



Gdańsk University of Technology
Faculty of Electronics,
Telecommunications and Informatics



Department:

Department of Geoinformatics

Author:

Tomasz Mrugalski

Studies type:

M. Sc. Studies

M. Sc. Dissertation

Dissertation topic:

**Trajectory determination and orbital
maneuvers in Earth vicinity**

Polish title:

**Wyznaczanie trajektorii i manewry
orbitalne w otoczeniu Ziemi**

Supervisor:

prof. dr hab. inż. Cezary Specht

Technical Supervisor:

dr inż. Mariusz Specht

Reviewer:

prof. dr hab. inż. Marek Moszyński

Gdańsk, 2021-11-02

Contents

List of Figures	vii
List of Tables	1
1. Introduction	3
1.1. Astrodynamics	3
1.2. Thesis overview and goals	3
1.3. History of orbital mechanics	4
1.3.1. Antiquity	4
1.3.2. Middle Ages	5
1.3.3. Age of Experiments	6
1.4. Orbits	7
1.4.1. Orbital elements	7
1.4.2. Orbit classification by shape	13
1.4.3. Orbit classification by altitude	14
1.4.4. Orbit classification by inclination	15
1.4.5. Special purpose orbits	15
1.4.6. Lagrangian points and exotic orbits	16
1.4.7. Energy perspective	19
1.4.8. Escape velocity	19
1.5. Spacetime Reference Frames	19
1.5.1. Reference Time	19
1.5.2. Coordinate Systems	20
1.6. Orbital maneuvers	21
1.6.1. Rocket Equation	21
1.6.2. In plane Maneuvers	22
1.6.3. Hohmann Transfer	22
1.6.4. Bi-elliptic Transfer	23
1.6.5. Rendezvous	24
1.6.6. Plane Change Maneuvers	25
1.7. Interplanetary trajectories	26
1.8. Orbital Perturbations	27
1.9. Miscellaneous Topics	28
1.9.1. Gauss's problem	28
1.9.2. Lambert's problem	28
1.9.3. Sphere of influence	29
1.9.4. Kessler syndrome	29
2. Related Work in Astrodynamics	31
2.1. Related work	31
2.2. Astrodynamics Books	31
2.3. Software tools	33

2.3.1.	System Tool Kit (STK)	33
2.3.2.	General Mission Analysis Tool (GMAT)	33
2.3.3.	Poliastro	34
2.3.4.	Cesium	35
2.3.5.	Python libraries	35
2.3.6.	Other software	36
2.4.	Notable papers	36
2.5.	Orbital File Formats	38
2.5.1.	Yuma format	39
2.5.2.	SEM almanac	39
2.5.3.	TLE format	40
2.5.4.	Orbit Mean-Elements Message Format	41
2.5.5.	MPCORB Format	43
3.	Developed Software	45
3.1.	Perylune Overview	45
3.1.1.	Design Assumptions	45
3.2.	Software architecture	46
3.3.	Software Layout	46
3.4.	Installation procedure	47
3.5.	User's Reference Manual	48
3.6.	Orbital Propagation using Cowell method	49
3.7.	Git repository	50
3.8.	Dependencies and Acknowledgements	50
3.9.	Example Charts	50
4.	Usage scenarios	53
4.1.	Problem 1: Georeferencing Satellite Images	53
4.1.1.	Calculating orbital position	53
4.1.2.	Converting TEME to Geodetic coordinates	55
4.1.3.	Georeferencing wide images	56
4.2.	Problem 2: Reviving or deorbiting old satellites	59
4.2.1.	Available Launch Services	59
4.2.2.	LEO Rendezvous Mission Design	62
4.3.	Problem 3: Are we all going to die on April 29, 2020? – Debunking fake news	66
4.3.1.	Obtaining ephemerids	67
4.4.	Problem 4: Interplanetary Transfer Windows	69
4.4.1.	Planet's Movement	71
4.4.2.	Launch Windows	75
4.4.3.	Hohmann transfer example	75
4.5.	Problem 5: Asteroid Survey	78
4.5.1.	Asteroid Discovery Process	78
4.5.2.	MPC Asteroids Database	78
4.5.3.	Asteroids Overview	79
4.5.4.	Osculating and Perturbed Orbits	80
4.5.5.	Asteroid Target Evaluation	80

4.6. Problem 6: Navigating with low power engine or propellantless cubesat	83
4.6.1. Simulating Ion Thrusters	84
4.6.2. Solar Sail	84
4.6.3. Solar Sail Performance Summary	87
4.6.4. Atmospheric drag	88
4.6.5. PW-Sat2 Sail	88
4.7. Problem 7: Website for tracking Polish Satellites	91
4.7.1. Web Interface Capabilities	91
4.7.2. Potential Future Improvements	93
5. Solution evaluation	95
5.1. Software Validation	95
5.2. Test Driven Development	95
5.2.1. Test Warnings	97
5.2.2. Testing Dependencies – Poliastro	97
5.3. Reporting and Fixing Bugs	97
5.4. Comparison with well-established sources	98
5.5. Appeal to the authority	98
5.6. Comparison with Geographical Data	98
5.7. Comparison with PW-Sat2 Data	99
5.8. Additional Verification	99
6. Achievements and Conclusions	101
6.1. Achieved Goals	101
6.2. Conclusions	102
6.3. Organizational Suggestions	103
6.4. Future work	104
Appendix A. Ephemerides For Asteroid 1998 OR₂	107
Bibliography	113
Index (EN)	119
Indeks (PL)	121

List of Figures

1.	Copernican model	6
2.	Conic sections with a plane	7
3.	A GPS orbit	8
4.	An orbit with high eccentricity	9
5.	Major (a) and minor (b) semi-axes	9
6.	Inclination	10
7.	Right ascension	10
8.	Argument of periapsis and true anomaly	12
9.	Eccentric anomaly	13
10.	Libration points	17
11.	Lissajous orbits	17
12.	NRHO orbits	18
13.	Orbital transfer example	23
14.	This diagram presents the Δv required for Hohmann (black line) and various bi-elliptic transfers (colored curves) between two circular orbit in function of their semi major axes. Source: [95].	24
15.	Orbital Rendezvous	25
16.	The Gauss's problem. Given the two known positions r_1 and r_2 and time of flight, there are exactly two orbital solutions that match the time of flight: the short way (left) and the long way (right).	28
17.	Bibliography	32
18.	STK Interace	34
19.	STK Orbit Design Tool	34
20.	General Mission Analysis Tool	35
21.	GPS Constellation	36
22.	Multiple Assists Trajectory proposed by project Lyra	38
23.	An example of orbital data for Navstar GPS satellite PRN-01 in Yuma format.	39
24.	An example of orbital data for a satellite in SEM format. The R0 to R8 are line numbers are not part of the format.	40
25.	An example of orbital data for a Światowid satellite in TLE (Two Line Element) format.	40
26.	OMM data has several echange formats defined.	41
27.	OMM data has several echange formats defined.	42
28.	An example of orbital data for a (1) Ceres asteroid in MPCORB format. The data has been wrapped (\) to increase readability. Original format expects one orbit per line.	43
29.	Perylune Directory Tree	46
30.	Inner Solar System Overview	50
31.	Side view of Inner Solar System	51
32.	Raw Satellite image	54
33.	Oblate Earth	56
34.	Geodetic projection	57
35.	Georeferencing Satellite images	58

36.	NOAA-17 Status	62
37.	Visualization of the initial orbits: GDASAT-1 just after orbital insertion (orange), the target orbit of NOAA-17 (red). Image generated using <i>Perylune</i> software.	63
38.	Visualization of the orbits after 2 maneuvers: J2 perturbations to sync RAAN and inclination change: GDASAT-1 (orange), the target orbit of NOAA-17 (red). Earth size and orbital altitude are to scale. Image generated using <i>Perylune</i> software.	65
39.	Sensationalist article about upcoming asteroid fly-by and a question whether it will kill all humanity. The title says “NASA warns: An asteroid speeding towards Earth. Will end of the world happen on April 29th, 2020?”. The publication date is March 23rd, 2020, or roughly a month before the closest approach.	66
40.	Web interface of the Horizons database hosted at SSD (Solar System Dynamics) at JPL (Jet Propulsion Laboratory). The Horizons database provides ephemerids for over a million asteroids, 3600 comets, 209 moons and minor planets, 8 planets and many other objects, such as barycenters, lagrangian points and more.	67
41.	The distance between 1998 OR2 and Earth is 6.3 million kms from Earth at the closest approach. As such, it does not pose any danger.	69
42.	1998 OR2 asteroid (orange) trajectory in the Earth vicinity (blue dot). The Moon’s orbit (green line) is shown for scale. Asteroid’s and Moon’s trajectory plotted for period of 2020-04-20 to 2020-05-10.	69
43.	Mercury, Venus, and Mars distance from Earth for years 2012-2040	73
44.	Jupiter, Saturn, Uranus, Pluto distance from Earth for years 2012-2040	74
47.	Mars 2020	75
45.	Mars transfer window, taking advantage of the 2020 opposition.	76
46.	Venus transfer window, taking advantage of the 2021 opposition.	77
48.	Histogram of semi-major axes	79
49.	Histogram of eccentricity	80
50.	Scatter plot of eccentricity vs semi major axis	81
51.	Top 10 asteroids	83
52.	Constant acceleration perturber. This type of perturber is a good model for ion engine. In this particular example, it was configured to provide constant forward thrust. The initial osculating orbit has been marked with dashed red line. The perturbed trajectory is marked in green and shows a period of almost 16 hours, which is roughly 10 orbits.	84
53.	Solar sail perturbation for PW-Sat2 orbit.	86
54.	A change of the altitude over a period of $2 \cdot 10^6$ s (roughly 23 days) due to solar radiation pressure exerted on 4 m^2 sail on a 2,5 kg satellite.	87
55.	Zoomed in first several days of simulated operation. Each orbital cycle has slightly higher apoapsis (red line) and barely measurable increase in periapsis (green line). The average altitude grows slowly.	87
56.	Two popular atmospheric models are COESA 62 and COESA 76, often referred to as U.S. Standard Atmosphere. The earlier (COESA62) model covered altitudes from 0 km to 700 km (dashed red), while the update (COESA76) covers range of 0 km to 1000 km (solid blue), with some extra coverage for negative altitudes, which are useful for simulating phenomena in depressions. This diagram is roughly based on diagram from [59].	89
57.	Atmospheric drag impact on 100 kg satellite moving on 250 km equatorial orbit.	90
58.	Comparison of actual and simulated PW-Sat2 orbit deterioration due to atmospheric drag. The results of <i>Perylune</i> simulation was marked in dashed red.	91
59.	Several views of the website developed.	92

-
60. Comparison of available and simulated PW-Sat2 orbit deterioration due to atmospheric drag. . . . 94
 61. The *Perylune* software uses Travis-CI platform to run tests automatically. The tests are written using standard pytest framework, the default recommended solution for all Python software. . . . 96

List of Tables

1.	Common apsis names in English and Polish	11
2.	Shape of Earth as defined in popular Geodetic Systems	55
3.	Existing and prospective orbital launch services. The first section lists Polish satellites along with the launch providers used. Second section lists an example university cubesat. The third section contains a list of low cost launch providers.	61
4.	Calculated burns needed to synchronize initial (GDASAT-1 launch) to target (NOAA-17) orbits. . .	65
5.	Known major and dwarf planets in the Solar System. Data calculated in Perylune, using ephemerids from NASA Horizons database. All parameters calculated for 2020-07-15.	70
6.	Hohmann transfers to all major and dwarf planets and two example asteroids. All departures from Earth. v_{h1} is heliocentric velocity of the departing planet (Earth, before first burn). v_{hoh1} is a velocity required to enter transfer orbit (after first burn). v_{h2} is heliocentric velocity of the target body (velocity after second burn). v_{hoh2} is the arrival velocity of the transfer orbit (before second burn). Δv_{hoh1} and Δv_{hoh2} are Δv requirements for first and second Hohmann burns. All values calculated for average Sun distance of the target.	72
7.	Top 20 asteroids that are easiest to reach from Earth. The Δv in relation to spacecraft at C3=0 (escape velocity from Earth system). The list is ordered by increasing total Δv , which assumes rendezvous and two Hohmann burns. For fly-by missions, only the first Hohmann burn is necessary.	82
8.	List of popular atmosphere models	88

1. Introduction

The first chapter formulates the goals of this thesis. It also introduces the topic of astrodynamics, the history of its development, introduces modern notations, reviews orbit types and orbital maneuvers.

1.1. Astrodynamics

The orbital mechanics (*pol.* mechanika orbitalna), or astrodynamics (*pol.* astrodinamika), is a branch of applied science concerned with the study of the motion of natural (planets, asteroids, comets, etc.) and man-made bodies (probes, spacecraft, rockets, etc.) in outer space. The motion of such objects is most typically predicted using Newton's laws of motions and law of universal gravitation [86], but other forces, such as propulsive maneuvers, solar pressure, Earth oblateness, solar wind, and other phenomena are taken into consideration as well.

1.2. Thesis overview and goals

The primary goal of this thesis is to provide a software framework for solving selected problems in astrodynamics. The envisaged applications are two-fold. First, the framework, together with this thesis, can be used to research related topics. Second, the studied problems are chosen in a way so they could be directly applicable to space missions being considered in Poland in the near future.

The following topics were researched, and their solutions are described in this thesis:

Problem 1: Georeferencing satellite images. In a separate project [69], the author constructed a satellite ground station that can receive weather images from NOAA satellites. The acquired data without additional information make the satellite images hard to read due to frequent cloud cover. Georeferencing and overlaying country and land contours would increase the readability of the images. This requires calculating the reference position of an image obtained from an Earth observation satellite, based on its known properties (orbital elements, sensor dimensions, optical properties), converting to geodetic reference systems, such as WGS-84, and then overlaying country borders, grid, and other types of data.

Problem 2: Reviving or deorbiting old satellites. Many older satellites still have fully functional electronic and optical systems, but they can no longer conduct their primary missions due to running out of fuel. The idea is to propose a trajectory for a small satellite that would perform a rendezvous maneuver and dock with an old dysfunctional satellite and act as its strap-on engine. This can regain control over the satellite if its onboard systems are still functional or deorbit if they are not. This problem focuses on on-orbit navigation. It requires designing several maneuvers: achieving orbit after launch, inclination change, orbit raise using Hohmann transfer, chase, and rendezvous maneuvers.

Problem 3: Debunking fake news. News media are often publishing alarming articles about the upcoming asteroids close fly-bys with varying levels of inaccuracy and fake sensationalism. The recent close flyby of asteroid 1998 OR-2 on 29 April 2020 was a good example. This problem aims to present an easy to follow step by step explanation of how to calculate ephemerides for the upcoming close passes of asteroids and comets, including current known uncertainties, and compare it to the predicted closest distance.

Problem 4: Interplanetary transfer windows. The difficulty of reaching a place in a Solar system is not expressed in the distance but in the relative change of velocity needed to get there. Due to the bodies being in constant movement, the difficulty changes over time. There are specific configurations where reaching one planet from another are easier. Such periods of favorable configurations are called transfer windows. For example, the transfer windows for Earth-Mars are open roughly once every two years. One of the goals here will be to calculate charts for choosing optimal departure times.

Problem 5: Asteroid Survey. As of today, there are close to 1 million asteroids known in the Solar System. Many of them belong to a NEA (Near Earth Asteroid) class. This problem aims to review existing known NEA asteroids and assess the difficulty of reaching them. This problem brings in the additional complexity of reaching Earth escape velocity, changing the frame of reference from geocentric to heliocentric, changing inclination and other orbital parameters to match those of the target.

Problem 6: Navigating with low force engine CubeSat. The economic reality implies that Poland is currently incapable of launching any satellites larger than CubeSats. This form factor is too small to have any conventional chemical propulsion. However, several possible alternative propulsion mechanisms can be taken into consideration. One of them is a solar sail that uses solar radiation pressure to generate a small but constant acceleration. This has already been demonstrated with PW-Sat and PW-Sat2. Another more ambitious one is an ion engine. It uses a high electromagnetic field to accelerate ionized noble gas (e.g., Xenon). The characteristic of ion propulsion is low thrust, long burn durations, and high specific impulse.

A solar sail could be used to perform some maneuvers, such as raising the perigee and apogee of the orbit. One complication is that the force vector always points directly outwards from the Sun. This would imply the sail would have to change orientation in various sections of its orbit around Earth. However, that should be doable with magnetorquer, an ingenious mechanism that generates magnetic dipole that interfaces with Earth's magnetic field, thus providing torque and eventually rotating the spacecraft.

This problem aims to investigate if a CubeSat mission could use a solar sail for actual navigation. In a sense, such a mission would be a follow-up to the PW-Sat2 mission that proved that small solar sails could be deployed and used in space.

Problem 7. Website for observing Polish satellites. This idea requires predicting future fly-overs of several satellites, calculating their ground track, and choosing those close passes that are close enough to specific observer positions. An additional difficulty is to take into account whether the satellite is in Earth's shadow or not. The most favorable conditions are when the observer is in the shadow, but the satellite is still in the Sunlight.

1.3. History of orbital mechanics

1.3.1. Antiquity

The history of astrodynamics (*pol.* historia astrodyamiki) is as old as the history of mankind. Our earliest ancestors were intrigued by the celestial sphere since its earliest days. The oldest successful attempts to record observed sky phenomena are carvings in ancient caves made many millennia ago. Although interesting from the historical and perhaps artistic point, they hold no scientific value. The first steps towards understanding the rules governing objects in the sky were taken in ancestry. Surprisingly many of the inventions and notations invented in ancient times survived and are still in use today. Babylonian concept of sexagesimal system (*pol.* system sześćdziesiątkowy), a system using 60 as a base, is still in common use. The full angle is represented as six parts of 60 degrees; an hour has 60 minutes, which further splits into 60 seconds. The same is true for degrees split into arc minutes and then arc seconds.

Another excellent example of ancient notation that withstood the trial of time is an invention of Claudius Ptolemy [57]. The Almagest, dated ~140CE, is a fascinating discourse about fundamental phenomena, such as

parallax, solar and lunar eclipses, basic spherical trigonometry, and much more. Surprisingly enough, one of the concepts introduced is clearly incorrect, but nevertheless is still being frequently used in modern times. Back then, there were four planets known: Venus, Mars, Jupiter and Saturn, as those are easily visible by the naked eye. The model proposed putting Earth in the middle, with translucent seven spheres around it: four for known planets, one for Sun, another one for Moon, and the final seventh was reserved for the fixed, immovable stars and was considered the perfect place. A popular saying "I'm in seventh heaven" (*pol.* jestem w siódmym niebie) comes from this archaic model.

Another concept that the Almagest is most known for is the star catalog that lists over 1000 stars. Ptolemy segregated visible stars into 6 magnitudes, with 1 being the brightest and 6 the faintest. This concept survived and the magnitudo scale (*pol.* magnitudo) is one of the essential scales in astronomy today. Although it has gotten a more precise definition (an object of magnitude $n + 1$ is $\sqrt[5]{100}$ times fainter than n), the scale is still faithful to the fundamentals laid out by Ptolemy back in the 2nd century in ancient Greece.

1.3.2. Middle Ages

Mikołaj Kopernik made the next major step forward in understanding heavenly bodies in his "De revolutionibus orbium coelestium" (Latin for On the Revolutions of the Heavenly Spheres) [31], published in 1543. The proposed heliocentric model (*pol.* model heliocentryczny) was a radical change compared to prior dominating geocentric model (*pol.* model geocentryczny). The most famous diagram presenting something we call Solar system today comes from its Book I and is presented in Fig. 1. An observant reader will immediately notice that the figure also presents the Copernican model's biggest flaw that caused much trouble for its author and almost led to its rejection – Copernicus incorrectly assumed that the orbits are circular. As we know today, the orbits are ellipses, but many brighter planets have low eccentricity, so their orbits are almost circular. However, the discrepancies between the circular model and actual planets movements were considered a serious argument against the Copernican model back when it was proposed.

Another frequent misconception is that Copernicus was the first one to propose a heliocentric model. Technically, the first book that discussed the concept was "Narratio Prima" [44], which was published in 1539, four years before "On the Revolutions...". However, its author Jerzy Joachim Retyk was a Copernicus friend and was merely reporting on unpublished works of Copernicus. However, Aristarchus of Samos proposed a heliocentric model in 270 BC. Sadly, his writings on this concept did not survive to modern times.

The next improvement came roughly 7 decades later with "Harmonices Mundi" (Latin for Harmonies of the World) [30], published in 1619 by Johannes Kepler. It defines laws of orbital motion that are considered fundamental for the modern-day understanding of orbital mechanics. The major novelty was the realization that orbits are ellipses (first law), not circles. The laws also defined the change of orbital velocity (second law) and the relation between orbital periods (third law). Another significant contribution was the introduction of the concept of inertia. The advances in the understanding of the universe caused some disturbance. Book I of related "Epitome of Copernican Astronomy" was listed on an index of books forbidden by an inquisition.

Moving on, the next significant step forward came with the book of "Philosophiæ Naturalis Principia Mathematica" (Latin for Mathematical Principles of Natural Philosophy), by Isaac Newton [51]. Better known as "Principia", the book is considered the most important work in the science history. It introduces Newton's laws of motion, which laid the foundations for classical mechanics, the law of universal gravitation, and derived Kepler's laws of orbital motion (which Kepler obtained empirically). In the orbital mechanics' context, it is worth pointing out that Newton was also the first person who calculated the escape velocity for Earth [88].

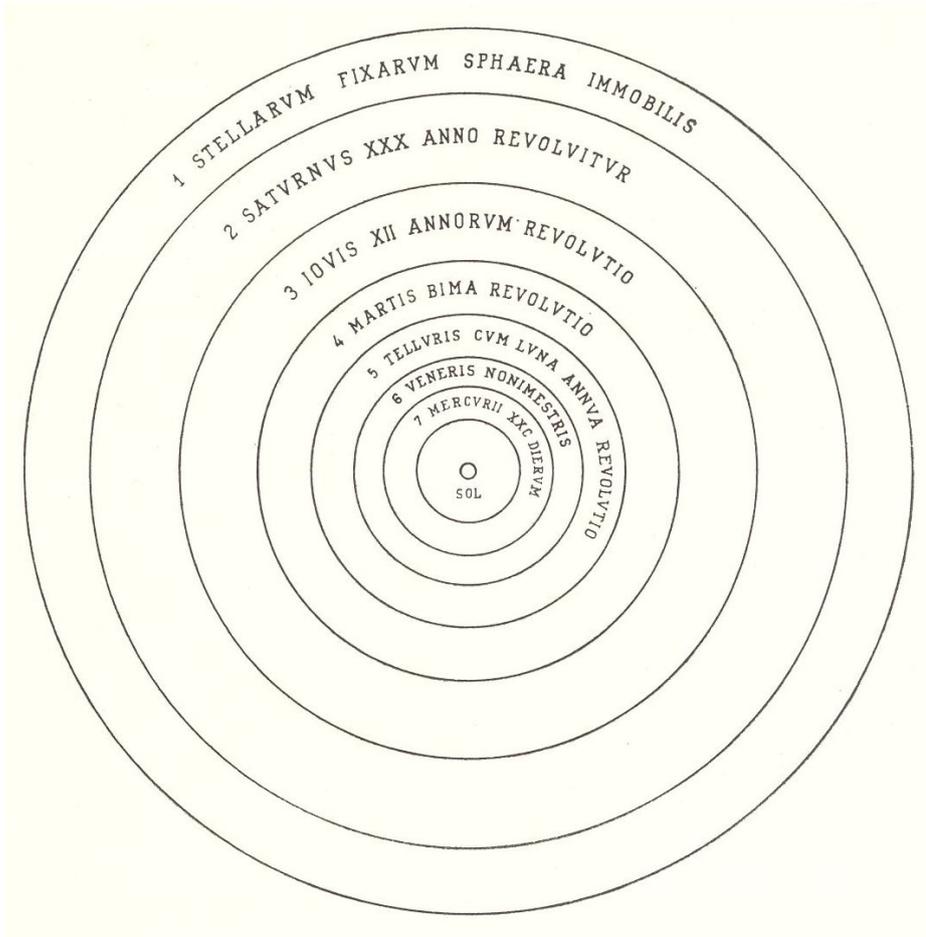


Figure 1: The first heliocentric model, proposed by Mikołaj Kopernik. This famous illustration shows both its greatest contribution and its flaw at the same time. The orbits are incorrectly assumed to be circular, not elliptical. Source: “De revolutionibus orbium coelestium” [31]

1.3.3. Age of Experiments

Until the beginning of the 20th century, the study of orbital mechanics was a purely observational field. That has been forever changed with new propulsion concepts based on expelling matter that we know as rockets (*pol.* rakietą) today. Konstantin Ciolkowski and his “Exploring world spaces with rocket devices” [13] published in 1903 introduced not only the rocket equation (*pol.* równanie rakietowe) that is a foundation of the modern rocket industry but also explained how to use it to conquer space and gave many great practical ideas, such as an aerodynamic tunnel. That is even more impressive, given that the book was published before the first powered flight of the Wright brothers. See Section 1.6.1 for a brief discussion.

