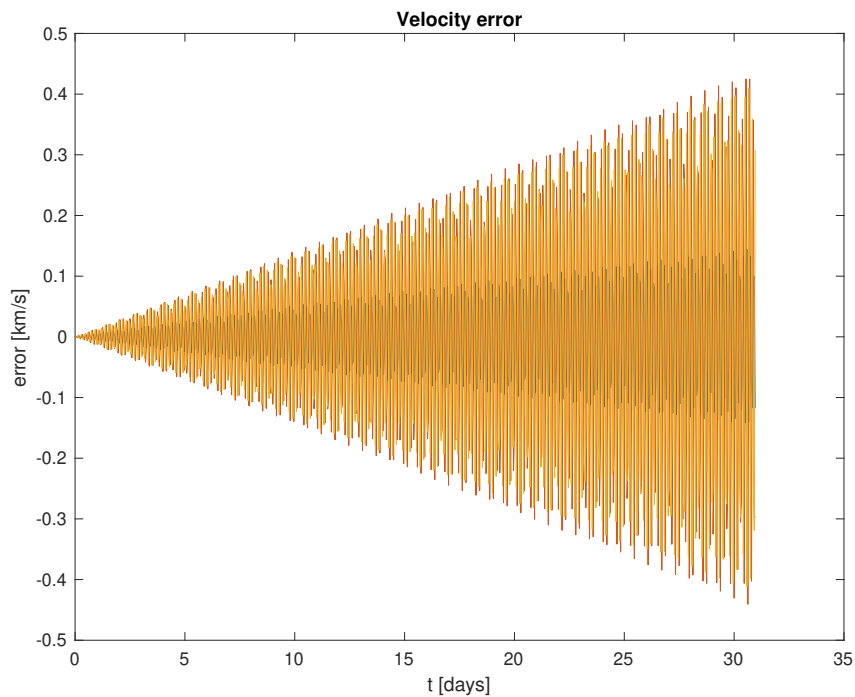
Nevertheless, as in past cases, the difference will be appreciated in the decimals of some orbital parameters, which will be presented in the next section.



**Figure 43:** LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations



**Figure 44:** LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations



**Figure 45:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations

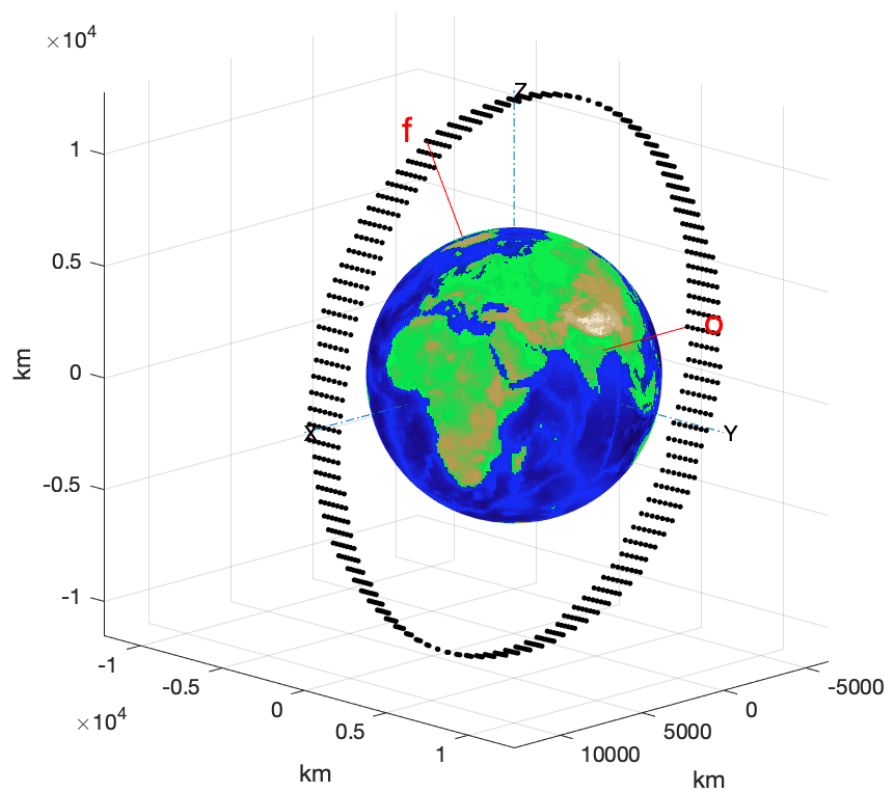
## 6. Perturbation due to Solar System Planets

If the addition of lunar and solar gravity perturbations hardly influenced the results obtained, then the introduction of the gravity of solar system planets as perturbations will be far from noticeable.

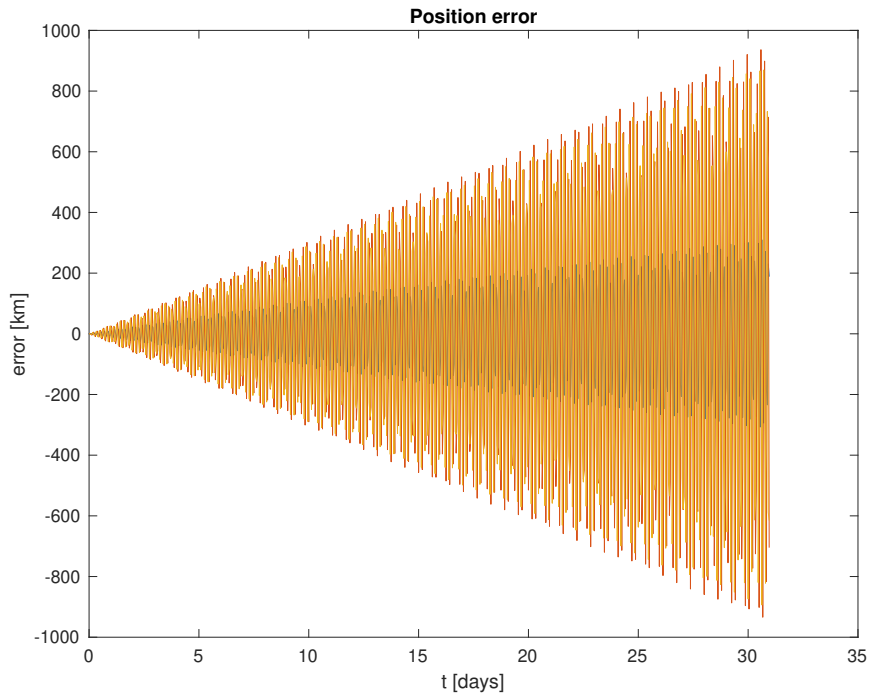
The only coefficients needed are the gravitational parameters of each planet, which have already been presented. By adding all the expressions for the perturbing accelerations of Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto, even though it is not regarded as a solar system planet, into the equations of motion, and integrating them, the state vector for each time instant is obtained, and the results are plotted from Figure 46 to 48.

As it was expected, the plots have not changed with respect to previous cases, being approximately the same since the introduction of relativistic effects, what is more, even the orbital parameters remain the same as the ones obtained including all perturbations up to solar gravity, as it will be seen later on in Table 2.

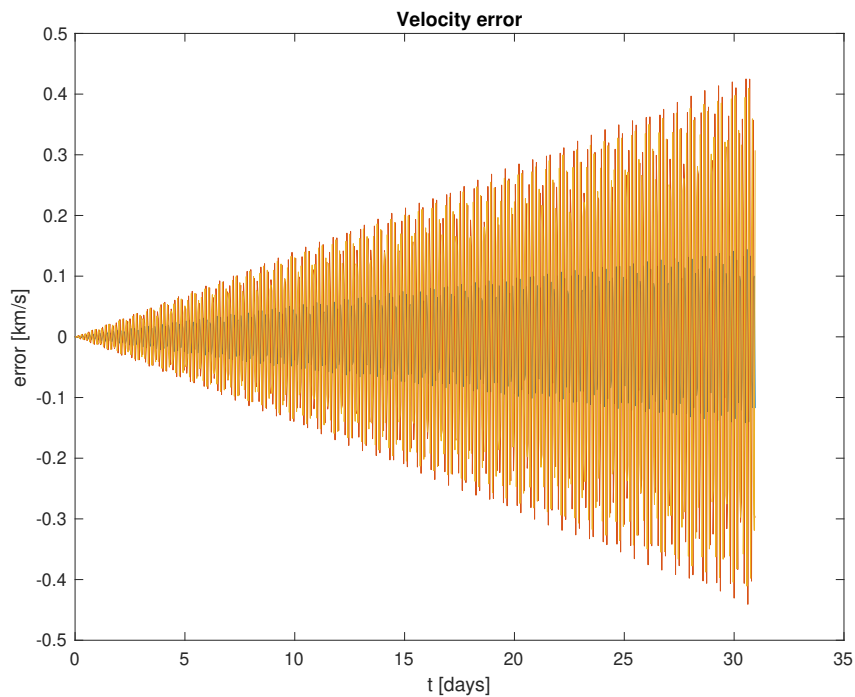
Hence, it can be concluded that the addition of these expressions in the differential equations is not worth the computational cost and time for the case of study of one month, since the effect of introducing them is not visible when it comes to the results obtained. Notwithstanding, if the integration was carried out for a longer period of time, the influence of these perturbations would definitely be more noticeable.



**Figure 46:** LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations



**Figure 47:** LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations



**Figure 48:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations

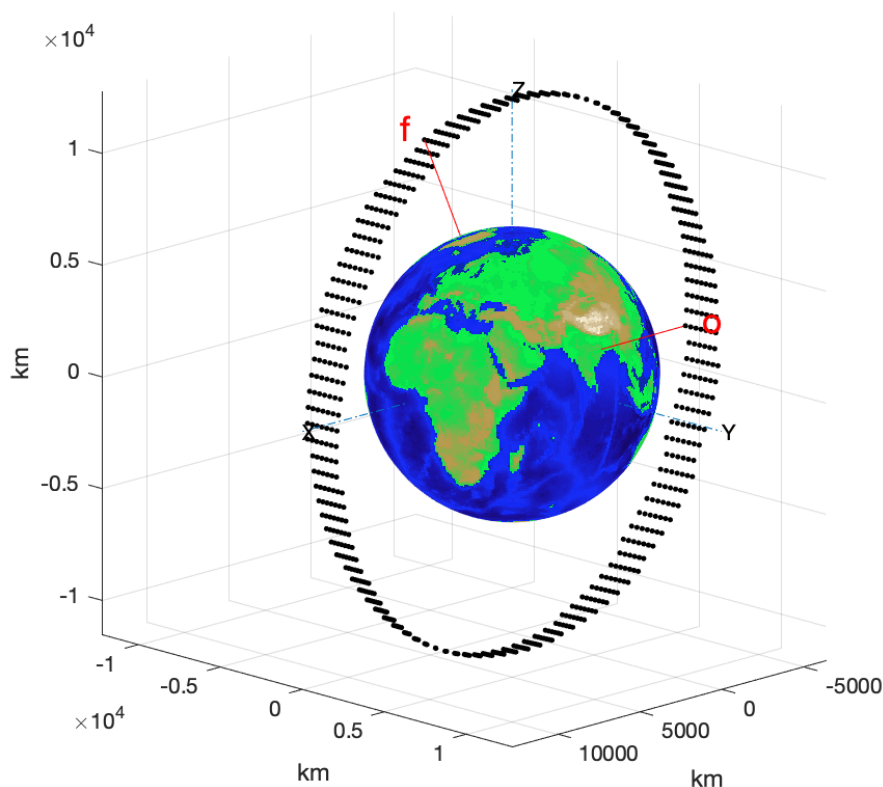
## 7. Solar Radiation Pressure

In the last place, the perturbing acceleration due to solar radiation pressure is added to the fundamental equation in *rates.m*. For that purpose, some parameters must be defined for the satellite which is going to be analyzed, LAGEOS-1 in this case. These coefficients are the radiation pressure coefficient,  $C_R$ , as well as the cross-sectional absorbing area of the satellite,  $A_S$ , which is considered to be the same as the one used for atmospheric drag. These parameters are defined hereafter.

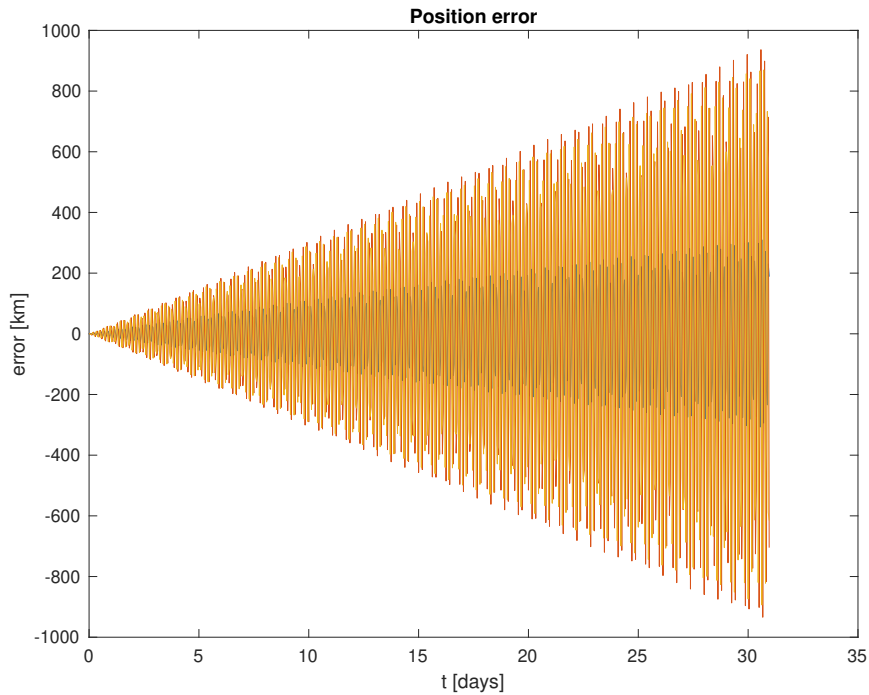
- $C_R = 1.142$  [17]
- $A_S = \pi \cdot (0.6/2 \text{ m})^2 = 0.2827 \text{ m}^2$ , since the satellite has a diameter of 60 cm [8]

The graphs obtained by solving the problem, once all perturbations studied in this document are included, are shown from Figure 49 to 51. Once again, the addition of this perturbation is hardly noticeable, since its order of magnitude is around  $10^{-9}$  times the initial acceleration of the satellite, according to Figure 13.

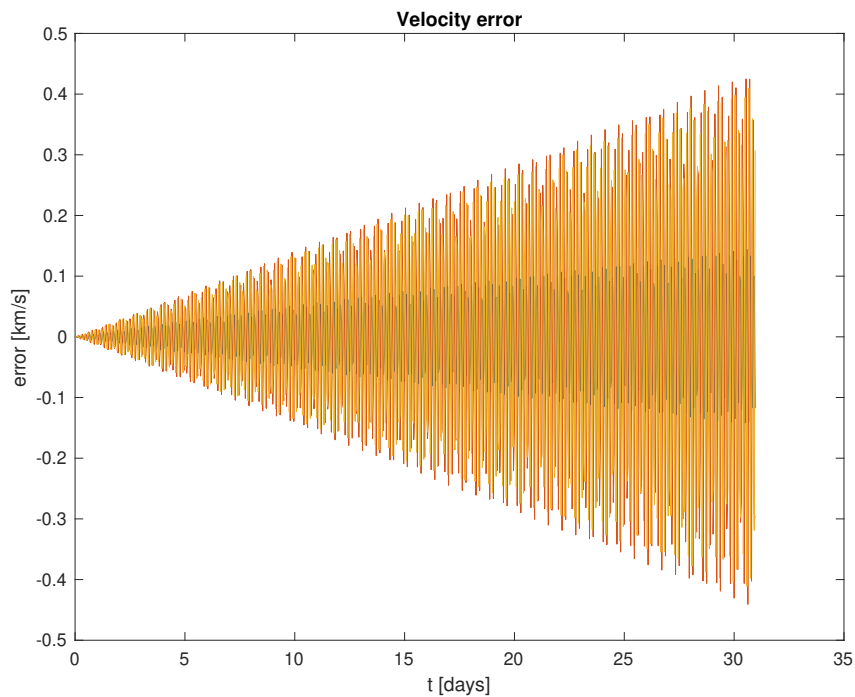
Even though the plots do not change with respect to the ones obtained in previous cases, it can actually be seen how the orbital parameters differ to some extent from the previous ones, which will be explained later on in Table 2.



**Figure 49:** LAGEOS-1 trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations



**Figure 50:** LAGEOS-1 absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations



**Figure 51:** LAGEOS-1 absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations

### 3.2.2 International Space Station orbit

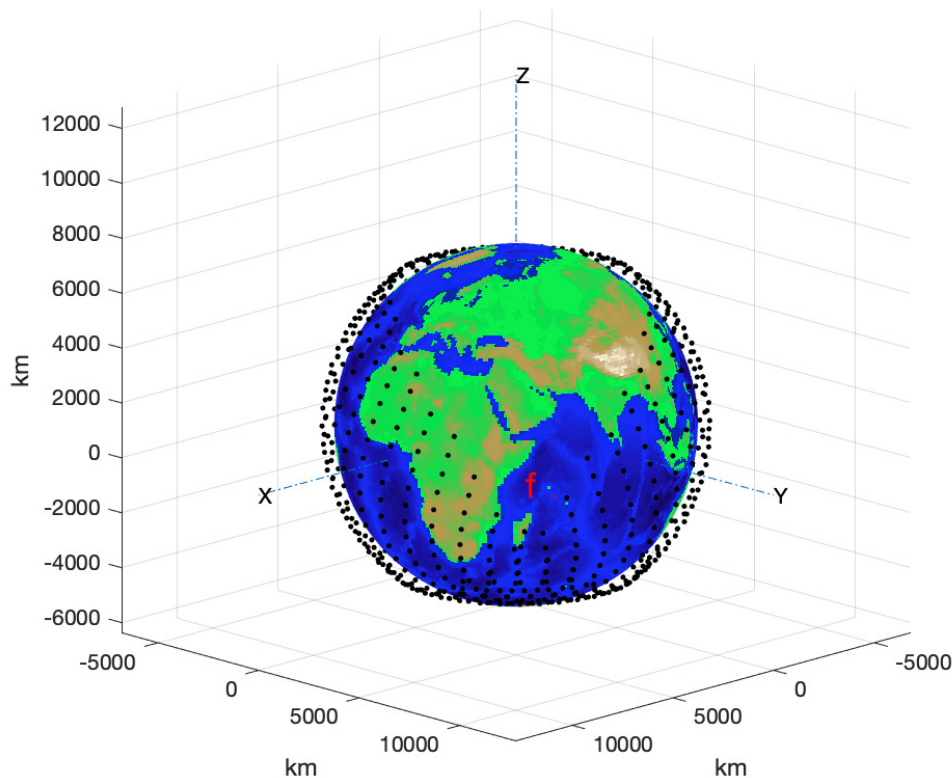
#### 1. Gravitational Perturbations

Moving on to the case of the International Space Station, the results obtained when introducing the gravitational perturbations due to zonal harmonics from  $J_2$  to  $J_6$ , which have already been defined, are shown from Figure 52 to 54.

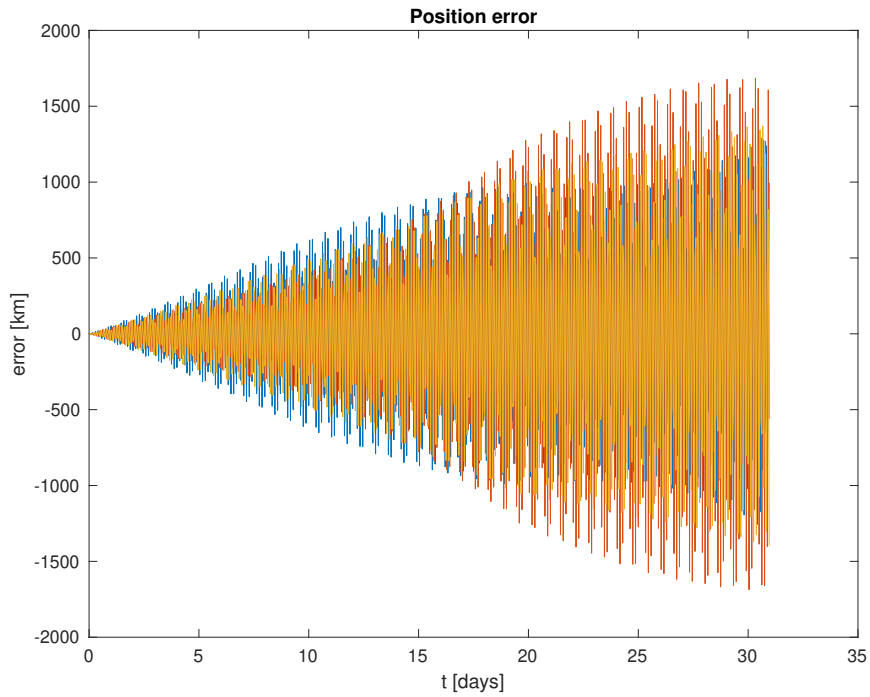
By first comparing the trajectory plot obtained in this case to the one without perturbations in Figure 10, a huge difference can be appreciated. The path followed by the station throughout the month for which the solution is calculated is not included in one orbital plane, but in several planes around the Earth, approaching the real trajectory shown in Figure 9. It appears that way due to the fact that the Earth is rotating below the ISS, and in the plots, the Earth is shown in a fixed position, and the movement of the station relative to it is depicted.

Concerning the state vector calculated in this case, by comparing Figure 11 and 12 to Figure 53 and 54, it can be observed that the error has been reduced for one order of magnitude with respect to the initial results obtained. The position error has decreased from around 10000 km in day 31, to approximately 1600 km. Similarly, the velocity error has experienced a reduction from about 11000 m/s to 2000 m/s, which, while still being a non-negligible value, is relatively low.

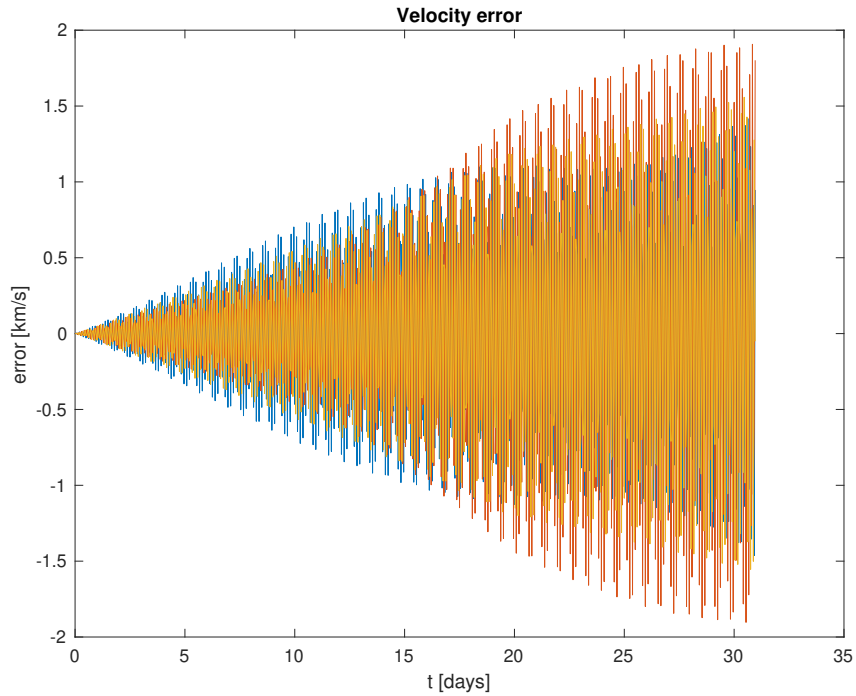
Regarding the shape of the error plots, they are not absolutely triangular as in the case of LAGEOS-1, just one coordinate has this shape, whereas the two remaining ones present a more kind of rounded shape.



**Figure 52:** ISS trajectory during May 2021 introducing gravitational perturbations



**Figure 53:** ISS absolute position error during May 2021 introducing gravitational perturbations



**Figure 54:** ISS absolute velocity error during May 2021 introducing gravitational perturbations



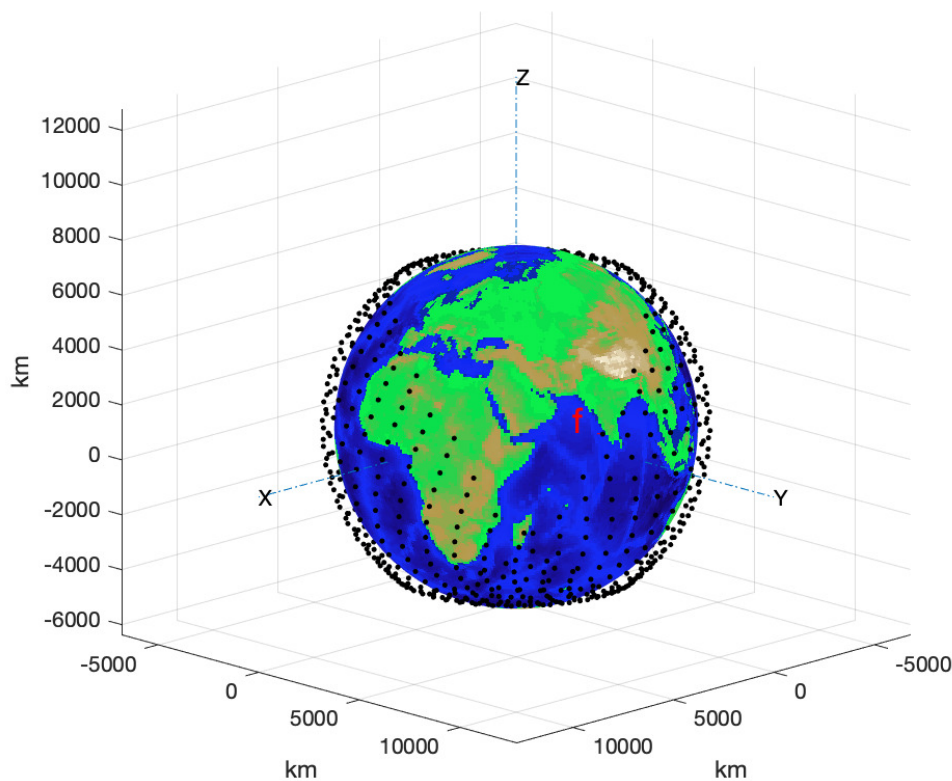
## 2. Atmospheric Drag Perturbation

When it comes to introducing the perturbing acceleration due to atmospheric drag, two coefficients must be defined for the case of the ISS. These parameters are specified hereunder [18]. It is worth noticing that the ISS, in contrast with LAGEOS-1, is not a spherical satellite, and therefore its cross-sectional surface is not constant and independent from the cutting plane, and then it can present a range of values. However, for the purpose of the present document, this area will be assumed to be around  $1000 \text{ m}^2$  [18].

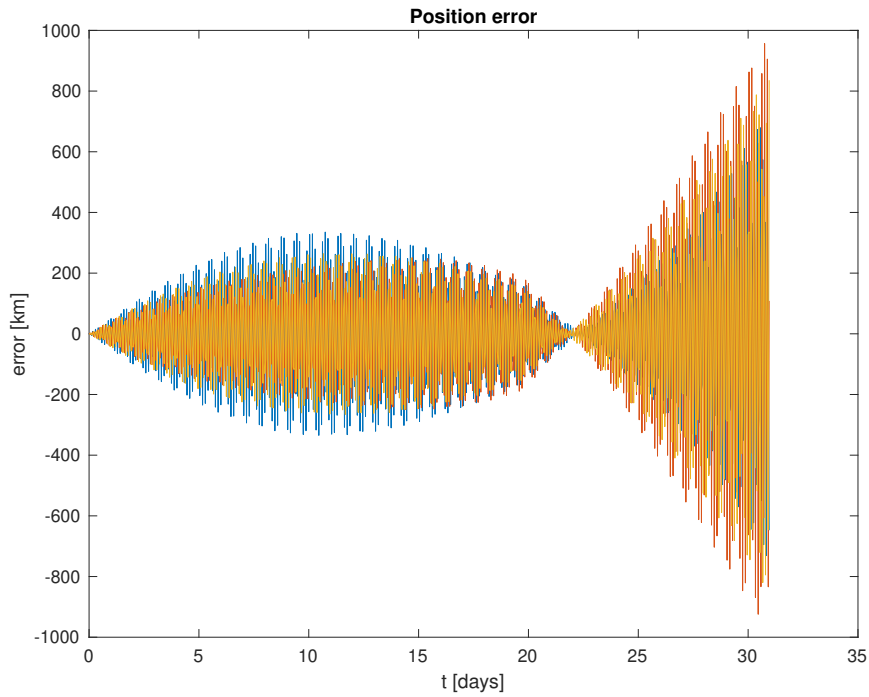
- $C_D = 2.07$
- $A \approx 1000 \text{ m}^2$ , it varies between 700 and  $2300 \text{ m}^2$

Regarding the resulting plots, shown from Figure 55 to 57, it can be observed how the solution in this case is more precise than in the previous case. Even though this is hardly visible in the trajectory plot, the error plots show an important reduction from 1600 km to less than 1000 km concerning the position error, and from around 2000 m/s to about half this value when it comes to the speed of the satellite. This relevant improvement of the results when adding the drag perturbation lies in the fact that, unlike LAGEOS-1, the station is located at around 400 km above the Earth surface, in a region where the order of magnitude of the atmospheric drag perturbing acceleration is large enough for an evolution to be noticed, according to Figure 13.

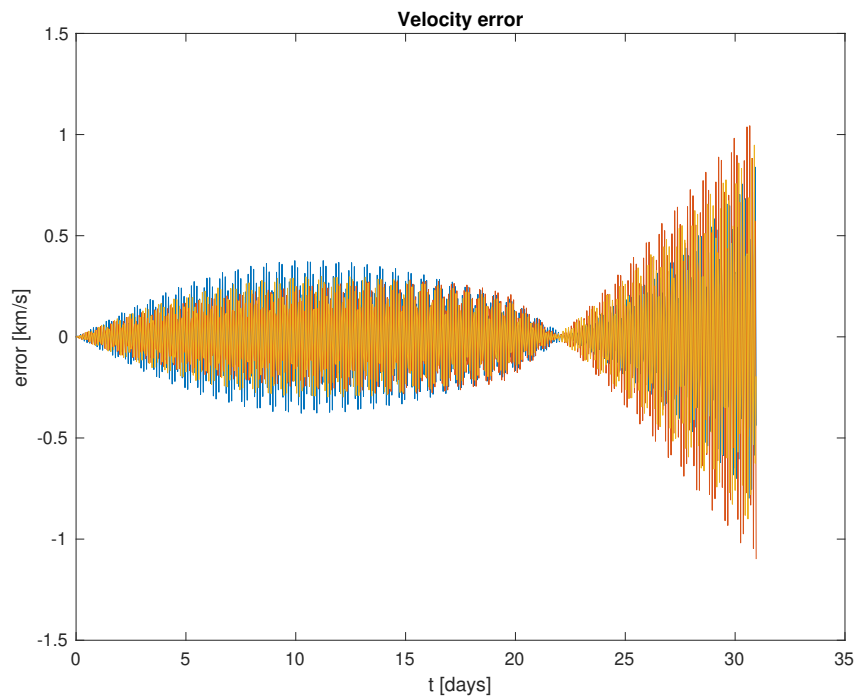
Furthermore, it is worth observing the shape of the error plots, which is completely different from the ones seen so far, in which the error was rather cumulative. In this case, though, the error starts to increase and, at a certain point, it decreases again until 0 in day 22, and then starts to grow again in a linear way.



**Figure 55:** ISS trajectory during May 2021 introducing gravitational and atmospheric drag perturbations



**Figure 56:** ISS absolute position error during May 2021 introducing gravitational and atmospheric drag perturbations



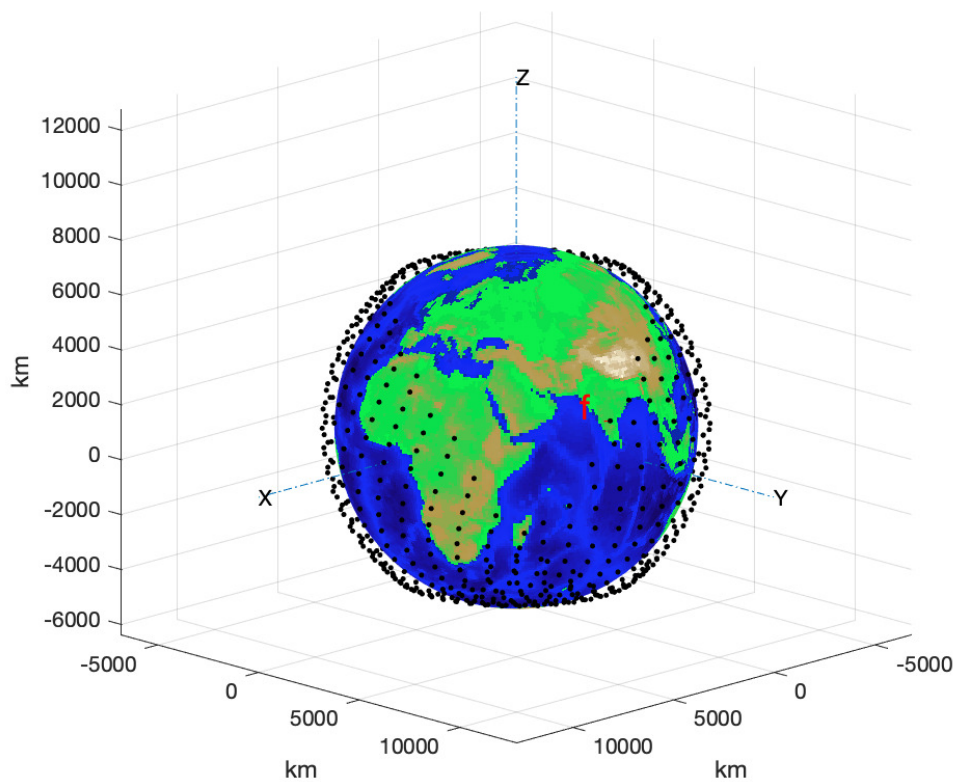
**Figure 57:** ISS absolute velocity error during May 2021 introducing gravitational and atmospheric drag perturbations

### 3. Relativistic Effects

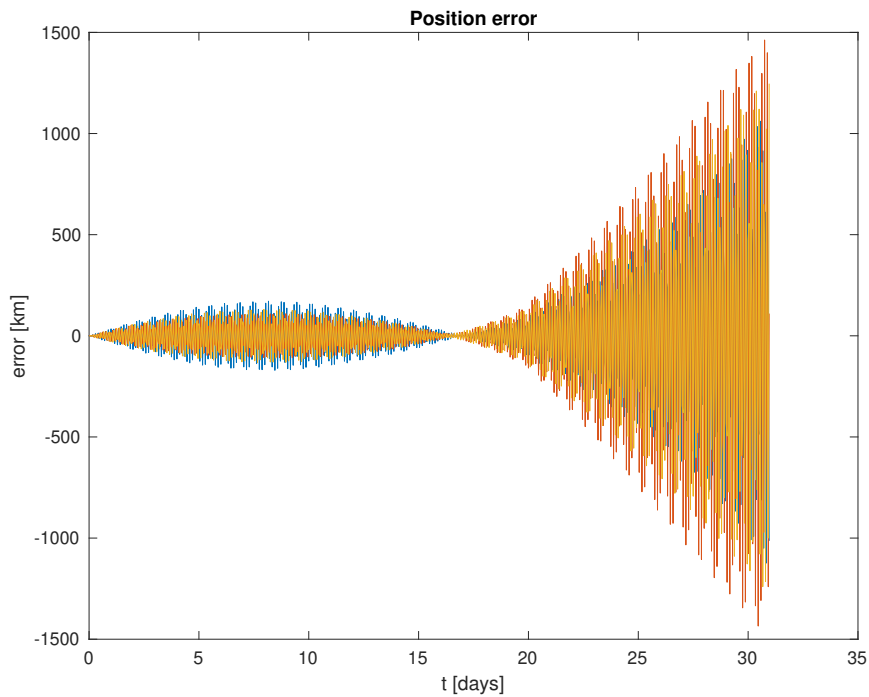
The results of accounting for the different time synchronization between the ISS and the planet are obtained and plotted from Figure 58 to 60. The only coefficient to include in the expression of the perturbing acceleration is the speed of light.

The trajectory plot obtained looks really similar to that of the previous case. Nonetheless, some differences can be appreciated in the state vector error plots. Even though the shape is quite similar to that of the previous error graphs, in this case, the error obtained during the first 16 days is reduced, keeping the increasing-decreasing circular shape during that period. However, during the second fortnight of May, the error increases rapidly and in a linear way, reaching even higher values at day 31 than in the previous case in which relativistic effects were not considered. It goes from 1000 km to around 1500 km in terms of position, and from about 1000 m/s to 1500 m/s in terms of speed.

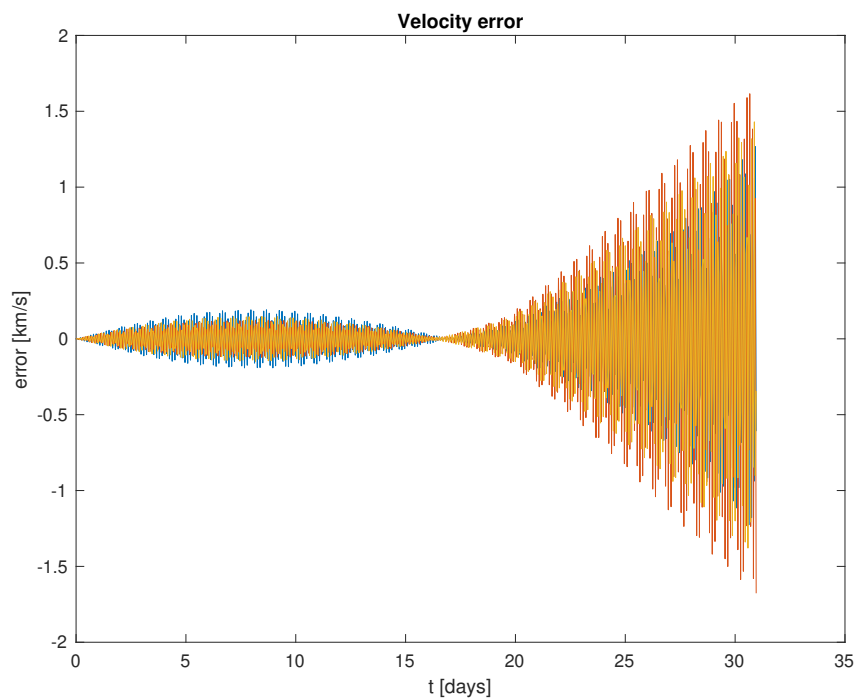
This suggests that the expression for introducing relativistic effects, given by Equation (61) to (62), might only be effective for the first 15-20 days of the month, since the addition of this perturbation causes a non-negligible increase in the error of the calculated state vector of the satellite.



**Figure 58:** ISS trajectory during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations



**Figure 59:** ISS absolute position error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations



**Figure 60:** ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag and relativistic perturbations

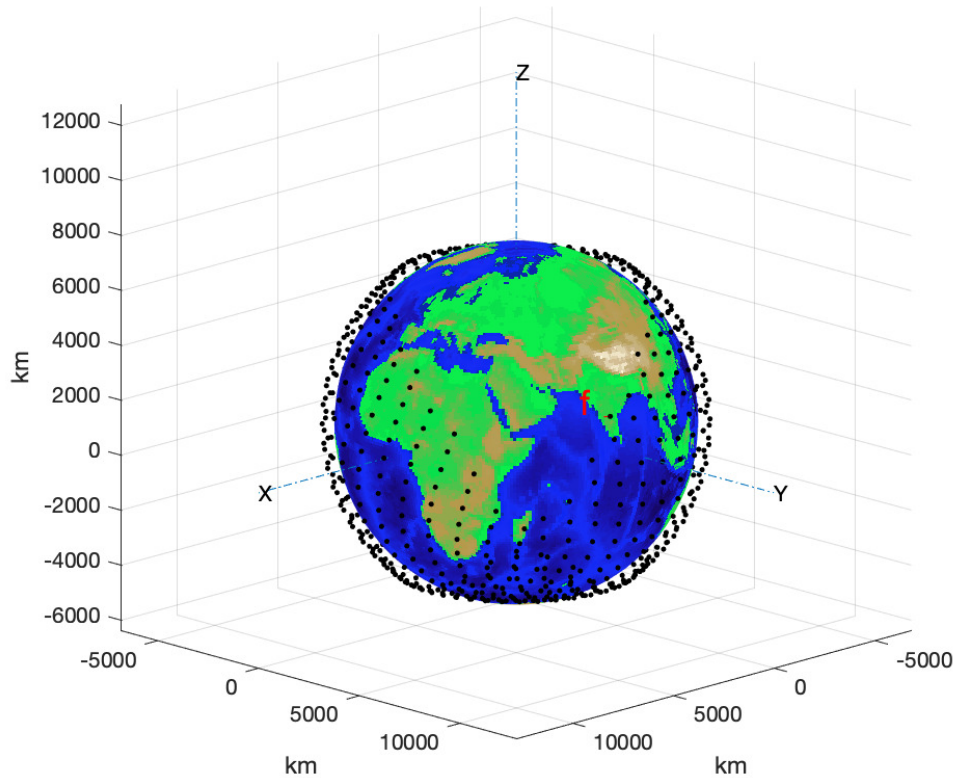
#### 4. Lunar Gravity Perturbation

As it was already stated when studying the case of LAGEOS-1 satellite, the perturbing acceleration due to the gravity of the Moon is around 7 orders of magnitude lower than the initial acceleration of the satellite, according to Figure 13, and hence, a minute reduction of the error should be expected.

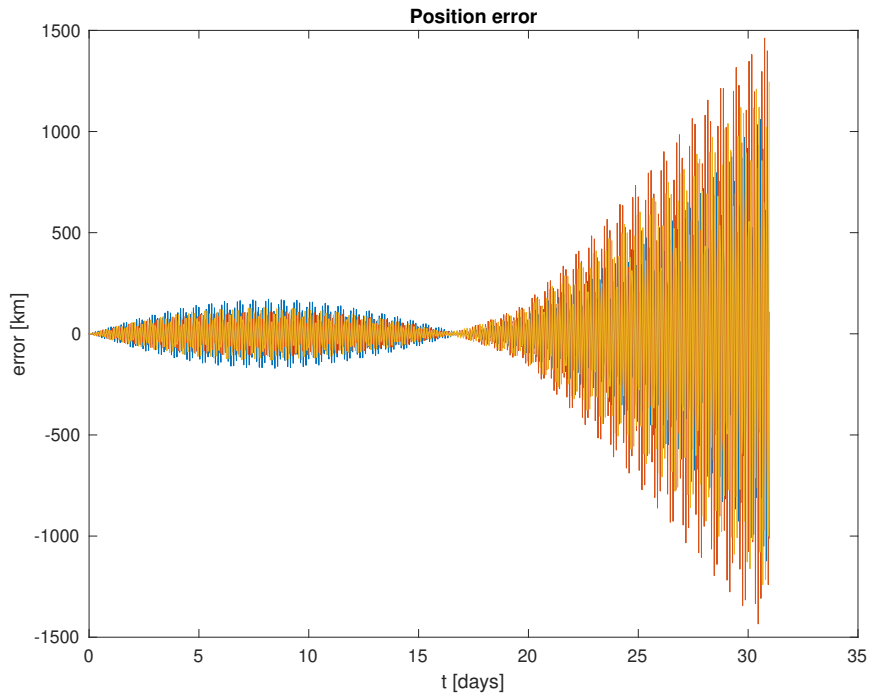
By comparing the trajectory plot, as well as the error plots, with the ones corresponding to the previous case, no differences are observed between both cases. It is worth remembering that the same happened when studying LAGEOS-1. The lunar gravity perturbation is, indeed, negligible when it comes to the state vector obtained.

Notwithstanding, as it will be seen in the orbital parameters section, some of them change slightly when introducing this perturbing acceleration.

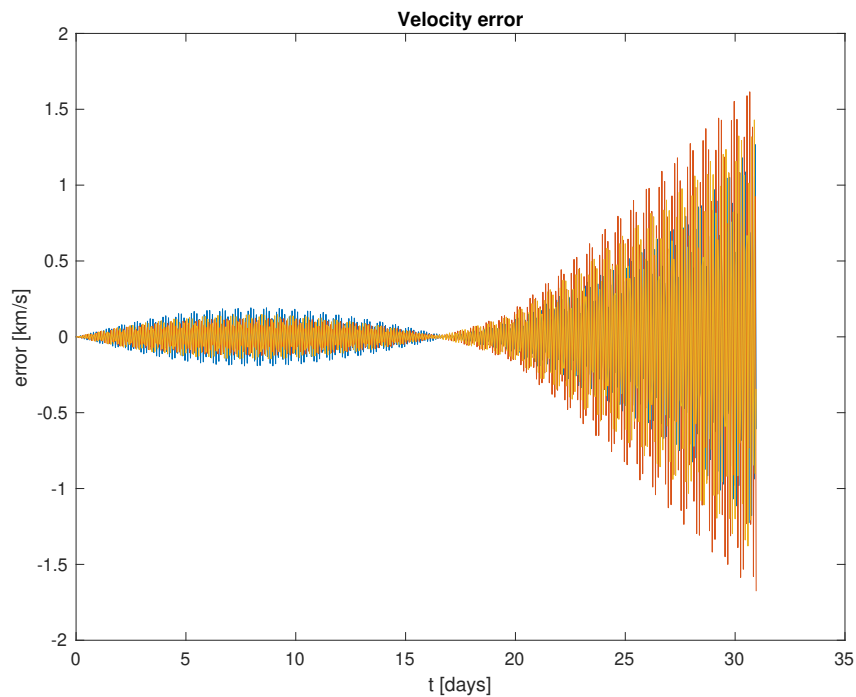
As it can be appreciated, the shape of the error plots remains the same as in the previous case, with really low error values during the first fortnight of May, reaching a null error around day 16, and a sudden steep increase during the second part of the month.