Significant progress has been made in Germany during World War 2. The V-2 rocket was sadly used mostly as a weapon that delivered deadly explosives. However, it significantly expanded humanity’s understanding of powered suborbital flight. While the great majority of flights were configured for the horizontal range to reach distant targets such as London, V-2 could have been and were launched vertically. On 20 June 1944, a V-2 rocket variant with designations MW 18014 reached an altitude of 176 km. At that time, the Kármán line (*pol.* linia Kármána) was not defined yet, but nevertheless, it became the first man-made object that crossed the boundary of space.

After the war, Werner von Braun continued his work in the United States, where he headed the space program that culminated in the first manned Moon landing. Von Braun was also interested in expanding the scope of applications of rockets. In 1949, he published “Project Mars, a technical tale” [82], which is considered a first realistic proposal for interplanetary trajectories.

The last major historic contribution that should be noted was made by Buzz Aldrin. While his most known

feat was being the second man ever to set foot on the Moon, his another substantial contribution was his doctoral thesis “Line-of-Sight Guidance Techniques for Manned Orbital Rendezvous” [4], published in 1963. It laid out the principles of precise orbital maneuvers needed for two spacecraft to meet.

1.4. Orbits

In a general case, an orbit is one of the conic curves, i.e. one of the curves obtained as the intersection of the plane and the cone. In particular, this could be a circle, ellipse, hyperbola or parabola. Visual representation of those shapes is presented in Fig. 2.

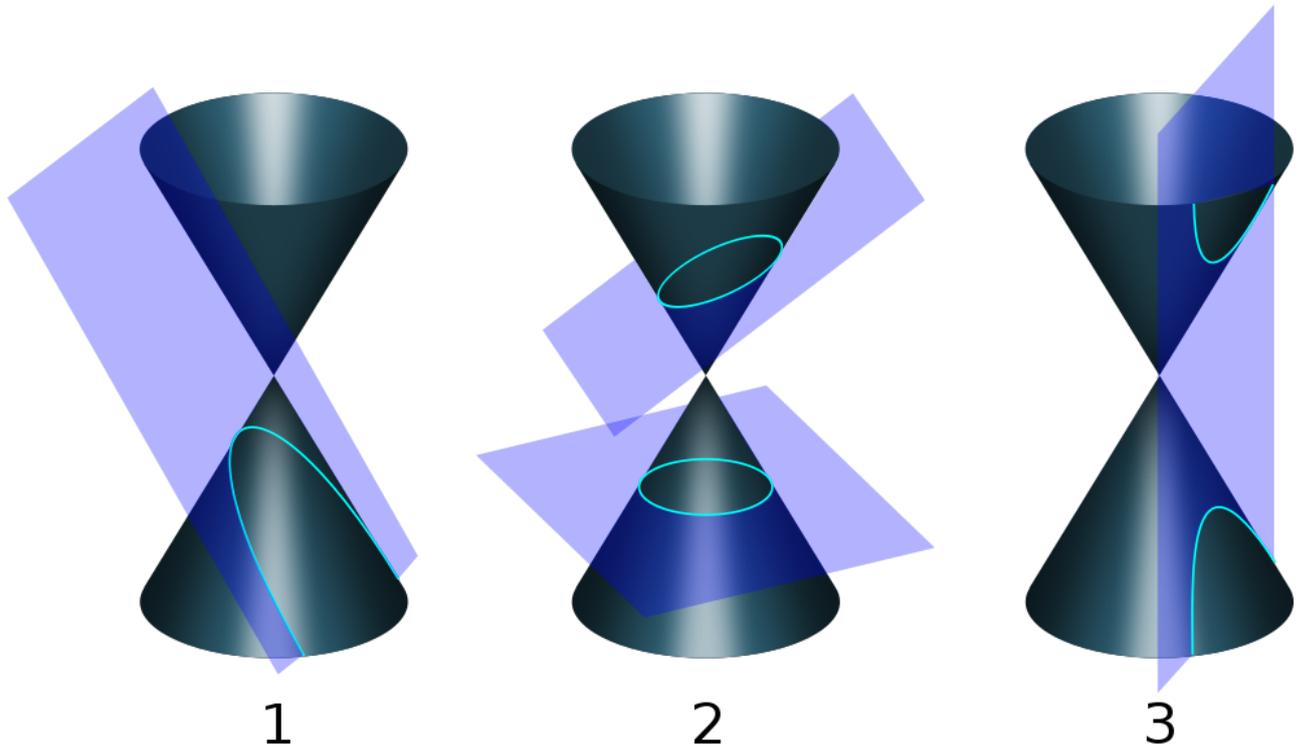


Figure 2: Orbit shapes are so called conic sections, i.e. the shapes created by intersecting a cone with a plane. Depending on the angle, the shape could be 1) parabola, 2) circle or ellipse, 3) hyperbola. Source: wikipedia [87]

Regardless of its possible varied shapes, five parameters are needed to define the shape of an orbit. The sixth parameter is needed if the location of an object along that orbit has to be specified. Those six are commonly referred to as orbital elements (*pol. elementy orbitalne*) or keplerian elements (*pol. elementy keplerowskie*). Those parameters are discussed in detail in the following paragraphs.

Many additional parameters may be used to specify optional data, such as a time when the orbit was specified or describe how an orbit changes over time or is affected by various phenomena. The activity of determining why the actual satellite’s orbit differs from the mathematical orbit is called orbital perturbation analysis. See Section 1.8.

1.4.1. Orbital elements

The five base orbital elements are eccentricity, major semi-axis, inclination, right ascension of the ascending node, and the argument of periapsis.

Eccentricity (*pol. mimośród*) — This single parameter is denoted with e . Its value changes from 0 (perfectly circular orbit) to infinity. It is defined as:

$$e = \sqrt{\frac{a^2 - b^2}{a^2}} \quad (1.1)$$

where a and b are major semi-axis and minor semi-axis. Those are well defined and easily understood for circles and ellipses, but not for parabola and hyperbola. Value $e = 0$ defines a circle, values in range $(0..1)$ define an ellipse. Those two are commonly called closed orbits as the bodies on such orbits can repeat their flight indefinitely given the absence of external forces. Those orbit types describe almost all satellites, moons, planets, asteroids, and the majority of comets. An example of an orbit with very low eccentricity (nearly circular) of a Navstar GPS satellite is presented in Fig. 3. Higher eccentricity is presented in Fig. 4 and Fig. 5.

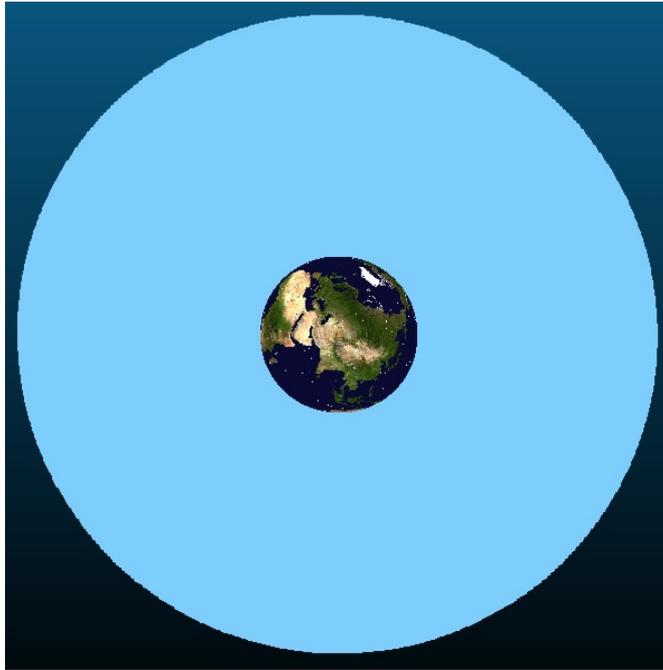


Figure 3: This image presents typical GPS orbit which has very low eccentricity (0.005) and is almost circular. Screen capture done in CloudCompare software.

The third orbit type is a parabola, and it is defined for $e = 1$. When moving away from the main body, it is called escape orbit (*pol.* orbita!ucieczkowa). When moving towards the main body, it is called capture orbit. That is mostly a theoretical concept, as it is impossible to achieve eccentricity of precisely 1. Usually, it is slightly less than 1 (orbit becomes highly elliptic) or slightly over 1 (orbit becomes hyperbolic).

Fourth type of orbits ($e > 1$) is a hyperbolic orbit (*pol.* orbita hiperboliczna). Similar to a parabola, those are open orbits and can be flown only once. As such, those are sometimes referred to as trajectories, not orbits. Hyperbolic orbits are used to describe planetary flybys, gravitational slingshots, and objects visiting from interstellar space. One natural example of this class is 'Oumuamua, an asteroid discovered in 2017 with an eccentricity $e > 1.2$. That is the first confirmed object of interstellar origin [54]. As such, it received designation 1I, the first object in a new class.

Major semi-axis (*pol.* pó!o! wielka) — An ellipse has two symmetry axes – longer (major) and shorter (minor). A major semi-axis, denoted with a , defines half of the longer axis of an ellipse. As an interesting observation, it is worth noting that an ellipse can be defined with any 2 out of 3 parameters: a, b, e . Given any two, the third one can be calculated. For orbital calculations the eccentricity is crucial, so only one semi-axis is needed and by convention a is used. Thus b is rarely used. The major and minor semi-axes are presented in Fig. 5.

This parameter is often imprecisely called the semi-major axis. The eccentricity and major semi-axis are the only parameters needed to describe the shape of an orbit. In certain limited cases where only the shape is important, but not its location, an orbit may be described with just a and e . For example, when discussing comets or asteroids,

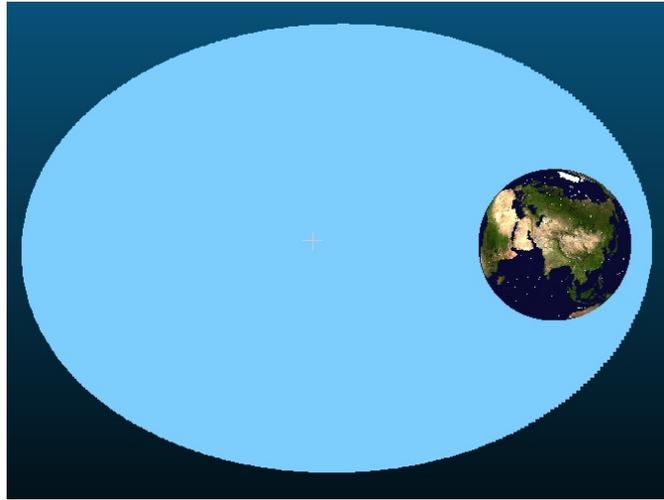


Figure 4: For very low e values the concept of eccentricity is difficult to present as the orbits look almost ideally circular. This image was created using a modified GPS almanac ($e = 0.7$) and CloudCompare software.

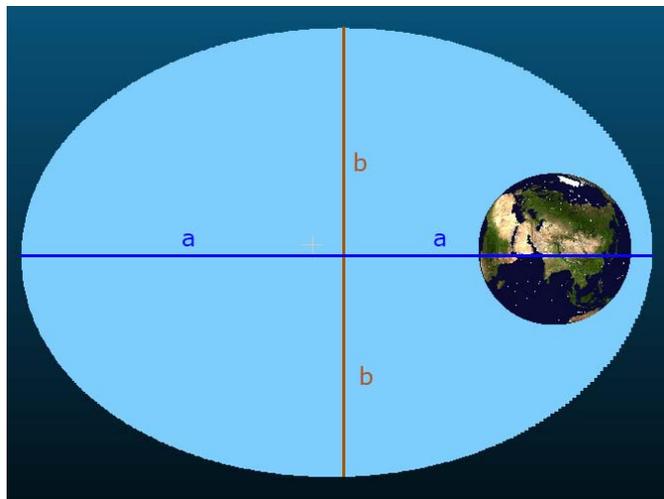


Figure 5: Major (a) and minor (b) semi-axes.

it is possible to classify them, calculate minimal and maximum Sun distance and roughly assess whether an object could cross Earth orbit using just a and e parameters.

Inclination (*pol.* inklinacija) – A body moving on an orbit moves on so-called orbital plane. Each central body being orbited, such as Earth, has an equator. An equator extended to infinity forms an equatorial plane. An angle between the equatorial plane and the orbital plane is called inclination. It is typically denoted with i . The inclination has been presented in Fig. 6. While in practice, most satellites and man-made objects in space are on orbits with an inclination between 0 and 90° , technically, an orbit could have an inclination up to 180° . However, it is very inefficient, thus not used. Orbits that have an inclination 0 and 180° are located on the equatorial plane, and the object would move over the equator. All geostationary satellites are using inclination 0. Orbits with $i \in (0..90^\circ)$ are said to be prograde (moving in the same direction as Earth surface). The orbits with $i \in (90..180^\circ)$ are said to be retrograde (moving in the opposite direction as equator). For $i = 90^\circ$ the orbit is called polar (*pol.* orbita polarna), because the satellite will pass over North and South poles. Earth monitoring satellites often use polar or near-polar orbits. Retrograde orbits ($i > 90^\circ$) are rarely used.

Most natural objects, such as planets and asteroids, were created from accretion disk when the Solar system was forming. While the dust matter collapsed into large bodies, the angular momentum was retained. As a result, most natural objects are rotating around the Sun in roughly the same direction. However, there are some exceptions.

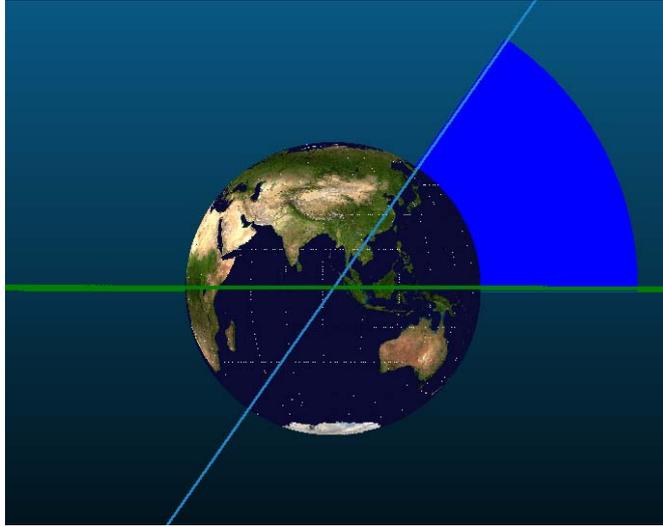


Figure 6: Orbital plane (blue line) forms certain angle with equatorial plane (green). The inclination is the angle measured from equatorial to orbital plane.

Triton, the largest Neptune's moon, and Phoebe, one of many Saturn's moons, are on retrograde orbits. The leading theory explaining this oddity proposes that those were not formed in their current place, but instead were formed elsewhere and then captured by their planets.

There are even fewer man-made objects using retrograde orbits, due to significant requirements for Δv (see Section 1.6.1) to achieve those orbits and associated increased fuel and mass requirements. One example is a series of Ofeq satellites launched from Israel. Due to political stress in the region, it was unacceptable to launch rockets in the eastern direction, as it could easily be misinterpreted as a military attack and could have triggered a war. Also, Israelis wanted to avoid potential debris to be investigated to prevent technology transfer. Launching westwards over the Mediterranean sea was a better choice.

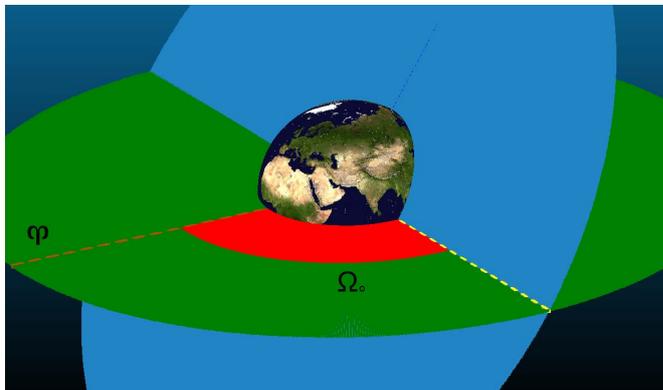


Figure 7: Aries point (Υ) and right ascension of the ascending node (Ω_0)

Right ascension of ascending node (*pol.* rektascensja węzła wstępującego), also called longitude of the ascending node (*pol.* długość węzła wstępującego), or simply RAAN, is often designated with Ω_0 . Some several additional points and angles have to be defined before RAAN can be explained. First, a distinguished direction is needed. For Earth coordinates this distinguished direction was agreed to be a prime meridian, a meridian crossing Greenwich observatory in London. However, since Earth rotates, using prime meridian as a rotating frame reference would complicate calculations tremendously. A different point was selected called Aries point (*pol.* punkt Barana). The Earth completes an orbit around the Sun every year. As observed from Earth, the Sun's apparent movement completes a full circle across the background of stars. That path is called an ecliptic (*pol.* ekliptyka). By definition, it is coplanar with Earth's orbit around the Sun.

Body	Periapsis (eng/pol)	Apoapsis (eng/pol)
Earth	perigee/perygeum	apogee/apogeum
Moon	perilune or perycynthion/perylune	apocynthion/apolune
Sun	perihelion/peryhelium	aphelion/aphelium
Jupiter	peryjove/peryjowium	apojove/apjowium

Table 1: Common apsis names in English and Polish

Earth's rotation is tilted roughly $23^{\circ}27'$ and thus the equatorial plane and ecliptic plane create precisely that angle. During its precession throughout the year, the Sun spends six months over northern and remaining six months over the southern hemisphere. The place an object (Sun in this example) passes equatorial plane from southern to northern hemisphere is called ascending node (*pol.* węzeł wstępujący). The opposite transition (from northern to the southern hemisphere) is called descending node (*pol.* węzeł zstępujący). Technically, the Aries point is an ascending node of the Sun. More intuitively, this is a point where Sun is observed on the sky during vernal or spring equinox (*pol.* równonoc wiosenna). Surprisingly enough, although the point took its name from Aries constellation (*pol.* Baran), it is currently located in Pisces (*pol.* Ryby) due to Earth precession. The orbital plane crossing the equatorial plane forms two points: ascending node (where an object passes from South to North) and descending node (where an object passes in the opposite direction). The right ascension of the ascending node is defined as an angle between the Aries point and the ascending node as measured on the equatorial plane. Although it defines an angle, and thus could be represented in radians ($0..2\pi$) or degrees ($0..360^{\circ}$), by convention (especially in the astronomy community) it is often specified in hours, minutes and seconds. Using hh:mm:ss notation greatly simplifies astronomical calculations, e.g. when calculating raising, culmination and setting times. The Aries point and the RAAN angle are presented in Fig. 7.

Argument of periapsis (ω) (*pol.* argument perygeum) – each orbit has its closest point to the body being orbited called periapsis and closed orbits ($e < 1$) also have a point of farthest distance called apoapsis. Apoapsis and periapsis are general terms and can be applied to any body being orbited. However, some frequently used bodies have their own suffixes. For geocentric orbits it is *-gee*, for lunar it is *-lune*, for Mars it is *-areion*, for Jupiter it is *-jove*. More commonly used names are listed in Table 1.

Therefore when dealing with objects around Earth, this parameter is usually referenced to as argument of perigee. However, it should be kept in memory that the name is Earth specific and should not be used for objects orbiting other bodies. The argument of periapsis defines an angle on the orbital plane between ascending node and a periapsis (perigee for Earth) and is typically denoted with ω . Expressed in degrees ($0..360^{\circ}$) or radians ($0..2\pi$). A graphical interpretation is presented in Fig. 8 as an angle in orange.

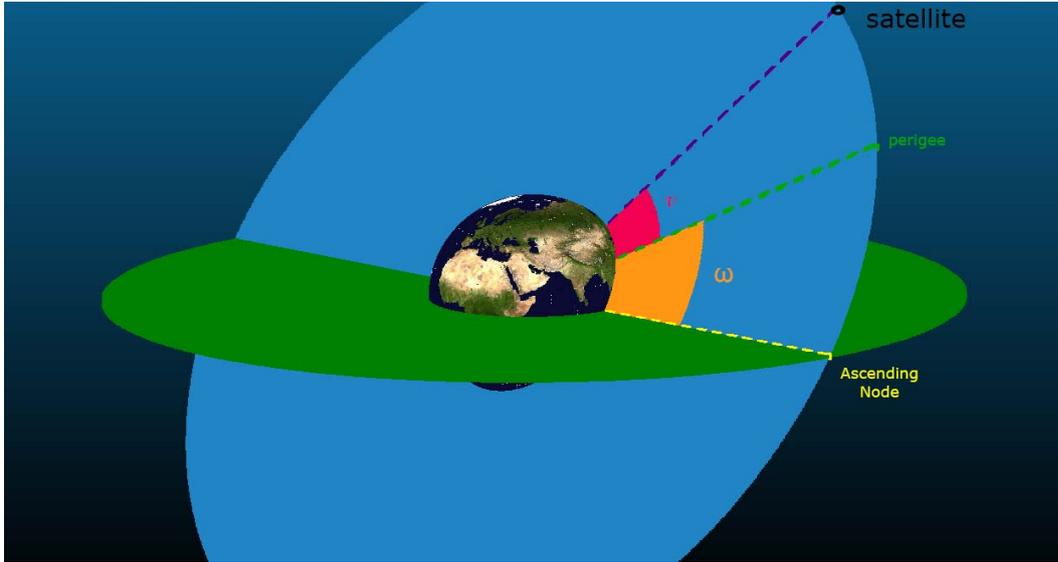


Figure 8: argument of periapsis (ω) and true anomaly (ν)

True anomaly (*pol.* anomalia prawdziwa) is usually denoted with ν (Greek letter nu), Θ or f . Previous parameters (major semi-axis a and eccentricity e) defined a shape of an orbit. The next two parameters (inclination i and RAAN Ω_0) defined the orbital plane, i.e. the plane the orbit and the moving body is located on. With the argument of periapsis (ω), it precisely defines an orbit in 3D space, i.e. a trajectory of an object moving through space. A sixth parameter is needed to locate the object on this trajectory. The true anomaly is an angular parameter that defines the position of a body moving along an orbit. It is the angle measured on an orbital plane between the direction of periapsis and the body's current position. It is presented in Fig. 8 above, in light red color.

It is worth pointing out that second Kepler's law dictates that a line connecting a body moving on an orbit of a central body sweeps out equal areas during equal time intervals. The most profound implication of this fact is that orbiting bodies change their linear and angular velocity on non-circular orbits as they progress throughout the orbit. Therefore it is often inconvenient to specify true anomaly and mean anomaly is provided instead. **Mean anomaly** (*pol.* anomalia średnia) M is the fraction of orbit's period that has elapsed since the moving body passed periapsis and is expressed as an angle. It defines an angular distance the body would have moved if it was on a circular orbit with the same period as the actual orbit.

Another related parameter is **eccentric anomaly** (*pol.* anomalia ekscentryczna) E , which is somewhat tricky to explain. First, we need to define an auxiliary circle (*pol.* koło opisane) of radius a , which is an outer bound of the orbit that touches the orbit in its periapsis and apoapsis points. The eccentric anomaly defines an angle between lines connecting periapsis with orbit focal point and a perpendicular line to the major axis and connecting to the point that passes through a given point P and lies on the outer circle. The visual representation of the eccentric anomaly is presented in Fig. 9.