**Figure 61:** ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations



**Figure 62:** ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations



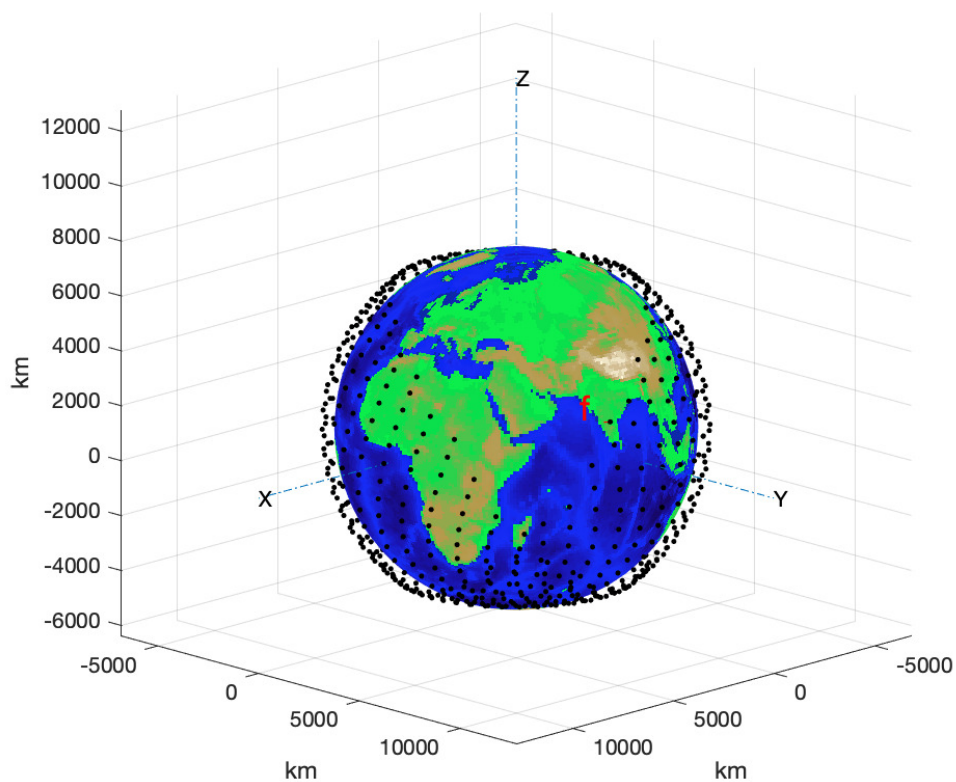
**Figure 63:** ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic and lunar gravity perturbations

## 5. Solar Gravity Perturbation

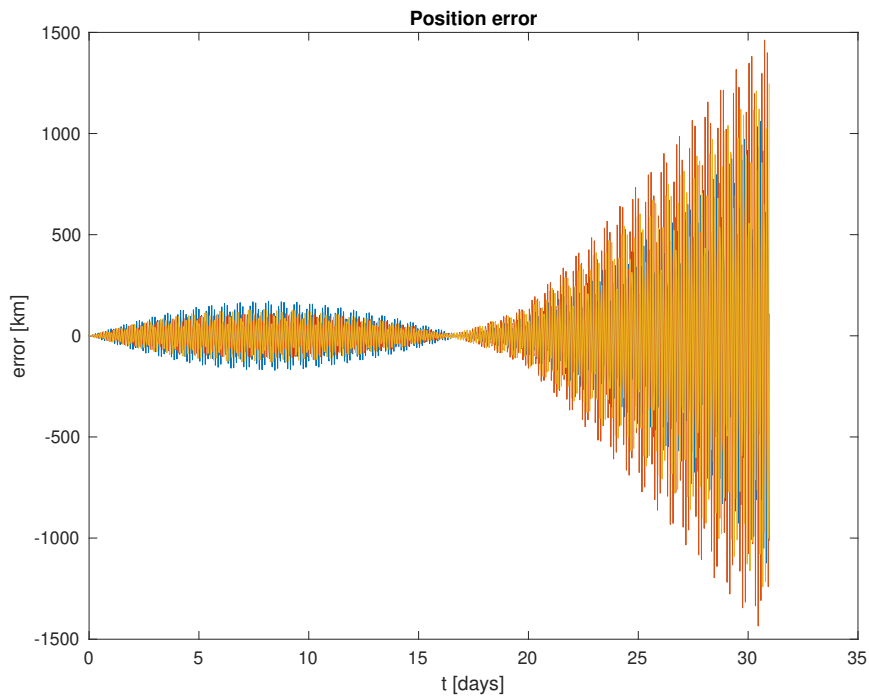
The order of magnitude of the perturbing acceleration due to the gravity of the Sun is really similar to that of the Moon, indeed it is even lower because of the larger distance between Sun and Earth.

Therefore, the results hardly experience a change with respect to not introducing the perturbation due to solar gravity, and then the error plots are identical to the ones obtained in the previous case, both in shape and numerical values, as it was expected.

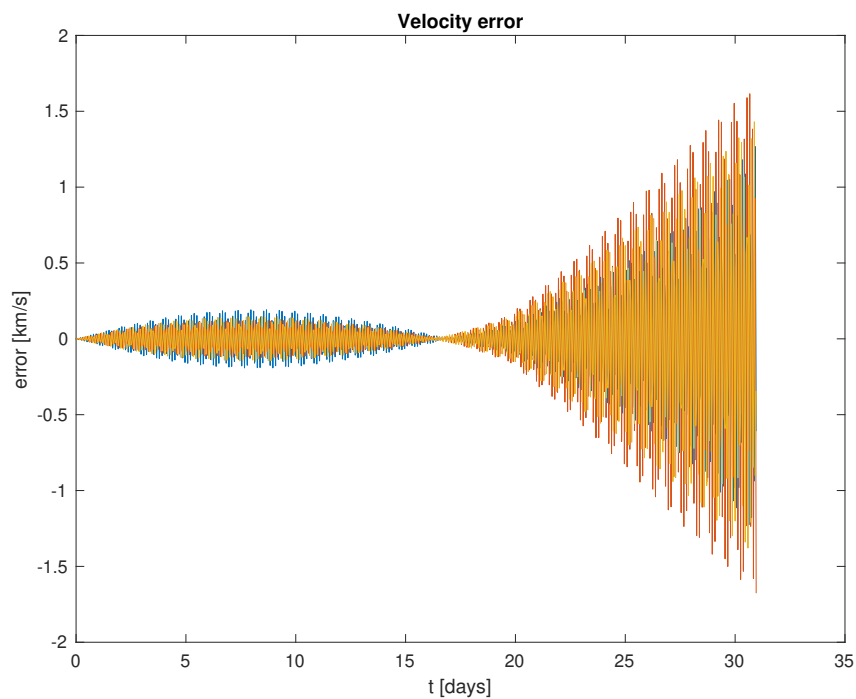
Nevertheless, when comparing the values of the orbital parameters with and without accounting for such perturbing acceleration, a slight difference in some decimal values can be appreciated, and thus it is worth considering the gravity of the Sun when calculating an orbit around the Earth.



**Figure 64:** ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations



**Figure 65:** ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations



**Figure 66:** ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity and solar gravity perturbations

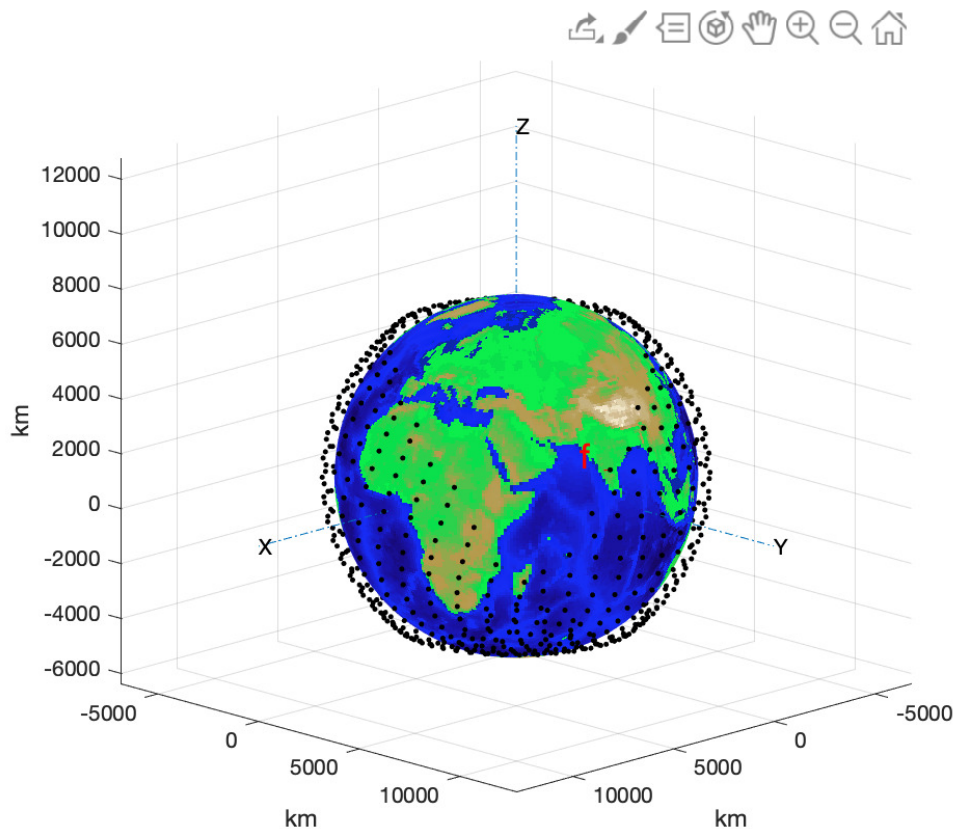


## 6. Perturbation due to Solar System Planets

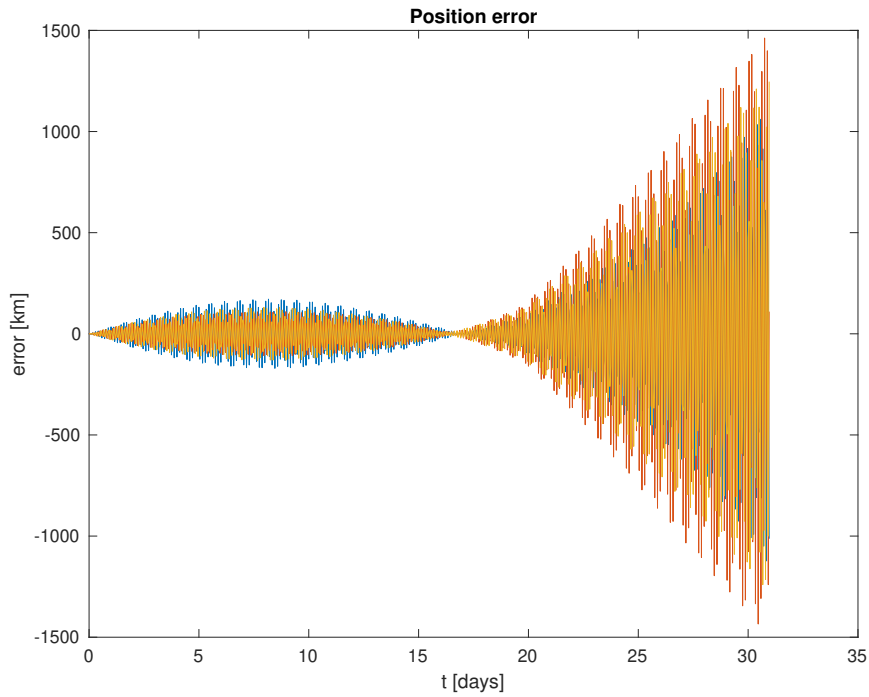
When taking into consideration the gravity of each of the planets in the solar system (plus Pluto) in the resolution of the two-body problem, the improvement of the results is expected to be absolutely unnoticeable, since the Moon and Sun hardly made a change when they were inserted, and since when studying LAGEOS-1 satellite the introduction of the gravity of all these bodies did not make any difference over the results.

This can be corroborated by just looking at the graphs from the previous case, both the trajectory and error plots, and observing that they are all truly identical to the ones obtained in this case. What is more, the orbital parameters do not change at all, not even the lowest decimal.

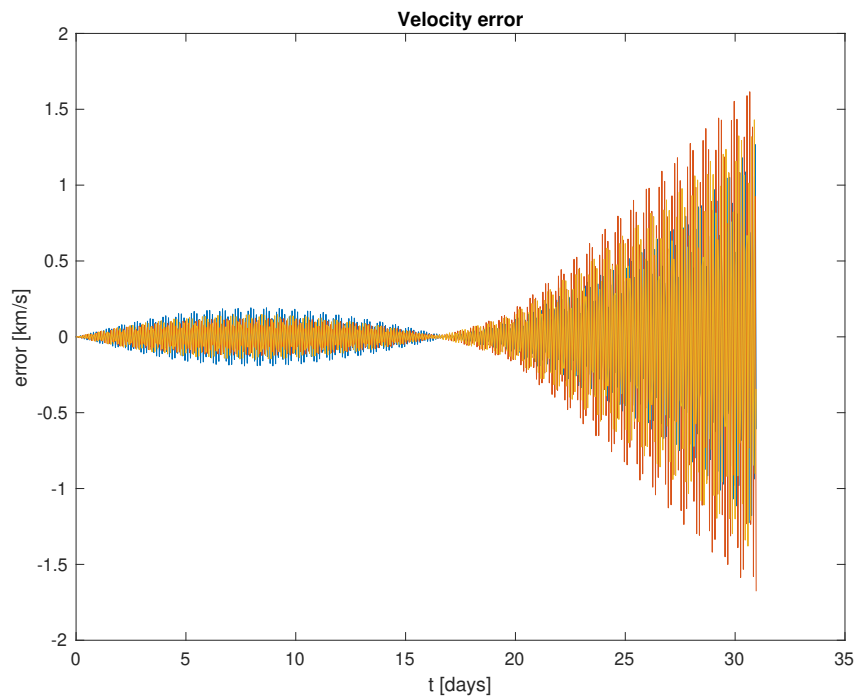
Having studied and introduced the perturbing acceleration of the different bodies in the solar system for two different cases of application, and having seen that both of them provided a null improvement of the results, it can be stated, as in the case of LAGEOS-1, that including them is not worth all the costs it implies.



**Figure 67:** ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations



**Figure 68:** ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations



**Figure 69:** ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity and solar system planets' gravity perturbations

## 7. Solar Radiation Pressure

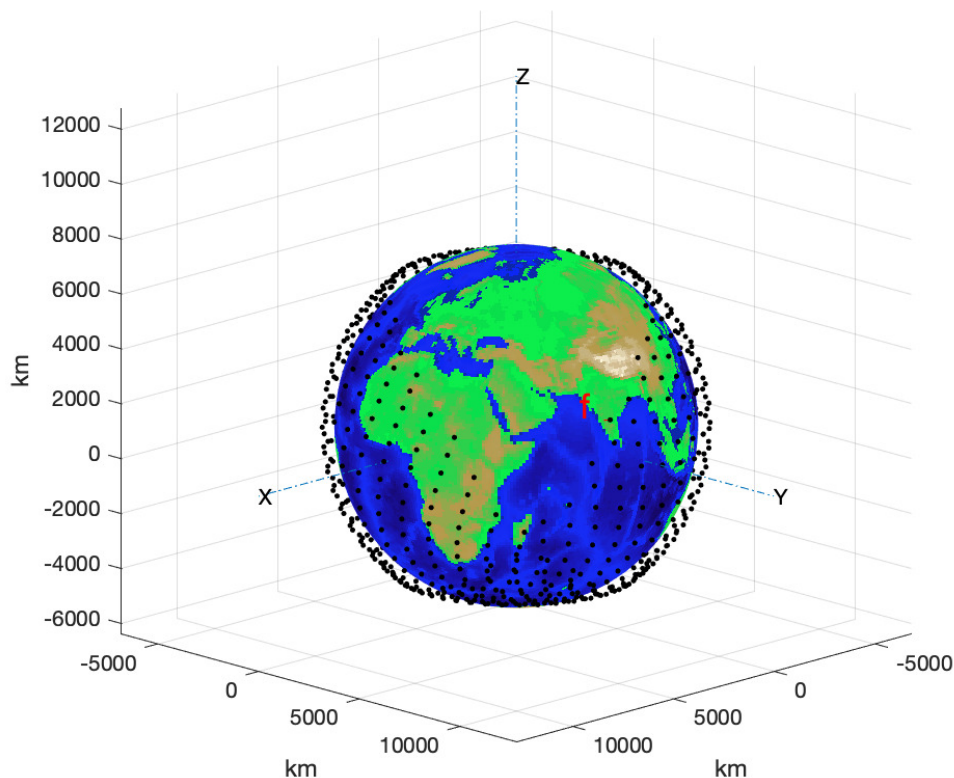
This is the last case that will be analyzed, and SRP is the last perturbing acceleration that will be introduced to the differential equation for obtaining the state vector of the ISS for every hour during the month of May, 2021.

In order to perform the addition of this perturbation, two coefficients are required to be specifically defined for the ISS. It is worth commenting that the radiation pressure coefficient has been assumed to be the average value between 1 and 2, due to the lack of information about this coefficient for the ISS, in such a way that it will be assumed to behave partially as a black body, absorbing and emitting some radiation, and partially as a reflecting body, reflecting some of the received radiation. Moreover, the absorbing area is considered to be the same as the one used when introducing the atmospheric drag perturbation.

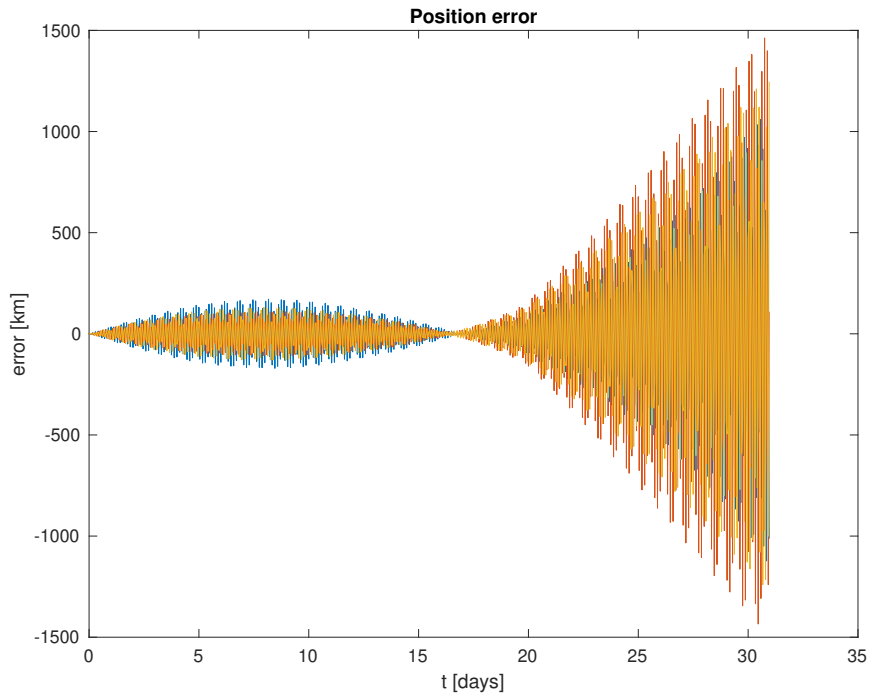
- $C_R = 1.5$
- $A_S \approx 1000 \text{ m}^2$ , it varies between 700 and 2300  $\text{m}^2$

As for the trajectory and error plots, no difference is appreciated between the current plots and the ones from Figure 68 and 69, since the magnitude of this perturbing acceleration is around  $10^{-9}$  times the acceleration of the satellite without accounting for perturbations. Even the shape of these plots is identical since the introduction of relativistic effects in the calculations.

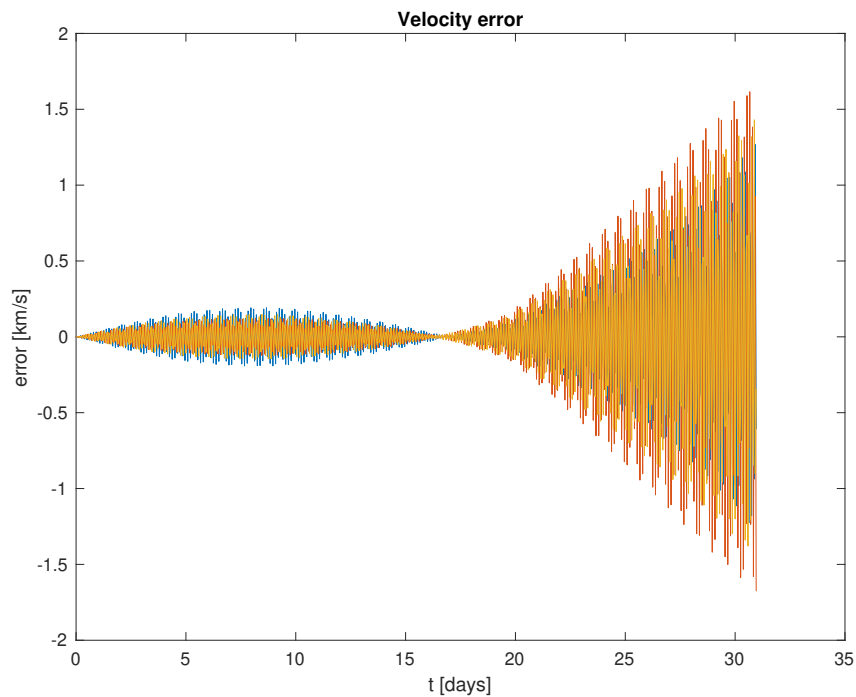
Nonetheless, by observing and comparing the orbital parameters in Table 4, some slight differences regarding the lowest decimal figures can be appreciated.



**Figure 70:** ISS trajectory during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations



**Figure 71:** ISS absolute position error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations



**Figure 72:** ISS absolute velocity error during May 2021 introducing gravitational, atmospheric drag, relativistic, lunar gravity, solar gravity, solar system planets' gravity and SRP perturbations

## 4 Orbital Parameters

When analyzing the results for both the ISS and the satellite LAGEOS-1 in the previous chapter, it has been observed that the trajectory and error plots obtained when adding some perturbations did not affect nor modify the graphs at all in some cases, and hence, no differences between a case studied and the previous one were noticeable.

With the aim of performing a comparison amongst all cases of study in a quantitative manner, the parameters that define the orbit of the satellite are exposed in the present section. From Equation (78) to Equation (97), the expressions that allow to obtain the different orbital parameters, including perigee and apogee radius, as well as period and time from perigee, amongst others, are defined.

- Specific energy,  $\epsilon$  ( $km^2/s^2$ )

$$\epsilon = \frac{v^2}{2} - \frac{\mu}{r} \quad (78)$$

- Semimajor axis,  $a$  ( $km$ )

$$a = -\frac{\mu}{2\epsilon} \quad (79)$$

- Specific angular momentum,  $\vec{h}$  ( $km^2/s$ )

$$\vec{h} = \vec{r} \times \vec{v} \quad (80)$$

- Eccentricity,  $\vec{e}$  ( $-$ )

$$\vec{C} = \vec{v} \times \vec{h} - \mu \frac{\vec{r}}{r} \longrightarrow \vec{e} = \frac{\vec{C}}{\mu} \quad (81)$$

- Perigee radius,  $r_p$  ( $km$ )

$$r_p = a(1 - e) \quad (82)$$

- Apogee radius,  $r_a$  ( $km$ )

$$r_a = a(1 + e) \quad (83)$$

- Orbital parameter,  $p$  ( $km$ )

$$p = \frac{h^2}{\mu} \quad (84)$$

- Tangential velocity,  $v_p$  ( $km/s$ )

$$v_p = \frac{h}{r} \quad (85)$$

- Flight path angle,  $\gamma$  ( $^\circ$ )

$$\begin{cases} \gamma = \arccos\left(\frac{h}{r \cdot v}\right) & \vec{r} \cdot \vec{v} \geq 0 \\ \gamma = -\arccos\left(\frac{h}{r \cdot v}\right) & \vec{r} \cdot \vec{v} < 0 \end{cases} \quad (86)$$

- Radial velocity,  $v_r$  ( $km/s$ )

$$v_r = v_p \cdot \tan \gamma \quad (87)$$

- Inclination,  $i$  ( $^\circ$ )

$$i = \arccos\left(\frac{h_z}{h}\right) \quad (88)$$

- Right Ascension of the Ascending Node,  $\Omega$  ( $^\circ$ )

$$\vec{k} = (0, 0, 1) \longrightarrow \vec{n} = \vec{k} \times \vec{h} \quad (89)$$

$$\begin{cases} \Omega = \arccos\left(\frac{n_x}{n}\right) & n_y \geq 0 \\ \Omega = 360 - \arccos\left(\frac{n_x}{n}\right) & n_y < 0 \end{cases} \quad (90)$$

- Argument of perigee,  $\omega$  ( $^\circ$ )

$$\begin{cases} \omega = \arccos\left(\frac{\vec{n} \cdot \vec{e}}{n \cdot e}\right) & e_z \geq 0 \\ \omega = 360 - \arccos\left(\frac{\vec{n} \cdot \vec{e}}{n \cdot e}\right) & e_z < 0 \end{cases} \quad (91)$$

- True anomaly,  $\theta$  ( $^\circ$ )

$$\begin{cases} \theta = \arccos\left(\frac{\vec{e} \cdot \vec{r}}{e \cdot r}\right) & \vec{r} \cdot \vec{v} \geq 0 \\ \theta = 360 - \arccos\left(\frac{\vec{e} \cdot \vec{r}}{e \cdot r}\right) & \vec{r} \cdot \vec{v} < 0 \end{cases} \quad (92)$$

- Eccentric anomaly,  $E$  ( $^\circ$ )

$$E = 2 \arctan\left(\sqrt{\frac{1-e}{1+e}} \tan \frac{\theta}{2}\right) \quad (93)$$

- Mean anomaly,  $M$  ( $^\circ$ )

$$M = E - e \sin E \quad (94)$$

- Mean motion,  $n_{av}$  ( $rad/s$ )

$$n_{av} = \sqrt{\frac{\mu}{a^3}} \quad (95)$$

- Period,  $T$  ( $s$ )

$$T = \frac{2\pi}{n_{av}} \quad (96)$$

- Time from perigee,  $t - t_p$  ( $s$ )

$$t - t_p = \frac{M}{n_{av}} \quad (97)$$

In order to be able to readily compare and contrast the different cases of study regarding the addition of each of the perturbing accelerations seen along the present document, the orbital parameters have been obtained for both LAGEOS-1 and the International Space Station and shown in Tables 1 - 4.

The first two tables correspond to LAGEOS-1 results, whereas the last two ones contain the data for the ISS. Each column in the tables refers to each of the cases studied in previous sections, corresponding to the addition of one perturbation at a time.

Moreover, the numerical values are obtained with the data concerning the final position of the satellite in its trajectory, that is, for day 31 May, 2021 at 23:00:00. It is worth mentioning that it has been done that way in order to choose the same time instant for all cases, for the purpose of comparing results and extracting some relevant conclusions.

## 4.1 LAGEOS-1

ORBITAL PARAMETERS	2-Body	+ J2-J6	+ Drag	+ Relativity
Specific energy, $\epsilon$ ( $km^2/s^2$ )	-16.2372	-16.241	-16.241	-16.2414
Semimajor axis, $a$ ( $km$ )	12273.8	12270.9	12270.9	12270.6
Specific angular momentum, $h$ ( $km^2/s$ )	69943.1	69934.9	69934.9	69934
Eccentricity, $e$ (-)	0.00428044	0.00409392	0.00409368	0.00410411
Perigee radius, $r_p$ ( $km$ )	12221.2	12220.7	12220.7	12220.2
Apogee radius, $r_a$ ( $km$ )	12326.3	12321.1	12321.1	12320.9
Orbital parameter, $p$ ( $km$ )	12273.6	12270.7	12270.7	12270.4
Tangential velocity, $v_p$ ( $km/s$ )	5.71725	5.71082	5.71081	5.71015
Radial velocity, $v_r$ ( $km/s$ )	0.0158214	0.0203132	0.0203226	0.0207836
Flight path angle, $\gamma$ ( $^\circ$ )	0.158555	0.203798	0.203893	0.208542
Inclination, $i$ ( $^\circ$ )	109.724	109.726	109.726	109.726
RAAN, $\Omega$ ( $^\circ$ )	257.419	267.947	267.947	267.948
Argument of perigee, $\omega$ ( $^\circ$ )	351.294	342.362	342.354	342.136
True anomaly, $\theta$ ( $^\circ$ )	40.4367	60.5272	60.5803	62.6887
Eccentric anomaly, $E$ ( $^\circ$ )	40.2779	60.3232	60.3762	62.48
Mean anomaly, $M$ ( $^\circ$ )	40.2752	60.3197	60.3726	62.4763
Mean motion, $n_{av}$ ( $rad/s$ )	0.000464293	0.000464456	0.000464457	0.000464475
Period, $T$ ( $s$ )	13532.8	13528.1	13528	13527.5
Time from perigee, $t - t_p$ ( $s$ )	5548.34	8119.22	8233.16	12762.1

**Table 1:** Evolution of LAGEOS-1 orbital parameters by adding perturbations I

By having a look at Table 1, it contains the orbital parameters for LAGEOS-1 for the initial two-body problem without accounting for perturbations. Then, gravitational perturbations regarding the Earth's zonal harmonics from  $J_2$  to  $J_6$  are added, followed by the introduction of the atmospheric drag perturbation and the relativistic effects due to the relative movement between the Earth and the satellite and the de-synchronization of time.

As it was expected and seen when analyzing the error plots, the addition of gravitational perturbations is what actually makes the greatest difference amongst the results. Since the order of magnitude of this perturbing acceleration approaches that of the initial acceleration of the satellite, and it is higher than that of any other perturbation, the accuracy and improvement of the results can be clearly discerned.

The addition of drag and relativity perturbations does not have such a relevant impact on the results, since the orbital parameters do not change much with respect to the second column in the table. Indeed, the column corresponding to the addition of atmospheric drag looks pretty similar to the previous one, meaning that the introduction of such perturbation is hardly relevant, as it was expected due to the fact that LAGEOS-1 orbit is around 6000 km above the Earth surface, where the air density is minute.

ORBITAL PARAMETERS	+ Moon	+ Sun	+ Planets	+ SRP
Specific energy, $\epsilon$ ( $km^2/s^2$ )	-16.2415	-16.2415	-16.2415	-16.2415
Semimajor axis, $a$ ( $km$ )	12270.5	12270.5	12270.5	12270.5
Specific angular momentum, $h$ ( $km^2/s$ )	69933.9	69933.9	69933.9	69933.9
Eccentricity, $e$ ( $-$ )	0.00410565	0.00410569	0.00410569	0.00410584
Perigee radius, $r_p$ ( $km$ )	12220.2	12220.2	12220.2	12220.2
Apogee radius, $r_a$ ( $km$ )	12320.9	12320.9	12320.9	12320.9
Orbital parameter, $p$ ( $km$ )	12270.3	12270.3	12270.3	12270.3
Tangential velocity, $v_p$ ( $km/s$ )	5.71014	5.71015	5.71015	5.71014
Radial velocity, $v_r$ ( $km/s$ )	0.0208029	0.0208	0.0208	0.0208062
Flight path angle, $\gamma$ ( $^\circ$ )	0.208736	0.208707	0.208707	0.208769
Inclination, $i$ ( $^\circ$ )	109.729	109.737	109.737	109.737
RAAN, $\Omega$ ( $^\circ$ )	267.956	267.957	267.957	267.957
Argument of perigee, $\omega$ ( $^\circ$ )	342.151	342.185	342.185	342.156
True anomaly, $\theta$ ( $^\circ$ )	62.7502	62.7336	62.7336	62.7624
Eccentric anomaly, $E$ ( $^\circ$ )	62.5412	62.5247	62.5247	62.5534
Mean anomaly, $M$ ( $^\circ$ )	62.5376	62.5211	62.5211	62.5498
Mean motion, $n_{av}$ ( $rad/s$ )	0.000464476	0.000464477	0.000464477	0.000464477
Period, $T$ ( $s$ )	13527.5	13527.5	13527.5	13527.5
Time from perigee, $t - t_p$ ( $s$ )	12893.9	12858.3	12858.3	12920.1

**Table 2:** Evolution of LAGEOS-1 orbital parameters by adding perturbations II

In Table 2, the LAGEOS-1 problem continues to be developed, exposing the orbital parameters that correspond to the addition of the gravity of the Moon, the Sun, and the rest of planets in the solar system, including Pluto. Furthermore, the last column corresponds to the introduction of solar radiation pressure perturbing acceleration to the problem.

Regarding the inclusion of lunar and solar gravity, really slight changes in the results can be appreciated, particularly in decimal figures, since the order of magnitude of these perturbations lies around  $10^{-7}$  times the initial acceleration, which is practically negligible.

In spite of this fact, the improvement is more noticeable than in the case of the addition of solar system planets, as well as SRP as perturbing accelerations, since the order of magnitude of these perturbations is actually lower. Indeed, the column corresponding to the addition of solar gravity is identical to that of the addition of the planets' gravity, and the same thing happened when analyzing the error and trajectory plots. This fact suggests that the computational cost and the time it takes to obtain the results from the numerical integration when adding the gravity of all bodies in the solar system is not worth it, since no evolution in the error nor in the orbital parameters is appreciated.

The numerical values in the solar radiation pressure column, on the other hand, experience a change, even if it is just in decimal figures. Therefore, even though this perturbation is 9 orders of magnitude lower than the initial acceleration of the satellite, it is worth accounting for it.



## 4.2 International Space Station

ORBITAL PARAMETERS	2-Body	+ J2-J6	+ Drag	+ Relativity
Specific energy, $\epsilon$ ( $km^2/s^2$ )	-29.3261	-29.2886	-29.2965	-29.299
Semimajor axis, $a$ ( $km$ )	6795.71	6804.41	6802.57	6802
Specific angular momentum, $h$ ( $km^2/s$ )	52044.7	52078	52070.9	52068.8
Eccentricity, $e$ (-)	0.000606268	0.00137348	0.00175322	0.0017773
Perigee radius, $r_p$ ( $km$ )	6791.59	6795.07	6790.64	6789.91
Apogee radius, $r_a$ ( $km$ )	6799.83	6813.76	6814.5	6814.09
Orbital parameter, $p$ ( $km$ )	6795.71	6804.4	6802.55	6801.98
Tangential velocity, $v_p$ ( $km/s$ )	7.65517	7.65962	7.66398	7.66463
Radial velocity, $v_r$ ( $km/s$ )	-0.00326375	-0.00860201	-0.00962048	-0.00954703
Flight path angle, $\gamma$ ( $^\circ$ )	-0.0244279	-0.064345	-0.0719226	-0.0713673
Inclination, $i$ ( $^\circ$ )	51.5662	51.5952	51.5932	51.5913
RAAN, $\Omega$ ( $^\circ$ )	215.799	62.4425	62.4128	62.41
Argument of perigee, $\omega$ ( $^\circ$ )	106.794	46.0752	61.2103	64.6242
True anomaly, $\theta$ ( $^\circ$ )	224.662	305.085	314.204	315.435
Eccentric anomaly, $E$ ( $^\circ$ )	-135.313	-54.8508	-45.7243	-44.4941
Mean anomaly, $M$ ( $^\circ$ )	-135.313	-54.8497	-45.723	-44.4928
Mean motion, $n_{av}$ ( $rad/s$ )	0.00112696	0.0011248	0.00112525	0.00112539
Period, $T$ ( $s$ )	5575.36	5586.07	5583.8	5583.1
Time from perigee, $t - t_p$ ( $s$ )	2588.62	1510.51	4036.81	5129.47

**Table 3:** Evolution of ISS orbital parameters by adding perturbations I

Moving on to the case of study of the International Space Station, which orbits at around 400 km above the Earth surface, it is interesting to appreciate how, in the same way as for LAGEOS-1, the gravitational perturbations are the ones which most influence the results, due to their relatively high order of magnitude, most notably  $J_2$ .

In contrast with LAGEOS-1, the atmospheric drag perturbation does influence the results in this case, since the ISS is much closer to the Earth surface than LAGEOS-1 is, and at such altitude, the friction caused due to the atmosphere is more noticeable. That is the reason for the shape of the perturbing acceleration due to drag in Figure 13.

Relativistic effects also affect the orbit, but to a lesser extent compared to gravitational and atmospheric drag perturbations. In general, it can be stated that all of the first three perturbations introduced in the calculation of the ISS orbit are quite relevant and greatly contribute to the improvement of the results and consequent reduction of the state vector error.

Even so, it is worth remembering that, when analyzing the error plots for the addition of relativistic effects, during the first fortnight of May, the error was reduced, whereas it considerably increased during the second part of the month, as seen in Figure 59 and 60.

ORBITAL PARAMETERS	+ Moon	+ Sun	+ Planets	+ SRP
Specific energy, $\epsilon$ ( $km^2/s^2$ )	-29.299	-29.299	-29.299	-29.299
Semimajor axis, $a$ ( $km$ )	6802	6802	6802	6802
Specific angular momentum, $h$ ( $km^2/s$ )	52068.8	52068.8	52068.8	52068.8
Eccentricity, $e$ ( $-$ )	0.00177714	0.00177722	0.00177722	0.00178017
Perigee radius, $r_p$ ( $km$ )	6789.92	6789.91	6789.91	6789.89
Apogee radius, $r_a$ ( $km$ )	6814.09	6814.09	6814.09	6814.11
Orbital parameter, $p$ ( $km$ )	6801.98	6801.98	6801.98	6801.98
Tangential velocity, $v_p$ ( $km/s$ )	7.66463	7.66463	7.66463	7.66465
Radial velocity, $v_r$ ( $km/s$ )	-0.00954836	-0.00954901	-0.00954901	-0.00956496
Flight path angle, $\gamma$ ( $^\circ$ )	-0.0713773	-0.0713822	-0.0713822	-0.0715013
Inclination, $i$ ( $^\circ$ )	51.5926	51.5927	51.5927	51.5927
RAAN, $\Omega$ ( $^\circ$ )	62.4082	62.4048	62.4048	62.4048
Argument of perigee, $\omega$ ( $^\circ$ )	64.6329	64.6395	64.6395	64.64
True anomaly, $\theta$ ( $^\circ$ )	315.421	315.42	315.42	315.42
Eccentric anomaly, $E$ ( $^\circ$ )	-44.5071	-44.5082	-44.5082	-44.5086
Mean anomaly, $M$ ( $^\circ$ )	-44.5059	-44.5069	-44.5069	-44.5074
Mean motion, $n_{av}$ ( $rad/s$ )	0.00112539	0.00112539	0.00112539	0.00112539
Period, $T$ ( $s$ )	5583.1	5583.1	5583.1	5583.1
Time from perigee, $t - t_p$ ( $s$ )	5117.88	5116.93	5116.93	5116.51

**Table 4:** Evolution of ISS orbital parameters by adding perturbations II

To finish off, in Table 4, the influence of the gravity of the Moon, the Sun and the rest of solar system planets is included in the shape of perturbations. The resulting orbital parameters obtained for the final position of the satellite are shown.

In the case of the ISS, these last perturbations with such a low order of magnitude have a very low impact on the results, since they are all quite similar and only really slight differences can be appreciated in the last decimal figures. In fact, some parameters do not even change at all when adding these four last perturbing accelerations.

It can be observed, as in the case of LAGEOS-1, that the introduction of the planets' gravity has null influence over the results and the error plots. Therefore, having corroborated this fact with two examples, it can be definitely stated that it is not worth to include this perturbation, since it does not compensate for all the computational cost and time invested.

## 5 Graphical Interface: Orbit Determination

The purpose of this section is to carefully present and explain the design of a user-friendly interface created with the purpose of allowing the user to dynamically select the amount and type of perturbations to account for in the calculations, and display the results, in the shape of a trajectory plot, the state vector error graphs, one for position and one for velocity, and the orbital parameters calculated in the previous section. It also allows for a  $\Delta v$  impulse to take place at a particular time instant throughout the month of May, 2021, setting a given value for the specific impulse,  $I_{sp}$ , with the aim of performing a change in the perigee/apogee of the orbit, or to change the orientation of the orbital plane. It has been created with MATLAB™ App Designer.

The applet is named **Orbit Determination** and can be summed up as: *'Calculation of a satellite's geocentric orbit introducing perturbations and  $\Delta v$  impulses'*. The description of the graphical interface is as follows: *'This applet is designed to obtain the orbit of a satellite, as well as the position and velocity errors and its orbital parameters. It allows the user to select the kind of satellite it is desired to work with, the amount of perturbations included in the calculations depending on the required accuracy, and the  $\Delta v$  value and precise instant at which the satellite is given an impulse in order to change its perigee/apogee or its plane. The calculation comprises from 1 May, 2021 at 00:00 until 31 May, 2021 at 23:00'*.

In order to explain how this applet works and how the results are displayed, several screenshots from the interface have been taken and included from Figure 73 to 78. In the first place, three steps must be followed in the corresponding order so that the orbit to be calculated is properly configured:

1. The user must select the type and amount of perturbations to be included by just ticking the corresponding boxes. For the purpose of the example shown, no perturbations are accounted for.
2. The user can configure, if desired, the addition of an impulse in order to change the trajectory of the satellite. In the sample case, the configuration can be seen, for instance, in Figure 73. It is worth mentioning that the time instant at which the impulse is given must be a multiple of 3600 s, expressed in hours, since that is the time interval applied for the integration. Furthermore, it can only be carried out starting from hour 3, so that the initial trajectory of the satellite, previous to the impulsive maneuver, can be calculated and added to the post-impulse trajectory.
3. Finally, one the two different cases of application studied along this paper must be selected for performing the calculations: either satellite LAGEOS-1 or the International Space Station, examples of GEO and LEO orbits, respectively.

Once the problem is configured and the satellite is selected, the applet starts running and it may take a while to obtain the results, depending on the number of perturbations included and the impulse given. Once MATLAB™ finishes the integration, the results are displayed in the different tabs located on the right-hand side of the interface panel.

The first tab shows the trajectory followed by the satellite along the month of May, 2021. In Figure 73, it can be appreciated how the trajectory has changed due to the  $\Delta v$  impulse introduced in hour 200. The exact location of the satellite at which the impulse has been given is marked in green with a  $\Delta v$  symbol.

The next two tabs show the position and velocity errors obtained. It is worth noticing that the shape of the error plots corresponds to the one studied along this paper up to the instant at which the impulse takes place. From that point on, calculating the error does not make sense, since the state vectors obtained from the calculation do not correspond anymore to the ones extracted from Horizons web interface.

From Figure 76 to 78, the orbital parameters for the final position of the satellite are obtained. The first two orbital parameters tabs refer to the ones explained in the previous section, adding also a new one: the propellant mass needed for executing the impulsive maneuver. This datum is calculated according to Tsiolkovski formula [19], shown in Equation (98).

$$\Delta m = m_i \left( 1 - \text{Exp} \left( - \frac{\Delta v}{I_{sp} \cdot g_0} \right) \right) \quad (98)$$