It is sufficient to specify one of the anomalies as others can be calculated from the Kepler equation:

$$M = E - e \cdot \sin(E) \quad (1.2)$$

This equation doesn't have a closed-form solution (a solution obtained in a finite number of operations). Therefore it is usually calculated using numeric methods (e.g. iteratively). True anomaly ν can be calculated from eccentric anomaly E using the following equation:

$$\cos \nu = \frac{\cos(E) - e}{1 - e \cdot \cos(E)} \quad (1.3)$$

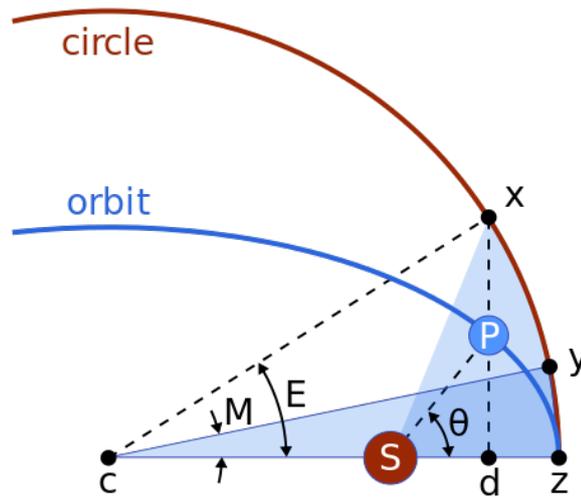


Figure 9: Eccentric anomaly E is an angle between d (a point on line of apsides) and a point x lying on both the auxiliary circle (brown) and a line perpendicular to line of apsides ($d-x$). source: wikipedia.org

It's possible to calculate the true anomaly from mean anomaly M , but it is more complex as it is based on Fourier expansion:

$$\nu = M + (2e - \frac{1}{4}e^3)\sin(M) + \frac{5}{4}e^2\sin(2M) + \frac{13}{12}e^3\sin(3M) + O(e^4) \quad (1.4)$$

where $O(x)$ is a Big-O notation that indicates the omitted terms are all of the order e^4 or smaller.

1.4.2. Orbit classification by shape

Circular orbit (*pol.* orbita kołowa) has no eccentricity ($e = 0$) and does not have designated periapsis or apoapsis. As such, the velocity and altitude are constant over time. The constant altitude makes them well suited for Earth observation tasks. A sensor on-board, such as an optical camera, or SAR (synthetic aperture radar), can be designed and optimized for specific imaging distance. Circular orbits are also used when the velocity variance is desired to be as low as possible. One notable example is the GNSS systems. It is worth pointing out that there are no perfectly circular orbits in practice, just elliptical ones with very low eccentricity.

Elliptical orbit (*pol.* orbita eliptyczna) has eccentricity lower than one ($e \in (0, 1)$). This is by far the most common type for both natural and man-made objects. Elliptical orbits have periapsis (*pol.* peryapsis) (the lowest altitude or closest approach point, where the velocity is highest) and apoapsis (*pol.* apoapsis) (the highest altitude or farthest point, where the velocity is lowest). This type of orbit is the easiest to achieve as there is no need to conduct circularisation burn when inserting an object into orbit with a rocket. The elliptical orbits are also used as intermediate stages between other orbits. For example, a Hohmann transfer (see Section 1.6.3) from circular LEO to GEO has an intermediate step of elliptical orbit with a periapsis at LEO and apoapsis at GEO altitude. See Section 1.4.3 for an explanation of acronyms such as LEO or GEO. The object on elliptical orbit changes its velocity over time. This is sometimes useful. For example, it is possible to design an orbit in such a way that the satellite spends more time over certain regions. A good example could be Molnyia orbits, which Russians used for communication satellites. They were optimised to spend as much time as possible over Russian territory.

Parabolic orbit (*pol.* orbita paraboliczna) has eccentricity of exactly one ($e = 1$) and is a border between closed orbits (circular and elliptical) and open orbits (hyperbolic). The parabolic trajectory is the minimum energy trajectory that allows escaping a system, e.g. the Earth system. When leaving a system, the orbit is called escape orbit (*pol.* orbita ucieczkowa). When arriving a system, it is called capture orbit (*pol.* orbita przechwytyjąca).

This kind of orbit does not exist in practice, as the energy is always slightly larger or slightly smaller than this theoretical boundary. One curious interpretation of the parabolic trajectory is that its total energy is zero, which is sometimes expressed as $C3 = 0$. The interpretation for this is that the object moving away at a distance r from the body is slowing down, but the gravity is getting weaker at the pace of r^2 . The departing object would stop at infinity. Obviously, that is a purely theoretical model that assumes the absence of any other bodies.

Hyperbolic orbit (*pol.* orbita hiperboliczna) has eccentricity of more than one ($e > 1$) and, together with parabolic, is an example of open trajectory. The naming convention of escape and capture orbits can be applied to hyperbolic, too. This type is used when departing a system (such as leaving Earth vicinity) or when visiting a system (such as Jupiter flyby). There are few natural bodies that move along such trajectories, but they are known to exist. Two most famous hyperbolic objects are 'Oumuamua and I2/Borisov. Both are proven to be interstellar visitors to the Solar System.

Radial trajectory (*pol.* trajektoria radialna) is a special case of degenerated orbit. The orbiting body moves directly away from or towards the centre of the orbited body. Such trajectories are seldom used, as they don't have any practical value yet. However, they can be used to model certain phenomena, such as throwing something directly up or explaining why launching a rocket straight up would not achieve orbit, regardless of how powerful the rocket is. An interesting property is that such trajectories always cross the central body's surface. In the future, they may be used to dispose of unwanted materials on bodies such as the Sun.

1.4.3. Orbit classification by altitude

A very common classification segregates orbits depending on their altitude, especially for Earth orbits. This is useful nomenclature for orbits that are circular or slightly elliptical, i.e. their periapsis and apoapsis are not radically different. The LEO or **Low Earth Orbit** (*pol.* niska orbita wokółziemiska) is typically defined as an orbit with an altitude of 2000 km or lower. While the Kármán line (*pol.* linia Kármána) set at 100 km is commonly considered a boundary of space, the actual atmosphere extends far beyond this line. In general, orbits with periapsis lower than 185 km are considered unstable. The Earth atmosphere is discussed in detail in Section 4.6.4. As such, a practical approach is to define LEO as orbits of periapsis as low as 185 km and apoapsis no greater than 2000 km. A great majority of satellites are in LEO orbits. This is particularly popular orbit for Earth Observation satellites. Since LEO is the easiest orbit to achieve, most missions with low energy budgets use this type of orbits. The orbital period is typically many times per day. For example, the International Space Station (ISS) is using 408 x 410km orbit and has an orbital period of 92 minutes. This translates to roughly 15,5 orbits per day.

The next category is MEO or **Medium Earth orbit** (*pol.* orbita!średnia okółziemiska). It is typically defined as an orbit with an altitude over 2000 km and less than 35 786 km. This orbit is frequently used by GNSS systems. In particular, orbits of altitude of 20200 km are popular, as the orbital period for such an altitude is 12 hours. This is useful for getting a repeating daily pattern.

There is one distinct altitude of 35 786 km that defines a separate class of orbits. For this altitude the orbital period matches a sidereal day, which is 23h 56m 04s. Any orbit that has this altitude will appear as not moving eastwards or westwards as observed from the Earth surface. Such class of orbits is called **geosynchronous orbit** (*pol.* orbita geosynchroniczna) and are often abbreviated as GSO (*pol.* GSO). GSO with a non-zero inclination will appear as moving roughly around the north-south direction on figure 8 shape. A special case of GSO with zero inclination is called **geostationary orbit** (*pol.* orbita geostacjonarna), often designated as GEO. It has a major advantage to rotate at the same pace as Earth and not move in the north-south direction. As such, GEO satellites appear stationary as observed from Earth. This brings a major advantage of being able to use fixed (non-tracking) directional antennas. Once set up, they can maintain good reception for a long time, assuming the orbital drift is negligible. This orbit is very popular with communication satellites, in particular satellite TV, radio and other communication. There are so many satellites in GEO orbit that they form a ring around Earth. Since the plane

and altitude is pre-determined for GEO satellites, there is a simplified notation that describes the object's location. It uses a longitude of the sub-satellite point. For example, a Hot Bird 10 satellite is located at 33° East.

Due to orbital drift, satellites in GEO need to conduct small correction maneuvers to remain in GEO orbit. The amount of remaining fuel is the typical limiting factor for the operational lifetime of GEO satellites. Once a satellite approaches the end of its lifetime, it is moved away from GEO not to clutter this precious space.

Finally, orbit above GEO is considered HEO or **High Earth Orbit** (*pol.* wysoka orbita okołoziemska). Such orbits are used infrequently, as the energy requirements to reach them from Earth are even larger than for GEO. The HEO orbits are used for missions that require considerable distance from Earth. There are few satellites in HEO orbits. Most of them are deep space observatories, especially those affected by Earth's magnetosphere.

The preceding paragraphs described typical orbits around Earth. Similar classification can be used for other bodies. For example, NASA uses LLO or Low Lunar Orbit (*pol.* niska orbita wokółksiężycowa) to designate orbits around the Moon with an altitude lower than 100km. Such low orbits on the Moon are possible because it has almost no atmosphere and there's negligible friction while flying at that altitude.

1.4.4. Orbit classification by inclination

Orbits are often classified or characterised by other parameters. One of the essential ones is inclination. An orbit that is on the equatorial plane has zero inclination and is called **equatorial orbit** (*pol.* orbita równikowa). Geostationary orbit is an equatorial orbit with an altitude of 35 786 km. Orbits that have a high inclination, i.e. close to 90° are passing at or near the poles. Such an orbit is called **polar orbit** (*pol.* orbita polarna). This type of orbits is particularly well suited for Earth Observation, as the satellite passes over almost all area. In general, the satellite can pass over a land area up to the latitude of its inclination. For example, for a satellite to cover whole Poland area, it needs to be placed on an orbit with an inclination of at least 54°50' (the longitude of the northernmost point in Poland – Cape Rozewie near Jastrzębia Góra).

Objects that circumvent Earth the same direction as its rotation have inclination between 0° and 90° are said to be on **prograde orbit** (*pol.* orbita prograde). Objects that circumvent in the opposite direction as Earth's rotation have inclination between 90° and 180° are said to be in the **retrograde orbit** (*pol.* orbita wsteczna). The prograde/retrograde naming convention is also convenient to describe orbital maneuvers. The prograde direction means in the direction of the movement ("accelerating forward"), while retrograde means the opposite direction ("breaking").

The great majority of satellites are using prograde orbits as launching eastwards decreases the Δv requirements due to the additive effect of Earth's rotation. The effect is most potent on the equator and non-existent on poles. Also, objects moving in prograde direction appear to move slower when observed from the surface, so most activities, such as imaging or communication, is more manageable. However, if the mission has special requirements, a retrograde orbit provides some unique benefits.

Real orbits drift over time due to small external factors. This is discussed in Section 1.8 in more detail. For now, it is worth pointing out that there are two special values of inclinations for Earth orbits. The first one is 98°, which causes the orbital plane to rotate slowly at the rate of exactly one revolution per year. This is useful for Sun-synchronous orbits, which is discussed in the following section. Another special value of inclination is 63.4°, which causes the orbit not to drift at all. This value is called **critical inclination** (*pol.* inklinacja krytyczna) and is used by Molniya and Tundra orbits, also to be discussed in the next section.

1.4.5. Special purpose orbits

There are several uncommon or special purpose orbits. While they all fall into the above classifications, they have specific properties that allow the satellites to achieve their mission goals better. They're listed roughly in the order of popularity.

Many Earth observation missions can benefit from observing the same area at the same local solar time, e.g. an air quality mission would be interested in obtaining measurements that are consistently on the same time of the day, so the impact of external patterns, such as industry operation throughout the day or people heating their homes with stoves in the evenings, is minimised. To achieve this goal, it is necessary to take advantage of perturbations. This phenomenon is described in more detail in Section 1.8. Briefly, certain external factors, such as Earth not being perfectly round or solar radiation pressure, cause the spacecraft's orbit to slightly change over time. This impact is small but measurable. To achieve the goal of fly-over at the constant time of day, the orbit has to precede slowly. The ascending node is expected to conduct a full circle over the year, at exactly the same rate as Sun's apparent movement over the sky. Such an orbit is called **Sun-synchronous** (*pol.* orbita heliosynchroniczna) and is often abbreviated as SSO.

An SSO can be chosen to always fly in the dawn/dusk. The Sun is almost always visible, which is of particular importance for Sun observation missions. Interestingly enough, the SSO requires perturbations strong enough to provide rotating momentum equal to one full revolution per year. The major part of such perturbations in the case of Earth comes from its oblateness. It is possible to construct Sun-synchronous orbits around Mars, because it is also oblate, but not Venus, which is almost perfectly spherical [98].

Another type of special orbits are **Molniya** (*pol.* Molnia) and **Tundra orbits**. They take the name from Russian series of communication satellites. Both orbits are trying to solve the problem of radio communication in high latitude areas, such as northern parts of Russia. The orbits take advantage of several factors. First, they use critical inclination ($i = 63.4^\circ$), so they have a zero-drift over time (i.e. the orbit does not shift eastwards or westwards). The next aspect here is that they are highly eccentric ($e \approx 0.74$), which results in a significant difference in periapsis (6650 km) and apoapsis (46550 km). The velocity is also radically different. As a result, the spacecraft spends a disproportionately large part of its orbital period near its apoapsis. Together with the critical altitude, the apoapsis always falls over the same chosen area of the Earth. As a result, each of Molniya satellites had 8 operational hours out of its 12h orbital period. A constellation of only three satellites in principle provided permanent coverage over the desired area of northern Russia.

The Tundra orbit uses a very similar approach but has a period of a full sidereal day (23h 56m 4s), instead of half sidereal day as Molniya does.

1.4.6. Lagrangian points and exotic orbits

All of the orbits discussed so far were solutions of a 2-body problem, i.e. they assumed a major body, such as Earth, is being orbited by a much smaller object, such as a spacecraft and any external influences, such as Sun's or Moon's gravity have only minor impact and is modelled by perturbations. This is a good approach in case of considering the relative proximity of the major body. This is often conveniently modelled as a **sphere of influence** (SOI) (*pol.* sfera wplywu). However, when the distance is large enough, or there is another heavy body in proximity, the classical 2-body problem is insufficient, and the 3-body problem has to be considered. The two most common examples are the Earth-Sun and Earth-Moon systems. One particular aspect of the 3-body system is that there are points where the gravitational forces of the two major bodies are equal. Depending on the configuration, the forces can reinforce each other or mostly cancel one another. Those points are called **lagrangian points** (*pol.* punkt Lagrange'a) or **libration points** (*pol.* punkt libracyjny). Calculating libration points is out of scope for this thesis, as they involve dealing with complex differential equations with no analytical solutions and require numerical methods. The reader interested in the details is encouraged to read Chapter 2.12 of [15]. An overview of the libration points of the Earth-Moon system is presented in Fig. 10.

A **halo orbit** (*pol.* orbita halo) is an orbit around L_1 , L_2 or L_3 Lagrange points. Although there is no mass there, the attracting forces of two other large bodies in the vicinity make this point can be orbited. The halo orbits are usually unstable and require stationkeeping, although the magnitude of corrective maneuvers tend to be

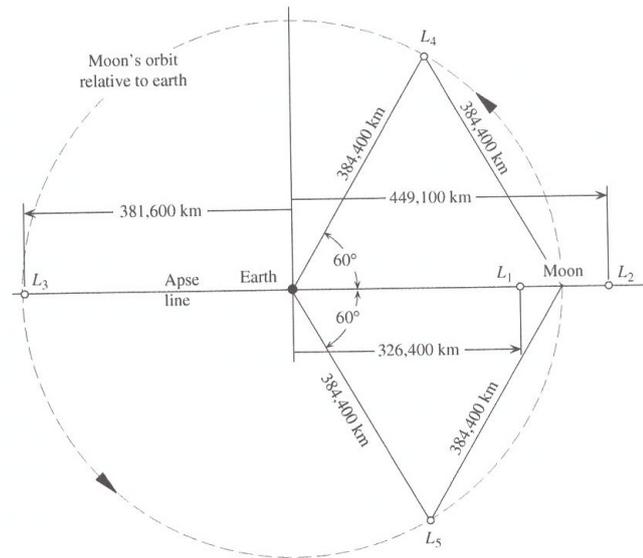


Figure 10: Any given 3-body system, such as Earth-Moon depicted here, has libration points where gravitational forces of two major bodies are equal. A small body, such as spacecraft, positioned in any lagrangian point would seem to orbit the body the observer is located at. Source: [15]

low. The concept of halo orbit was first proposed in 1968 by Farquhar in his PhD thesis [18]. There are very few spacecraft that used such an orbit. Herschel Space Observatory by ESA uses 800 000 km average halo orbit around L_2 of the Sun-Earth system. The spacecraft on average remains at a distance of 1.5 million km from Earth.

A **Lissajous orbit** (*pol.* orbita Lissajous) is a quasi-periodic orbital trajectory around Lagrangian points. Its major flaw is that it is not periodic, meaning that each evolution is slightly different from the previous one. However, in return, the major advantage to halo orbit is that it does not require stationkeeping maneuvers, and thus is much cheaper for maintaining in the long term. In most applications, the L_4 and L_5 points are considered stable. [90] claims the orbits around L_4 and L_5 points can last few millions of years. Ignoring perturbations by other planets, they can be stable for billions of years. An example of a Lissajous trajectory is shown in Fig. 11. There are several spacecraft that use this kind of orbit: ACE, SOHO, DSCOVR, WMAP, Genesis, Herschel and Planck observatories, Gaia, two THEMIS spacecraft and Queqiao.

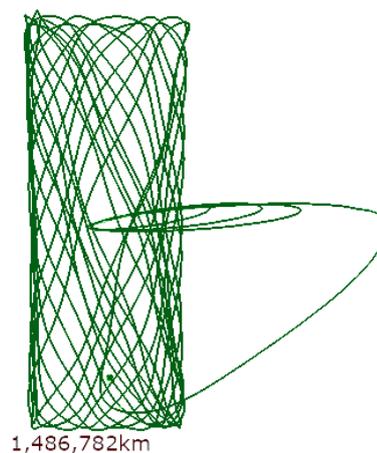


Figure 11: Lissajous orbit is quasi-stable orbit around Lagrangian points. An example of Wilkinson Microwave Anisotropy Probe orbiting around L_2 libration point of the Earth-Moon system. Source: [90]

A **Near-rectilinear Halo Orbit** (*pol.* prawie prostoliniowa orbita halo) is a class of halo orbits. It has recently gained popularity after NASA announced [45] in 2020 that NRHO orbit will be used for the Lunar Gateway,

a new manned space station to be built in the general vicinity of the Moon. The Gateway is expected to be positioned around L_1 point of the Earth-Moon system. After this announcement, the orbit became of high interest to many. Advanced Space is working on CAPSTONE mission, a 12U CubeSat to be launched to NRHO orbit. Three companies (Blue Origin, Dynetics and SpaceX) were picked by NASA to provide launch services and lunar lander architecture. Many variants being discussed involve rendezvous with the Gateway at NRHO orbit. The NRHO concept is studied in details by Zimovan in [100].

There are several attributes of NRHO that makes it attractive. It is easier to get to from Earth, compared to LLO. This makes the general supply chain from Earth much more sustainable. NRHO passes over the Moon poles, which are the most attractive prospective sites for first lunar bases. Due to the absence of the atmosphere, any ice deposited on the Moon slowly evaporates when exposed to direct sunlight. However, there are craters near the poles, that remain in permanent shadow, which prevents the evaporation, and current results strongly suggest they have rich deposits of water ice. On the other hand, the top of the crater rims is in permanent sunlight, which offers tremendous benefits. It is worth pointing out that the Moon is tidally locked to Earth, which means that one Lunar day is 27 Earth days and 7 hours and the average lunar night lasts over 13 days. The difficulty of surviving the lunar night by a prospective base is substantial.

There are several other advantages of NRHO. The orbital plane is perpendicular to the Earth-Moon line, which means that objects on NRHO are never in the Moon shadow and can always maintain radio communication with Earth. Finally, the orbit has low escape velocity, which may become very useful for future Mars missions. The human capsule departing for Mars could be assembled at NRHO and be provisioned from the Moon base, in particular in the context of providing water and its products (oxygen and hydrogen).

The NRHO selected by NASA has a period of 7 days, periapsis of 3000 km and an apoapsis of 70000 km. A visualisation of the NRHO orbits is presented in Fig. 12. The reader interested in learning more about NRHO is encouraged to read [100].

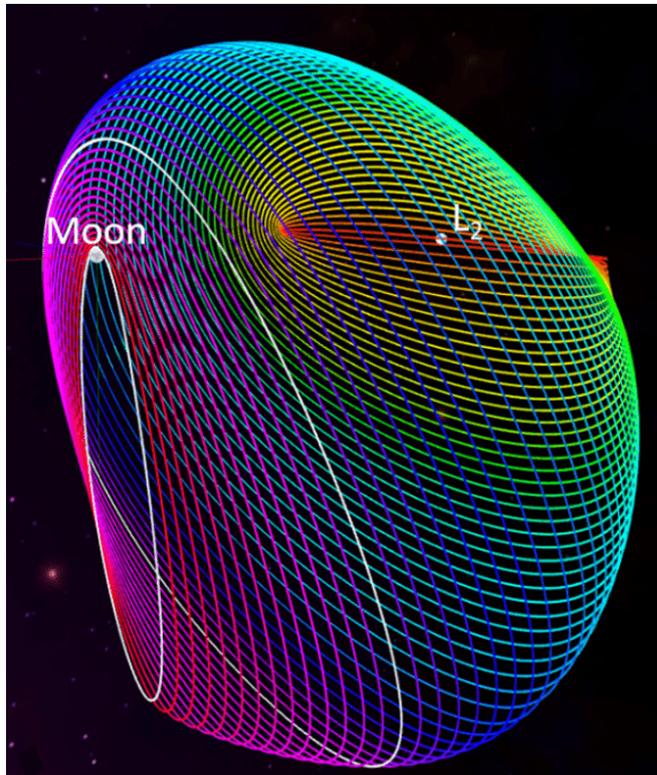


Figure 12: A family of Near Rectilinear Halo orbits around the Moon-Earth system. Source: [100]

1.4.7. Energy perspective

An alternative way to describe an orbit is to define the object's kinetic energy. As [5] shows, for a given orbit, the energy is dependent only on the major semi-axis:

$$E = -\frac{\mu}{2a} \quad (1.5)$$

where E is the total energy of an object, a is major semi-axis. See Section 1.6 for an explanation of μ . It is worth noting the zero point in this notation. The mechanical energy of a satellite moving in a closed (circular or elliptical) orbit is negative. It is zero for parabolic orbits and positive for hyperbolic orbits.

The energy perspective is discussed at length in [5], [6] and [15].

1.4.8. Escape velocity

Even though a gravitational field of a given body, such as Earth or Sun, extends to infinity, the strength of it decreases rapidly and thus only a finite amount of energy is required to escape it. The velocity needed to escape from a circular orbit of radius r is defined by the following equation:

$$V_{esc} = \sqrt{\frac{2\mu}{r}} \quad (1.6)$$

Based on this, we can formulate an interesting observation that for any given circular orbit, the Δv necessary to escape is $v_{esc} = \sqrt{2} \cdot V_{cir}$, or roughly 1.414 more than the current orbital speed. This is intuitive. The higher an orbit is, the slower its actual velocity is. Therefore, the farther away from the central body it is, the easier it becomes to escape.

1.5. Spacetime Reference Frames

A stable reference is needed to describe a location in three-dimensional space. Depending on the specific purpose, many reference frames allow locating objects in space. Furthermore, astrodynamics rarely deals with static objects. Objects in motion also require time reference. Given the rotating Earth, orbital motion around Sun and other movement types, the time reference is tightly coupled with the spatial reference. As such, it will be described first.