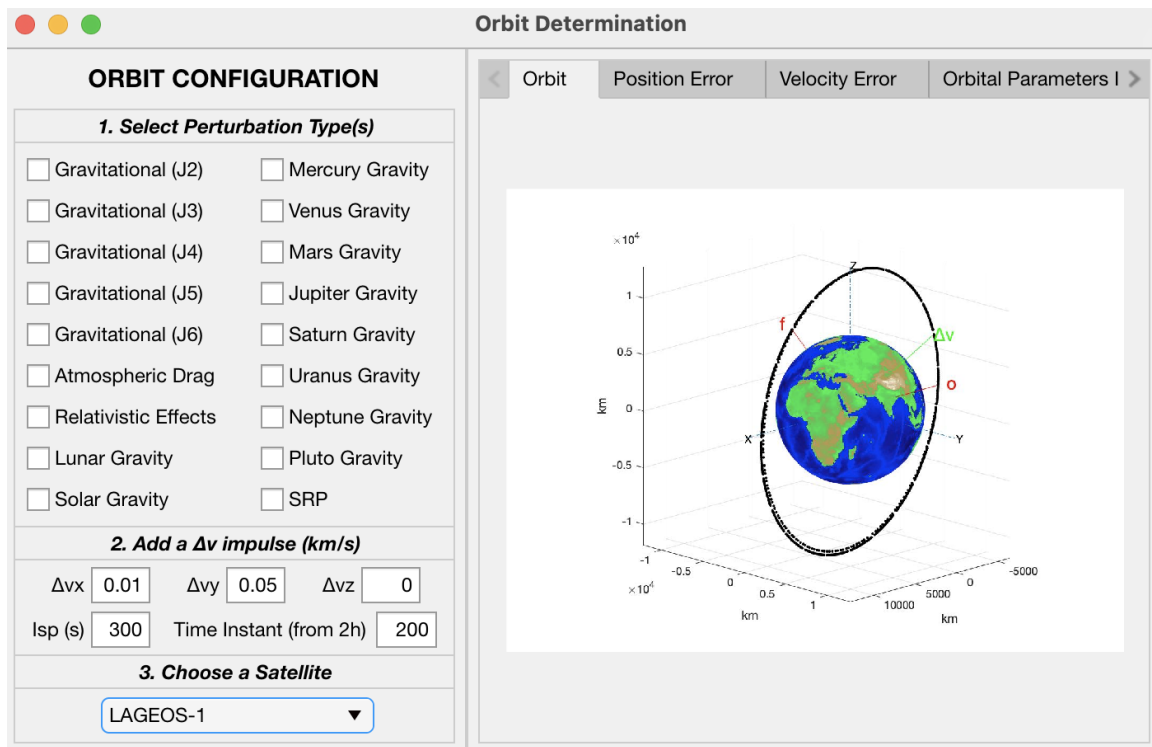


Figure 73: Orbit Determination: Orbit tab

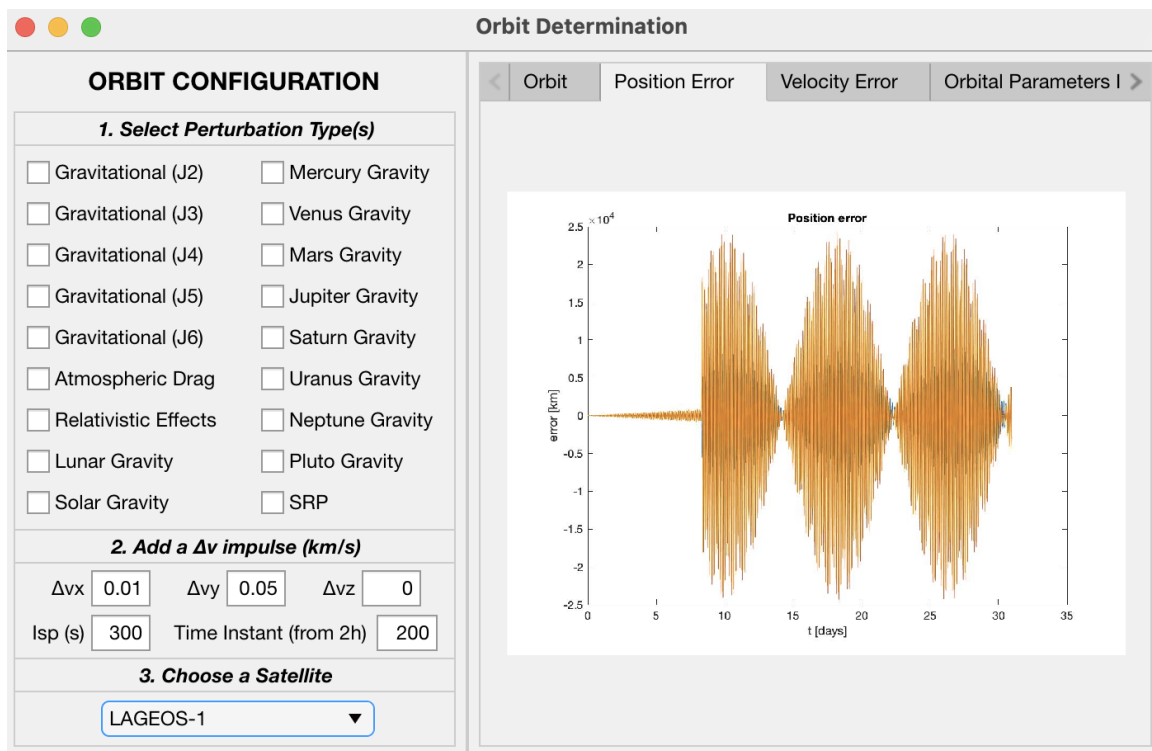


Figure 74: Orbit Determination: Position Error tab

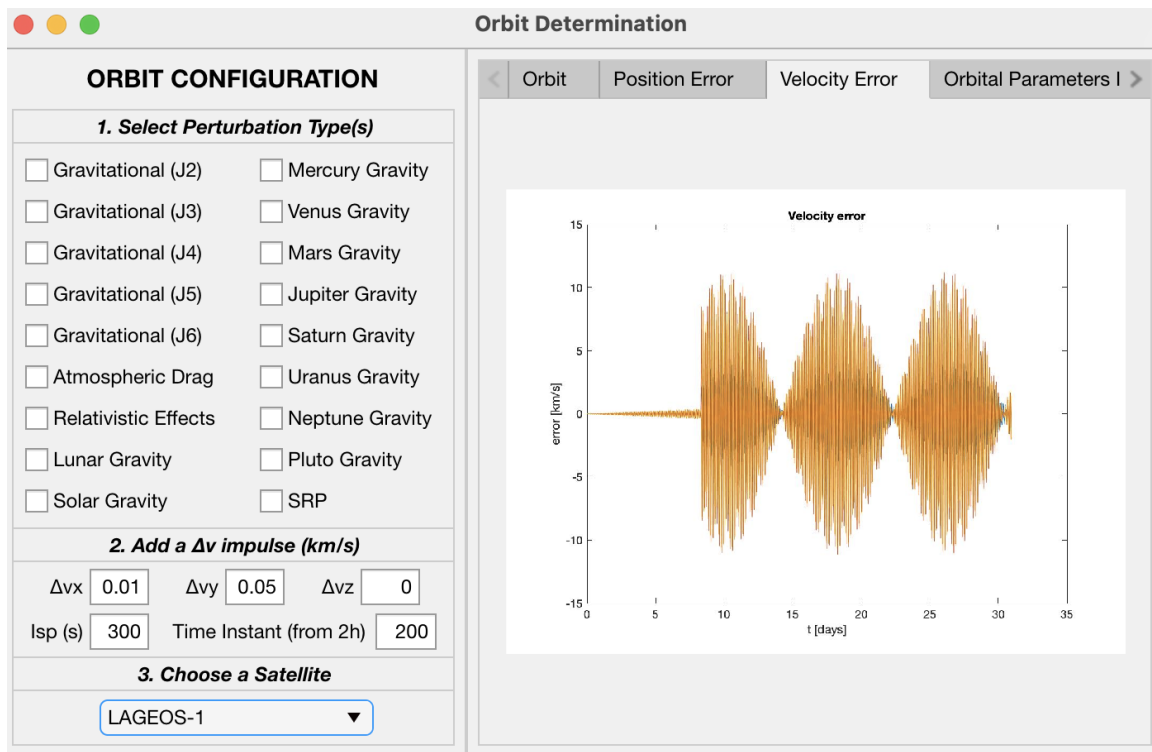


Figure 75: Orbit Determination: Velocity Error tab

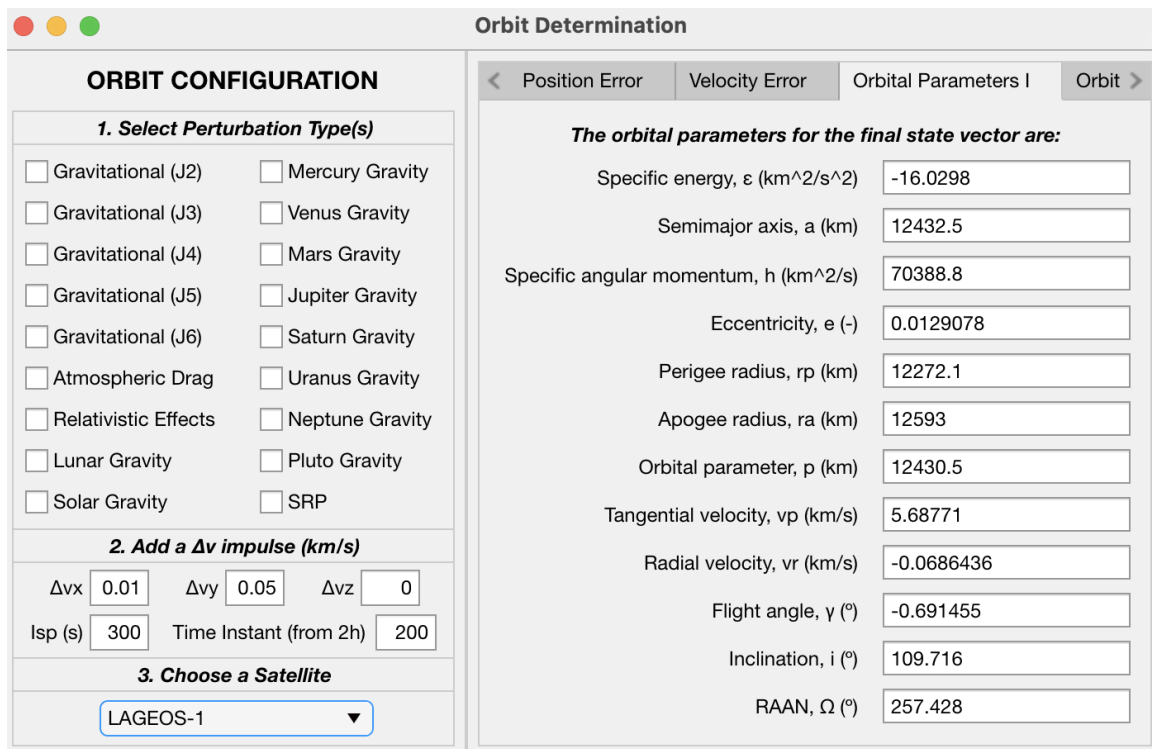


Figure 76: Orbit Determination: Orbital Parameters I tab

Figure 77: Orbit Determination: Orbital Parameters II tab

Figure 78: Orbit Determination: Orbital Parameters III tab

## 6 Conclusions

Once the main points in this document have been fully developed and completed, it is worth stating all the conclusions that have been reached throughout the development of this bachelor's degree thesis in Aerospace Engineering.

The aim of the current paper consisted on exposing and giving a detailed explanation of the process followed in order to obtain the state vector of a satellite orbiting around the Earth with the highest level of precision reachable, according to the temporal and computational limitations encountered along the process.

The problem initially consisted on analyzing the relative motion between two bodies, in particular the Earth and a satellite, setting as a system of reference the Earth-centered frame. For this purpose, two satellites, located on LEO and MEO orbits, were studied, obtaining their state vectors by numerically integrating the fundamental differential equation of relative motion making use of solver `ode45`, already implemented in MATLAB<sup>™</sup>. These bodies were the International Space Station and LAGEOS-1 satellite, and their trajectory was calculated for every hour inside the interval of one month, from 1 May, 2021 at 00:00 until 31 May, 2021 at 23:00.

Once the problem was defined and the results were obtained, the main goal was to start considering more than just the mutual gravitational attraction forces between both bodies, which was achieved by introducing perturbations, which consist of other forces in space that affect the state vector calculation and must be accounted for if more accurate results are required. These were considered by adding several perturbing accelerations to the right-hand side of the differential equation.

The first perturbations added were the gravitational perturbing accelerations caused due to the Earth's oblateness, represented by zonal harmonics from  $J_2$  to  $J_6$ . These were followed by the acceleration coming from the drag forces experienced by the satellite in orbit due to the air density in the Earth's atmosphere. Predictably, the lower the altitude of the orbit over the surface, the more it will be affected by atmospheric drag perturbing acceleration, as it happened for LAGEOS-1 and the ISS. The next perturbation introduced was the one caused by relativistic effects. These are accounted for due to the differences in altitude, as well as in gravitational field, between the Earth and the satellite, which make time de-synchronize. Furthermore, the gravity accelerations of the Moon, Sun and all the planets in the solar system, including Pluto, are introduced as perturbations, following an expression derived from accounting for a third body in the problem. Finally, the last perturbation included is the solar radiation pressure, according to which the photons emitted from the surface of the Sun contain energy which distorts the trajectory of the satellite by the time these photons reach its orbit.

By observing the results, in terms of graphs and orbital parameters, obtained throughout this document by adding each perturbing acceleration at a time, it can be concluded that the purpose of this project has been reached successfully, due to the fact that the error obtained in the state vector of the satellite during one month has considerably diminished with respect to the one obtained when only accounting for the mutual gravitational attraction between both bodies.

The perturbations were added based on their order of magnitude and consequent influence over the results, according to Figure 13. The first one is the one that has the closest order of magnitude to the initial acceleration of the satellite, and thus the one that mostly affects the orbit, apart from drag at low altitudes: the gravitational perturbation due to the Earth oblateness. The last one introduced, on the other hand, is the one with a lowest impact: solar radiation pressure. Notwithstanding, it has been proven that the perturbing acceleration due to the gravity of solar system planets has an even lower order of magnitude than SRP perturbation, which is  $10^{-9}$  times the acceleration of the satellite due to the Earth's gravitational attraction.

It is interesting to include one of the main conclusions reached throughout the analysis carried out in this paper. According to both the trajectory and state vector error plots, as well as the orbital parameters calculated, the introduction of the gravity of Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto has a null influence over the results obtained for the time interval considered and satellites under study, not being appreciated even a small difference in the last decimal figures of the orbital parameters.

This is the reason for stating that including this perturbation in the calculations is not worth in this particular case the computational cost and time devoted to it.

Regarding the limitations encountered throughout the process of including the perturbing accelerations in order to increase accuracy in the results, an important constraint is the time it takes the mathematical robot to perform the required integration. When it comes to the applet designed, when it is installed in MATLAB™ apps and it is executed, the time it will take to carry out the calculations will depend on the speed of the particular computer processing machine in which the applet is installed. It is worth mentioning that even more perturbations could be added to the two-body problem, in order to try to get an error reduction the closest possible to 0.

On a final note, it can be concluded that the Orbit Determination interface achieves the expectations and proposals mentioned at the beginning of this paper, while being at the same time user-friendly and useful to compare and contrast the addition of the different perturbations to the problem of the relative motion between two bodies.



## 7 Future Work

The aim of this section is to propose a continuation to the research and analysis carried out for this project, always looking for an improvement on the process and consequent error reduction and accuracy increase in the results.

In order to do so, a research job is proposed to be carried out, investigating about new ways and methodologies that can be followed and implemented so as to see an evolution of the solution.

In the first place, increasing the applet capacity so as to allow for the selection of any desired satellite could be a good suggestion for future tasks. Moreover, it could be interesting to apply the perturbations studied to satellites in different LEO, MEO and HEO (High Earth Orbit) orbits in order to compare the magnitude of the perturbations amongst the different cases of study. Furthermore, the calculation of perturbations should be optimized as far as possible so that the computational time can be reduced.

As a suggestion for further implementation, a possible perturbation to be accounted for and added to the ones already introduced is that of the seasonal Yarkovsky effect over LAGEOS-1 satellite.

The Yarkovsky effect is an effect caused by sunlight over small asteroids or satellites, which modifies its orbit in space. The side of the satellite that is facing the Sun absorbs the incident radiation, and when it rotates, this radiation is anisotropically emitted [20] as energy or heat, which gives the asteroid a small push, making it deviate from its orbit. This effect has two different components: the diurnal effect and the seasonal effect.

The diurnal effect takes over on large objects, wider than about 100 m. If the direction of rotation of the body around the Sun is the same as that of the object spinning on its axis, then the orbit is called *prograde*. What happens to an object which is on a prograde orbit is that it is pushed away from the Sun. If, by contrast, these directions are opposite to each other, the orbit is called *retrograde*. The Yarkovsky effect over objects in orbits of this type pushes them towards the Sun.

The seasonal effect, in contrast, has an influence over objects that are rather smaller than 100 m, or if its axis of rotation is close to  $90^\circ$ . With this effect, the object gets pushed inwards towards the Sun, no matter the sense of rotation. However, the magnitude of this perturbation is so minute that it actually takes a lot of time for some substantial changes to be noticed [21].

When it comes to analyzing the case of the LAGEOS-1 orbit around the Earth, this body is known to experience the seasonal Yarkovsky effect in the shape of thermal drag which distorts the trajectory calculated for the satellite. Specifically, it lowers its orbit by around one millimeter per day [1]. Even though these variations are hardly noticeable, when applied throughout one entire month, some improvement in the results must be appreciated.

Apart from the Yarkovsky effect, which is only applicable to LAGEOS-1 regarding the cases of study considered in the present thesis, a greater research is suggested to be carried out so that more perturbations or effects of the relative motion of the satellite around the Earth are found and implemented, achieving a greater accuracy in the results, and making sure to depict a precise trajectory for any space mission, in which the cost involved and time devoted to its preparation are factors to be seriously accounted for.

## 8 Technical Specifications

### 8.1 Standard Regulations

This section aims to expose the most relevant legislation concerning from safety and healthcare conditions to the regulations about the working framework for final degree and master thesis at the Polytechnic University of Valencia.

For the purpose of presenting the corresponding legislation, which is originally written in Spanish, the language of origin will be kept, citing the regulations as they appear in the *Boletín Oficial del Estado*.

#### 1. Real Decreto 488/1997, de 14 de abril, sobre disposiciones mínimas de seguridad y salud relativas al trabajo con equipos que incluyen pantallas de visualización [22].

##### Disposiciones mínimas

*Observación preliminar: las obligaciones que se establecen en el presente anexo se aplicarán para alcanzar los objetivos del presente Real Decreto en la medida en que, por una parte, los elementos considerados existan en el puesto de trabajo y, por otra, las exigencias o características intrínsecas de la tarea no se opongan a ello.*

*En la aplicación de lo dispuesto en el presente anexo se tendrán en cuenta, en su caso, los métodos o criterios a que se refiere el apartado 3 del artículo 5 del Real Decreto de los Servicios de Prevención.*

##### 1. Equipo

- (a) *Observación general: La utilización en sí misma del equipo no debe ser una fuente de riesgo para los trabajadores.*
- (b) *Pantalla: Los caracteres de la pantalla deberán estar bien definidos y configurados de forma clara, y tener una dimensión suficiente, disponiendo de un espacio adecuado entre los caracteres y los renglones. La imagen de la pantalla deberá ser estable, sin fenómenos de destellos, centelleos u otras formas de inestabilidad. El usuario de terminales con pantalla deberá poder ajustar fácilmente la luminosidad y el contraste entre los caracteres y el fondo de la pantalla, y adaptarlos fácilmente a las condiciones del entorno. La pantalla deberá ser orientable e inclinable a voluntad, con facilidad para adaptarse a las necesidades del usuario. Podrá utilizarse un pedestal independiente o una mesa regulable para la pantalla. La pantalla no deberá tener reflejos ni reverberaciones que puedan molestar al usuario.*
- (c) *Teclado: El teclado deberá ser inclinable e independiente de la pantalla para permitir que el trabajador adopte una postura cómoda que no provoque cansancio en los brazos o las manos. Tendrá que haber espacio suficiente delante del teclado para que el usuario pueda apoyar los brazos y las manos. La superficie del teclado deberá ser mate para evitar los reflejos. La disposición del teclado y las características de las teclas deberán tender a facilitar su utilización. Los símbolos de las teclas deberán resaltar suficientemente y ser legibles desde la posición normal de trabajo.*
- (d) *Mesa o superficie de trabajo: La mesa o superficie de trabajo deberán ser poco reflectantes, tener dimensiones suficientes y permitir una colocación flexible de la pantalla, del teclado, de los documentos y del material accesorio. El soporte de los documentos deberá ser estable y regulable y estará colocado de tal modo que se reduzcan al mínimo los movimientos incómodos de la cabeza y los ojos. El espacio deberá ser suficiente para permitir a los trabajadores una posición cómoda.*
- (e) *Asiento de trabajo: El asiento de trabajo deberá ser estable, proporcionando al usuario libertad de movimiento y procurándole una postura confortable. La altura del mismo deberá ser regulable. El respaldo deberá ser reclinable y su altura ajustable. Se pondrá un reposapiés a disposición de quienes lo deseen.*

## 2. Entorno

- (a) *Espacio: El puesto de trabajo deberá tener una dimensión suficiente y estar acondicionado de tal manera que haya espacio suficiente para permitir los cambios de postura y movimientos de trabajo.*
  - (b) *Iluminación: La iluminación general y la iluminación especial (lámparas de trabajo), cuando sea necesaria, deberán garantizar unos niveles adecuados de iluminación y unas relaciones adecuadas de luminancias entre la pantalla y su entorno, habida cuenta del carácter del trabajo, de las necesidades visuales del usuario y del tipo de pantalla utilizado. El acondicionamiento del lugar de trabajo y del puesto de trabajo, así como la situación y las características técnicas de las fuentes de luz artificial, deberán coordinarse de tal manera que se eviten los deslumbramientos y los reflejos molestos en la pantalla u otras partes del equipo.*
  - (c) *Reflejos y deslumbramientos: Los puestos de trabajo deberán instalarse de tal forma que las fuentes de luz, tales como ventanas y otras aberturas, los tabiques transparentes o translúcidos y los equipos o tabiques de color claro no provoquen deslumbramiento directo ni produzcan reflejos molestos en la pantalla. Las ventanas deberán ir equipadas con un dispositivo de cobertura adecuado y regulable para atenuar la luz del día que ilumine el puesto de trabajo.*
  - (d) *Ruido: El ruido producido por los equipos instalados en el puesto de trabajo deberá tenerse en cuenta al diseñar el mismo, en especial para que no se perturbe la atención ni la palabra.*
  - (e) *Calor: Los equipos instalados en el puesto de trabajo no deberán producir un calor adicional que pueda ocasionar molestias a los trabajadores.*
  - (f) *Emisiones: Toda radiación, excepción hecha de la parte visible del espectro electromagnético, deberá reducirse a niveles insignificantes desde el punto de vista de la protección de la seguridad y de la salud de los trabajadores.*
  - (g) *Humedad: Deberá crearse y mantenerse una humedad aceptable.*
3. *Interconexión ordenador/persona: Para la elaboración, la elección, la compra y la modificación de programas, así como para la definición de las tareas que requieran pantallas de visualización, el empresario tendrá en cuenta los siguientes factores:*
- (a) *El programa habrá de estar adaptado a la tarea que deba realizarse.*
  - (b) *El programa habrá de ser fácil de utilizar y deberá, en su caso, poder adaptarse al nivel de conocimientos y de experiencia del usuario; no deberá utilizarse ningún dispositivo cuantitativo o cualitativo de control sin que los trabajadores hayan sido informados y previa consulta con sus representantes.*
  - (c) *Los sistemas deberán proporcionar a los trabajadores indicaciones sobre su desarrollo.*
  - (d) *Los sistemas deberán mostrar la información en un formato y a un ritmo adaptados a los operadores.*
  - (e) *Los principios de ergonomía deberán aplicarse en particular al tratamiento de la información por parte de la persona.*

## 2. NORMATIVA MARCO DE TRABAJOS FIN DE GRADO Y FIN DE MÁSTER UNIVERSITAT POLITÈCNICA DE VALÈNCIA [23]

### Artículo 2. *Ámbito de aplicación*

1. *La presente normativa será de aplicación a las enseñanzas impartidas por la UPV conducentes a la obtención de los títulos de Grado y Máster Universitario de carácter oficial y validez en todo el territorio nacional (en adelante títulos oficiales).*
2. *La presente normativa será de aplicación a los estudiantes de la UPV de los títulos oficiales interuniversitarios, salvo que en el correspondiente convenio o en la memoria de verificación de dichos títulos se establezcan explícitamente otras disposiciones al respecto.*

3. *Las Estructuras Responsables de los Títulos oficiales (ERT) podrán desarrollar sus propias normativas o reglamentos que complementen lo que se indica en la presente Normativa Marco, que en todo caso prevalecerá en caso de conflicto entre ambas. Las normas o reglamentos que establezcan las ERT deberán ser aprobadas por la Comisión Académica del Consejo de Gobierno de la UPV con anterioridad a su entrada en vigor.*
4. *Las ERT darán publicidad a la presente normativa y, cuando haga el caso, a las normativas propias complementarias, a través de los medios que consideren oportunos con el objeto de que sean conocidas por todos los estudiantes.*
5. *Los TFG y TFM de los títulos oficiales que habiliten para el ejercicio de las profesiones reguladas se registrarán por lo dispuesto en la correspondiente Orden Ministerial que establece los requisitos para la verificación del título, sin perjuicio de la aplicación, con carácter complementario, de lo que se indique en la presente Normativa Marco.*
6. *La denominación TFG o TFM empleada en esta Normativa Marco debe entenderse de aplicación incluso en aquellas titulaciones en las que las órdenes ministeriales dispongan denominaciones alternativas para señalar módulos o materias a las que, de forma genérica, se hace referencia como TFG o TFM en el RD 1393/2007.*

### **Artículo 3. Naturaleza de los TFG y TFM**

1. *Los TFG y TFM deberán estar orientados a la aplicación y evaluación de competencias asociadas al título.*
2. *En el caso de los TFG y en el de los TFM de títulos que habiliten para el ejercicio de profesiones reguladas deberán tener una orientación profesional. En el resto de casos, el TFM podrá tener orientación profesional o investigadora.*
3. *Los TFG y TFM consistirán en la realización de un trabajo o proyecto original en el que queden de manifiesto conocimientos, habilidades y competencias adquiridas por el estudiante a lo largo de sus estudios y, expresamente, las competencias asociadas a la materia TFG o TFM, tal y como se indique en la memoria de verificación.*
4. *La originalidad del trabajo a que se hace referencia en el punto anterior, debe entenderse sin menoscabo de que pueda ser parte independiente e individual de un trabajo integral desarrollado de manera conjunta entre estudiantes de una misma titulación o de diferentes titulaciones y ERT. En cualquier caso, la defensa del TFG y del TFM debe ser individual.*
5. *La materia TFG y TFM podrá organizarse mediante actividades de docencia reglada en forma de seminario, taller o similar; mediante trabajo autónomo y tutelado del estudiante; o mediante una mezcla de ambas.*
6. *El alcance, contenido y nivel de exigencia de los TFG y TFM deberá adecuarse a la asignación de ECTS que dicha materia haya recibido en la memoria de verificación. A tal efecto, las ERT velarán para que el tiempo de dedicación requerido para la realización del TFG o TFM se adecúe al número de ECTS asignados al mismo y pueda ser evaluado en el periodo académico previsto en la estructura del plan de estudios.*
7. *Como cualquier otra materia de un plan de estudios, los TFG y TFM deberán disponer de una Guía Docente en la que, con los contenidos y alcance que determine la UPV y de acuerdo con lo indicado en la memoria de verificación del título, deberán constar todos aquellos aspectos que orienten el trabajo del estudiante. En la Guía docente deberá especificarse qué parte de los ECTS asignados a la materia TFG o TFM de desarrollarán mediante docencia reglada y qué parte mediante trabajo autónomo y tutelado del estudiante. Los directores académicos del título serán los responsables de la confección de dicha Guía, salvo que la totalidad de los ECTS se organicen mediante docencia reglada, en cuyo caso la responsabilidad de la Guía recaerá en el departamento al que se asigne dicha docencia.*

## 8.2 Work Conditions

The working environment in which the present thesis has been carried out meets all the specifications given by the laws and regulations exposed in the previous subsection. It basically consists of a quiet neat room, in which it is possible to attain the required level of concentration, avoiding distractions to the extent possible. In this room, a desk and an armchair with a high back for greater comfort are placed. The light is adequate and the room temperature is adjustable by means of an air conditioning device.

The main computing machine which has been used for the development of this project is a laptop, connected to the internet through a 600 mbps symmetrical connection.

Regarding the hardware and software used for the purpose of performing the research work, the resolution of the numerical integration of each of the cases of study and the display of results, these are presented hereafter.

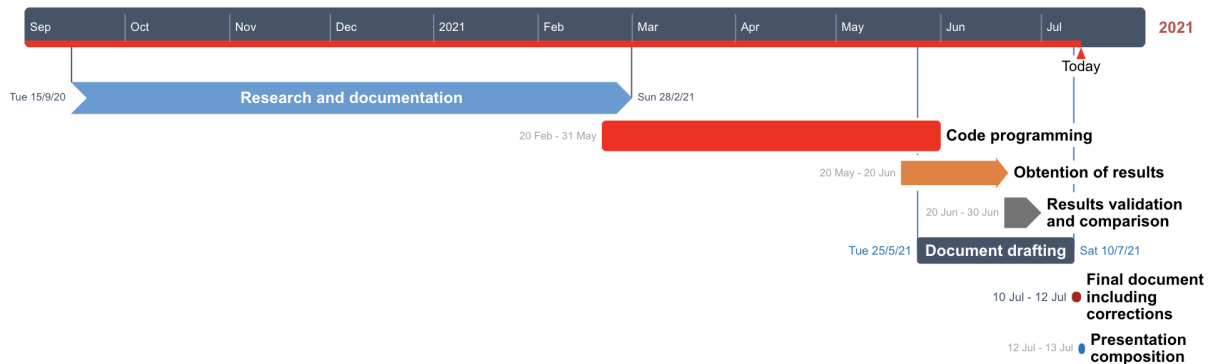
- **Hardware:** MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)
  - Operative System macOS Big Sur, version 11.2.3 (20D91)
  - Processor 2,4 GHz Quad-Core Intel Core i5
  - Memory 8 GB 2133 MHz LPDDR3
  - Startup Disk Macintosh HD
  - Graphics Intel Iris Plus Graphics 655 1536 MB
  - Serial Number C02Z52UALVDG
  - Built-in Retina Display 13,3-inch (2560 × 1600)
  - Storage 251 GB Flash Storage
- **Software:**
  - MATLAB™ R2019b: UPV student license
  - Microsoft Office® 365 package: UPV student license
  - Overleaf®: LaTeX open source

## 9 Project Budget

The purpose of this section is to specify the total budget resulting from the development of this final degree project, accounting for the cost of software licenses, the cost of depreciation of the required hardware and the personnel costs, as well.

It is worth mentioning that the rates of depreciation introduced are the ones dictated by the Inland Revenue, according to [24].

Before getting into detail about the costs of the different issues related to the project, it is interesting to include a Gantt Chart of the process of developing and completing the bachelor's degree thesis, which can be seen in Figure 79. It shows a timeline of the different tasks in which the project has been divided, and the amount of time devoted to the realisation of each of them along the academic course 2020-2021.



**Figure 79:** Gantt Chart of the final degree project

The project budget can be classified, as it has already been stated, into personnel, hardware and software costs.

### 9.1 Personnel Costs

These costs account for the working hours of all researchers included in the process, which are the author of the project and the tutor in charge. The cost per hour of each of both contributors is different, according to the position and experience of the person at issue. The price per hour of the graduated aerospace engineer is estimated to be 20 €, whereas for the supervisor, the cost is around 40 €/h.

Concept	Hours	Price per hour of graduate engineer (€)	Subtotal (€)
Research and documentation	350	20	7000
Code programming	200	20	4000
Obtention of results	20	20	400
Results validation and comparison	30	20	600
Document drafting	200	20	4000
Final document	20	20	400
Presentation composition	15	20	300
<b>Total</b>	<b>835</b>	<b>20</b>	<b>16700</b>

**Table 5:** Breakdown of graduate engineer expenses

Concept	Hours	Price per hour of supervisor (€)	Subtotal (€)
Tutorial hours	15	40	600
Revision of code	10	40	400
Revision of document and presentation	20	40	800
Total	45	40	1800

**Table 6:** Breakdown of supervisor expenses

In Table 5 and 6, each task is described and the hours needed to perform it are estimated in order to calculate the total personnel costs involved. It is worth commenting that the code programming task includes both the development of MATLAB™ code and the design and programming of the applet interface.

## 9.2 Hardware Costs

In this subsection, the cost of purchasing the hardware needed, a MacBook Pro® laptop in this case, is detailed in Table 7, in which the depreciation costs must be taken into consideration.

Hardware	Cost (€)	Depreciation rate (%)	Total Usage Time (months)	Subtotal (€)
MacBook Pro®	2099	25	10	437.29

**Table 7:** Breakdown of hardware costs

## 9.3 Software Costs

The final costs to be taken into account for estimating the total project budget are those related to software licenses, for which the depreciation rates are also considered, since they are only used for a certain amount of time. The breakdown of prices is shown in Table 8. It is interesting to add that the price of licenses is expressed as an annual cost.

Software	Cost (€)	Depreciation rate (%)	Total Usage Time (months)	Subtotal (€)
MATLAB™ R2019b	800	33	10	220
Microsoft Office® 365	69	33	10	18.98
Overleaf®	0	33	10	0
Total	869	33	10	238.98

**Table 8:** Breakdown of software costs

## 9.4 Total Budget

The last step consists of adding up all costs obtained throughout this section, and applying the corresponding percentage due to commercial benefits and taxes. The result is exposed in Table 9, being the project worth a final market price of **24595.49 €**.

<b>Concept</b>	<b>Amount (€)</b>
Graduate engineer costs	16700
Supervisor costs	1800
Hardware costs	437.29
Software costs	238.98
<b>Total industrial costs</b>	<b>19176.27</b>
Industrial benefit (6 %) [25]	1150.58
<b>Total costs with benefit</b>	<b>20326.85</b>
IVA (21 %)	4268.64
<b>Total market cost</b>	<b>24595.49</b>

**Table 9:** Total costs including taxes



## References

- [1] Elizabeth Zubritsky. *Now 40, NASA's LAGEOS Set the Bar for Studies of Earth*. NASA's Goddard Space Flight Center, Greenbelt, Md., 2017.
- [2] NASA Jet Propulsion Laboratory. HORIZONS Web-Interface. <https://ssd.jpl.nasa.gov/horizons.cgi>. [Online; accessed 16-June-2021].
- [3] Instructor Robert Frost and Flight Controller in the Flight Operations Directorate at NASA. *Why can you see the space station on some days but not on others?* Quora, December 24, 2019.
- [4] Charles D. Brown. *Spacecraft Mission Design*. American Institute of Aeronautics and Astronautics, Inc., 1801 Alexander Bell Drive, Reston, VA 20191, 2 edition, 1998.
- [5] Howard D. Curtis. *Orbital Mechanics For Engineering Students*. Elsevier, The Boulevard, Langford Lane, Kidlington, Oxford, OX5 1GB, UK, 3 edition, 2014.
- [6] Matthias Meschede Mark Wieczorek. *SHTools: Tools for Working with Spherical Harmonics*. Geochemistry, Geophysics, Geosystems, AGU and the Geochemical Society, 2018.
- [7] Will Campbell. *Earth-sized Sphere with Topography*. MATLAB Central File Exchange, July 8, 2021.
- [8] Herbert J. Kramer. *Observation of the Earth and Its Environment: Survey of Missions and Sensors*. Springer, 4 edition, 2002.
- [9] Vladimir A. Chobotov. *Orbital Mechanics*. American Institute of Aeronautics and Astronautics, Inc., 1801 Alexander Bell Drive, Reston, Virginia 20191-4344, 3 edition, 2002.
- [10] R.H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA, Education Series, Reston, Virginia, revised edition, 1999.
- [11] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press/Springer, 3 edition, 2007.
- [12] Ibrahim Sanad. *Design of Remote Sensing Satellite Orbit*. ResearchGate, 2013.
- [13] John L. Junkins Hanspeter Schaub. *Analytical Mechanics of Space Systems*. American Institute of Aeronautics and Astronautics, Inc., 1801 Alexander Bell Drive, Reston, VA 20191-4344, 2 edition, 2009.
- [14] Željko Hećimović. Relativistic effects on satellite navigation. *Tehnički vjesnik/Technical Gazette*, February, 2013.
- [15] Dr. Elmar Wasle Dr. Bernhard Hofmann-Wellenhof, Dr. Herbert Lichtenegger. *GNSS – Global Navigation Satellite Systems*. SpringerWienNewYork, 2008.
- [16] A Milani, A M Nobill, and P Farinella. *Non-gravitational perturbations and satellite geodesy*. CRC Press, United States, 1 edition, 1987.
- [17] Daniel Kucharski Akihisa Hattori, Toshimichi Otsubo. *Precise modeling of solar radiation pressure acceleration for spherical geodetic satellites*. SOKENDAI (The Graduate University for Advanced Studies), Hitotsubashi University, Space Environment Research Centre (SERC), 2018.
- [18] *ATV: a very special delivery - Lesson notes*. European Space Agency.
- [19] Santiago Emmanuel Moll López. *Maniobras Impulsivas*. Universitat Politècnica de València, 2020.
- [20] S. R. Chesley D. Vokrouhlicky, A. Milani. *Yarkovsky Effect on Small Near-Earth Asteroids: Mathematical Formulation and Examples*. Academic Press, 2000.
- [21] Josie A. Peters. *The Yarkovsky Effect : What is it and why should I care?* 2016.
- [22] JAVIER ARENAS BOCANEGRA. *Real Decreto 488/1997, de 14 de abril, sobre disposiciones mínimas de seguridad y salud relativas al trabajo con equipos que incluyen pantallas de visualización*. Agencia Estatal, Gobierno de España, 1997.

- [23] *NORMATIVA MARCO DE TRABAJOS FIN DE GRADO Y FIN DE MÁSTER UNIVERSITAT POLITÈCNICA DE VALÈNCIA*. Universitat Politècnica de València, 2013.
- [24] *Tabla de coeficientes de amortización lineal*. Agencia Tributaria, Gobierno de España, 2021.
- [25] Álvaro Osta Carbonell. *La administración y su "imposición" en el ICIO*. Hayderecho, Expansión, 2019.

# Appendices

## A MATLAB code

### A.1 General code

Listing 1: Atmosphere function for drag perturbation [5]

```

1 function density = atmosphere(height)
2
3 % ATMOSPHERE calculates density for altitudes from sea level through
4   1000 km using exponential interpolation.
5
6 % Geometric altitudes (km):
7
8 hache = [0 25 30 40 50 60 70 80 90 100 110 120 130 140 150 180 200 250
9   300 350 400 450 500 600 700 800 900 1000];
10
11 % Corresponding densities (kg/m^3) from USSA76:
12
13 erre = [1.225 4.008e-2 1.841e-2 3.996e-3 1.027e-3 3.097e-4 8.283e-5
14   1.846e-5 3.416e-6 5.606e-7 9.708e-8 2.222e-8 8.152e-9 3.831e-9 2.076e
15   -9 5.194e-10 2.541e-10 6.073e-11 1.916e-11 7.014e-12 2.803e-12 1.184e
16   -12 5.215e-13 1.137e-13 3.070e-14 1.136e-14 5.759e-15 3.561e-15];
17
18 % Scale heights (km):
19
20 Hache = [7.310 6.427 6.546 7.360 8.342 7.583 6.661 5.927 5.533 5.703
21   6.782 9.973 13.243 16.322 21.652 27.974 34.934 43.342 49.755 54.513
22   58.019 60.980 65.654 76.377 100.587 147.203 208.020];
23
24 % Handle altitudes outside of the range:
25
26 if height > 1000
27     height = 1000;
28 elseif height < 0
29     height = 0;
30 end
31
32 % Determine the interpolation interval:
33
34 for j = 1:27
35     if height >= hache(j) && height < hache(j+1)
36         i = j;
37     end
38 end
39
40 if height == 1000
41     i = 27;
42 end
43
44 % Exponential interpolation:
45
46 density = erre(i)*exp(-(height - hache(i))/Hache(i));
47
48 end

```

Listing 2: Function to draw an earth-sized sphere [7]

```

1 function [xx,yy,zz] = earth_sphere(varargin)
2 %EARTH_SPHERE Generate an earth-sized sphere.
3 % [X,Y,Z] = EARTH_SPHERE(N) generates three (N+1)-by-(N+1)

```

```

4 % matrices so that SURFACE(X,Y,Z) produces a sphere equal to
5 % the radius of the earth in kilometers. The continents will be
6 % displayed.
7 %
8 % [X,Y,Z] = EARTH_SPHERE uses N = 50.
9 %
10 % EARTH_SPHERE(N) and just EARTH_SPHERE graph the earth as a
11 % SURFACE and do not return anything.
12 %
13 % EARTH_SPHERE(N,'mile') graphs the earth with miles as the unit
    rather
14 % than kilometers. Other valid inputs are 'ft' 'm' 'nm' 'miles' and '
    AU'
15 % for feet, meters, nautical miles, miles, and astronomical units
16 % respectively.
17 %
18 % EARTH_SPHERE(AX,...) plots into AX instead of GCA.
19 %
20 % Examples:
21 %     earth_sphere('nm') produces an earth-sized sphere in nautical miles
22 %
23 %     earth_sphere(10,'AU') produces 10 point mesh of the Earth in
24 %     astronomical units
25 %
26 %     h1 = gca;
27 %     earth_sphere(h1,'mile')
28 %     hold on
29 %     plot3(x,y,z)
30 %     produces the Earth in miles on axis h1 and plots a trajectory
    from
31 %     variables x, y, and z
32 %     Clay M. Thompson 4-24-1991, CBM 8-21-92.
33 %     Will Campbell, 3-30-2010
34 %     Copyright 1984-2010 The MathWorks, Inc.
35 %% Input Handling
36 [cax,args,nargs] = axescheck(varargin{:}); % Parse possible Axes input
37 error(nargchk(0,2,nargs)); % Ensure there are a valid number of inputs
38 % Handle remaining inputs.
39 % Should have 0 or 1 string input, 0 or 1 numeric input
40 j = 0;
41 k = 0;
42 n = 50; % default value
43 units = 'km'; % default value
44 for i = 1:nargs
45     if ischar(args{i})
46         units = args{i};
47         j = j+1;
48     elseif isnumeric(args{i})
49         n = args{i};
50         k = k+1;
51     end
52 end
53 if j > 1 || k > 1
54     error('Invalid input types')
55 end

```

```

56 %% Calculations
57 % Scale factors
58 Scale = {'km' 'm' 'mile' 'miles' 'nm'
           'au' 'ft'};
59         1      1000 0.621371192237334 0.621371192237334 0.539956803455724
           6.6845871226706e-009 3280.839895};
60 % Identify which scale to use
61 try
62     myscale = 6378.1363*Scale{2,strcmpi(Scale(1,:),units)};
63 catch %#ok<*CTCH>
64     error('Invalid units requested. Please use m, km, ft, mile, miles,
           nm, or AU')
65 end
66
67 % -pi <= theta <= pi is a row vector.
68 % -pi/2 <= phi <= pi/2 is a column vector.
69 theta = (-n:2:n)/n*pi;
70 phi = (-n:2:n)'/n*pi/2;
71 cosphi = cos(phi); cosphi(1) = 0; cosphi(n+1) = 0;
72 sintheta = sin(theta); sintheta(1) = 0; sintheta(n+1) = 0;
73 x = myscale*cosphi*cos(theta);
74 y = myscale*cosphi*sintheta;
75 z = myscale*sin(phi)*ones(1,n+1);
76 %% Plotting
77 if nargout == 0
78     cax = newplot(cax);
79     % Load and define topographic data
80     load('topo.mat','topo','topomap1');
81     % Rotate data to be consistent with the Earth-Centered-Earth-Fixed
82     % coordinate conventions. X axis goes through the prime meridian.
83     % http://en.wikipedia.org/wiki/Geodetic\_system#
           Earth\_Centred\_Earth\_Fixed\_.28ECEF\_or\_ECF.29\_coordinates
84     %
85     % Note that if you plot orbit trajectories in the Earth-Centered-
86     % Inertial, the orientation of the continents will be misleading.
87     topo2 = [topo(:,181:360) topo(:,1:180)]; %#ok<NODEF>
88
89     % Define surface settings
90     props.FaceColor = 'texture';
91     props.EdgeColor = 'none';
92     props.FaceLighting = 'phong';
93     props.Cdata = topo2;
94     % Create the sphere with Earth topography and adjust colormap
95     surface(x,y,z,props,'parent',cax)
96     colormap(topomap1)
97 % Replace the calls to surface and colormap with these lines if you do
98 % not want the Earth's topography displayed.
99 %     surf(x,y,z,'parent',cax)
100 %     shading flat
101 %     colormap gray
102
103 % Refine figure
104 axis equal
105 xlabel(['X [' units ']]')
106 ylabel(['Y [' units ']]')

```

```

107     xlabel(['Z [' units ']]')
108     view(127.5,30)
109 else
110     xx = x; yy = y; zz = z;
111 end
112 end

```

## A.2 Interpolating polynomials

**Listing 3:** Script to obtain the interpolating polynomial of the position of the Moon as seen from the Earth

```

1  clc
2  close all
3  clear all
4
5  hours = 3600;
6  t0 = 0;
7  tf = 31*24*hours;
8  tspan = (t0:1*hours:(tf-1));
9
10 load('moonmatrix.mat')
11 rm = horizonsresultsMOON(51:794, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rm(:,1), 20);
15 rmx = 0;
16
17 for j = 1:length(polx)
18     rmx = t.^(j-1)*polx(length(polx)+1-j)+rmx;
19 end
20
21 errorx = rm(:,1)-rmx;
22
23 poly = polyfit(tspan', rm(:,2), 20);
24 rmy = 0;
25
26 for l = 1:length(poly)
27     rmy = t.^(l-1)*poly(length(poly)+1-l)+rmy;
28 end
29
30 errory = rm(:,2)-rmy;
31
32 polz = polyfit(tspan', rm(:,3), 20);
33 rmz = 0;
34
35 for p = 1:length(polz)
36     rmz = t.^(p-1)*polz(length(polz)+1-p)+rmz;
37 end
38
39 errorz = rm(:,3)-rmz;
40
41 syms x y z
42 rmsx = rmx-x;
43 rmsy = rmy-y;
44 rmsz = rmz-z;
45

```

```

46 rmnorm = norm([rmx rmy rmz]);
47 rmsnorm = norm([rmsx rmsy rmsz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rmx),'r','LineWidth',2)
54 hold on
55 plot(tspan'/(24*hours), subs(rmy),'b','LineWidth',2)
56 hold on
57 plot(tspan'/(24*hours), subs(rmz),'g','LineWidth',2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Moon Position')
63 legend('r_{moon,x}','r_{moon,y}','r_{moon,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rmnorm),'r','LineWidth',2)

```

**Listing 4:** Script to obtain the interpolating polynomial of the position of the Sun as seen from the Earth

```

1 clc
2 close all
3 clear all
4
5 hours = 3600;
6 t0 = 0;
7 tf = 31*24*hours;
8 tspan = (t0:1*hours:(tf-1));
9
10 load('sunmatrix.mat')
11 rs = horizonsresultsSUN(50:793, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rs(:,1), 20);
15 rsx = 0;
16
17 for j = 1:length(polx)
18     rsx = t.^(j-1)*polx(length(polx)+1-j)+rsx;
19 end
20
21 errorx = rs(:,1)-rsx;
22
23 poly = polyfit(tspan', rs(:,2), 20);
24 rsy = 0;
25
26 for l = 1:length(poly)
27     rsy = t.^(l-1)*poly(length(poly)+1-l)+rsy;
28 end
29
30 errory = rs(:,2)-rsy;
31
32 polz = polyfit(tspan', rs(:,3), 20);

```

```

33 rsz = 0;
34
35 for p = 1:length(polz)
36     rsz = t.^(p-1)*polz(length(polz)+1-p)+rsz;
37 end
38
39 errorz = rs(:,3)-rsz;
40
41 syms x y z
42 rssx = rsx-x;
43 rssi = rsi-y;
44 rssi = rsz-z;
45
46 rsnorm = norm([rsx rsi rsi]);
47 rsnorm = norm([rssx rssi rssi]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rsx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rsi), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rsz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Sun Position')
63 legend('r_{sun,x}', 'r_{sun,y}', 'r_{sun,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rsnorm), 'r', 'LineWidth', 2)

```

**Listing 5:** Script to obtain the interpolating polynomial of the position of Mercury as seen from the Earth

```

1 clc
2 close all
3 clear all
4
5 hours = 3600;
6 t0 = 0;
7 tf = 31*24*hours;
8 tspan = (t0:1*hours:(tf-1));
9
10 load('mercurymatrix.mat')
11 rmercury = horizonsresultsMERCURY(50:793, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rmercury(:,1), 20);
15 rmercuryx = 0;
16
17 for j = 1:length(polx)
18     rmercuryx = t.^(j-1)*polx(length(polx)+1-j)+rmercuryx;
19 end

```



```

20
21 errorx = rmercury(:,1)-rmercuryx;
22
23 poly = polyfit(tspan', rmercury(:,2), 20);
24 rmercuryy = 0;
25
26 for l = 1:length(poly)
27     rmercuryy = t.^(l-1)*poly(length(poly)+1-l)+rmercuryy;
28 end
29
30 errory = rmercury(:,2)-rmercuryy;
31
32 polz = polyfit(tspan', rmercury(:,3), 20);
33 rmercuryz = 0;
34
35 for p = 1:length(polz)
36     rmercuryz = t.^(p-1)*polz(length(polz)+1-p)+rmercuryz;
37 end
38
39 errorz = rmercury(:,3)-rmercuryz;
40
41 syms x y z
42 rmercurysx = rmercuryx-x;
43 rmercurysy = rmercuryy-y;
44 rmercurysz = rmercuryz-z;
45
46 rmercurynorm = norm([rmercuryx rmercuryy rmercuryz]);
47 rmercurysnorm = norm([rmercurysx rmercurysy rmercurysz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rmercuryx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rmercuryy), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rmercuryz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Mercury Position')
63 legend('r_{mercury,x}', 'r_{mercury,y}', 'r_{mercury,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rmercurynorm), 'r', 'LineWidth', 2)

```

**Listing 6:** Script to obtain the interpolating polynomial of the position of Venus as seen from the Earth

```

1 clc
2 close all
3 clear all
4
5 hours = 3600;
6 t0 = 0;

```

```
7 tf = 31*24*hours;
8 tspan = (t0:1*hours:(tf-1));
9
10 load('venusmatrix.mat')
11 rvenus = horizonsresultsVENUS(50:793, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rvenus(:,1), 20);
15 rvenusx = 0;
16
17 for j = 1:length(polx)
18     rvenusx = t.^(j-1)*polx(length(polx)+1-j)+rvenusx;
19 end
20
21 errorx = rvenus(:,1)-rvenusx;
22
23 poly = polyfit(tspan', rvenus(:,2), 20);
24 rvenusy = 0;
25
26 for l = 1:length(poly)
27     rvenusy = t.^(l-1)*poly(length(poly)+1-l)+rvenusy;
28 end
29
30 errory = rvenus(:,2)-rvenusy;
31
32 polz = polyfit(tspan', rvenus(:,3), 20);
33 rvenusz = 0;
34
35 for p = 1:length(polz)
36     rvenusz = t.^(p-1)*polz(length(polz)+1-p)+rvenusz;
37 end
38
39 errorz = rvenus(:,3)-rvenusz;
40
41 syms x y z
42 rvenussx = rvenusx-x;
43 rvenussy = rvenusy-y;
44 rvenussz = rvenusz-z;
45
46 rvenusnorm = norm([rvenusx rvenusy rvenusz]);
47 rvenussnorm = norm([rvenussx rvenussy rvenussz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rvenusx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rvenusy), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rvenusz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
```

```

62 title('Venus Position')
63 legend('r_{venus,x}', 'r_{venus,y}', 'r_{venus,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rvenusnorm), 'r', 'LineWidth', 2)

```

**Listing 7:** Script to obtain the interpolating polynomial of the position of Mars as seen from the Earth

```

1  clc
2  close all
3  clear all
4
5  hours = 3600;
6  t0 = 0;
7  tf = 31*24*hours;
8  tspan = (t0:1*hours:(tf-1));
9
10 load('marsmatrix.mat')
11 rmars = horizonsresultsMARS(52:795, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rmars(:,1), 20);
15 rmarsx = 0;
16
17 for j = 1:length(polx)
18     rmarsx = t.^(j-1)*polx(length(polx)+1-j)+rmarsx;
19 end
20
21 errorx = rmars(:,1)-rmarsx;
22
23 poly = polyfit(tspan', rmars(:,2), 20);
24 rmarsy = 0;
25
26 for l = 1:length(poly)
27     rmarsy = t.^(l-1)*poly(length(poly)+1-l)+rmarsy;
28 end
29
30 errory = rmars(:,2)-rmarsy;
31
32 polz = polyfit(tspan', rmars(:,3), 20);
33 rmarsz = 0;
34
35 for p = 1:length(polz)
36     rmarsz = t.^(p-1)*polz(length(polz)+1-p)+rmarsz;
37 end
38
39 errorz = rmars(:,3)-rmarsz;
40
41 syms x y z
42 rmarssx = rmarsx-x;
43 rmarssy = rmarsy-y;
44 rmarssz = rmarsz-z;
45
46 rmarsnorm = norm([rmarsx rmarsy rmarsz]);
47 rmarssnorm = norm([rmarssx rmarssy rmarssz]);
48

```

```

49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rmarsx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rmarsy), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rmarsz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Mars Position')
63 legend('r_{mars,x}', 'r_{mars,y}', 'r_{mars,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rmarsnorm), 'r', 'LineWidth', 2)

```

**Listing 8:** Script to obtain the interpolating polynomial of the position of Jupiter as seen from the Earth

```

1 clc
2 close all
3 clear all
4
5 hours = 3600;
6 t0 = 0;
7 tf = 31*24*hours;
8 tspan = (t0:1*hours:(tf-1));
9
10 load('jupitermatrix.mat')
11 rjupiter = horizonsresultsJUPITER(51:794, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rjupiter(:,1), 20);
15 rjupiterx = 0;
16
17 for j = 1:length(polx)
18     rjupiterx = t.^(j-1)*polx(length(polx)+1-j)+rjupiterx;
19 end
20
21 errorx = rjupiter(:,1)-rjupiterx;
22
23 poly = polyfit(tspan', rjupiter(:,2), 20);
24 rjupitery = 0;
25
26 for l = 1:length(poly)
27     rjupitery = t.^(l-1)*poly(length(poly)+1-l)+rjupitery;
28 end
29
30 errory = rjupiter(:,2)-rjupitery;
31
32 polz = polyfit(tspan', rjupiter(:,3), 20);
33 rjupiterz = 0;
34
35 for p = 1:length(polz)

```

```

36     rjupiterz = t.^(p-1)*polz(length(polz)+1-p)+rjupiterz;
37 end
38
39 errorz = rjupiter(:,3)-rjupiterz;
40
41 syms x y z
42 rjupitersx = rjupiterx-x;
43 rjupitersy = rjupitery-y;
44 rjupitersz = rjupiterz-z;
45
46 rjupiternorm = norm([rjupiterx rjupitery rjupiterz]);
47 rjupitersnorm = norm([rjupitersx rjupitersy rjupitersz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rjupiterx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rjupitery), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rjupiterz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Jupiter Position')
63 legend('r_{jupiter,x}', 'r_{jupiter,y}', 'r_{jupiter,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rjupiternorm), 'r', 'LineWidth', 2)