1.5.1. Reference Time

To understand the concept of some reference frames, an idea of epoch and its dependency on time needs to be introduced. The astronomical community faces the problem of using the right ascension and declination (sky equivalent of longitude and latitude) to locate objects such as asteroids, planets, stars and galaxies. However, due to various phenomena, such as the proper movement of stars, Earth precession, Sun movement around the galaxy center and others, even the most static objects as seen from Earth – stars and galaxies – drift slowly. In most cases, the drift is negligible. However, certain applications must consider it. One practical case is a polar alignment of telescope mounts. It is of paramount importance to align the axis of rotation in such a way to point exactly at the Celestial North or South Pole. This is done using Polaris (*pol. gwiazda Polarna*). This star drifts slowly due to Earth's precession. Modern telescope mounts have special polar alignment scopes that have marked the position for the year 2000 and then corrections for coming years. The astronomical community decided to select a certain moment in time and use it as a reference. This reference was chosen to be 12:00 terrestrial time (TT) (*pol. czas ziemski*) on 1 January 2000. This particular instance in time is called **J2000** epoch. Many astronomical coordinates are said to be J2000, which means “as observed on 12:00 Terrestrial Time of 1 January 2000”.

1.5.2. Coordinate Systems

Various reference systems have been developed. Depending on the intended purpose, some are better suited for specific tasks than others. For example, for calculations that involve locations on the ground, it is usually more comfortable to use a reference system that is fixed to Earth and thus does not require considering Earth rotation and other Earth movements. On the other hand, when dealing with orbital trajectories around Earth, it is more convenient to use an inertial system that is fixed in space. The satellites and other spacecraft orbit around Earth barycenter, which is constant with regards to its rotational motion.

The following list is a small subset of all reference systems defined.

ECEF – Earth Centered Earth fixed is a cartesian coordinate system with its origin placed in the centre of mass of Earth, x and y axes are located in the equatorial plane with x pointing toward prime meridian and z axis towards the North Pole. Contrary to its name, the system rotates over time together with Earth. As such, it belongs to the non-inertial category. This system is handy for calculating points and describe the motion of objects on Earth surface of its close proximity (ships, planes, rockets in the lower atmosphere, etc.).

ECI – Earth Centered Inertial is a class of coordinate systems that has its origin in the centre of mass of Earth. It is fixed in space with respect to the stars and does not rotate with Earth. This family is convenient for describing points and motion of objects in Earth vicinity, such as Moon, satellites etc. The x and y lie in the equatorial plane, while the z axis is perpendicular and crosses the North Pole. One significant problem in the ECI family is the selection of a designated x axis. At least three ECI systems currently in frequent use vary with different definitions of the x axis: J2000, GCRF, and TEME.

J2000 – This is one of ECI coordinate systems that use Earth's mean equator and mean equinox as observed on J2000 epoch. As such, the coordinate shares the same name as the epoch designation. The x axis is aligned with the mean equinox as observed on J2000 epoch. This system is sometimes referred to as EME2000. It replaced an earlier system called M50, which used a similar approach, but an earlier date of 1 January 1950.

GRCF – Geocentric Celestial Reference Frame is another Earth-centered system that uses International Celestial Reference Frame (ICRF). The general relativity implies that there are no truly inertial frames in the vicinity of any bodies with non-zero mass. ICRF selected several extra-galactic bodies, such as quasars. Those are extremely distant objects, and due to their distance, they do not exhibit any measurable motion. As such, they are good, stable reference frames.

TEME – True Equator Mean Equinox is another coordinate system. This one is of particular importance in the field of space missions. The importance has a historical origin. Since the early 1960s, NORAD and other US agencies used TLE (Two Line Elements) format to specify satellite orbits and published SGP models to calculate orbital positions. The TLE format and SGP models use TEME reference system. As such, all of the space industry is using TEME everywhere.

ENU – This is a local tangent plane coordinate. It is a coordinate system with its origin located at the location of interest, e.g. a spacecraft. The x axis points East on the Earth, y points North and z points Up or away from the Earth centre. This reference frame moves along with an object, such as spacecraft. It is particularly useful for describing maneuvers or relations in the spacecraft's reference, such as describing orbital maneuvers or rendezvous operations when approaching other spacecraft.

ICRF and **HCRS** – International Celestial Reference Frame and Heliocentric Reference System are two very similar systems. The first was defined by IERS (International Earth Rotation and Reference Systems Service). IERS published an extraordinarily complex document [27] that define on 179 pages many systems and frames (ICRF, ICRS, ITRF, ITRS, BCRS, CIRS, GTRS) and relations between them. The ICRF and HCRS systems have its origin located at the Sun barycenter, with the axes aligned with those defined by ICRS. The HCRS is marginally simpler than ICRS because it uses barycenter of the Sun itself, rather than barycenter of the whole solar system. As such, it does not include the effects of aberration, which ICRS does. However, from the Earth

perspective, the difference is cited to be only 8 milli-arcseconds. Those two systems are useful for describing points and events happening in the Solar system, outside of Earth sphere of influence, such as interplanetary probes, planets and asteroid locations, gravity assists and similar.

The reader interested in studying those topics is encouraged to read [32] and [33] and then attempt to read [27].

1.6. Orbital maneuvers

Orbital maneuver is a velocity vector change of a spacecraft or other object in space, typically – but not always – performed using engines. Maneuvers can be achieved using engines – propulsive maneuver (*pol.* manewr napędowy), atmospheric drag – aerobreaking (*pol.* aerobreaking), gravity of massive bodies, such as planets – gravity assist (*pol.* asysta grawitacyjna) or other phenomena. A maneuver conducted using propulsive engines are often briefly called **burn**. There is no popular Polish short alternative to *maneuver*.

A maneuver can be considered as a velocity vector change. One handy abstraction is to operate on the current velocity vector and express the target orbits in terms of the desired velocity vector. The difference between current and desired vector is a Δv of the maneuver.

The orbital velocity is defined using the following formula, better known as vis viva equation (*pol.* równanie vis-viva):

$$v^2 = GM \left(\frac{2}{r} - \frac{1}{a} \right) \quad (1.7)$$

where v is velocity, G is universal gravitational constant, M is mass of the object being orbited (e.g. Earth), r is the current distance between centers of the orbiting bodies and a is an average of semi-major and semi-minor axes of the orbit. Since G and M are constant and frequently occur in many equations, they are conveniently replaced by a single constant:

$$GM = \mu \quad (1.8)$$

Furthermore, for circular orbits, $r = a$, so the equation simplifies to:

$$v = \sqrt{\mu \cdot \frac{1}{a}} \quad (1.9)$$

1.6.1. Rocket Equation

The **rocket equation** (*pol.* równanie rakietowe), sometimes referred to as Tsiolkovsky's equation (*pol.* wzór Ciołkowskiego) is a fundamental equation that defines the basic operation of a rocket that expels fuel, e.g. conducts propulsive maneuver. The equation was formulated by Konstantin Tsiolkovsky and published in 1903 in [13]. It defines the final velocity of a rocket that expelled mass (usually in the form of exhaust gases or plasma) in the opposite direction.

$$\Delta v = I_{sp} \cdot g_0 \ln \frac{m_0}{m_f} \quad (1.10)$$

where Δv (**delta-v**) is the change of velocity, I_{sp} is the specific impulse, expressed in seconds, g_0 is the standard gravity, m_0 is the initial mass and m_f is the final mass. The I_{sp} and g_0 are sometimes substituted with a single variable:

$$v_e = I_{sp} \cdot g_0 \quad (1.11)$$

where v_e is the effective exhaust velocity.

1.6.2. In plane Maneuvers

The **prograde burn** (*pol. manewr prograde*) is a burn "forward" (acceleration), i.e. in the same direction as the velocity vector. This type of burn raises the anti-point, i.e. point located at 180° apart on the orbit. This maneuver does not change the orbital plane. Typically conducted at periapsis (raises apoapsis) or apoapsis (raises periapsis). When conducted at other places, it will generally rotate the orbit on the orbital plane, changing the argument of periapsis (ω) and moving periapsis or apoapsis points and their absolute altitudes. This maneuver is typically used to raise orbits.

The **retrograde burn** (*pol. manewr wsteczny*) is a burn in the opposite direction as the velocity vector. It is the opposite of a prograde burn and is sometimes described as burning backward (breaking) or slowing down. This maneuver does not change the orbital plane. This type of burn lowers the anti-point. It can be used to lower the opposite point of the orbit. Typically conducted at periapsis (lowers apoapsis) or apoapsis (lowers periapsis). When conducted at other locations along the trajectory, it will generally rotate the orbit on the orbital plane, changing the argument of periapsis (ω) and moving periapsis or apoapsis points and their absolute altitudes. This maneuver can be used as deorbit burn (*pol. manewr deorbitacyjny*).

The **chase maneuver** (*pol. manewr pościgowy*) is a maneuver conducted when two spacecraft are sharing the same orbit, and the one that is behind is interested in catching up, usually for the purpose of docking. Depending on the capabilities, either the leading spacecraft can slow down (by temporarily moving to a higher orbit, which has slower velocity), or the chasing spacecraft can speed up (by temporarily moving to a lower orbit, which has faster velocity). A series of catch up maneuvers are part of a larger rendezvous and proximity operations campaign, to be discussed in the later sections. See [4], [8], [83] for extensive discussions regarding rendezvous problem.

The **Hohmann transfer** and **Bi-elliptic transfer** are also in-plane maneuvers, but due to their importance, they are described in dedicated sections.

1.6.3. Hohmann Transfer

The Hohmann transfer is a set of two maneuvers that take a spacecraft from one circular orbit to another coplanar circular orbit. The Hohmann transfer in the general case is the transfer that takes the least possible amount of propellant (Δv), although bi-elliptic transfer may be more efficient in some cases. Hohmann transfer is by far the most frequently used maneuver due to its efficiency and a broad application to raising or lowering apses, catching up or slowing down to get in proximity with various objects, interplanetary transfers, and more.

The Hohmann transfer is described in detail in every book related to astrodynamics, including [6], [15], [70], [83] and others.

For example, we can calculate how much Δv is required to move a spacecraft from a circular LEO orbit with an altitude of 300km to GEO. The following assumes that the inclination of the starting orbit is zero (i.e. the orbit is on the equatorial plane). First, we need to calculate the orbital velocity for circular LEO orbit:

$$v_{LEO} = \sqrt{\mu \cdot \left(\frac{1}{R_e + r} \right)} \approx 7723.29[m/s] \quad (1.12)$$

where R_e is Earth radius and r is an altitude. Since we want the destination to be circular, we can calculate the velocity of the destination orbit:

$$v_{GEO} = \sqrt{\mu \cdot \left(\frac{1}{R_e + 35786} \right)} \approx 3073.68[m/s] \quad (1.13)$$

In most cases, the orbital change is done using Hohmann transfers. For cases when better alternatives are available, see Section 1.6.4 about bi-elliptic transfers.

Hohmann transfer from circular to circular orbit is particularly easy, as Hohmann transfers are initiated during periapsis or apoapsis. However, if the departing orbit is circular, any orbital position can be considered both apoapsis and periapsis. The first burn will turn the orbit into highly elliptical with a periapsis of 300km and apoapsis of 35786km , often called GTO (Geostationary Transfer Orbit). Since this orbit is elliptical, its velocity changes depending on the spacecraft's position. For periapsis and apoapsis, it is respectively:

$$v_{p(LEO \Rightarrow GSO)} = \sqrt{\mu \left(\frac{2}{r_p} - \frac{2}{r_p + r_a} \right)} \approx 10148[m/s] \quad (1.14)$$

$$v_{a(LEO \Rightarrow GSO)} = \sqrt{\mu \left(\frac{2}{r_a} - \frac{2}{r_p + r_a} \right)} \approx 1607[m/s] \quad (1.15)$$

To relocate a spacecraft from LEO to GSO, the Δv required is $v_{p(LEO \Rightarrow GSO)} - v_{LEO}$, which is roughly 2424m/s . Once the spacecraft reaches its apogee, it should perform a circularization maneuver. It will have a velocity of 1607m/s and needs to increase it to 3073m/s , thus expending another 1466m/s . By summing all maneuvers together, we get the total Δv expenditure of the LEO (300km) to GEO transfer to be 3891m/s . This situation has been presented in Fig. 13.

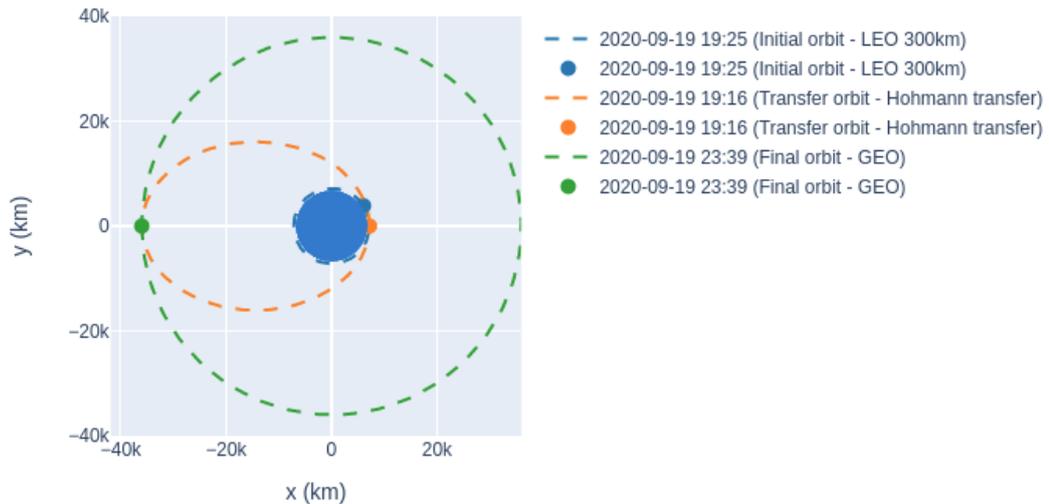


Figure 13: This diagram shows steps necessary to move from LEO to geostationary orbit. Note that although both the original and destination orbits are circular, the intermediate Hohmann transfer orbit is elliptical. Generated using Perylune software.

The discussion above makes a strong assumption that both departure and destination orbits are coplanar. Both Hohmann and bi-elliptic transfers require coplanar departure and target orbits.

1.6.4. Bi-elliptic Transfer

The bi-elliptic transfer (*pol. manewr dwueliptyczny*) is an orbital maneuver that conducts three burns, rather than two as in Hohmann transfer. The first burn moves the spacecraft to a trajectory with very high apoapsis. Once that point is reached, the spacecraft has comparatively low velocity. As such, changing the velocity substantially is relatively cheap. The second burn puts the spacecraft into a Hohmann trajectory towards the final orbit. The third burn is conducted once the final periapsis is reached to circularize the orbit. The bi-elliptic transfer is more efficient than Hohmann if the ratio of semi-major axes of departing and target orbits is equal or greater than 11.94. This transfer generally takes much longer to complete than Hohmann. The comparison of relative efficiency in function of this ratio is presented in Fig. 14.

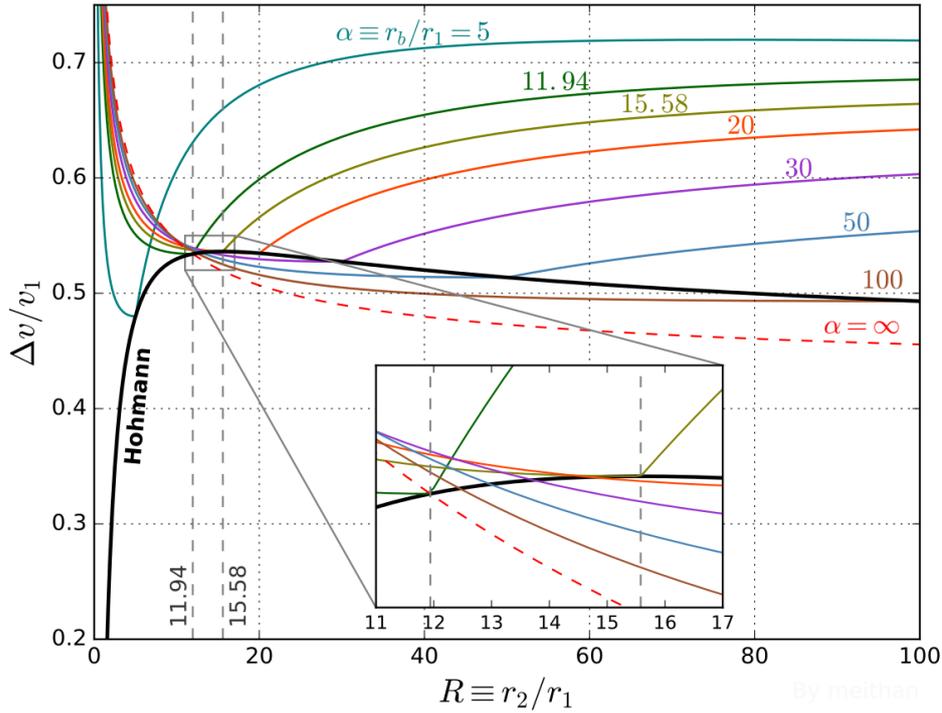


Figure 14: This diagram presents the Δv required for Hohmann (black line) and various bi-elliptic transfers (colored curves) between two circular orbit in function of their semi major axes. Source: [95].

1.6.5. Rendezvous

The rendezvous (*pol.* rendez-vous) is a set of orbital maneuvers where two spacecraft physically touch or get in very close proximity with minimal relative velocity. A rendezvous requires matching orbital velocities and orbital vectors. There are many practical applications for this maneuver: docking or berthing a spaceship to a station, attaching a new satellite to an old one, assembling larger structures in space, and many more. Also, for some bodies with a small mass, such as Mars' moons Phobos and Deimos, the maneuver of landing on their surface looks more like rendezvous, rather than typical entry, descent and landing maneuvers.

The rendezvous is counter-intuitive and requires a good understanding of astrodynamics. For example, imagine a situation depicted in Fig. 15. There is a space station and a ship attempts to dock to it. Let assume that both ship and the station are on a circular orbit of the same radius r . What maneuvers should the ship conduct to dock? The intuitive answer – to fire its thrusters directly towards the station – is incorrect. By firing its thrusters in the prograde direction, the ship will extend the opposite point of the orbit, thus effectively moving to an elliptic orbit with periapsis equal to r , but its apoapsis being $r + \Delta h$. As the ship and station travel along their orbits, the ship would slowly drift away to a higher altitude, and after half an orbit it would reach a maximum distance of Δh from its previous orbital path. After completing the full orbit, it would return to its original altitude and distance to the station.

This is the mistake that US astronaut Jim McDivitt made when attempting to perform the first rendezvous maneuver in history. He tried to maneuver his Gemini 4 capsule to match the Titan II's upper stage. Despite multiple attempts, the maneuver was unsuccessful [97].

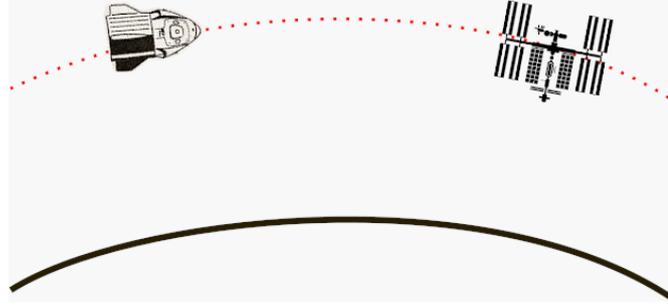


Figure 15: One object, such as a capsule, approaching another object, such as a space station, needs to perform rendezvous maneuver to dock or berth.

The proper way to approach the target is to move the approaching ship to a lower orbit first. Lower orbit has a higher orbital velocity, so the chasing ship will "catch up" with the target. Once it is close, it should then raise its own orbit to that of the target's. This is counter-intuitive as the first maneuver (move to lower orbit) requires firing engines in the retrograde direction, i.e. away from the target. The various phases of rendezvous are well described in [83], [84], and to some lesser degree in [70].

1.6.6. Plane Change Maneuvers

Many orbital maneuvers, most notably Hohmann and bi-elliptic transfers, require the initial and final orbits to be coplanar, i.e. both be on the same plane. This requires both orbits to have the same inclination and the same RAAN. To achieve such an orbit, a class of maneuvers is needed that involves acceleration in a direction that's not on the orbital plane. Such maneuvers are called **out of plane maneuvers** or **plane change maneuvers** (*pol.* manewr zmiany płaszczyzny orbity). There are several notable basic maneuvers in this class: inclination change maneuver, which alters the inclination, and plane rotation maneuver, which changes the RAAN.

1.6.6.1. Inclination Change

The goal of the inclination change maneuver (*pol.* manewr zmiany inklinacji) is to change the inclination of the orbit while leaving all other parameters intact. The general equation for Δv cost in any orbital maneuver is a direct application of law of cosines (*pol.* twierdzenie cosinusów), better known as cosine rule. It also applies to the inclination change.

$$\Delta V^2 = V_1^2 + V_2^2 - 2V_1V_2 \cdot \cos(i) \quad (1.16)$$

The inclination change, as with any other out of plane maneuvers, is costly. To better understand its cost, let assume we want to do a pure inclination change from one circular orbit to another circular orbit with the same radius. In this case $V_1 = V_2 = V$ and the equation can be simplified to:

$$\Delta V = V\sqrt{2 \cdot (1 - \cos(i))} \quad (1.17)$$

For example, let us assume a scenario of launching a satellite from Leba Proving Grounds aiming for an equatorial LEO orbit of 300km. Leba's latitude is 54°45'N. An orbital velocity of circular 300km LEO orbit is 7723.29 m/s. As such, the inclination change would require 7102 m/s, which is an absurdly high value. As shown in the earlier section, it is far easier to reach GEO orbit. That is why launch locations located at high latitudes rarely launch to equatorial orbits and instead opt to specialize in polar orbits.

Let us hypothetically assume that Poland would build a launch complex at its southernmost point, which is near Opolonek peak in Bieszczady mountains. Its latitude is 49°00'N. The Δv required to achieve equatorial orbit from that location is 6405 m/s. It is still very unfavorable, but almost 700 m/s less than from Leba. In any case, Poland is poorly situated for equatorial launches. As such, we as a nation have two possible development paths.

First, we can focus on missions that use near-polar orbits as polar launches are easier to achieve from high latitudes. Secondly, for missions that require near-equatorial orbits (GEO and beyond Earth orbit), we cannot depend on domestic capabilities and must rely on third party launch sites.

The inclination change should be done on the intersection of departing and target orbital planes. For most cases, this is on the line of nodes, i.e. when the spacecraft passes either ascending or descending node. This implies that to perform the inclination change, the RAAN must be synchronized between departing and target orbits first.

Practical implementation aspects of the inclination change maneuvers are discussed in Section 3.5.

1.6.6.2. Plane Rotation

Another out-of-plane maneuver is orbital plane rotation (*pol.* rotacja płaszczyzny orbity), sometimes also called apse line rotation (*pol.* rotacja linii apsyd). It assumes that the plane rotates around the orbited body's (e.g. Earth) North-South axis. Such a maneuver is sometimes required to synchronize ascending and descending nodes. However, this maneuver is somewhat uncommon for several reasons. First, like any other out-of-plane maneuvers, it is costly, so best avoided if possible. Secondly, the maneuver can be skipped altogether by carefully choosing the launch moment. The intention to avoid plane rotation maneuver is why many missions involving existing spacecraft (such as docking with ISS) have instantaneous launch window, i.e. they need to launch at a specific time of day. If there are any delays, launch operations declare scrub and need to reschedule for another time, when the RAAN of the launched payload coincides with the RAAN of the intended orbit.

The third reason is a bit more complicated. It is possible to take advantage of natural perturbations, particularly J₂, to rotate the orbital plane slowly. That is a natural process that alters orbital parameters over time. In particular, the Earth oblateness and the gravitational field's unevenness causes the orbits to drift slowly. The RAAN increases slowly, and both apoapsis and periapsis degrade very slowly. That is a complex process that takes into consideration many variables. However, there are good models available that allow simulating this drift, thus taking advantage of it. For implementation details, see Section 3.5. For general discussion about perturbations, see Section 1.8.

1.7. Interplanetary trajectories

Interplanetary trajectories is a rich, complex, and well-studied area. An excellent introductions are available in [15], [6] and [84]. The following very brief description provides only a high-level overview of the problems involved.