```

**Listing 9:** Script to obtain the interpolating polynomial of the position of Saturn as seen from the Earth

```

1  clc
2  close all
3  clear all
4
5  hours = 3600;
6  t0 = 0;
7  tf = 31*24*hours;
8  tspan = (t0:1*hours:(tf-1));
9
10 load('saturnmatrix.mat')
11 rsaturn = horizonsresultsSATURN(51:794, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rsaturn(:,1), 20);
15 rsaturnx = 0;
16
17 for j = 1:length(polx)
18     rsaturnx = t.^(j-1)*polx(length(polx)+1-j)+rsaturnx;
19 end
20
21 errorx = rsaturn(:,1)-rsaturnx;
22

```

```

23 poly = polyfit(tspan', rsaturn(:,2), 20);
24 rsaturny = 0;
25
26 for l = 1:length(poly)
27     rsaturny = t.^(l-1)*poly(length(poly)+1-l)+rsaturny;
28 end
29
30 errory = rsaturn(:,2)-rsaturny;
31
32 polz = polyfit(tspan', rsaturn(:,3), 20);
33 rsaturnz = 0;
34
35 for p = 1:length(polz)
36     rsaturnz = t.^(p-1)*polz(length(polz)+1-p)+rsaturnz;
37 end
38
39 errorz = rsaturn(:,3)-rsaturnz;
40
41 syms x y z
42 rsaturnsx = rsaturnx-x;
43 rsaturnsy = rsaturny-y;
44 rsaturnsz = rsaturnz-z;
45
46 rsaturnnorm = norm([rsaturnx rsaturny rsaturnz]);
47 rsaturnsnorm = norm([rsaturnsx rsaturnsy rsaturnsz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rsaturnx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rsaturny), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rsaturnz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Saturn Position')
63 legend('r_{saturn,x}', 'r_{saturn,y}', 'r_{saturn,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rsaturnnorm), 'r', 'LineWidth', 2)

```

**Listing 10:** Script to obtain the interpolating polynomial of the position of Uranus as seen from the Earth

```

1 clc
2 close all
3 clear all
4
5 hours = 3600;
6 t0 = 0;
7 tf = 31*24*hours;
8 tspan = (t0:1*hours:(tf-1));
9

```

```

10 load('uranusmatrix.mat')
11 ruranus = horizonsresultsURANUS(51:794, 3:5);
12
13 syms t
14 polx = polyfit(tspan', ruranus(:,1), 20);
15 ruranusx = 0;
16
17 for j = 1:length(polx)
18     ruranusx = t.^(j-1)*polx(length(polx)+1-j)+ruranusx;
19 end
20
21 errorx = ruranus(:,1)-ruranusx;
22
23 poly = polyfit(tspan', ruranus(:,2), 20);
24 ruranusy = 0;
25
26 for l = 1:length(poly)
27     ruranusy = t.^(l-1)*poly(length(poly)+1-l)+ruranusy;
28 end
29
30 errory = ruranus(:,2)-ruranusy;
31
32 polz = polyfit(tspan', ruranus(:,3), 20);
33 ruranusz = 0;
34
35 for p = 1:length(polz)
36     ruranusz = t.^(p-1)*polz(length(polz)+1-p)+ruranusz;
37 end
38
39 errorz = ruranus(:,3)-ruranusz;
40
41 syms x y z
42 ruranussx = ruranusx-x;
43 ruranussy = ruranusy-y;
44 ruranussz = ruranusz-z;
45
46 ruranusnorm = norm([ruranusx ruranusy ruranusz]);
47 ruranussnorm = norm([ruranussx ruranussy ruranussz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(ruranusx), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(ruranusy), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(ruranusz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Uranus Position')
63 legend('r_{uranus,x}', 'r_{uranus,y}', 'r_{uranus,z}')
64

```

```
65 figure(2)
66 plot(tspan'/(24*hours), subs(ruranusnorm), 'r', 'LineWidth', 2)
```

**Listing 11:** Script to obtain the interpolating polynomial of the position of Neptune as seen from the Earth

```
1  clc
2  close all
3  clear all
4
5  hours = 3600;
6  t0 = 0;
7  tf = 31*24*hours;
8  tspan = (t0:1*hours:(tf-1));
9
10 load('neptunematrix.mat')
11 rneptune = horizonsresultsNEPTUNE(51:794, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rneptune(:,1), 20);
15 rneptunex = 0;
16
17 for j = 1:length(polx)
18     rneptunex = t.^(j-1)*polx(length(polx)+1-j)+rneptunex;
19 end
20
21 errorx = rneptune(:,1)-rneptunex;
22
23 poly = polyfit(tspan', rneptune(:,2), 20);
24 rneptuney = 0;
25
26 for l = 1:length(poly)
27     rneptuney = t.^(l-1)*poly(length(poly)+1-l)+rneptuney;
28 end
29
30 errory = rneptune(:,2)-rneptuney;
31
32 polz = polyfit(tspan', rneptune(:,3), 20);
33 rneptunez = 0;
34
35 for p = 1:length(polz)
36     rneptunez = t.^(p-1)*polz(length(polz)+1-p)+rneptunez;
37 end
38
39 errorz = rneptune(:,3)-rneptunez;
40
41 syms x y z
42 rneptunesx = rneptunex-x;
43 rneptunesy = rneptuney-y;
44 rneptunesz = rneptunez-z;
45
46 rneptunenorm = norm([rneptunesx rneptunesy rneptunesz]);
47 rneptunesnorm = norm([rneptunesx rneptunesy rneptunesz]);
48
49 t = tspan';
50
51 % Plot
```



```

52 figure(1)
53 plot(tspan'/(24*hours), subs(rneptunex), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rneptuney), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rneptunez), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Neptune Position')
63 legend('r_{neptune,x}', 'r_{neptune,y}', 'r_{neptune,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rneptunenorm), 'r', 'LineWidth', 2)

```

**Listing 12:** Script to obtain the interpolating polynomial of the position of Pluto as seen from the Earth

```

1  clc
2  close all
3  clear all
4
5  hours = 3600;
6  t0 = 0;
7  tf = 31*24*hours;
8  tspan = (t0:1*hours:(tf-1));
9
10 load('plutomatrix.mat')
11 rpluto = horizonsresultsPLUTO(47:790, 3:5);
12
13 syms t
14 polx = polyfit(tspan', rpluto(:,1), 20);
15 rplutox = 0;
16
17 for j = 1:length(polx)
18     rplutox = t.^(j-1)*polx(length(polx)+1-j)+rplutox;
19 end
20
21 errorx = rpluto(:,1)-rplutox;
22
23 poly = polyfit(tspan', rpluto(:,2), 20);
24 rplutoy = 0;
25
26 for l = 1:length(poly)
27     rplutoy = t.^(l-1)*poly(length(poly)+1-l)+rplutoy;
28 end
29
30 errory = rpluto(:,2)-rplutoy;
31
32 polz = polyfit(tspan', rpluto(:,3), 20);
33 rplutoz = 0;
34
35 for p = 1:length(polz)
36     rplutoz = t.^(p-1)*polz(length(polz)+1-p)+rplutoz;
37 end
38

```

```

39 errorz = rpluto(:,3)-rplutoz;
40
41 syms x y z
42 rplutosx = rplutox-x;
43 rplutosy = rplutoy-y;
44 rplutosz = rplutoz-z;
45
46 rplutonorm = norm([rplutox rplutoy rplutoz]);
47 rplutosnorm = norm([rplutosx rplutosy rplutosz]);
48
49 t = tspan';
50
51 % Plot
52 figure(1)
53 plot(tspan'/(24*hours), subs(rplutox), 'r', 'LineWidth', 2)
54 hold on
55 plot(tspan'/(24*hours), subs(rplutoy), 'b', 'LineWidth', 2)
56 hold on
57 plot(tspan'/(24*hours), subs(rplutoz), 'g', 'LineWidth', 2)
58 hold off
59 grid on
60 xlabel('t [days]')
61 ylabel('distance [km]')
62 title('Pluto Position')
63 legend('r_{pluto,x}', 'r_{pluto,y}', 'r_{pluto,z}')
64
65 figure(2)
66 plot(tspan'/(24*hours), subs(rplutonorm), 'r', 'LineWidth', 2)

```

### A.3 Code for LAGEOS-1

Listing 13: Main function for LAGEOS-1 [5]

```

1 function orbitLAGEOS
2 clc
3
4 global y02 hours mu statematrix r0 v0 R t0 tend m2 deltavx deltavy
   deltavz timpulse
5
6 hours = 3600;
7 G = 6.6742e-20;
8
9 % Input data:
10 % Earth:
11 m1 = 5.972e24;
12 R = 6378;
13 m2 = 406.965;
14 load('lageosmatrix.mat')
15 statematrix = horizonsLAGEOS127completo(35:778, 3:8);
16 r0 = [statematrix(1,1), statematrix(1,2), statematrix(1,3)];
17 v0 = [statematrix(1,4), statematrix(1,5), statematrix(1,6)];
18 t0 = 0;
19 tend = 31*24*hours;
20
21 % Numerical integration:
22 mu = G*(m1 + m2);

```

```

23 options = odeset('RelTol', 1e-13, 'AbsTol', 1e-13);
24
25 if deltavx == 0 && deltavy == 0 && deltavz == 0
26     tspan = (t0:1*hours:(tend-1));
27     y0 = [r0 v0]';
28     [t,y] = ode45(@ratesLAGEOS, tspan, y0, options);
29 else
30     tspan1 = (t0:1*hours:(timpulse-1));
31     y01 = [r0 v0]';
32     [t1,y1] = ode45(@ratesLAGEOS, tspan1, y01, options);
33
34     tspan2 = (timpulse:1*hours:(tend-1));
35     y02 = [y1(end,1) y1(end,2) y1(end,3) y1(end,4)+deltavx y1(end,5)+
36           deltavy y1(end,6)+deltavz]';
37     [t2,y2] = ode45(@ratesLAGEOS, tspan2, y02, options);
38
39     t = [t1; t2];
40     y = [y1; y2];
41
42 % Output the results:
43 outputLAGEOS(t,y)
44
45 return
46 end

```

**Listing 14:** Rates function containing the differential equation for LAGEOS-1 [5]

```

1 function dydt = ratesLAGEOS(t,f)
2
3 global mu R m2 pJ2 pJ3 pJ4 pJ5 pJ6 pDrag pRel pMoon pSun pMercury pVenus
4     pMars pJupiter pSaturn pUranus pNeptune pPluto pSRP
5
6 J2 = 1082.63e-6;
7 J3 = -2.52e-6;
8 J4 = -1.61e-6;
9 J5 = -0.15e-6;
10 J6 = 0.57e-6;
11 wE = [0 0 7.2921159e-5]';
12 CD = 20; % https://www.researchgate.net/publication/328586902
13     _On_Testing_Frame -
14     Dragging_with_LAGEOS_and_a_Recently_Announced_Geodetic_Satellite
15 A = pi*(0.6/2)^2; % Diameter: 60 cm
16 c = 299792458/1000; % Speed of light in km/s
17 mumoon = 4902.800066; % km^3/s^2
18 musun = 132712440041.93938; % km^3/s^2
19 mumercury = 22031.86855; % km^3/s^2
20 muvenus = 324858.592; % km^3/s^2
21 mumars = 42828.375214; % km^3/s^2
22 mujupiter = 126686531.900; % km^3/s^2
23 musaturn = 37931206.159; % km^3/s^2
24 muuranus = 5793951.322; % km^3/s^2
25 mun Neptune = 6835099.97; % km^3/s^2
26 mupluto = 869.96; % km^3/s^2
27
28 x = f(1);

```

```
26 y = f(2);
27 z = f(3);
28 vx = f(4);
29 vy = f(5);
30 vz = f(6);
31
32 r = norm([x y z]);
33
34 % Atmospheric drag:
35 vrel = [vx vy vz]-cross(wE,[x y z]);
36 vrelnorm = norm(vrel);
37
38 % Lunar gravity:
39 rmx = ...
40 rmy = ...
41 rmz = ...
42 rmnorm = norm([rmx rmy rmz]);
43
44 rmsx = rmx-x;
45 rmsy = rmy-y;
46 rmsz = rmz-z;
47 rmsnorm = norm([rmsx rmsy rmsz]);
48
49 % Solar gravity:
50 rsx = ...
51 rsy = ...
52 rsz = ...
53 rsnorm = norm([rsx rsy rsz]);
54
55 rssx = rsx-x;
56 rssid = rsy-y;
57 rssid = rsz-z;
58 rssidnorm = norm([rssx rssid rssid]);
59
60 % Mercury gravity:
61 rmercuryx = ...
62 rmercuryy = ...
63 rmercuryz = ...
64 rmercurynorm = norm([rmercuryx rmercuryy rmercuryz]);
65
66 rmercurysx = rmercuryx-x;
67 rmercurysy = rmercuryy-y;
68 rmercurysz = rmercuryz-z;
69 rmercurysnorm = norm([rmercurysx rmercurysy rmercurysz]);
70
71 % Venus gravity:
72 rvenusx = ...
73 rvenusy = ...
74 rvenusz = ...
75 rvenusnorm = norm([rvenusx rvenusy rvenusz]);
76
77 rvenussx = rvenusx-x;
78 rvenussy = rvenusy-y;
79 rvenussz = rvenusz-z;
80 rvenussnorm = norm([rvenussx rvenussy rvenussz]);
```

```
81
82 % Mars gravity:
83 rmarsx = ...
84 rmarsy = ...
85 rmarsz = ...
86 rmarsnorm = norm([rmarsx rmarsy rmarsz]);
87
88 rmarssx = rmarsx-x;
89 rmarssy = rmarsy-y;
90 rmarssz = rmarsz-z;
91 rmarssnorm = norm([rmarssx rmarssy rmarssz]);
92
93 % Jupiter gravity:
94 rjupiterx = ...
95 rjupitery = ...
96 rjupiterz = ...
97 rjupiternorm = norm([rjupiterx rjupitery rjupiterz]);
98
99 rjupitersx = rjupiterx-x;
100 rjupitersy = rjupitery-y;
101 rjupitersz = rjupiterz-z;
102 rjupitersnorm = norm([rjupitersx rjupitersy rjupitersz]);
103
104 % Saturn gravity:
105 rsaturnx = ...
106 rsaturny = ...
107 rsaturnz = ...
108 rsaturnnorm = norm([rsaturnx rsaturny rsaturnz]);
109
110 rsaturnsx = rsaturnx-x;
111 rsaturnsy = rsaturny-y;
112 rsaturnsz = rsaturnz-z;
113 rsaturnsnorm = norm([rsaturnsx rsaturnsy rsaturnsz]);
114
115 % Uranus gravity:
116 ruranusx = ...
117 ruranusy = ...
118 ruranusz = ...
119 ruranusnorm = norm([ruranusx ruranusy ruranusz]);
120
121 ruranussx = ruranusx-x;
122 ruranussy = ruranusy-y;
123 ruranussz = ruranusz-z;
124 ruranussnorm = norm([ruranussx ruranussy ruranussz]);
125
126 % Neptune gravity:
127 rneptunex = ...
128 rneptuney = ...
129 rneptunez = ...
130 rneptunenorm = norm([rneptunex rneptuney rneptunez]);
131
132 rneptunesx = rneptunex-x;
133 rneptunesy = rneptuney-y;
134 rneptunesz = rneptunez-z;
135 rneptunesnorm = norm([rneptunesx rneptunesy rneptunesz]);
```

```

136
137 % Pluto gravity:
138 rplutox = ...
139 rplutoy = ...
140 rplutoz = ...
141 rplutonorm = norm([rplutox rplutoy rplutoz]);
142
143 rplutosx = rplutox-x;
144 rplutosy = rplutoy-y;
145 rplutosz = rplutoz-z;
146 rplutosnorm = norm([rplutosx rplutosy rplutosz]);
147
148 % Solar Radiation Pressure:
149 th = acosd(dot([rsx rsy rsz],[x y z])/(rsnorm*r));
150 th1 = acosd(R/r);
151 th2 = acosd(R/rsnorm);
152
153 if th1+th2 <= th
154     nu = 0; % Shadow function: 0 if in shadow, 1 if not
155 else
156     nu = 1;
157 end
158
159 c2 = 299792458; % Speed of light in m/s
160 CR = 1.142; % Radiation pressure coefficient: 1 if black body, 2 if
    every radiation is reflected --> https://cdis.nasa.gov/lw21/docs/2018/presentations/Session2\_Hattori\_presentation.pdf
161 AS = pi*(0.6/2)^2; % Diameter: 60 cm
162 S = 1367; % Solar constant (nominal 1367 W/m^2)
163
164 % Total acceleration (including perturbations):
165 ax = -mu*x/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(x*((5*z^2/(r^2))-1))) +
    pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(5*x/r)*(7*(z/r)^3-3*(z/r)) + pJ4
    *(5/8)*J4*(mu/r^2)*(R/r)^4*(x/r)*(3-42*(z/r)^2+63*(z/r)^4) + pJ5*(J5
    /8)*(mu/r^2)*(R/r)^5*(3*x/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - pJ6*
    J6/16*(mu/r^2)*(R/r)^6*(x/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)
    ^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(1) -
    pRel*3*mu^2/c^2*x/r^3 + pMoon*mumoon*(rmsx./(rmsnorm.^3)-rmx./(rnorm
    .^3)) + pSun*musun*(rssx./(rssnorm.^3)-rsx./(rsnorm.^3)) + pMercury*
    mumercury*(rmercurysx./(rmercurysnorm.^3)-rmercuryx./(rmercurynorm
    .^3)) + pVenus*muvenus*(rvenussx./(rvenussnorm.^3)-rvenusx./(
    rvenusnorm.^3)) + pMars*mumars*(rmarssx./(rmarssnorm.^3)-rmarsx./(
    rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersx./(rjupitersnorm.^3)-
    rjupiterx./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsx./(
    rsaturnsnorm.^3)-rsaturnx./(rsaturnnorm.^3)) + pUranus*muuranus*(
    ruranussx./(ruranussnorm.^3)-ruranusx./(ruranusnorm.^3)) + pNeptune*
    muneptune*(rneptunesx./(rneptunesnorm.^3)-rneptunex./(rneptunenorm
    .^3)) + pPluto*mupluto*(rplutosx./(rplutosnorm.^3)-rplutox./(
    rplutonorm.^3)) - pSRP*nu*S/c2*CR*AS/m2/1000*(rssx/rssnorm);
166 ay = -mu*y/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(y*((5*z^2/(r^2))-1))) +
    pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(5*y/r)*(7*(z/r)^3-3*(z/r)) + pJ4
    *(5/8)*J4*(mu/r^2)*(R/r)^4*(y/r)*(3-42*(z/r)^2+63*(z/r)^4) + pJ5*(J5
    /8)*(mu/r^2)*(R/r)^5*(3*y/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - pJ6*
    J6/16*(mu/r^2)*(R/r)^6*(y/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)
    ^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(2) -

```

```

pRel*3*mu^2/c^2*y/r^3 + pMoon*mumoon*(rmsy./(rmsnorm.^3)-rmy./(rmnorm
.^3)) + pSun*musun*(rssy./(rssnorm.^3)-rsy./(rsnorm.^3)) + pMercury*
mumercury*(rmercurysy./(rmercurysnorm.^3)-rmercuryy./(rmercurynorm
.^3)) + pVenus*muvenus*(rvenussy./(rvenussnorm.^3)-rvenusy./(
rvenusnorm.^3)) + pMars*mumars*(rmarssy./(rmarssnorm.^3)-rmarsy./(
rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersy./(rjupitersnorm.^3)-
rjupitery./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsy./(
rsaturnsnorm.^3)-rsaturny./(rsaturnnorm.^3)) + pUranus*muuranus*(
ruranussy./(ruranussnorm.^3)-ruranusy./(ruranusnorm.^3)) + pNeptune*
muneptune*(rneptunesy./(rneptunesnorm.^3)-rneptuney./(rneptunenorm
.^3)) + pPluto*mupluto*(rplutosy./(rplutosnorm.^3)-rplutoy./(
rplutonorm.^3)) - pSRP*nu*S/c^2*CR*AS/m2/1000*(rssy/rssnorm);
167 az = -mu*z/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(z*((5*z^2/(r^2))-3))) +
pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(1-10*(z/r)^2+35/3*(z/r)^4) + pJ4
*(5/8)*J4*(mu/r^2)*(R/r)^4*(z/r)*(15-70*(z/r)^2+63*(z/r)^4) + pJ5*(J5
/8)*(mu/r^2)*(R/r)^5*(693*(z/r)^6-945*(z/r)^4+315*(z/r)^2-15) - pJ6*
J6/16*(mu/r^2)*(R/r)^6*(z/r)*(245-2205*(z/r)^2+4851*(z/r)^4-3003*(z/r
)^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(3) -
pRel*3*mu^2/c^2*z/r^3 + pMoon*mumoon*(rmsz./(rmsnorm.^3)-rmz./(rmnorm
.^3)) + pSun*musun*(rssz./(rssnorm.^3)-rsz./(rsnorm.^3)) + pMercury*
mumercury*(rmercurysz./(rmercurysnorm.^3)-rmercuryz./(rmercurynorm
.^3)) + pVenus*muvenus*(rvenussz./(rvenussnorm.^3)-rvenusz./(
rvenusnorm.^3)) + pMars*mumars*(rmarssz./(rmarssnorm.^3)-rmarsz./(
rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersz./(rjupitersnorm.^3)-
rjupiterz./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsz./(
rsaturnsnorm.^3)-rsaturnz./(rsaturnnorm.^3)) + pUranus*muuranus*(
ruranussz./(ruranussnorm.^3)-ruranusz./(ruranusnorm.^3)) + pNeptune*
muneptune*(rneptunesz./(rneptunesnorm.^3)-rneptunez./(rneptunenorm
.^3)) + pPluto*mupluto*(rplutosz./(rplutosnorm.^3)-rplutoz./(
rplutonorm.^3)) - pSRP*nu*S/c^2*CR*AS/m2/1000*(rssz/rssnorm);
168
169 dydt = [vx vy vz ax ay az]';
170
171 end

```

Listing 15: Output function for LAGEOS-1 [5]

```

1 function outputLAGEOS(t,y)
2
3 global y02 hours statematrix r0 R rmax rmin imax imin v_at_rmax
v_at_rmin m2 mu chi a h e rp ra p vp vr gama inc Omega omega theta E
M averagen T tminustp deltavx deltavy deltavz Isp deltam
4
5 for i = 1:length(t)
6 vecr(i,:) = [y(i,1) y(i,2) y(i,3)];
7 r(i) = norm(vecr(i,:));
8 vecv(i,:) = [y(i,4) y(i,5) y(i,6)];
9 v(i) = norm(vecv(i,:));
10 end
11
12 [rmax, imax] = max(r);
13 [rmin, imin] = min(r);
14
15 v_at_rmax = norm([y(imax,4) y(imax,5) y(imax,6)]);
16 v_at_rmin = norm([y(imin,4) y(imin,5) y(imin,6)]);
17

```

```

18 finalpos = [y(length(t),1), y(length(t),2), y(length(t),3)];
19 finalvel = [y(length(t),4), y(length(t),5), y(length(t),6)];
20
21 %% Obtain the orbital parameters from the state vector:
22 for i = 1:length(t)
23     chi(i) = v(i)^2/2-mu/r(i);
24     a(i) = -mu/(2*chi(i));
25     vech(i,:) = cross(vecr(i,:),vecv(i,:));
26     h(i) = norm(vech(i,:));
27     vecC(i,:) = cross(vecv(i,:),vech(i,:))-mu*vecr(i,)/r(i);
28     vece(i,:) = vecC(i,)/mu;
29     e(i) = norm(vece(i,:));
30     rp(i) = a(i)*(1-e(i));
31     ra(i) = a(i)*(1+e(i));
32     p(i) = h(i)^2/mu;
33     vp(i) = h(i)/r(i);
34     if dot(vecr(i,:),vecv(i,))>=0
35         gama(i) = acos(h(i)/(r(i)*v(i)))*180/pi;
36     else
37         gama(i) = -acos(h(i)/(r(i)*v(i)))*180/pi;
38     end
39     vr(i) = vp(i)*tand(gama(i));
40     inc(i) = acosd(vech(i,3)/h(i));
41     vecn(i,:) = cross([0 0 1],vech(i,:));
42     n(i) = norm(vecn(i,:));
43     if vecn(i,2)>=0
44         Omega(i) = acosd(vecn(i,1)/n(i));
45     else
46         Omega(i) = 360-acosd(vecn(i,1)/n(i));
47     end
48     if vece(i,3)>=0
49         omega(i) = acosd(dot(vecn(i,:),vece(i,))/(n(i)*e(i)));
50     else
51         omega(i) = 360-acosd(dot(vecn(i,:),vece(i,))/(n(i)*e(i)));
52     end
53     if dot(vecr(i,:),vecv(i,))>=0
54         theta(i) = acosd(dot(vece(i,:),vecr(i,))/(e(i)*r(i)));
55     else
56         theta(i) = 360-acosd(dot(vece(i,:),vecr(i,))/(e(i)*r(i)));
57     end
58     E(i) = 2*atand(sqrt((1-e(i))/(1+e(i)))*tand(theta(i)/2));
59     M(i) = E(i)-e(i)*sind(E(i)); % Kepler's equation
60     averagen(i) = sqrt(mu/(a(i)^3));
61     T(i) = 2*pi/averagen(i);
62     tminustp(i) = M(i)/averagen(i);
63     while tminustp(i) < 0
64         tminustp(i) = tminustp(i)+T(i);
65     end
66     while tminustp(i) > T(i)
67         tminustp(i) = tminustp(i)-T(i);
68     end
69 end
70
71 %% Find the propellant mass needed with Tsiolkovski formula:
72 deltav = [deltavx deltavy deltavz];

```



```

73 deltam = m2*(1-exp(-norm(deltav)/(Isp*0.00981)));
74
75 %% Plot the results:
76 % Draw the planet
77 h1 = figure(1);
78 h1.Visible = 'off';
79 earth_sphere('km')
80
81 % Draw and label the X, Y and Z axes
82 line([0 2*R], [0 0], [0 0], 'LineStyle', '-. '); text(2*R, 0, 0, 'X')
83 line([0 0], [0 2*R], [0 0], 'LineStyle', '-. '); text(0, 2*R, 0, 'Y')
84 line([0 0], [0 0], [0 2*R], 'LineStyle', '-. '); text(0, 0, 2*R, 'Z')
85
86 % Plot the orbit, draw a radial to the starting point and label the
      starting point (o) and the final point (f)
87 hold on
88 plot3(y(:,1), y(:,2), y(:,3), 'k. ')
89 line([0 r0(1)], [0 r0(2)], [0 r0(3)], 'Color', 'r')
90 line([0 y(end,1)], [0 y(end,2)], [0 y(end,3)], 'Color', 'r')
91 text(y(1,1)-1000, y(1,2), y(1,3), 'o', 'FontSize', 16, 'Color', 'r')
92 text(y(end,1)+2000, y(end,2)+500, y(end,3)+1000, 'f', 'FontSize', 16, '
      Color', 'r')
93
94 if deltax ~ = 0 || deltay ~ = 0 || deltaz ~ = 0
95     line([0 y02(1)], [0 y02(2)], [0 y02(3)], 'Color', 'g')
96     text(y02(1), y02(2), y02(3), ' v ', 'FontSize', 16, 'Color', 'g')
97 end
98
99 % Select a view direction (a vector directed outward from the origin)
100 view([1, 1, .4])
101
102 % Specify some properties of the graph
103 grid on
104 axis equal
105 xlabel('km')
106 ylabel('km')
107 zlabel('km')
108 hold off
109 saveas(h1, 'orbit', 'jpg')
110
111 % Plot the error with respect to HORIZONS results:
112 abserr = y-statematrix;
113 h2 = figure(2);
114 h2.Visible = 'off';
115 plot(t/(24*hours), abserr(:,1:3))
116 xlabel('t [days]')
117 ylabel('error [km]')
118 title('Position error')
119 saveas(h2, 'positionerror', 'jpg')
120
121 h3 = figure(3);
122 h3.Visible = 'off';
123 plot(t/(24*hours), abserr(:,4:6))
124 xlabel('t [days]')
125 ylabel('error [km/s]')

```

```

126 title('Velocity error')
127 saveas(h3,'velocityerror','jpg')
128
129 end

```

#### A.4 Code for ISS

Listing 16: Main function for ISS [5]

```

1 function orbitISS
2 clc
3
4 global y02 hours mu statematrix r0 v0 R t0 tend m2 deltavx deltavy
   deltavz timpulse
5
6 hours = 3600;
7 G = 6.6742e-20;
8
9 % Input data:
10 % Earth:
11 m1 = 5.972e24;
12 R = 6378;
13 m2 = 419725;
14 load('ISSmatrix.mat')
15 statematrix = horizonsresultsISS(40:783,3:8);
16 r0 = [statematrix(1,1), statematrix(1,2), statematrix(1,3)];
17 v0 = [statematrix(1,4), statematrix(1,5), statematrix(1,6)];
18 t0 = 0;
19 tend = 31*24*hours;
20
21 % Numerical integration:
22 mu = G*(m1 + m2);
23 options = odeset('RelTol', 1e-13, 'AbsTol', 1e-13);
24
25 if deltavx == 0 && deltavy == 0 && deltavz == 0
26     tspan = (t0:1*hours:(tend-1));
27     y0 = [r0 v0]';
28     [t,y] = ode45(@ratesISS, tspan, y0, options);
29 else
30     tspan1 = (t0:1*hours:(timpulse-1));
31     y01 = [r0 v0]';
32     [t1,y1] = ode45(@ratesISS, tspan1, y01, options);
33
34     tspan2 = (timpulse:1*hours:(tend-1));
35     y02 = [y1(end,1) y1(end,2) y1(end,3) y1(end,4)+deltavx y1(end,5)+
   deltavy y1(end,6)+deltavz]';
36     [t2,y2] = ode45(@ratesISS, tspan2, y02, options);
37
38     t = [t1; t2];
39     y = [y1; y2];
40 end
41
42 % Output the results:
43 outputISS(t,y)
44
45 return

```

46 `end`

**Listing 17:** Rates function containing the differential equation for ISS [5]

```

1 function dydt = ratesISS(t,f)
2
3 global mu R m2 pJ2 pJ3 pJ4 pJ5 pJ6 pDrag pRel pMoon pSun pMercury pVenus
   pMars pJupiter pSaturn pUranus pNeptune pPluto pSRP
4
5 J2 = 1082.63e-6;
6 J3 = -2.52e-6;
7 J4 = -1.61e-6;
8 J5 = -0.15e-6;
9 J6 = 0.57e-6;
10 wE = [0 0 7.2921159e-5]';
11 CD = 2.07; % https://www.esa.int/Education/Space\_In\_Bytes/
   ATV\_a\_very\_special\_delivery\_-\_Lesson\_notes
12 A = 1000; % m2
13 c = 299792458/1000; % Speed of light in km/s
14 mumoon = 4902.800066; % km3/s2
15 musun = 132712440041.93938; % km3/s2
16 mumercury = 22031.86855; % km3/s2
17 muvenus = 324858.592; % km3/s2
18 mumars = 42828.375214; % km3/s2
19 mujupiter = 126686531.900; % km3/s2
20 musaturn = 37931206.159; % km3/s2
21 muuranus = 5793951.322; % km3/s2
22 mun Neptune = 6835099.97; % km3/s2
23 mupluto = 869.96; % km3/s2
24
25 x = f(1);
26 y = f(2);
27 z = f(3);
28 vx = f(4);
29 vy = f(5);
30 vz = f(6);
31
32 r = norm([x y z]);
33
34 % Atmospheric drag:
35 vrel = [vx vy vz]-cross(wE,[x y z]);
36 vrelnorm = norm(vrel);
37
38 % Lunar gravity:
39 rmx = ...
40 rmy = ...
41 rmz = ...
42 rmnorm = norm([rmx rmy rmz]);
43
44 rmsx = rmx-x;
45 rmsy = rmy-y;
46 rmsz = rmz-z;
47 rmsnorm = norm([rmsx rmsy rmsz]);
48
49 % Solar gravity:
50 rsx = ...

```

```
51 rsy = ...
52 rsz = ...
53 rsnorm = norm([rsx rsy rsz]);
54
55 rssx = rsx-x;
56 rssid = rsy-y;
57 rssid = rsz-z;
58 rssidnorm = norm([rssx rssid rssid]);
59
60 % Mercury gravity:
61 rmercuryx = ...
62 rmercuryy = ...
63 rmercuryz = ...
64 rmercurynorm = norm([rmercuryx rmercuryy rmercuryz]);
65
66 rmercurysx = rmercuryx-x;
67 rmercurysy = rmercuryy-y;
68 rmercurysz = rmercuryz-z;
69 rmercurysnorm = norm([rmercurysx rmercurysy rmercurysz]);
70
71 % Venus gravity:
72 rvenusx = ...
73 rvenusy = ...
74 rvenusz = ...
75 rvenusnorm = norm([rvenusx rvenusy rvenusz]);
76
77 rvenussx = rvenusx-x;
78 rvenussy = rvenusy-y;
79 rvenussz = rvenusz-z;
80 rvenussnorm = norm([rvenussx rvenussy rvenussz]);
81
82 % Mars gravity:
83 rmarsx = ...
84 rmarsy = ...
85 rmarsz = ...
86 rmarsnorm = norm([rmarsx rmarsy rmarsz]);
87
88 rmarssx = rmarsx-x;
89 rmarssy = rmarsy-y;
90 rmarssz = rmarsz-z;
91 rmarssnorm = norm([rmarssx rmarssy rmarssz]);
92
93 % Jupiter gravity:
94 rjupiterx = ...
95 rjupitery = ...
96 rjupiterz = ...
97 rjupiternorm = norm([rjupiterx rjupitery rjupiterz]);
98
99 rjupitersx = rjupiterx-x;
100 rjupitersy = rjupitery-y;
101 rjupitersz = rjupiterz-z;
102 rjupitersnorm = norm([rjupitersx rjupitersy rjupitersz]);
103
104 % Saturn gravity:
105 rsaturnx = ...
```

```
106 rsaturny = ...
107 rsaturnz = ...
108 rsaturnnorm = norm([rsaturnx rsaturny rsaturnz]);
109
110 rsaturnsx = rsaturnx-x;
111 rsaturnsy = rsaturny-y;
112 rsaturnsz = rsaturnz-z;
113 rsaturnsnorm = norm([rsaturnsx rsaturnsy rsaturnsz]);
114
115 % Uranus gravity:
116 ruranusx = ...
117 ruranusy = ...
118 ruranusz = ...
119 ruranusnorm = norm([ruranusx ruranusy ruranusz]);
120
121 ruranussx = ruranusx-x;
122 ruranussy = ruranusy-y;
123 ruranussz = ruranusz-z;
124 ruranussnorm = norm([ruranussx ruranussy ruranussz]);
125
126 % Neptune gravity:
127 rneptunex = ...
128 rneptuney = ...
129 rneptunez = ...
130 rneptunenorm = norm([rneptunex rneptuney rneptunez]);
131
132 rneptunesx = rneptunex-x;
133 rneptunesy = rneptuney-y;
134 rneptunesz = rneptunez-z;
135 rneptunesnorm = norm([rneptunesx rneptunesy rneptunesz]);
136
137 % Pluto gravity:
138 rplutox = ...
139 rplutoy = ...
140 rplutoz = ...
141 rplutonorm = norm([rplutox rplutoy rplutoz]);
142
143 rplutosx = rplutox-x;
144 rplutosy = rplutoy-y;
145 rplutosz = rplutoz-z;
146 rplutosnorm = norm([rplutosx rplutosy rplutosz]);
147
148 % Solar Radiation Pressure:
149 th = acosd(dot([rsx rsy rsz],[x y z])/(rsnorm*r));
150 th1 = acosd(R/r);
151 th2 = acosd(R/rsnorm);
152
153 if th1+th2 <= th
154     nu = 0; % Shadow function: 0 if in shadow, 1 if not
155 else
156     nu = 1;
157 end
158
159 c2 = 299792458; % Speed of light in m/s
160 CR = 1.5; % Radiation pressure coefficient: 1 if black body, 2 if every
```

```

radiation is reflected --> mean value
161 AS = 1000; % m^2
162 S = 1367; % Solar constant (nominal 1367 W/m^2)
163
164 % Total acceleration (including perturbations):
165 ax = -mu*x/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(x*((5*z^2/(r^2))-1))) +
    pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(5*x/r)*(7*(z/r)^3-3*(z/r)) + pJ4
    *(5/8)*J4*(mu/r^2)*(R/r)^4*(x/r)*(3-42*(z/r)^2+63*(z/r)^4) + pJ5*(J5
    /8)*(mu/r^2)*(R/r)^5*(3*x/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - pJ6*
    J6/16*(mu/r^2)*(R/r)^6*(x/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)
    ^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(1) -
    pRel*3*mu^2/c^2*x/r^3 + pMoon*mumoon*(rmsx./(rmsnorm.^3)-rmx./(rmnorm
    .^3)) + pSun*musun*(rssx./(rssnorm.^3)-rsx./(rsnorm.^3)) + pMercury*
    mumercury*(rmercurysx./(rmercurysnorm.^3)-rmercuryx./(rmercurynorm
    .^3)) + pVenus*muvenus*(rvenussx./(rvenussnorm.^3)-rvenusx./(
    rvenusnorm.^3)) + pMars*mumars*(rmarssx./(rmarssnorm.^3)-rmarsx./(
    rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersx./(rjupitersnorm.^3)-
    rjupiterx./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsx./(
    rsaturnsnorm.^3)-rsaturnx./(rsaturnnorm.^3)) + pUranus*muuranus*(
    ruranusx./(ruranusnorm.^3)-ruranusx./(ruranusnorm.^3)) + pNeptune*
    muneptune*(rneptunesx./(rneptunesnorm.^3)-rneptunex./(rneptunenorm
    .^3)) + pPluto*mupluto*(rplutosx./(rplutosnorm.^3)-rplutox./(
    rplutonorm.^3)) - pSRP*nu*S/c2*CR*AS/m2/1000*(rssx/rssnorm);
166 ay = -mu*y/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(y*((5*z^2/(r^2))-1))) +
    pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(5*y/r)*(7*(z/r)^3-3*(z/r)) + pJ4
    *(5/8)*J4*(mu/r^2)*(R/r)^4*(y/r)*(3-42*(z/r)^2+63*(z/r)^4) + pJ5*(J5
    /8)*(mu/r^2)*(R/r)^5*(3*y/r)*(35*z/r-210*(z/r)^3+231*(z/r)^5) - pJ6*
    J6/16*(mu/r^2)*(R/r)^6*(y/r)*(35-945*(z/r)^2+3465*(z/r)^4-3003*(z/r)
    ^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(2) -
    pRel*3*mu^2/c^2*y/r^3 + pMoon*mumoon*(rmsy./(rmsnorm.^3)-rmy./(rmnorm
    .^3)) + pSun*musun*(rssy./(rssnorm.^3)-rsy./(rsnorm.^3)) + pMercury*
    mumercury*(rmercurysy./(rmercurysnorm.^3)-rmercuryy./(rmercurynorm
    .^3)) + pVenus*muvenus*(rvenussy./(rvenussnorm.^3)-rvenusy./(
    rvenusnorm.^3)) + pMars*mumars*(rmarssy./(rmarssnorm.^3)-rmarsy./(
    rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersy./(rjupitersnorm.^3)-
    rjupitery./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsy./(
    rsaturnsnorm.^3)-rsaturny./(rsaturnnorm.^3)) + pUranus*muuranus*(
    ruranussy./(ruranusnorm.^3)-ruranusy./(ruranusnorm.^3)) + pNeptune*
    muneptune*(rneptunesy./(rneptunesnorm.^3)-rneptuney./(rneptunenorm
    .^3)) + pPluto*mupluto*(rplutosy./(rplutosnorm.^3)-rplutoy./(
    rplutonorm.^3)) - pSRP*nu*S/c2*CR*AS/m2/1000*(rssy/rssnorm);
167 az = -mu*z/r^3 + pJ2*(((3/2)*(J2*mu*R^2)/(r^5))*(z*((5*z^2/(r^2))-3))) +
    pJ3*(1/2)*J3*(mu/r^2)*(R/r)^3*(1-10*(z/r)^2+35/3*(z/r)^4) + pJ4
    *(5/8)*J4*(mu/r^2)*(R/r)^4*(z/r)*(15-70*(z/r)^2+63*(z/r)^4) + pJ5*(J5
    /8)*(mu/r^2)*(R/r)^5*(693*(z/r)^6-945*(z/r)^4+315*(z/r)^2-15) - pJ6*
    J6/16*(mu/r^2)*(R/r)^6*(z/r)*(245-2205*(z/r)^2+4851*(z/r)^4-3003*(z/r)
    ^6) - pDrag*(1/2)*atmosphere(r-R)*10^3*vrelnorm*(CD*A/m2)*vrel(3) -
    pRel*3*mu^2/c^2*z/r^3 + pMoon*mumoon*(rmsz./(rmsnorm.^3)-rmz./(rmnorm
    .^3)) + pSun*musun*(rssz./(rssnorm.^3)-rsz./(rsnorm.^3)) + pMercury*
    mumercury*(rmercurysz./(rmercurysnorm.^3)-rmercuryz./(rmercurynorm
    .^3)) + pVenus*muvenus*(rvenussz./(rvenussnorm.^3)-rvenusz./(
    rvenusnorm.^3)) + pMars*mumars*(rmarssz./(rmarssnorm.^3)-rmarisz./(
    rmarsnorm.^3)) + pJupiter*mujupiter*(rjupitersz./(rjupitersnorm.^3)-
    rjupiterz./(rjupiternorm.^3)) + pSaturn*musaturn*(rsaturnsz./(
    rsaturnsnorm.^3)-rsaturnz./(rsaturnnorm.^3)) + pUranus*muuranus*(

```

```

    ruranussz./(ruranussnorm.^3)-ruranusz./(ruranusnorm.^3) + pNeptune*
    muneptune*(rneptunesz./(rneptunesnorm.^3)-rneptunez./(rneptunenorm
    .^3)) + pPluto*mupluto*(rplutosz./(rplutosnorm.^3)-rplutoz./(
    rplutonorm.^3)) - pSRP*nu*S/c2*CR*AS/m2/1000*(rssz/rssnorm);
168
169 dydt = [vx vy vz ax ay az]';
170
171 end

```

Listing 18: Output function for ISS [5]

```

1 function outputISS(t,y)
2
3 global y02 hours statematrix r0 R rmax rmin imax imin v_at_rmax
   v_at_rmin m2 mu chi a h e rp ra p vp vr gama inc Omega omega theta E
   M averagen T tminustp deltavx deltavy deltavz Isp deltam
4
5 for i = 1:length(t)
6     vecr(i,:) = [y(i,1) y(i,2) y(i,3)];
7     r(i) = norm(vecr(i,:));
8     vecv(i,:) = [y(i,4) y(i,5) y(i,6)];
9     v(i) = norm(vecv(i,:));
10 end
11
12 [rmax, imax] = max(r);
13 [rmin, imin] = min(r);
14
15 v_at_rmax = norm([y(imax,4) y(imax,5) y(imax,6)]);
16 v_at_rmin = norm([y(imin,4) y(imin,5) y(imin,6)]);
17
18 finalpos = [y(length(t),1), y(length(t),2), y(length(t),3)];
19 finalvel = [y(length(t),4), y(length(t),5), y(length(t),6)];
20
21 % Obtain the orbital parameters from the state vector:
22 for i = 1:length(t)
23     chi(i) = v(i)^2/2-mu/r(i);
24     a(i) = -mu/(2*chi(i));
25     vech(i,:) = cross(vecr(i,:),vecv(i,:));
26     h(i) = norm(vech(i,:));
27     vecC(i,:) = cross(vecv(i,:),vech(i,:))-mu*vecr(i,:)/r(i);
28     vece(i,:) = vecC(i,+)/mu;
29     e(i) = norm(vece(i,:));
30     rp(i) = a(i)*(1-e(i));
31     ra(i) = a(i)*(1+e(i));
32     p(i) = h(i)^2/mu;
33     vp(i) = h(i)/r(i);
34     if dot(vecr(i,:),vecv(i,:))>=0
35         gama(i) = acos(h(i)/(r(i)*v(i)))*180/pi;
36     else
37         gama(i) = -acos(h(i)/(r(i)*v(i)))*180/pi;
38     end
39     vr(i) = vp(i)*tand(gama(i));
40     inc(i) = acosd(vech(i,3)/h(i));
41     vecn(i,:) = cross([0 0 1],vech(i,:));
42     n(i) = norm(vecn(i,:));
43     if vecn(i,2)>=0

```

```

44     Omega(i) = acosd(vecn(i,1)/n(i));
45 else
46     Omega(i) = 360-acosd(vecn(i,1)/n(i));
47 end
48 if vece(i,3)>=0
49     omega(i) = acosd(dot(vecn(i,:),vece(i,))/(n(i)*e(i)));
50 else
51     omega(i) = 360-acosd(dot(vecn(i,:),vece(i,))/(n(i)*e(i)));
52 end
53 if dot(vecr(i,:),vecv(i,))>=0
54     theta(i) = acosd(dot(vece(i,:),vecr(i,))/(e(i)*r(i)));
55 else
56     theta(i) = 360-acosd(dot(vece(i,:),vecr(i,))/(e(i)*r(i)));
57 end
58 E(i) = 2*atand(sqrt((1-e(i))/(1+e(i)))*tand(theta(i)/2));
59 M(i) = E(i)-e(i)*sind(E(i)); % Kepler's equation
60 averagen(i) = sqrt(mu/(a(i)^3));
61 T(i) = 2*pi/averagen(i);
62 tminustp(i) = M(i)/averagen(i);
63 while tminustp(i) < 0
64     tminustp(i) = tminustp(i)+T(i);
65 end
66 while tminustp(i) > T(i)
67     tminustp(i) = tminustp(i)-T(i);
68 end
69 end
70
71 %% Find the propellant mass needed with Tsiolkovski formula:
72 deltav = [deltavx deltavy deltavz];
73 deltam = m2*(1-exp(-norm(deltav)/(Isp*0.00981)));
74
75 %% Plot the results:
76 % Draw the planet
77 h1 = figure(1);
78 h1.Visible = 'off';
79 earth_sphere('km')
80
81 % Draw and label the X, Y and Z axes
82 line([0 2*R], [0 0], [0 0], 'LineStyle', '-'); text(2*R, 0, 0, 'X')
83 line([0 0], [0 2*R], [0 0], 'LineStyle', '-'); text(0, 2*R, 0, 'Y')
84 line([0 0], [0 0], [0 2*R], 'LineStyle', '-'); text(0, 0, 2*R, 'Z')
85
86 % Plot the orbit, draw a radial to the starting point and label the
    starting point (o) and the final point (f)
87 hold on
88 plot3(y(:,1), y(:,2), y(:,3), 'k.')
89 line([0 r0(1)], [0 r0(2)], [0 r0(3)], 'Color', 'r')
90 line([0 y(end,1)], [0 y(end,2)], [0 y(end,3)], 'Color', 'r')
91 text(y(1,1)-1000, y(1,2), y(1,3), 'o', 'FontSize', 16, 'Color', 'r')
92 text(y(end,1)+2000, y(end,2)+500, y(end,3)+1000, 'f', 'FontSize', 16, '
    Color', 'r')
93
94 if deltavx ~= 0 || deltavy ~= 0 || deltavz ~= 0
95     line([0 y02(1)], [0 y02(2)], [0 y02(3)], 'Color', 'g')
96     text(y02(1), y02(2), y02(3), ' v ', 'FontSize', 16, 'Color', 'g')

```



```
97 end
98
99 % Select a view direction (a vector directed outward from the origin)
100 view([1, 1, .4])
101
102 % Specify some properties of the graph
103 grid on
104 axis equal
105 xlabel('km')
106 ylabel('km')
107 zlabel('km')
108 hold off
109 saveas(h1,'orbit','jpg')
110
111 % Plot the error with respect to HORIZONS results:
112 abserr = y-statematrix;
113 h2 = figure(2);
114 h2.Visible = 'off';
115 plot(t/(24*hours),abserr(:,1:3))
116 xlabel('t [days]')
117 ylabel('error [km]')
118 title('Position error')
119 saveas(h2,'positionerror','jpg')
120
121 h3 = figure(3);
122 h3.Visible = 'off';
123 plot(t/(24*hours),abserr(:,4:6))
124 xlabel('t [days]')
125 ylabel('error [km/s]')
126 title('Velocity error')
127 saveas(h3,'velocityerror','jpg')
128
129 end
```