During an interplanetary mission, the spacecraft needs to exceed its departing planet's escape velocity, e.g. Earth. In a hypothetical case of a spacecraft reaching precisely the escape velocity, it is no longer bound by the departing body's gravity. It can be considered moving on a heliocentric orbit that almost exactly matches that of the departing body. In the heliocentric frame of reference, the spacecraft then needs to perform a Hohmann transfer to reach the target body's orbit. In principle, this is an ordinary Hohmann transfer, however in practice it is more complicated for two significant reasons. First, the planets do not have the same inclinations, so the escape burn, first Hohmann burn, and the inclination change burn are often combined into one maneuver. Secondly, reaching the target orbit on its own is insufficient. The target body and the spacecraft need to be in almost the same place in orbit, e.g. have very similar anomalies. This topic is discussed in detail in Section 4.4.

The combined escape, first Hohmann transfer and inclination burns are together called **injection burn**. There is no Polish equivalent. The name comes from the Apollo missions, where Trans-Lunar Injection burns were critical milestones of each mission. The Apollo's naming convention is used to this day, with the destination being used in the name. For example, for Mars there is TMI (Trans-Mars Injection), and missions for Venus need to conduct TVI (Trans-Venus Injection).

1.8. Orbital Perturbations

Typical ion thrusters burns are expressed in hours or days, and sometimes months or even years. This has an interesting implication. So far, all the maneuvers discussed were approximated as an instantaneous change in velocity. In reality, the change is never instantaneous, as it would require infinite force. However, for burns that take seconds or minutes, this is often an approximation that is good enough. Sadly, this is not the case for low thrust propulsion. It requires models and processing that account for gradual change over longer periods. Typically, this problem is solved with various dedicated integration methods that perform orbital propagation but consider small additional forces. Those external forces are called perturbers (*pol.* *perturbacje*). They can model many external influences, such as the gravity of a third body, e.g. Moon and Sun in case of a satellite circling the Earth, Jupiter in case of Sun orbit, solar wind when investigating asteroids, and more. It is also a good way to model low thrust propulsion.

The **atmospheric drag** (*pol.* *tarcie atmosferyczne*) is one of the primary phenomena that is usually modeled using perturbers. See Section 4.6.4 for a detailed discussion about atmospheric models and atmospheric drag. Another phenomenon that influences spacecraft on LEO is the non-spherical nature of the Earth's gravitational field. Several aspects play a role here. The major one comes from the oblateness, i.e. Earth is not a perfect sphere but instead has a bulge around the equator. Its equatorial radius is slightly larger than its pole radius. Also, the mass distribution is not ideal and varies. Finally, the last aspect related to the atmosphere is its dynamism. The atmosphere is a dynamic system that constantly changes its distribution and center of mass.

Another important class of perturbations comes from other heavenly bodies – **Sun and Lunar gravity** when dealing with orbits around Earth. In some use cases, also impact of other massive planets – Jupiter and Saturn – is taken into consideration. This influence is mostly negligible on satellites around Earth, however for interplanetary missions or asteroids, especially those in the main asteroid belt, it is often considered. Influence of other planets and major asteroids, such as (4) Vesta, which has a radius exceeding 500km, is typically considered when the trajectory is generally in the vicinity of the objects. For example, mission planners for a mission to the asteroid belt would likely consider perturbations from Mars, Jupiter and several most massive asteroids.

Solar radiation pressure is another effect that applies to all bodies. However, since its minimal net force, it is usually considered only for light-weight objects, such as small spacecraft or asteroid or when considering very long periods.

Other phenomenons are sometimes considered as well. **Yarkovsky effect** (*pol.* *efekt Jarkowskiego*) is an effect caused by solar radiation on slowly rotating bodies, such as small asteroids. The side exposed to the Sun warms up. Once that area rotates into the shadow, it starts to cool down by emission of thermal photons. The force is small but steady. Depending on the rotation direction, the effect causes the object to spiral inwards or outwards. This effect impacts all rotating objects in principle, but it is observable only on rocks and small asteroids between 10 cm and 10 km. The effect was first measured over a time span of 12 years between 1991 and 2003 on asteroid 6489 Golevka. [91].

Another related phenomena is **Yarkovsky–O'Keefe–Radzievskii–Paddack effect** or YORP effect (*pol.* *efekt YORP*) for short. There are three fundamental ways of how Solar radiation interacts with the body. First, part of the radiation from the Sun is absorbed. Secondly, part of the radiation is reflected and diffused. Thirdly, the radiation is emitted as thermal radiation. As photons possess momentum, each of those aspects changes the angular momentum. This effect is very weak, but it is persistent, and for long periods, it can yield noticeable results. Similar to the Yarkovsky effect, this effect is observable only on reasonably small asteroids. This mechanism is briefly described in [96].

The **Poynting–Robertson effect** (*pol.* *efekt Poyntinga–Robertsona*) is another weak effect caused by Solar radiation. The Sun radiation appears to be coming from a slightly forward direction due to the aberration of light caused by the body's relative movement. The aberration angle is very small but non-zero. This means that besides

the much higher solar radiation pressure acting outwards, another much smaller force slows down the object. This effect is minimal and affects mostly interplanetary dust particles of the size between 20-200 μm .

Finally, two bodies are said to be in **orbital resonance** (*pol.* rezonans orbitalny) when their orbital periods form a fraction, expressed by small integers, such as 1:2 or 2:3. The resonance means that the bodies experience periodic gravitational influence on each other. Depending on the proportion of bodies, they may form a stable resonance if both are of roughly similar mass (e.g., resonant asteroids). Alternatively, if one body is vastly heavier than the other, such as an asteroid and a planet, the lighter object is pushed out of resonance. This explains why the asteroid distribution in the Solar system is not uniform, and there are gaps. See Section 4.5.3 for details.

1.9. Miscellaneous Topics

The following section briefly discusses various topics related to astrodynamics.

1.9.1. Gauss's problem

The **Gauss's method** (*pol.* metoda Gauss'a), or **Gauss's problem** (*pol.* problem Gauss'a) is an algorithm for obtaining preliminary orbit determination based on two known positions r_1 and r_2 and a time of flight. The problem is presented graphically in Fig. 16. The full walkthrough of the solution is outside of the scope of this thesis. A step by step solution is available in [93]. A much better explanation is available in Chapter 5 and Appendix C of [6].

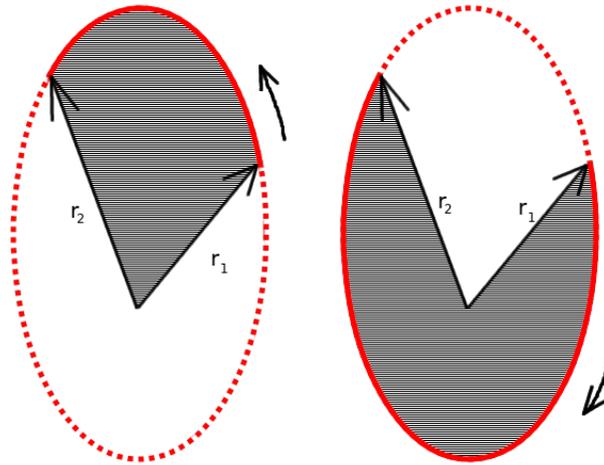


Figure 16: The Gauss's problem. Given the two known positions r_1 and r_2 and time of flight, there are exactly two orbital solutions that match the time of flight: the short way (left) and the long way (right).

1.9.2. Lambert's problem

The problem of determination of an orbit from two position vectors and a time of flight between them was formulated by Johann Lambert in 18th century and then later solved by Joseph-Louis Lagrange. It is an important tool for solving many problems in astrodynamics. The problem is formulated as follows in [92]:

The transfer time of a spacecraft moving on a conic trajectory is a function of only the sum of distances of the initial position P_1 and final position P_2 from the origin of the force and the major semi-axis of the conic.

This problem can be also formulated more formally as a boundary value problem of the following differential equation:

$$\vec{r}'' = -\mu \cdot \frac{\hat{r}}{r^3} \quad (1.18)$$

where \vec{r}'' is the acceleration vector, \hat{r} is the unit vector and r is distance of the moving object from the center of the orbited body. μ is defined with the following expression:

$$\mu = G \cdot (m_1 + m_2) \quad (1.19)$$

where G is a gravitational constant, m_1 is the weight of the orbited body and m_2 is the weight of the spacecraft. Since in most cases m_2 is insignificant compared to m_1 , it is usually ignored. The expression is then simplified to:

$$\mu = G \cdot m_1 \quad (1.20)$$

The solution to this problem is complex. A brief overview of the solution is available in [92]. It is recommended to read [94] beforehand, as the former is missing many designation explanations. A much more detailed annotated solution with examples is available in Section 5.3 of [15].

1.9.3. Sphere of influence

Throughout the whole interplanetary flight, the spacecraft is influenced by the departing planet's gravity, target planet, and the Sun. However, the influence is not equal. The dominant force for most of the time during the interplanetary flight is that of the Solar System's central body – the Sun. The influence of departing and destination planets overcome that of Sun's only during relatively brief periods. As such, one very convenient concept is a **sphere of influence** (*pol.* sfera wpływu), which determines which single body – departing body (such as Earth), target body (such as Mars) or Sun – has the dominating influence. Other bodies' influence can be modeled as perturbations or neglected altogether, depending on the precision required. This model greatly simplifies all calculations. The flight trajectory is then approximated using conic sections (*pol.* krzywe stożkowe). See Fig. 2 for conic shape examples. The method of approximating the trajectory this way is called **patched conics approximation** (*pol.* aproksymacja krzywymi stożkowymi). This method is discussed in detail in Section 8.3 of [6].

1.9.4. Kessler syndrome

A **Kessler syndrome** (*pol.* syndrom Kesslera) is a theoretical scenario in which a collision of two satellites trigger a chain reaction of thousands of debris that hit other satellites that continue generating more debris. The final effect is predicted to make the LEO entirely unusable for an extended time. The concept was proposed by Donald Kessler in 1978. Several purposeful collisions were conducted to demonstrate anti-satellite capabilities: China shooting down FENGYUN-1C satellite, US Navy shooting down its own NROL-21 satellite, India. There was one accidental collision of defunct Iridium 33 and Cosmos 2251 satellites. Each collision creates hundreds of debris tracked from Earth and likely many more that are too small to track.

Currently, all satellites are required to have a capability to safely dispose at the end of life, either by deorbiting or moving to a graveyard orbit. The problem of space debris is an ongoing concern. The SOCRATES project [77] provides a report of potential upcoming close passes of known satellites and debris.

2. Related Work in Astrodynamics

The second chapter describes the current state of the art in the broad topic of orbital mechanics. It can be roughly split into four sections. The first one is a review of current, up to date books. The second one is an attempt to describe the latest papers and articles in related fields. The third section is an overview of existing software, its capabilities, and limitations. The fourth one is a selection of orbital related file and data exchange formats.

2.1. Related work

The astrodynamics is a rich and complex topic. Its current development can be split into several areas. As with any other major field of study, there are many books available with some considered references. As numerical methods make an essential part of the field, many different programs, tools, software libraries, and suites are available. Finally, as with any other field, there are many conferences, journals, and similar types of publications. This field is too large to be reviewed in its entirety. As such, the following sections contain a subjective, partial selection.

2.2. Astrodynamics Books

Astrodynamics is not a popular topic in Poland. The literature about astrodynamics was not easily accessible. Therefore the author spent a substantial amount of time looking for books. Most of them had to be purchased in the US and shipped overseas, as they were not available in Polish libraries or bookstores. The incomplete set of books the author managed to get is presented in Fig. 17.

The first book that comes in any reputable list dedicated to orbital mechanics is *Fundamentals of Astrodynamics* by Roger Bate, Donald Mueller, and Jerry White [5]. The initial 1st edition has been published in 1971 and became a de facto standard in the industry due to its ability to explain complex topics using accessible language. It is often affectionally called *BMW* due to the authors' acronyms. The book covers the topics of two-body orbital mechanics, discusses orbit determination from observations, basic maneuvers, lunar and interplanetary trajectories, and perturbations. Sadly, the age of this book is noticeable. It uses imperial units (nautical miles, square feet, etc.), and the equations typesetting is horrible. Fortunately, after 49 years, a second edition [6] has been published in 2020, which addresses most of the issues and contains many updates. If possible, it is strongly recommended to obtain the second edition.

The second fundamental book in the field is *Orbital Mechanics for Engineering Students* by Howard Curtis [15]. This extensive (over 750 pages) book provides an excellent, comprehensive introduction to orbital mechanics. The initial refreshment of background math (vector arithmetics, notations, numerical integration, etc.) is very useful. The topics are supplemented with many examples and exercises with solutions. The two-body problem, orbits, and orbital maneuvers, including orbital rendezvous, are discussed in detail. Several chapters are dedicated to topics specific to satellite design – rigid body dynamics, satellite attitude maintenance, rocket dynamics – that, while fascinating subjects on their own – have nothing to do with orbital mechanics. The last extensive (70+ pages) chapter is dedicated to orbital perturbations.



Figure 17: A subset of books the author managed to obtain: Fundamentals of Astrodynamics, 1st ed. [5], 2nd ed. (not shown) [6], Astronomical Algorithms [37], Mission Geometry [84], Space Mission Analysis and Design, 3rd ed. [70], Spacecraft systems engineering [19], Orbital Mechanics for Engineering Students [15].

The third popular book in the field is *Mission Geometry; Orbit and Constellation Design and Management* by James Wertz [84]. Although not strictly limited to orbital mechanics, it focuses on mission and satellite designs, which orbital mechanics being only part of. James Wertz is a well-known persona in the industry, as he is the primary author of SMAD (to be discussed later in this section). The goal of this work is to provide the means to design specific orbits to fulfill mission goals. It is heavy on Earth Observation aspects and assumes the focus on Earth, although there are some limited elements dedicated to interplanetary aspects. This publication's solid areas are ground track and coverage, extensive background on satellite constellations, and excellent annotated bibliography. The last aspect is of particular use when starting research in a given field. Many chapters end with a list of related books, with a brief paragraph explaining each referenced item's strong and weak points.

The fourth book is *Space Mission Analysis and Design*, third edition by James Wertz and Wiley Larson. The book is almost universally referenced to as *SMAD*. This nickname is so popular that the fourth edition has a subtitle *New SMAD*. This book is an absolute standard in satellite and aerospace engineering. It is a compendium of all topics related to designing a space mission: defining mission requirements, selecting appropriate orbit to fulfill the mission, design the satellite and all aspects of its operation, and more. It is a great book, and it is highly recommended that every person interested in space engineering read it. Surprisingly enough, although a bit dated (published in 1999 and republished many times since), the third edition is more popular than its latest fourth edition (published in 2011). The New SMAD is no longer considered a handbook. Weighing around 2.5 kg, it is simply too large to be carried around.

The fifth book is *System GPS* by prof. Cezary Specht [74]. This book is dedicated to specialized orbital mechanics application: Guidance and Navigation Satellite Systems. The problems of determining orbital and on the ground positions are the major aspects being thoroughly discussed. From the orbital mechanics perspective, the most interesting is Chapter VI, dedicated to satellite movement in Earth's gravity field. Sadly, the book is out of print since its initial publication in 2007. While the book is focused on Navstar GPS, chapter XII provides an insight into the evolution of the GALILEO system.

Another book of some interest from astrodynamics's perspective is *Spacecraft Systems Engineering* by Peter Fortescue [19], Graham Swinerd and John Stark. The primary topic of study is the process of designing, building, launching, and operating a satellite with related subjects: thermal, electrical, propulsion, and altitude systems. Chapters 4 (Celestial Mechanics), 5 (Mission Analysis, discussing orbits and transfers), and 14 (discussing ground track and communication windows with ground stations) are relevant to the topic of astrodynamics. However, given much better alternatives available, this book is not recommended.

The last book worth mentioning is *Astronomical Algorithms* by Jean Meeus [37]. The intended audience is the astronomers' community, which causes the problems to be described strictly from the point of Earth observations. However, some problems discussed make it well suited for solving specific problems. The book is split into 58 chapters, each dedicated to the numeric solution of a specific problem, such as calculating Julian Date, precession, nutation, positions of planets etc. While impressive on their own, many problems, such as calculating stellar magnitude, have no astrodynamics application.

Finally, the great missing absent on this list is *Fundamentals of Astrodynamics and Applications* by David Vallado [79]. This book is considered a modern successor to *BMW*, with a particular focus on modern numerical solutions and their implementation aspects. In the industry, the book is referenced simply as *Vallado*. Sadly, the author of this thesis could not obtain access to it until the thesis was complete. Similar to other books, this one is hard to find and is usually unavailable. In the rare periods when it is available, the on-line price is often upwards of 400 USD. However, judging by the on-line comments and numerous references in recent papers, publications, and software code, this book is highly recommended if one manages to get hold of it.

2.3. Software tools

There are many software tools, libraries and environments that have some application in astrodynamics. The following sections discuss a selection of them.

2.3.1. System Tool Kit (STK)

The most well-known software suite related to astrodynamics is *System Tool Kit* or simply *STK*, published by Analytical Graphics Inc. [1]. This is a powerful software described as multi-domain mission-level software for system design, operations, and analysis. This is commercial software. While there are free demo licenses that are not time-limited, they limit the software features. More advanced features need to be licensed (paid) separately. The software has a very aggressive privacy policy (7 pages long) that require sending back many data. This may be a serious problem for organizations that desire confidentiality (e.g., military, commercial projects). *STK* is mostly addressed at Earth mission planners and does not support missions that involve other bodies. There are extensive on-line training materials available, which are a significant help for newcomers. Also, some webinars are free of charge, although they require prior reservation. This software's desktop nature makes it difficult to use for an on-line service, although there are good exporting capabilities to Cesium using CZML format. Some components are military-oriented (ships, missiles, locations referred to as targets). An example of the *STK* software running is presented in Fig. 18. One particularly nice feature is the orbit design tool for Earth Observation missions. It shows the ground track and allows tweaking orbital parameters. A screenshot is presented in Fig. 19.

2.3.2. General Mission Analysis Tool (GMAT)

Another popular tool is *General Mission Analysis Tool* (or *GMAT*) [20], developed as open-source by NASA, private industry, and a group of public and private contributors. This open-source software is most likely the most advanced tool available to the scientific community. It is an extensive environment with different gravity and Earth curvature models, a great variety of bodies available (Earth, Luna, Mars, and many more), a selection of

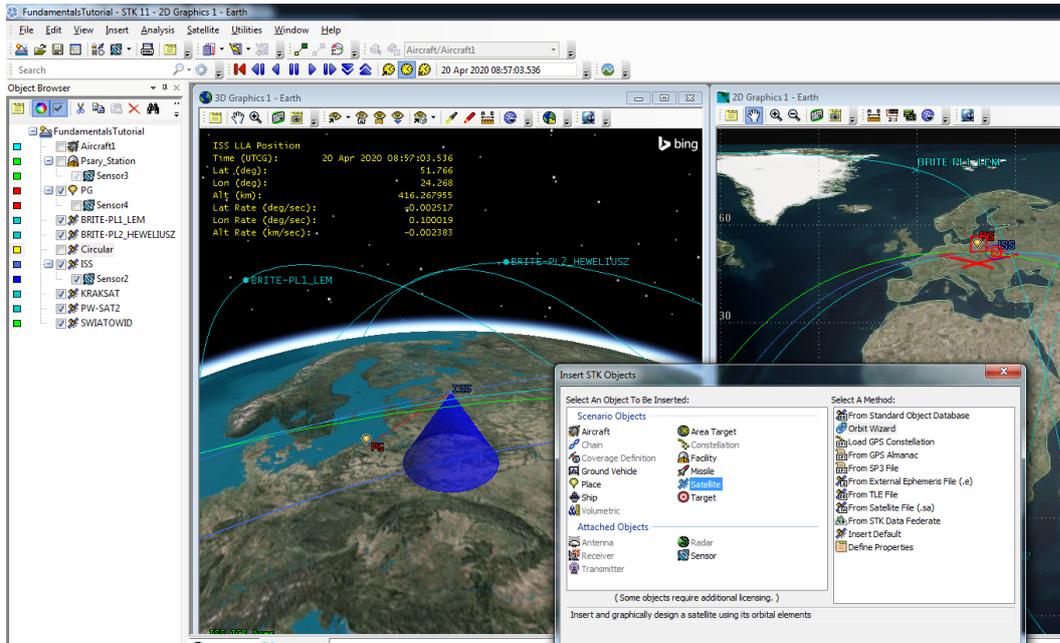


Figure 18: STK interface.

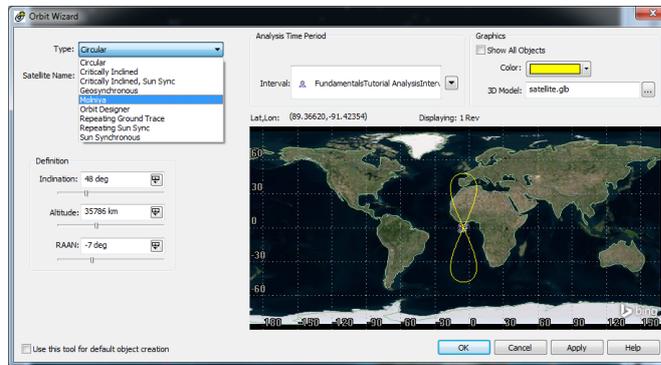


Figure 19: One of the particularly nice features of STK is its orbit design tool. It shows the ground track of the orbit, while allowing tweaking some orbital parameters. This is very useful for Earth observation missions planning.

atmospheric models, coordinate systems, notations etc. One exciting feature is a solver – a tool that attempts to adjust initial orbit and burns to achieve the intended target orbit. The software is more focused on interplanetary missions than just Earth vicinity. While by default it offers 8 major planets, one dwarf (Pluto) and Earth-Moon (Luna), there is a capability to add additional bodies. The *Add Moon* option is not something that is seen often. The software was used by NASA in several actual interplanetary missions. While the default mode of operation is graphical, there is also a command-line interface available, which can be used to automate specific tasks, such as web service backend operations. GMAT provides several APIs (Application Programming Interface) that can be used to integrate it with other systems. The software is available for Windows, Linux, and macOS. An example screenshot of an Earth-Luna mission is presented in Fig. 20.

2.3.3. Poliastro

Poliastro is an open-source collection of Python functions, published under MIT license [59]. The software offers analytical and numerical orbit propagation, conversion between coordinate frames, Hohmann and bi-elliptic maneuvers computations, extensive plotting capabilities, initial orbit determination, and much more. The learning curve is somewhat steeper compared to other graphical tools, such as STK or GMAT. However, many tutorials explain how to achieve common tasks, such as calculating Hohmann transfers, propagating osculating and perturbed orbits, etc. The software is written in Python and can easily integrate with a great selection of other Python

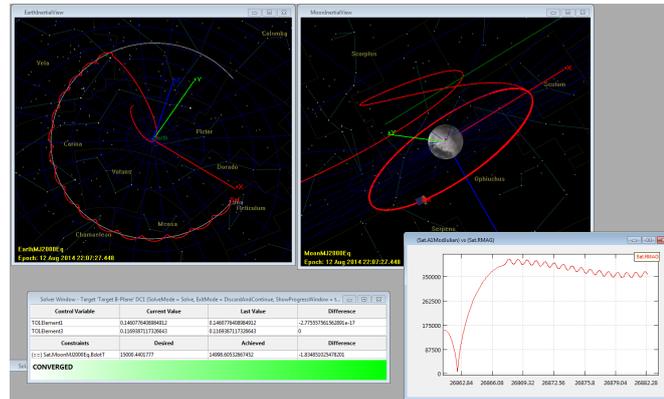


Figure 20: The General Mission Analysis Tool (GMAT) is a powerful software made available by NASA.

libraries, making it by far the most extensible tool. One instrumental capability of *Poliastro* is its handling of units. Most parameters specified in the high-level API require the parameters to be expressed with units, e.g., velocity is expressed as 7.7 km/s (in Python: $7.7 * \text{u.km/u.s}$). This makes it much easier to spot mistakes and make sure the conversions are correct. Thanks to Python portability, the software can run on Windows, Linux, and macOS. This software has been used as a base for the *Perylune* software. The great majority of diagrams and charts in this thesis were produced with some involvement of the *Poliastro* software. The software has been initially developed by Juan Luis Cano Rodriguez while studying at the Politecnico di Milano. While the original author is still heavily involved, there is now a healthy community of contributors developing the software. The author of this thesis joined the *Poliastro* project and contributed several improvements with more planned in the near future. *Poliastro* uses the Github platform for collaboration, making it very accessible for enthusiasts, volunteers, and the academic community in general.

2.3.4. Cesium

Cesium library [12] is an Earth visualization library suitable for interactive web experience, focused 3D geospatial visualizations. AGI, the company behind *STK*, developed this JavaScript library. One important aspect of *Cesium* is the Cesium Markup Language or CZML, CZML(*pol.* CZML) a JSON-based format that allows exchanging 3D data, including its variation over time. This format was specifically designed to handle orbital and atmospheric trajectories, land visualization, and other similar data. The library itself is open source, as is the CZML format. There is an optional paid *Cesium* Ion service that provides additional services, but it is unnecessary for most orbital focused applications. Thanks to the CZML format being open, a growing list of software tools can use it. The author used the *Cesium* library to develop a web site for visualizing Polish satellites. The trajectory was calculated using *Perylune* software that exported results using CZML syntax that is then visualized using *Cesium*. See Section 4.7 for details. An example visualization of GPS constellation using CZML and *Cesium* is presented in Fig. 21.

2.3.5. Python libraries

Python is a popular language, particularly in the research community, due to its simplicity and flexibility and an extensive list of libraries. One can say there is a snowball effect in progress – the language is being chosen for new projects, which are later published, further increasing the attractiveness of the environment. The following is a selection of a few out of many thousands of projects available.

The *SGP4* models are orbital propagation models developed a long time ago and used for TLE data format (see Section 2.5.3) back in the 1960s. Since then, the models were updated by T.S.Kelso [2]. The original models are implemented in Fortran, but they were ported to many languages, including Python. There is a *tle-tools* library that handles orbital data in TLE format. Another interesting library is *AstroPy*, an astronomy oriented library that provides a capability to conduct many calculations. One particularly useful capability is the units module, which

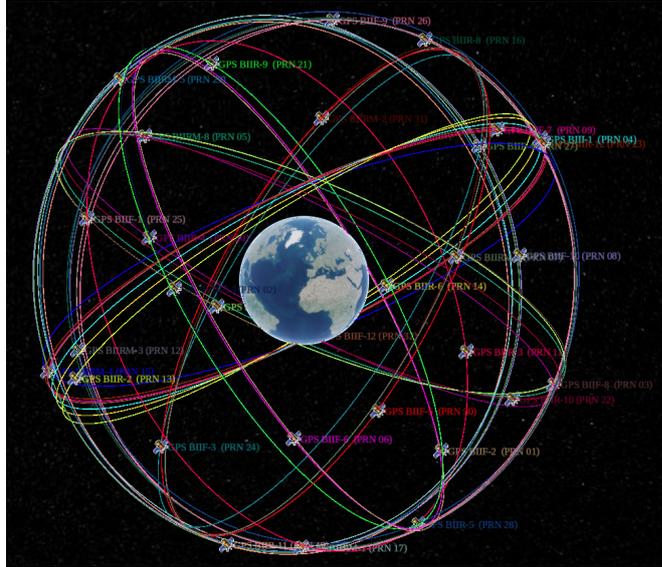


Figure 21: A constellation of GPS satellites presented using Cesium library. The data has been calculated using Perylune software and exported using CZML format.

allows specifying scalars and vectors with units, checking if values have expected units, and conducting conversions, if possible. This library is used by *Poliastro*, which in turn is used by *Perylune*. *Orbital predictor* is another Python tool dedicated to calculating orbital propagation to predict satellite fly-overs. It is handy for calculating incoming flyovers. The author of this thesis uses it in his Svarog [69] project to predict upcoming transmission opportunities to receive orbital images from NOAA satellites.

2.3.6. Other software

The software described in the earlier sections is just a subjective selection. There are many other software solutions available. The *Optimum Interplanetary Trajectory Software (OITS)* [55] is a purpose-written software, developed by Adam Hibberd, which analyzes challenging interplanetary missions, focusing on chasing two known interstellar objects – I1/'Oumuamua asteroid and I2/Borisov comet.

NASA publishes *The Spice Toolkit* software [49], often called simply *Spice*. This is an older software that provides C, Fortran, IDL, Matlab, and Java interfaces. Spice is a collection of APIs and functions with additional tools developed to achieve specific goals. One important aspect is the kernels or reasonably large data sets that can calculate planetary and asteroid ephemerides, perturbed orbits, and other values. The Navigation and Ancillary Information Facility (NAIF), a department of Jet Propulsion Laboratory, publishes kernels that are used by Spice and other software, such as *OITS*, *Astropy* or *Poliastro*.

Other software that the author did not investigate is *Orekit* (a mature Java library focused on Earth-centric scenarios), *Skyfield* (a python software that computes positions for the stars, planets, and satellites around Earth), and *Stellarium* (a user friendly desktop software for visualization of the night sky with some ability to track satellites).

2.4. Notable papers

The following list is a small subset that piqued the author's interest. The first area is rendezvous and proximity operations (RPO). The most profound publication is the *Line-of-Sight Guidance Techniques for Manned Orbital Rendezvous* by Edwin Aldrin [4]. His Ph.D. thesis was completed two months before he joined NASA and Apollo project. While he is commonly known under his nickname Buzz Aldrin, his Apollo colleagues called him Dr. Rendezvous. He introduced the concept of catch up and waiting orbits, and practically defined the PRO procedures later

used in the Apollo program. Although dated (published in 1963), it is still a relevant publication. Currently, much of the RPO activities are conducted autonomously. The automation aspect is discussed in *Autonomous Rendezvous and Docking Technologies – Status and Prospects* by James Wertz and Robert Bell [83]. In this paper, the authors propose to split the procedure into 8 phases: separate orbits, drift orbits (out of sight), drift orbit (insight), proximity operations A and B, docking, joint maneuvers, separation, and escape. Depending on the nature of the mission, some or all phases apply. With space missions becoming more complex, the importance and prevalence of RPO will increase. A good example supporting this observation is *CubeSat based Rendezvous, Proximity Operations, and Docking in the CPOD Mission* by John Bowen et al. [8]. In this 2015 publications, authors describe in detail a CubeSat Proximity Operations Demonstration (CPOD), where two 3U CubeSats demonstrated formation flying techniques, proposed delta GPS mechanism, inter-satellite radio ranging, a cold gas propulsion system for in-orbit maneuvering, and many other aspects. This paper is recommended for people interested in challenging but doable CubeSat missions.

Reference systems are another area that developed in recent years. International Astronomical Union (IAU) spent a substantial time to define new, stable reference systems. The issues with the old systems and the new proposals are presented in a short paper named *Comparison of Old and New Concepts: Reference Systems* by Jean Kovalevsky [32]. One of the problems with heliocentric systems mentioned is the implication of all objects in the solar system orbiting not around the Sun but around the mass's barycenter. This is a subtle but observable difference. There are periods when the barycenter of the Solar System is located outside of the Sun. The paper discusses the Barycentric Celestial Reference System (BCRS) and its relation to ICRS and ICRF reference systems. A much larger (16 pages) paper named *Major Concepts of Recent Celestial and Terrestrial Reference Systems* by Jan Kryński [33] expands on the problems and additionally delves into the problem of the reference time.

Another area where orbital mechanics continues to evolve is the data exchange formats. In a paper called *Comparative analysis of Satellite Data Formats (Almanacs) in GPS System (pol. Analiza porównawcza formatów danych satelitarnych (almanac) w systemie GPS)* Cezary Specht, Marcin Skóra and Mariusz Specht provide a detailed comparison between two most popular orbital data exchange formats dedicated to GPS systems. GPS specific formats offer higher precision needed in localization and navigation services. Although the de facto standard still seems to be the TLE format (see Section 2.5.3 for details), this format is old and is no longer adequate for modern systems. *Two-Line Element Sets – Practice and Use* by David Vallado and Paul Cefola [80] discusses TLE origins and its limitations that are getting more and more serious. While the TLE format remained the most popular in 2020, its successor – OMM – gains popularity rapidly. See Section 2.5.4 for details.

Many topics discussed so far can be considered an applied science. There are developments in fundamental research as well. In his paper *Revisiting Lambert's problem*, Dario Izzo proposes a new method of solving Lambert's problem using a novel approach. In particular, it allows considering multiple revolutions to find optimal solutions, which is a substantial benefit compared to other, more classical solutions. This may enable different interplanetary trajectories that, although slower, have lower Δv requirements. Another interesting paper that also can be considered fundamental research is *Near Rectilinear Halo Orbits and Their Application in Cis-Lunar Space* by Emily Zimovan, Kathleen Howell and Diane Davis [100]. Zimovan, a graduate student at Purdue University, studied lagrangian points L_1 and L_2 in the Earth-Luna system and described NRHO orbits. The benefits of this type of orbits have been considered by NASA, and the upcoming new Gateway space station will use this orbit type. The NRHO orbits are discussed in Section 1.4.6.

Finally, in recent years, a serious breakthrough in the astronomy community has been made as two interstellar objects have been observed. Both objects were observed only briefly before they became too dim to be observed even by the most powerful telescopes. Nevertheless, their trajectories have been determined with high precision. Naturally, there is a huge interest in studying those objects further. The paper titled *Project Lyra: Sending a Spacecraft to 1I/'Oumuamua (former A/2017 U1), the Interstellar Asteroid* by Andreas Hein, Nikolaos Periaklis,

Adam Hibberd et al. [23] discusses the difficulties of planning a rendezvous mission that could chase and eventually catch the interstellar asteroid. They use the *OITS* software discussed earlier that is particularly well suited for optimizing complex, multiple gravity assists trajectories. The somewhat modified team of Adam Hibberd, Nikolaos Perakis, and Andreas Hein further expanded on that idea in their another paper *Sending a Spacecraft to Interstellar Comet C/2019 Q4 (Borisov)* [24], where they discuss their proposal for a similar mission, but aimed at the second known interstellar object – I2/Borisov comet. Both papers, while fascinating theoretically, share the same problem. It is unrealistic to assume that such a mission would be designed, proposed, approved, and implemented in the very short time needed to match the expected launch window. This problem has been addressed in their follow-up paper *Project Lyra: Catching 1I/'Oumuamua – Mission Opportunities after 2024* [25], with launch dates considered to be in 2024 or later. Similar to earlier papers, this one also proposes trajectories with numerous gravity assists. It also investigates a very unorthodox concept of doing Sun gravity assist. The closer a spacecraft could get to the Sun's surface, the stronger the Oberth effect, but obviously, there are thermal limits on how close it could get. Authors proposed a new naming convention to describe multiple assists trajectories using the bodies used for gravity assists. Their craziest (but workable!) trajectory called E-E-V-E-M-E-J-IP-J-1I that visits Venus, Mars, Earth (4 times), Jupiter is presented in Fig. 22.

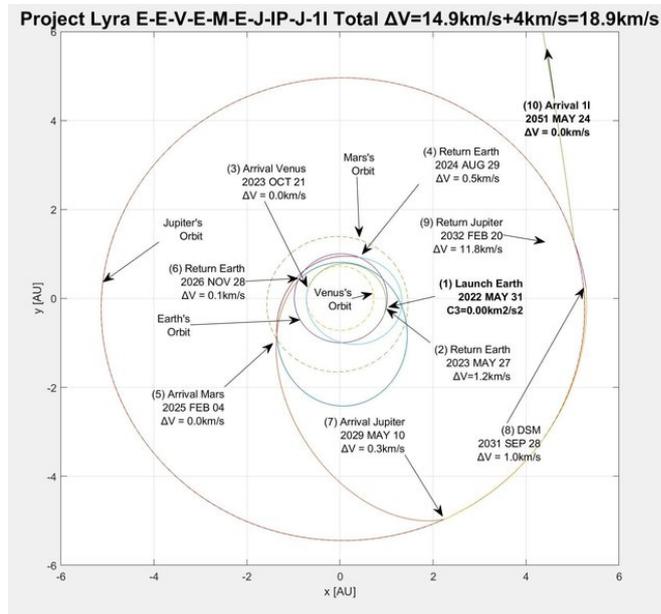


Figure 22: An unorthodox trajectory proposed by Project Lyra. The goal of this mission is to catch an interstellar asteroid. It assumes multiple gravity assists around Earth, Venus, Mars and Jupiter. Source: [25].

2.5. Orbital File Formats

There are many file formats in use that convey orbital information. Many of them are dedicated to specific purposes, e.g. GNSS applications require very high precision and are concerned with MEO orbits or lower. There are two dedicated formats for GNSS applications – Yuma and SEM. The Yuma and SEM formats comparison is available in [75].

In general, an almanac (*pol.* almanach) is a periodic publication, usually yearly, that provides a list of events for the upcoming period. In astronomy and space engineering, an almanac contains a list of astronomical events or provides necessary information on how to calculate them. The following sections describe various almanac formats and briefly discuss their primary area of application.

2.5.1. Yuma format

Yuma (*pol.* Yuma) format was designed to be human-readable and is used in GNSS applications. The data density is low, as there is much repetitive, explanatory text. It is well suited for the specific application it was designed for – a limited (less than 100) navigational satellites, most of them in nearly perfectly circular MEO orbits. Compared to other data formats, this one also has a rate of RAAN change defined, typically determined using perturbations.

```

***** Week 6 almanac for PRN-01 *****
ID:                01
Health:            000
Eccentricity:      0.8843898773E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9759017993
Rate of Right Ascen(r/s): -0.7737465154E-008
SQRT(A) (m 1/2):   5153.610840
Right Ascen at Week(rad): -0.2855338956E+001
Argument of Perigee(rad): 0.704684783
Mean Anom(rad):    -0.7098451155E+000
Af0(s):            -0.2288818359E-004
Af1(s/s):          -0.7275957614E-011
week:              6

```

Figure 23: An example of orbital data for Navstar GPS satellite PRN-01 in Yuma format.

Yuma format takes its name from Yuma Proving Grounds, a US military range where GPS receivers were tested when Navstar GPS was developed. Since its primary purpose is to provide information about GPS satellites, it contains several GPS related parameters. The syntax is easily human-readable as each line starts with a textual description of the parameter followed by its units. A series of 14 lines define each satellite. The first one started with a series of asterisks, which can be considered a name. The second line defines an ID of the satellite (integer). The third line describes the current state (0 – healthy, 255 – broken or undergoing maintenance). The third line defines eccentricity in scientific notation. Fourth is a time of applicability (a number of seconds since the beginning of the week when the almanac was created). Orbital inclination is specified in radians. The seventh line defines the rate of change in the measurement of the angle of right ascension. The eighth line defines the square root of the major semi-axis. The ninth line defines RA at the beginning of the week. The tenth line defines argument of perigee, an angular measurement along the orbital path measured from the ascending node to the point of perigee, measured in the direction of the SV’s motion. The eleventh line defines mean anomaly. Lines twelfth and thirteenth contain information about satellite clock bias (in seconds) and clock drift in seconds per second. Finally, the last line defines the current week, counted from the last epoch. GPS syntax uses 10 bits to specify week numbers, thus storing numbers between 0 and $2^{10}-1 = 1023$. The previous epoch started on August 22, 1999, while the current one began on April 6, 2019. Celestrak provides a reasonable description of YUMA format in [10], but the actual standard is defined in [21]. The almanac in this format can be downloaded from several places, such as [9].

2.5.2. SEM almanac

Another almanac format used in the GPS industry seems to be System Effectiveness Model or SEM. Compared to Yuma format, SEM looks more compact (does not have any human-readable text) and is better suited for software tools. It also uses scientific notation with more decimal points so it can provide more precise data in principle. See [11] for easy to follow descriptions. Note that it was defined by the same document [21] as Yuma

format. A beginning of a SEM almanac is presented in Fig. 24. As this is GPS specific format, it assumes that all orbits are Earth-centered.

```

31 CURRENT.ALM
6 589824

R0
R1 1
R2 63
R3 0
R4 8.85391235351562E-03 1.06410980224609E-02 -2.50656739808619E-09
R5 5.15360644531250E+03 -9.09348249435425E-01 2.24659562110901E-01
R6 3.31409454345703E-01 -2.47955322265625E-05 -7.27595761418343E-12
R7 0
R8 11

```

Figure 24: An example of orbital data for a satellite in SEM format. The R0 to R8 are line numbers are not part of the format.

2.5.3. TLE format

The almanacs discussed so far were dedicated to providing information about GNSS (GPS, specifically) satellites. However, GNSS satellites are a small fraction: less than a hundred out of 19625 objects currently tracked and orbiting Earth [43]. There is a need to share information in a standard way so different organizations and researchers can easily use them. Contrary to its name, the definition takes three lines as there is an additional line with the name. The format is sometimes referred to as 3LE.

```

SWIATOWID
1 44426U 98067QL 20349.98776826 .00089630 00000+0 59507-3 0 9995
2 44426 51.6361 136.4358 0004442 265.5329 194.8797 15.74706152 82799

```

Figure 25: An example of orbital data for a Światowid satellite in TLE (Two Line Element) format.

Many US institutions (such as NASA, NORAD, and DOD) came up with a format called **TLE** or Two Line Element. The format is described in [78]. An example entry is shown in Fig. 25. The format was defined in the 1960s and was originally formatted to print online printers, display correctly on 80 column text terminals, and even be stored on punch cards [43]. This format is by far the most popular in the industry. However, it has several limitations. See the next section about OMM for a discussion of TLE problems leading to its replacement format be defined.

The initial, unnumbered line contains the body name. In the example used in Fig. 25. The first line contains the following parameters after the line number (always 1): the satellite catalog number (also known as NORAD catalog number), which is a 5 digits number assigned by United States Space Command; one letter designating classification (U stands for unclassified); Epoch date and Julian Day fraction; first derivative of mean motion (often called ballistic coefficient), the second derivate of mean motion (usually 0); radiation pressure coefficient (so-called **BSTAR** or drag term); element number (999 in this example) followed by a checksum number (5 in this example). The second line starts with line number (2) followed by satellite number, inclination in degrees, Right Ascension of the Ascending Node (in degrees), eccentricity (a leading decimal should be added; the example specifies an the eccentricity of 0.0004442), argument of perigee (also in degrees), mean anomaly (degrees), mean motion (number of orbits completed per sidereal day), revolutions completed at epoch and is concluded with another checksum.

```

CCSDS_OMM_VERS = 2.0
CREATION_DATE =
ORIGINATOR =

OBJECT_NAME = SWIATOWID
OBJECT_ID = 1998-067QL
CENTER_NAME = EARTH
REF_FRAME = TEME
TIME_SYSTEM = UTC
MEAN_ELEMENT_THEORY = SGP/SGP4

EPOCH = 2020-12-14T23:42:23.177664
MEAN_MOTION = 15.74706152
ECCENTRICITY = .0004442
INCLINATION = 51.6361
RA_OF_ASC_NODE = 136.4358
ARG_OF_PERICENTER = 265.5329
MEAN_ANOMALY = 194.8797

EPHEMERIS_TYPE = 0
CLASSIFICATION_TYPE = U
NORAD_CAT_ID = 44426
ELEMENT_SET_NO = 999
REV_AT_EPOCH = 8279
BSTAR = .59507E-3
MEAN_MOTION_DOT = .8963E-3
MEAN_MOTION_DDOT = 0

```

(a) OMM data in KVM format.

```

[
{
"OBJECT_NAME": "SWIATOWID",
"OBJECT_ID": "1998-067QL",
"EPOCH": "2020-12-14T23:42:23.177664",
"MEAN_MOTION": 15.74706152,
"ECCENTRICITY": 0.0004442,
"INCLINATION": 51.6361,
"RA_OF_ASC_NODE": 136.4358,
"ARG_OF_PERICENTER": 265.5329,
"MEAN_ANOMALY": 194.8797,
"EPHEMERIS_TYPE": 0,
"CLASSIFICATION_TYPE": "U",
"NORAD_CAT_ID": 44426,
"ELEMENT_SET_NO": 999,
"REV_AT_EPOCH": 8279,
"BSTAR": 0.00059507,
"MEAN_MOTION_DOT": 0.0008963,
"MEAN_MOTION_DDOT": 0
}
]

```

(b) OMM data in JSON format.

Figure 26: OMM data has several exchange formats defined.

2.5.4. Orbit Mean-Elements Message Format

TLE data exchange format has been used since the beginning of the space age. It was initially designed when computer memory was a precious commodity, line printers and punch cards were used to store information. The data format is fixed length, so the choices made in the early 1960s to save up precious memory affect the precision of calculations conducted 60 years later. Some problems, such as the year 2000 problem, was mitigated by adding additional rules (if year XX is greater than 60, it is interpreted as 19XX; otherwise it is 20XX).

There was significant resistance to adopting any alternative due to the enormous popularity of the TLE format. However, the industry's primary reason to work together on a successor was the Norad ID field's limit. This 5 digit number is being assigned to each tracked object in space. At the time of publication of this thesis, the highest number assigned was below 50000. However, in March 2020, a new Space Fence facility under the US Space Force's auspices was commenced. It is an advanced radar system that is expected to track orbiting objects and space debris. The system is very sensitive and is expected to increase the number of tracked objects up to 10 fold. This means that the new objects' NORAD ID will exceed 100000. It will not be possible to publish its orbital data in TLE format that has reserved only 5 digits for TLE.

The new format has been defined by The Consultative Committee for Space Data Systems (CCSDS) [17] and it provides several exchange formats: KVM (key,value notation) as presented in Fig. 26a, JSON as presented in Fig. 26b, XML as presented in Fig. 27a, and CSV as presented in Fig. 27b. Depending on the intended use case, the most suitable data exchange format can be used. Another major flaw of TLE (its geocentric character) is address by OMM as well.

```

<?xml version="1.0" encoding="UTF-8"?>
<ndm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://sanaregistry.org/r/ndmxml/ndmxml-1.0-master.xsd">
  <omm id="CCSDS_OMM_VERS" version="2.0">
    <header>
      <CREATION_DATE />
      <ORIGINATOR />
    </header>
    <body>
      <segment>
        <metadata>
          <OBJECT_NAME>SWIATOWID</OBJECT_NAME>
          <OBJECT_ID>1998-067QL</OBJECT_ID>
          <CENTER_NAME>EARTH</CENTER_NAME>
          <REF_FRAME>TEME</REF_FRAME>
          <TIME_SYSTEM>UTC</TIME_SYSTEM>
          <MEAN_ELEMENT_THEORY>SGP4</MEAN_ELEMENT_THEORY>
        </metadata>
        <data>
          <meanElements>
            <EPOCH>2020-12-14T23:42:23.177664</EPOCH>
            <MEAN_MOTION>15.74706152</MEAN_MOTION>
            <ECCENTRICITY>.0004442</ECCENTRICITY>
            <INCLINATION>51.6361</INCLINATION>
            <RA_OF_ASC_NODE>136.4358</RA_OF_ASC_NODE>
            <ARG_OF_PERICENTER>265.5329</ARG_OF_PERICENTER>
            <MEAN_ANOMALY>194.8797</MEAN_ANOMALY>
          </meanElements>
          <tleParameters>
            <EPHEMERIS_TYPE>0</EPHEMERIS_TYPE>
            <CLASSIFICATION_TYPE>U</CLASSIFICATION_TYPE>
            <NORAD_CAT_ID>44426</NORAD_CAT_ID>
            <ELEMENT_SET_NO>999</ELEMENT_SET_NO>
            <REV_AT_EPOCH>8279</REV_AT_EPOCH>
            <BSTAR>.59507E-3</BSTAR>
            <MEAN_MOTION_DOT>.8963E-3</MEAN_MOTION_DOT>
            <MEAN_MOTION_DDOT>0</MEAN_MOTION_DDOT>
          </tleParameters>
        </data>
      </segment>
    </body>
  </omm>
</ndm>

```

(a) OMM data in XML (extensible markup language) format.

```

OBJECT_NAME,OBJECT_ID,EPOCH,MEAN_MOTION,ECCENTRICITY,INCLINATION,RA_OF_ASC_NODE,ARG_OF_PERICENTER,MEAN_ANOMALY,\
EPHEMERIS_TYPE, CLASSIFICATION_TYPE,NORAD_CAT_ID,ELEMENT_SET_NO,REV_AT_EPOCH,BSTAR,MEAN_MOTION_DOT,MEAN_MOTION_DDOT
SWIATOWID,1998-067QL,2020-12-14T23:42:23.177664,15.74706152,.0004442,51.6361,136.4358,265.5329,194.8797,\
0,U,44426,999,8279,.59507E-3,.8963E-3,0

```

(b) OMM in CSV (coma separate values) format. Both lines were wrapped (\) for readability.

Figure 27: OMM data has several exchange formats defined.

2.5.5. MPCORB Format

The 3 of 4 formats discussed so far (Yuma, SEM, TLE) are geocentric. That is a very reasonable approach, given that the bodies (satellites) they describe are orbiting Earth. However, that is not always true for all objects. A good example of a heliocentric format is MPCORB, defined by Minor Planet Center (MPC). While the absolute essentials are defined in [40], an extended commentary is available in [42]. It describes the orbits of minor planets, asteroids, comets, and similar objects in the Solar System. As of the time of writing this document, over a million objects are being tracked [39]. Since the total number of bodies is substantial, MPC provides several ways to access the data. First, usually only a small fraction of all known bodies are of interest. Typically, these are those asteroids that currently are in the vicinity of Earth. These are the so-called Near-Earth Asteroids (NEA). Out of that group, some are suspected of potentially not only crossing Earth's orbit, but do so in a close distance. That group is called Potentially Hazardous Asteroids (PHA). Another group of interest is objects of cometary nature. The status of whether an object is considered NEA or not may change over time. MPC provides various almanacs for objects currently passing near Earth, upcoming (ranging from +1 to +15 days), and past (from -1 to -15 days). There are also almanacs for objects that are in the process of being discovered. In the very early stages of discovery, there are usually too few observations or taken in insufficiently spaced intervals to determine orbits.

```
00001 3.34 0.12 K194R 77.37215 73.59764 80.30553 10.59407 0.0760091 0.21388522 \
2.7691652 0 MP0467603 6743 115 1801-2019 0.60 M-v 30h MPCLINUX 0000 (1) Ceres \
20190302 2458238.75384 A899 0F 1943 XB
```

Figure 28: An example of orbital data for a (1) Ceres asteroid in MPCORB format. The data has been wrapped (\) to increase readability. Original format expects one orbit per line.

MPCORB provides all of the mentioned almanacs in two formats: plain text and JSON. The data is the same, just the syntax changes. Plain text is more compact, which is a significant factor as the full almanac is enormous (173MB for text, 475MB for JSON). However, JSON is better suited for automated processing.

An example line describing a single body is presented in Fig. 28. The following section describes only the essential parameters and is based on [42] used in plain text notation. Columns 1-7 define a designation. Each object that is not a full planet is assigned a unique number. This number is roughly determining the order of bodies being discovered. The second parameter (columns 9-13, floating-point) determines absolute magnitude (H), i.e., the body's theoretical magnitude when located 1 AU from the Sun and observed from a distance of 1 AU. The third parameter (columns 15-19, floating-point) is the slope (G), which describes the surge of brightness when the object is near opposition). Both H and G form the so-called H-G system, described in [85]. The fourth parameter is epoch (columns 21-25), which is a date written in compact format [40]. For example, K194R means 2019-04-27. The fifth parameter (columns 27-35) defines mean anomaly at epoch, expressed in degrees. Columns 38-46 define argument of perihelion (a name for periapsis of objects orbiting the Sun) is a floating-point number the expressed the angle in degrees. Columns 49-57 define Right Ascension of the Ascending Node, also in degrees and a floating-point. Columns 60-68 define orbital inclination, measured in degrees and expressed as floating-point. All three parameters are specified for J2000.0 (a standard Epoch used in astronomy, Julian year 2000, January 1, 12:00, see Section 1.5.1). Orbital eccentricity (e) is specified in columns 71-79. The next parameter is mean daily motion (n) describes how fast a body is moving in the sky when observed from Earth and is expressed in degrees per day. It is useful to roughly estimate the body's location in the sky for observation planning. The next parameter is the semi-major axis and is expressed in AU. The column 106 describes U, uncertainty parameter, which describes how reliable the observational data is. The remaining parameters describe observations conducted so far or help assess how useful the data may be for long term planning. Columns 108-116 provide a reference number (it is not frequently used, the most common way to reference objects from this catalog is simply by its designation). Columns 118-112 specify the number of observations, and columns 124-126 specify the number of

oppositions when an object was observed. The opposition is a period when the body is particularly well suited for observations. When a previously known object is not observed during opposition, suspicion may be raised that either the original observations were incorrect, the object changed orbits, or may have been destroyed. For example, many Trans-Neptunian Objects, a group of asteroids from outer parts of the Solar System, are sometimes perturbed by outer planets. When passing close to the Sun they can hit its surface or be melted completely if made out of ice. The following two parameters are conditional. For single opposition orbits, they define (columns 128-131) the arc length in days, followed by the word 'days'. The parameters define the first (columns 128 to 131) and last (columns 133-136) year of observations for multiple-opposition orbits. The next parameter is r.m.s. residual, a root mean square difference between calculated position and its reported observed position. Expressed in " (seconds of angle). For detailed explanation, see [41]. The next two parameters define the coarse (columns 143-145) and precise (columns 147-149) indicator of perturbers. These values indicate whether a measurable impact of other large bodies has been observed. Note that major planets (Mercury–Neptune) are always accounted for in the calculations. If specified, the perturbers attempt to account for additional smaller but still non-negligible bodies, such as Ceres, Pallas, Vesta and others. Additional parameters are sometimes provided. See [42] for details.

3. Developed Software

This chapter describes the design assumptions, architectural decisions, and the actual software implemented.

3.1. Perylune Overview

Perylune is a periapsis (the lowest point in orbit) around Luna, better known as the Moon. It is also a name chosen for the library of tools developed during this master thesis, intended to aid various calculations related to orbits and astrodynamics in general. The software is written in Python 3 and uses several dependant Python packages to achieve its goals. The software is in principle text-driven and written in a portable way, and can run on Windows, Linux, macOS, and embedded environments as long as they can run Python 3.

As of the time of writing this thesis, the software provides the following functionalities:

- load, process and use many orbital data formats: TLE, Yuma, MPCORB (Minor Planets Center);
- import orbital data from several online databases: NASA HORIZONS (many Solar System bodies: major and dwarf planets, moons, asteroids, also NASA probes); CELESTRAK (TLE data for many Earth satellites and debris); MPC (Minor Planets Center, all known asteroids);
- calculates orbital burns, including Hohmann, prograde burns, pure inclination change. The burns are calculated using vector notation that is compatible with Poliastro software;
- study transfer windows: generate charts of body distances, such as Earth-Mars, Earth-Venus, and others, porkchop plots;
- aid investigation of interplanetary transfer windows: calculate velocities, generate distance and transit charts;
- a variety of auxiliary calculations, such as basic time calculations, Julian date, leap years, shadow related calculations, i.e., determine if the spacecraft is lit or in eclipse, DOP parameters for assessing GPS precision, and many more;
- provides a list of historic TLEs for Polish sats that are no longer in orbit, also provides the ability to merge several TLE data sources. This allows supplementing the current CELESTRAK database with historical data;
- provides several orbital perturbers: constant acceleration, solar sail, atmospheric drag;
- orbit inspection code, which allows detailed orbit presentation in textual format.

3.1.1. Design Assumptions

The following design assumptions and goals were kept in mind when implementing the software:

- **portable** – the software must run on as many platforms as possible. The consequence of this requirement was the selection of Python 3 as the language of choice, with a tiny, optional portion providing web interface written in TypeScript. It can run on Windows systems, which are still popular as the development platform. This is convenient for developers and potential users. It can run on Linux, which is the system of choice for server platforms. This may be useful when using Perylune as a web service backend. The software should also run on embedded platforms, such as Raspberry Pi. This gives the potential opportunity to run Perylune as an on-board software on a satellite in the future.
- **using industry standards** – The software should take advantage of as many existing standards as possible. For example, it must handle orbital data in TLE format, as this is the most popular format for decades.
- **easy adoption** – There should be as few barriers for adoption as possible. The software license should be permissive (eliminating legal barriers). The software should be easy to install and well documented (eliminating the barrier of entry for new users and developers).
- **do not reinvent the wheel** – There is a rich selection of libraries already developed. Instead of reinventing the same concept, it is better to take advantage of existing solutions.

3.2. Software architecture

The software follows the Python recommendations. The core software is provided as a Python 3 module, with the code dedicated to specific functionalities being split into separate files. There are many additional directories with input and output data, documentation, and various presentation related elements, such as Jupyter notebooks demonstrating certain operations.

The tree structure is presented in Fig. 29. The directories are described in the next section.

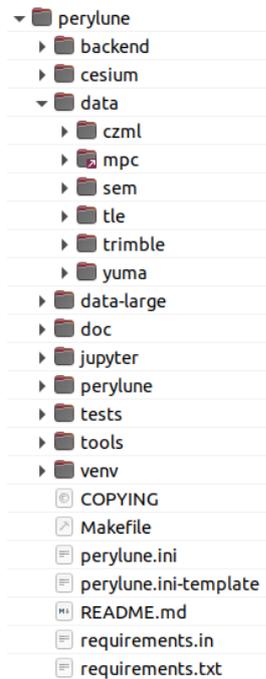


Figure 29: The dir tree overview of the Perylune software.

3.3. Software Layout

The `backend/` directory contains python scripts used to provide backend service for the web interface, along with integration scripts for Apache web server and its WSGI interface. The software uses `perylune` module to

retrieve current orbital data from Celestrak, then propagates orbits for Polish satellites and exports them in CZML format to be used by the Cesium library. This script is expected to be either run periodically, e.g., daily, to generate the update or on-demand, every time the user visits the website.

The `cesium/` directory contains all files necessary to build a web interface using the Cesium library. In particular, the `cesium/src/Utils` directory contains a set of utility functions: `cursor_label.ts` shows and updates label with geographical coordinates every time a cursor is moved, `czml_viewer.ts` provides the CZML file viewer with the ability to load CZML file, display list of satellites, zoom to current satellite's position, show TLE information, and others. The `reference_frame.ts` provides the ability to choose the reference frame – either ECI (inertial) or ECEF (fixed).

The `data/` and `data-large/` directories contain data files in various formats: `data/czml/` (Cesium Markup Language), a JSON format that describes graphical scene changing over time, which can be used by any CZML-compatible software; `data-large/mpc/` (Minor Planets Center) which holds orbital data for many asteroids; `data/sem/` – GPS almanac in SEM format; `data/tle/` – various orbital information using TLE format; `data/trimble/` – GPS almanacs in a format used by Trimble software; `data/yuma/` – GPS almanacs in Yuma format. Currently the MPC data is kept in `data-large/` directory with a link in `data/`. For reasons regarding this split, see Section 3.7.

The `doc/` directory contains documentation written in Markdown. The recommended way to view this documentation is via the Github web interface.

The `jupyter/` directory contains around 20 Jupyter notebooks that demonstrate many of the calculations conducted during this thesis. Jupyter is a visually attractive way of mixing Python code with graphically rich results presentation, including interactive charts and diagrams. The best way to view the files is to use the Github web interface.

The `perylune/` directory contains the core Python code. Its content is described in detail in Section 3.5.

The `tests/` directory contains `pytest` unit tests, along with any supplementary files that may be needed to run them.

The `tools/` directory contains a set of small tools that use `perylune` module to achieve individual specific goals, such as conducting orbital calculations for a proposed satellite during a Space Missions class, exporting NOAA satellites trajectories etc.

The `Makefile` is an optional file that can automate the installation on Linux systems. The usefulness of this script for anyone except the author is limited.

The `requirements.txt` is a file that can easily install the python dependencies. See Section 3.4 for details.

3.4. Installation procedure

The *Perylune* software is still in reasonably early development at the time of writing this thesis. The installation is not well tested. However, the principles are reasonably simple. The first step required is to get a Python 3 installation. The software has been tested with 3.6, 3.7, and 3.8, but newer versions are also very likely to work. The software is best obtained from GitHub using `git` or downloaded as a zip archive. The `pip` software from python is used to manage the dependencies.

To download the latest `perylune` software:

```
git clone https://github.com/tomaszmrugalski/perylune
```

The software will be downloaded to a new `perylune` directory. If there is an older version already checked out, it can be updated:

```
git pull
```

It is useful to create a Python virtual environment in many deployments, so all dependencies are installed locally. See [68] for details.

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

It may be necessary to export PYTHONPATH variable in some environments to point to the *Perylune* root directory.

```
export PYTHONPATH=.
```

3.5. User's Reference Manual

The code provided is a python 3 module. The functions provided can be used using standard python 3 import mechanism. Most functions have detailed documentation that explain their purpose, parameters and returned values. For example, to parse TLE orbital data and print its details using `load_tle` and `print_orb` functions, the following example can be used:

```
$ python
>>> from perylune.orbit_tools import load_tle, print_orb
>>> tle_text = """NOAA-17
... 1 27453U 02032A 20263.80942421 -.00000011 +00000-0 +13583-4 0 9998
... 2 27453 098.5909 208.3215 0011096 327.5463 032.5033 14.25072668948324"""
>>> orb = load_tle(tle_text)
>>> print_orb(orb)
```

will produce the following output:

```
7178 x 7194 km x 98.6 deg (GCRS) orbit around Earth (♁) at epoch 2020-09-19T19:25:34.251744000 (UTC)
a(α)=7186.3853km, b=7186.3809km, e=0.00, i=98.59deg raan(Ω)=208.32deg argp(ω)=327.55deg nu(ν)=32.57deg
period=6062.85s periapsis=7178.4113km(800.27km) apoapsis=7194.3594km(816.22km)
```

The `alamanc_yuma.py` file contains `AlmanacYuma` class that can be used to load Yuma almanac. The useful methods are `load` and `printAll`. The file also defines an auxiliary class `OrbitYuma`. A short example of usage is provided at the end of the file.

The `constants.py` file contains a small number of constants using `units` concept from `astropy` module. As of time of writing, this file is basic but is expected to grow in the future.

The `geom.py` file provides `solar_angle` function that calculates angle between Sun rays and the spacecraft vector in relation to Earth center. This is used to determine whether the spacecraft is in the light or in the eclipse.

The `gpsdop.py` file provides `GpsDop` class for calculating DOP (Dilution of Precision), used in GPS. It is able to calculate GDOP, PDOP, HDOP, VDOP, TDOP parameters.

The `horizons.py` file provides an interface to the NASA Horizons database. It can retrieve ephemerids, parse them, and plot various distance charts. A small example is provided at the end of the file.

The `interplanetary.py` file provides utility functions to calculate various aspects of interplanetary trajectories, such as escape velocity, heliocentric velocity, transfer velocities, Hohmann burns, and others. It can also export the data in CSV format and generate some charts.

The `mpc.py` file provides interface to Minor Planet Center database. In particular, it is able to parse the MPC files using `parse_txt` that can be later searched using `find_objects`.

The `orbit_tools.py` file provides an assortment of utility functions that print orbit details (`print_orb`), load TLE data (`load_tle`), change inclination from one orbit to another (`inc_change`), calculate orbital velocity of orbit (`calc_vel`), propagate orbits to various points, such as periapsis, ascending or descending nodes, performs plane change maneuver, returns cost of specific maneuver and more. This is by far the most extensive file.

The `orbitdb.py` file provides an interface to Celestrak database and can manage an arbitrary number of files and on-line locations that contain TLE data. The `OrbitDatabase` class provides `refresh_urls` method that checks if locally downloaded cached copies are up to date, and `get_name` and `get_norad` functions that search for a satellite by name or by NORAD ID.

The `perturbers.py` file provides implementations for several orbital perturber models in a format that is compatible with *Poliastro* software: `constant_accel` provides constant forward (positive values) or backward (negative values) acceleration, `solar_sail` provides a model of a solar sail being pushed by Sun's radiation pressure.

The `time.py` file provides several basic time-related functions.

The `tle.py` file is obsolete and should not be used.

The `utils.py` file provide various small utility functions related to geographical coordinates presentation, checking whether a string can be safely used as a filename and similar.

3.6. Orbital Propagation using Cowell method

As discussed in Section 1.9.2, the Keplerian orbits are the closed-form solutions of the two-body problem. This approach assumes that the body moves on osculating orbits (i.e., ideal, not perturbed by any other forces). While it is a good first approximation, often more precise results are needed. In such a case, the additional forces are modeled as perturbers. To account for perturbations, the eq. 1.18 gains additional element a_p .

$$r'' = -\mu \frac{r}{r^3} + a_p \quad (3.1)$$

The Cowell method reduces eq. 3.1 to first-order differential equations:

$$\begin{aligned} r' &= v \\ v' &= a_p - \mu \cdot \frac{r}{r^3} \end{aligned} \quad (3.2)$$

where r and v are the radius and velocity of an object with respect to the central body. For the purpose of numerical integration, the equations can be further split into a system of equations:

$$\begin{aligned} x' &= v_x & \hat{v}_x &= a_{px} - \mu \frac{x}{r^3} \\ y' &= v_y & \hat{v}_y &= a_{py} - \mu \frac{y}{r^3} \\ z' &= v_z & \hat{v}_z &= a_{pz} - \mu \frac{z}{r^3} \end{aligned} \quad (3.3)$$

where $r = \sqrt{x^2 + y^2 + z^2}$. With that approach, the numeric integration can take place.

Cowell's method is praised for its simplicity and robustness, as it can be used to model any external force: atmospheric drag, Moon, Sun, or other bodies' gravity, radiation pressure, etc. Furthermore, it allows for multiple perturbations by the simple addition of their a_p vectors. Some drawbacks of Cowell's method are discussed in [6], Chapter 9.2. The method is also discussed briefly in [15], Chapter 12.2.

The *Poliastro* library provides an easy to use optional `ad` function that takes time t_0 , u (a concatenation of r and v vectors) and k (which is equal to μ) parameters and return non-Keplerian acceleration. This interface has been used by all perturbers implemented in the *Perylune* software. See Section 4.6 for details.

There are alternatives to Cowell available: Farnocchia, Vallado, Mikkola, Markley, Pimienta, Gooding, and Danby. See [61] for a matrix of supported propagators and scenarios where they are applicable.

3.7. Git repository

The *Perylune* software is primarily available in its main git repository on GitHub, available under this URL: <https://github.com/tomaszmrugalski/perylune>. Some of the data files are large. Those are available under a separate repository: <https://github.com/tomaszmrugalski/perylune-data/>. This repository, if needed, should be cloned into `data-large/` directory. This is generally unnecessary unless MPC data is to be processed. As of the time of writing this thesis, the `perylune-data` repository is only needed for the asteroid survey.

3.8. Dependencies and Acknowledgements

This software makes extensive use of the *Poliastro* library [59], including its propagation models, orbit plotting capabilities, diagrams, and many other useful features. This software uses *tle-tools* python library to load and parse TLE data. This software uses the *Plotly* library to generate charts and plots of numeric data.

3.9. Example Charts

The *Perylune* software uses *Plotly* and *Poliastro* libraries to visualize various types of data, including orbital trajectories. Two basic examples presented in Fig. 30 and 31. A majority of figures in this thesis were produced using *Perylune* software.

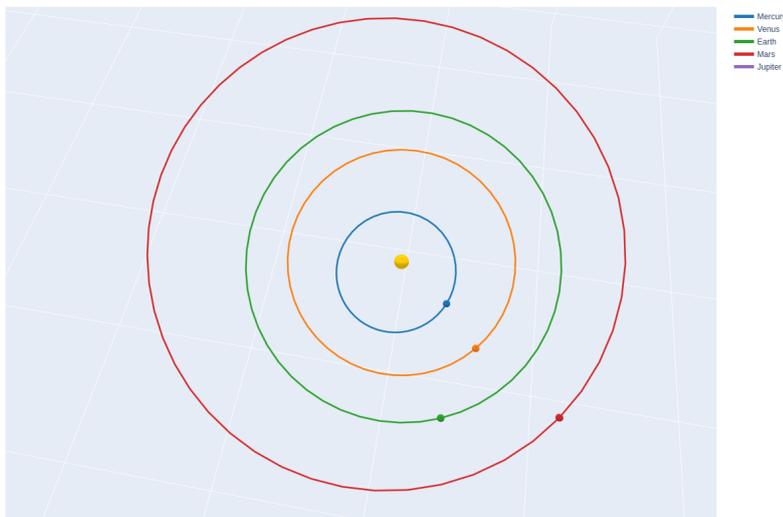


Figure 30: A top view of the inner Solar System with five innermost planets. Mercury’s eccentricity is clearly visible, while Venus’ and Mars is somewhat noticeable.

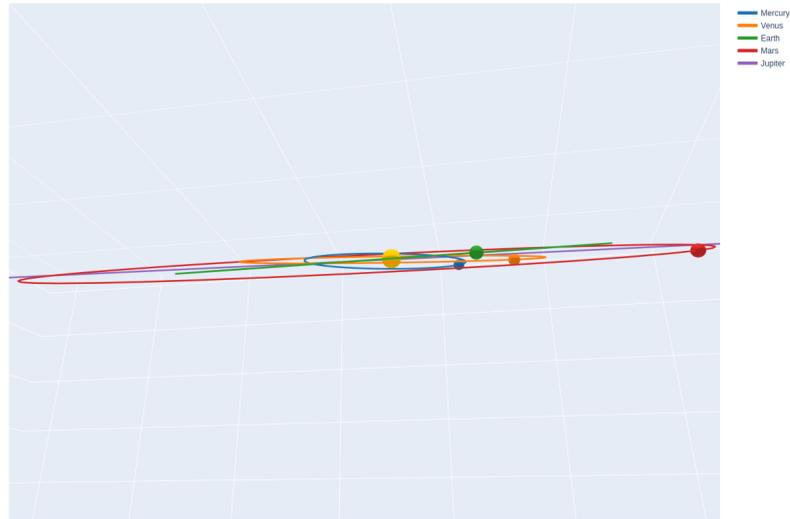


Figure 31: A side view of the inner Solar System with five innermost planets. The perspective chosen so the camera is exactly on the ecliptic (Earth's orbit appears as flat line). Inclination of various planets becomes visible.

4. Usage scenarios

This chapter describes solutions to problems defined in Introduction chapter: 1. georeferencing images received from NOAA satellite; 2. a series of orbital maneuvers necessary to reach old or defunct satellites along with a sketch of a low cost CubeSat mission with a review of currently available launch providers; 3. how to corroborate or discredit sensationalist articles about asteroids to hit Earth and how to obtain and interpret ephemerids; 4. discussion of interplanetary transfer windows for various planets, including Mars and Venus; 5. a survey of all currently known asteroids with a discussion regarding propulsion requirements for reaching them; 6. simulation results for perturbations induced by a sail of PW-Sat2 satellite with a novel proposal of using it as solar sail; 7. an overview of a developed interactive web service that demonstrates all current and historic Polish satellites, with 3D visualization and ability to predict fly-overs for Poland and other arbitrary regions of Earth.

4.1. Problem 1: Georeferencing Satellite Images

With the advent of cheap software-defined radio (SDR) hardware, it is possible to receive satellite VHF and UHF satellite transmissions. As part of a different assignment [69], the author designed and built a satellite ground station that can receive VHF transmissions in the 137MHz band. The transmissions convey Earth Observation images from NOAA satellites. The NOAA images are transmitted using APT encoding and contain the current view of the atmosphere. They are imaged in two IR bands: near IR and far IR, although the NOAA satellites have several sensors and can be reconfigured to transmit photos from different bands. In both cases, the images contain a scan along the fly-over path. The visual data on its own is hard to interpret because characteristic land features in Europe are often clouded, and the image is very wide (usually covers most of Europe). This issue can be seen in Fig. 32. The developed software should use image acquisition time and known orbital trajectory to provide georeferencing information. The calculated coordinates can be used to overlay country boundaries, geodetic grid, and other types of information. The practical goal is to increase the readability of images received in a working project. Prof. Marek Moszyński has proposed this task.

4.1.1. Calculating orbital position

The georeferencing procedure requires several parameters: orbital parameters, the time of transmission start t_{LOS} , length of transmission Δt and geometry of the optical sensor mounted on a satellite. In a general case, two timestamps are required: time of transmission start t_{AOS} for beginning, in the satellite industry called Acquisition of Signal or AOS, (*pol.* uzyskanie sygnału, but AOS abbreviation is in common use) and end of the transmission time t_{LOS} during Loss of Signal event, or LOS (*pol.* utrata sygnału, but LOS abbreviation is in common use). However, since NOAA satellites transmit their images continuously at the rate of two lines per second, the end of transmission time can be substituted with transmission duration d , which can be either expressed in seconds or number of lines expressing the height of the received image:

$$t_{LOS} = t_{AOS} + \frac{d}{2} \quad (4.1)$$

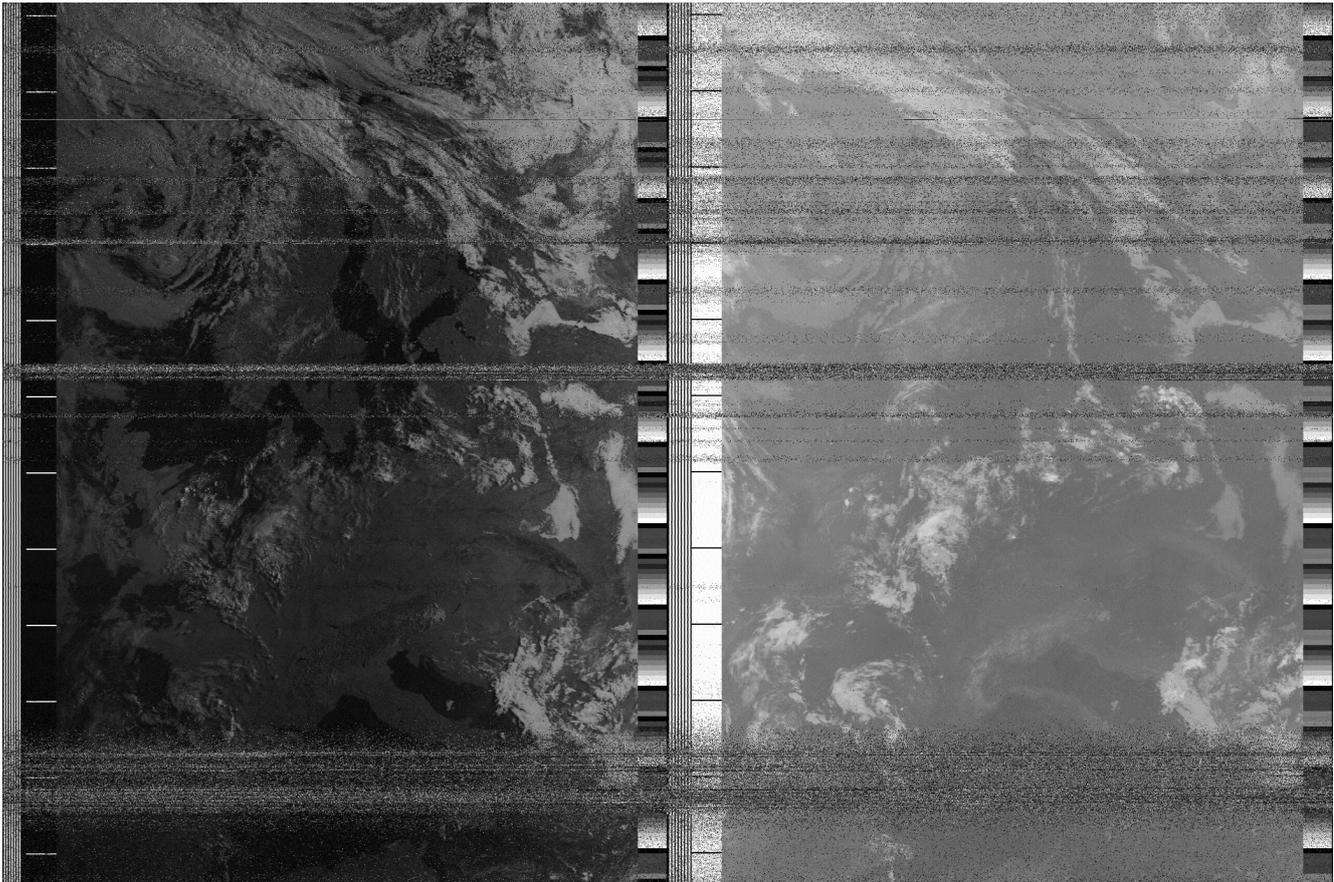


Figure 32: Image from NOAA-19 satellite, no georeferencing information. The Svarog project recorded over 3000 observations since its inception in Jan. 2020. This particular image was chosen because of a reasonably clear weather and good reception quality. The land contours are reasonably clearly seen. This will simplify the visual verification of the borders overlay.

Ellipsoid reference	Semi-major axis a	Semi-minor axis b	Inverse flattening $1/f$
WGS 72	6 378 135.0 m	\approx 6 356 750.520000 m	298.260000000
GRS 80	6 378 137.0 m	\approx 6 356 752.314140 m	298.257222100
WGS 84	6 378 137.0 m	\approx 6 356 752.314245 m	298.257223563

Table 2: Shape of Earth as defined in popular Geodetic Systems

The first step is to calculate the satellite position at AOS and LOS. This can be done using SGP4 models (see Section 2.3.5) and TLE orbital parameters (see Section 2.5.3). It is worth noting that the orbital parameters are not constant and evolve slowly over time. As a practical matter, it is essential to keep TLE parameters from the time when an observation was recorded. The SGP4 models take TLE parameters, and a timestamp expressed as Julian Date (see Section 1.5.1) and produce results in a Cartesian position and velocity versus Time Since TLE Epoch in the True Equator, Mean Equinox (TEME) coordinate system (see Section 1.5.2). Those TEME coordinates can be converted to Earth-Centered, Earth-Fixed (ECEF) reference frame. However, this needs to consider the time as one frame of reference is fixed, and the other is rotating. Depending on the Earth motion model, this may range from reasonably simple if only circular rotation is taken into account to surprisingly difficult if precession and nutation are also considered. The implementation author used (see Section 2.3.5 for details) is based on AIAA paper [2].

4.1.2. Converting TEME to Geodetic coordinates

Due to centrifugal forces, Earth is not perfectly round and is slightly bulged at the equator and flattened at the poles. An exaggerated view of the Earth cross-session is presented in Fig. 33. The Earth curvature is defined by two radii measured at different places: a and b . The proper term for a is a semi-major axis, but it is simply an equatorial radius. Conversely, the proper term for b is a semi-minor axis, but it is a radius measured at the poles. Polar radius, b , and equatorial radius a are not equal and are bound by a flattening parameter f .

$$b = a(1 - f) \quad (4.2)$$

As a practical convenience, the f value is a small fraction, so the value is often expressed as $1/f$ and is referred to as inverse flattening. Many geodetic systems define a , b and $1/f$ values. More popular ones are shown in Table 2. Surprisingly enough, the TLE format and the SGP4 models are based on WGS 72 ellipsoid reference, as the US Department of Defense used this model. For details, see [89]. Note the difference between WGS 72 and WGS 84 is only 1.8 meters, which is acceptable for most applications, except GNSS.

The curvature introduces a new phenomenon, which may not be obvious. For a spherical Earth, the local zenith (defined as right angle from the local horizon, or more colloquially "straight up"), the observer and Earth center always form a straight line. That is not the case on oblate Earth. The distinction between the angle between the equator and observer ϕ' and the angle between the equator and local zenith ϕ is an essential aspect of the coordinates conversion algorithm.

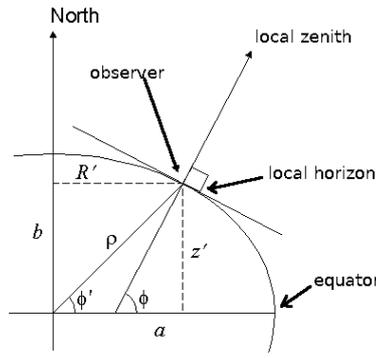


Figure 33: Oblate Earth. The Earth has a bulge around its equator. Image based roughly on [29].

Assuming the ECI position of the satellite to be $[x, y, z]$, the latitude on spherical Earth is

$$\phi' = \tan^{-1} \left[\frac{z}{\sqrt{x^2 + y^2}} \right] \quad (4.3)$$

and the longitude is:

$$\lambda_E = \tan^{-1} \left[\frac{y}{x} \right] - \Theta_g \quad (4.4)$$

where Θ_g is the Greenwich Mean Sidereal Time (GMST) (*pol.* średni czas gwiazdowy Greenwich), expressed in radians. In a general case, the satellite altitude would be defined by

$$h = \sqrt{x^2 + y^2 + z^2} - R_e \quad (4.5)$$

where R_e is the Earth radius. However, with the georeferencing problem at hand, we are mostly concerned with the on-surface projection and the satellite altitude is not a concern.

For an oblate Earth, the calculation is more complex and requires several iterations. To calculate the geodetic latitude of the sub-satellite point (*pol.* punkt podsatelitarny), the algorithm starts with an approximation of ϕ with ϕ' as calculated for round Earth and then goes through a series of iterations until the error is within the desired tolerance. The iteration is as follows:

$$\phi_i = \phi \quad (4.6)$$

$$C = \frac{1}{\sqrt{1 - e^2 \cdot \sin^2(\phi_i)}} \quad (4.7)$$

$$\phi = \tan^{-1} \left[\frac{z + aCe^2 \cdot \sin(\phi_i)}{R} \right] \quad (4.8)$$

and the approximation error is $|\phi - \phi_i|$ and R is a distance of the satellite from the Earth axis of rotation and can be calculated using the $R = \sqrt{x^2 + y^2}$ formula. It is worth noting that the algorithm converges very quickly. T.S. Kelso [29] gives an example of Mir station calculations. The iterations gave the following errors (in degrees): first 0.180537, second 0.000574, and third 0.000002.

4.1.3. Georeferencing wide images

One significant problem with NOAA satellite images is that they cover a wide area, which are a significant portion of the sphere. The image swath is roughly 2900km and the length depends on the reception quality, but it is many thousands kilometers. This means that the projection of geodetic coordinates to x, y coordinates of the

image cannot assume meridians or parallels to actually be parallel on the image. Given two coordinates of ϕ_1, λ_1 and ϕ_2, λ_2 , the distance $dist$ and azimuth az between them can be calculated using the following equations:

$$az = \tan^{-1} \left[\frac{\sin(\Delta\lambda)}{\cos(\phi_1) \cdot \tan(\phi_2) - \sin(\phi_2) \cdot \cos(\Delta\lambda)} \right] \quad (4.9)$$

where $\Delta\lambda = \lambda_2 - \lambda_1$. Similarly, the distance can be calculated using the following formula:

$$dist = \cos^{-1} [\sin(\phi_1) \cdot \sin(\phi_2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\Delta\phi)] \quad (4.10)$$

Note the resulting azimuth and distance are both expressed in radians.

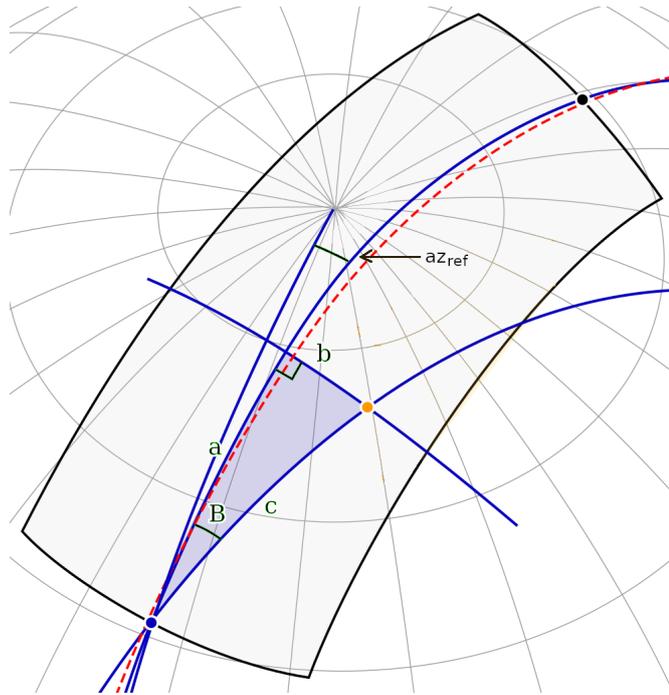


Figure 34: Projecting geodetic coordinates on the satellite image.

The preliminary step requires to calculate the satellite azimuth. This is calculated using subsatellite points for AOS and LOS, and converting to $\phi_{AOS}, \lambda_{AOS}$ and $\phi_{LOS}, \lambda_{LOS}$ as described in the earlier sections. Then the satellite azimuth can be calculated using the equations 4.9 and 4.10. This is the reference azimuth that corresponds to the column of pixels in the middle of the image. To calculate pixel position on the image, one needs to convert the ϕ, λ coordinates using the following steps:

1. calculate reference azimuth az_{ref} using $\phi_{AOS}, \lambda_{AOS}$ and $\phi_{LOS}, \lambda_{LOS}$
2. calculate azimuth az between AOS and the point being converted
3. calculate distance c between AOS and the point being converted
4. calculate $B = az - az_{ref}$
5. calculate a using $a = \tan^{-1} (\cos(B) \cdot \tan(c))$
6. calculate b using $b = \sin^{-1} (\sin(B) \cdot \sin(c))$
7. calculate horizontal pixel position using $x = -b/x_{res}$
8. calculate vertical pixel position using $y = a/y_{res}$

where x_{res} and y_{res} are the image dimensions, expressed in pixels. Finally, the NOAA images are actually two images with additional control/sync data on the sides, so an extra offset needs to be calculated separately for each image. This process is demonstrated in Fig. 34. This particular algorithm is roughly based on [53].

Using the algorithm above, any geodetic coordinates can be converted to x, y coordinates of the image. An example result is shown in Fig. 35. As proof of concept, three types of data were overlaid. First, a set of country borders were shown in yellow. The overlay map uses SHP files from The Natural Earth service [50] that were simplified using MapShaper service [35]. Second, a geodetic grid was overlaid to show better the Earth curvature (and demonstrate the inappropriateness of rectangular projection in the process). Finally, the third type of data was the ground station location, marked as a red dot.

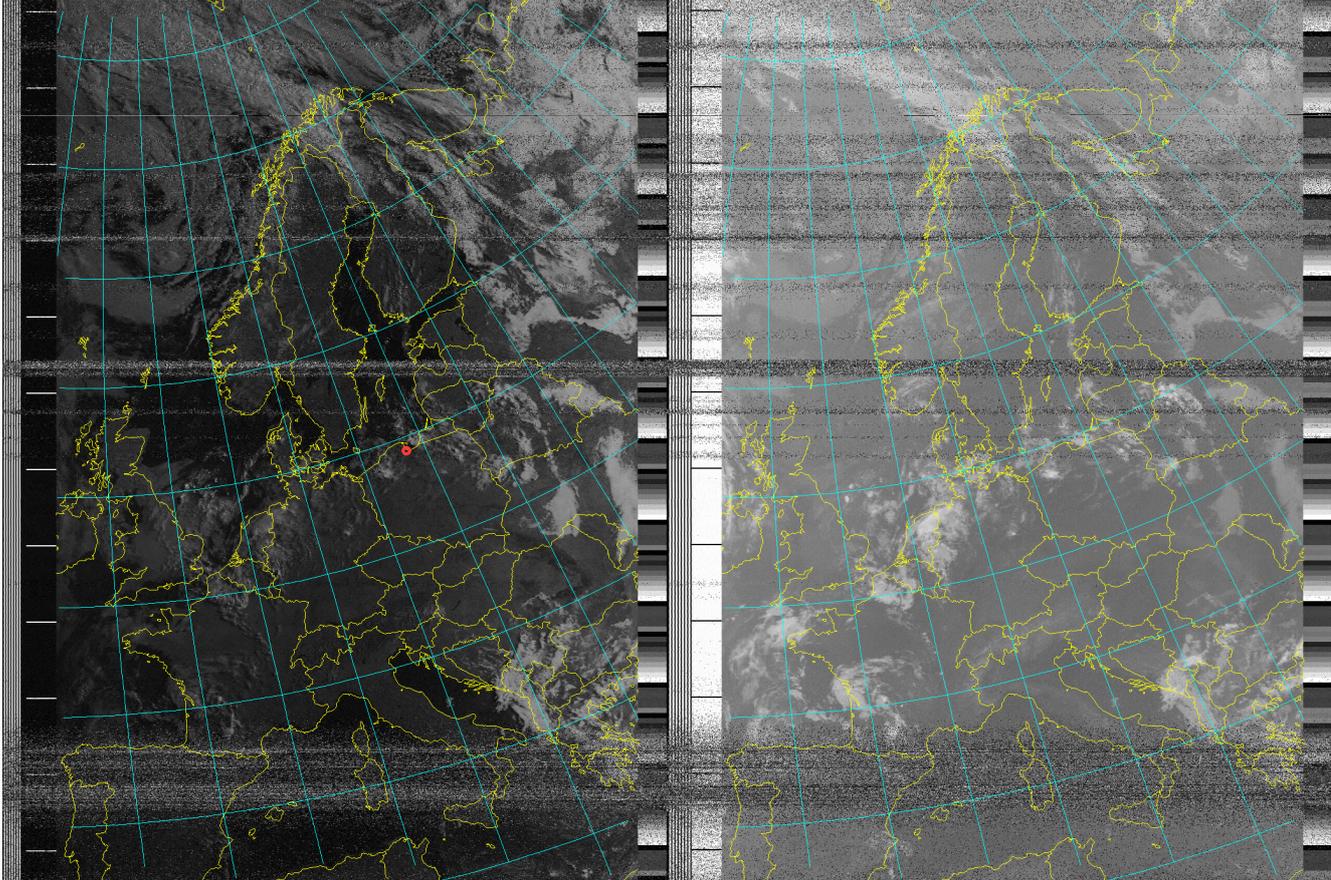


Figure 35: Georeferenced image from NOAA-19 satellite, data acquired in Svarog project. The country contours (yellow) make the image much easier to recognize, especially in cases of high cloud cover or faint contrast (right image). The geodetic grid (cyan) clearly shows the image distortion. Ground station location is shown in red.

Note: Research Idea

The author has implemented three algorithms of converting TEME to geodetic coordinate and two ways of converting them to x, y coordinates of the NOAA images. There are several alternative methods for calculating GMST. All the alternatives can be compared from the point of their impact on final georeferencing precision. The precision can be assessed easily. Several different geoids (e.g., WGS-72 and WGS-84) can be used and assessed whether it makes any difference at that scale. The reference can be obtained by selecting characteristic points, e.g., Hel peninsula in Poland, Gibraltar, the tip of the Italy "heel" with well-known coordinates, translate them to the image, and then calculate relative and absolute errors. This may be a potential topic for a research paper.

4.2. Problem 2: Reviving or deorbiting old satellites

Many older satellites still have fully functional electronic and optical systems, but they can no longer conduct their primary missions due to running out of fuel. A recent idea that seems to be particularly popular in the NewSpace community is to come up with a revival sat – a satellite that would rendezvous with existing 'dead' satellite, dock with it and then serve as a permanent new engine. An alternative version of such a mission could attach to a dead satellite and conduct its deorbit. Cleaning up orbital debris is an activity that is endorsed by NASA and ESA. It is also something that can potentially mitigate the threat of Kessler syndrome (see Section 1.9.4).

Designing such a mission would require several orbital maneuvers: matching orbital plane, synchronizing apogee and perigee, and then performing approach and rendezvous maneuvers. However, before any orbital mechanics can be applied, the first preliminary step is to evaluate available launch services. The choice of a rocket typically also determines the launch location, which determines the initial orbit parameters.

4.2.1. Available Launch Services

A launch vehicle selection is the most critical decision point during the mission design phase and has the most significant budgetary implications. Many organizations offer or soon be offering launch services, catering for the CubeSat business. A list of current Polish satellites and their launch vehicles and other potential alternatives has been presented in Table 3.

As of Q4 2020, the launch services available or expected to be available are also presented in Table 3. The total cost of launching a satellite has a very complex structure, and it is never as simple as price per mass. Other aspects that are often neglected are: who is doing the payload integration, what kind of services are provided for visiting teams (power and lab space only, clean room, communication, thermal control, other lab equipment, etc.), whether the payload is integrated early or late (some payloads, such as biological experiments require as late integration as possible). Many services catering to the CubeSat industry provide the standard offering, expressed in U (units). One U is a cube of 10 cm by 10 cm by 11 cm. Standard sizes are 1U, 2U, 3U, 6U, and 12U.

In most cases, the launch cost is confidential. Only very few companies publish the price tag. In some cases, the actual price can be deducted from project reports that are often funded by research grants that have public financial reporting requirements.

In the case of Światowid, the six mln PLN was quoted by [38] as a total cost to develop and launch a satellite. The author incorrectly claims that Światowid is the first Polish satellite, which makes him a less reliable source.

The RocketLab company, a relatively young but successful launch provider from New Zealand, offers ride-share missions. So far, they were able to put 55 satellites in orbit successfully. Launch Complex 1 (LC1) offers inclinations of 39° to 120°. The [58] says that the typical ride-share mission (a type of a mission where several smaller satellites share a common launcher to decrease its cost) is 500km Sun-synchronous orbit.

Spaceflight service is a reliable broker agency that uses multiple launchers [72] and offer routine launch services. It was created in 2010, and since its inception, it launched over 200 satellites, using Antares, Dnepr, Soyuz, Falcon 9, PSLV, Minotaur, Electron rockets, and also deployed satellites from ISS. As of the time of writing this thesis (Q4 2020), they offer launch services for CubeSats for each quarter of 2021. The price is strongly dependent on the size of the CubeSat. The dependency is almost linear with the factor of 50k USD per 1U. For example, a 1U CubeSat costs 50k USD to launch, while the 3U costs 145k to 150k USD. Every ride-share mission needs to have some form of the deployment mechanism, which will dispense each satellite separately. The cost includes a dispenser. Before and during the flight, it will provide an interface between the CubeSat and the launch vehicle, protect the CubeSat during the launch's stress, and fulfill its final task of releasing the satellite into space at the appropriate time. The CubeSat designs and dispensers are described in [46].

VELOX-1 is the first CubeSat designed and launched by the Nanyang Technological University of Singapore [81]. Its launch cost was quoted on an on-line forum [76] as 140k EUR. The validity of this number is unknown.

Satellite	Launch vehicle	Provider	Launch site	Orbit [km x km, inclination, NORAD ID]	Size, Price	Comments
PW-Sat	Vega	ESA	ELA-1, French Guiana	300x1023, 69°, [38083]	1U	first Polish sat
PW-Sat2	Falcon 9	SpaceX	Vandenberg, SLC-4E, USA	510x525, 94°, [43814]	2U	
Brite-PL (Lem)	Dnepr	ISC Kosmotrans (Ukraine)	Yasny LC-13, Russia	598x886, 97.8°, [39431]	8U	
Brite-PL (Hewelius)	Chang Zheng 4B	Shanghai Academy of Spaceflight Tech	Taiyuan, China	609x635, 97.9°, [40119]	8U	
Światowid	Antares 230	Orbital Science Corp	Wallops, USA	377x379, 51.6°, [44426]	1U, 6 mln PLN	first Polish commercial sat
KRAK'Sat	Antares 230	Orbital Science Corp	Wallops, USA	400x400, 51.6°, [44427]	1U	launch planned for 2022
WROSat	1U?	TBD	TBD	TBD	-	
VELOX-1	PSLV-C23	ISRO	...	634x651, 98.3°, [40057]	140k EUR	Singapore university [81]
-	various	Spaceflight.com	US(multi-provider)		50k USD/1U	
-	Electron	RocketLab	Launch Complex 1 (NZ)		77k USD/1U, 240k USD/3U	
-	Alpha	FireFly	USA		1000kg, 15mln USD (est.)	expected in 2021
-	Starship	SpaceX	USA		100tons, 2mln USD	fully reusable, est. 2022

Table 3: Existing and prospective orbital launch services. The first section lists Polish satellites along with the launch providers used. Second section lists an example university cubesat. The third section contains a list of low cost launch providers.

4.2.2. LEO Rendezvous Mission Design

This section outlines a sketch of a CubeSat mission that aims to dock with an existing LEO satellite that is decommissioned for some reason, such as running out of fuel, but is otherwise still functional. Another potential reason for servicing mission could be a hardware failure that requires a relatively small replacement part. An example of a satellite that experienced such a failure is NOAA-17. Its primary observational instrument, AVHRR, performed well since its launch in 2003 until one of its sub-components – a motor rotating a mirror – failed. Overall NOAA-17 status has been presented in Fig. 36. For the sake of this section discussion, it is assumed that a small CubeSat class recovery satellite has been built and is awaiting its launch. Such a satellite’s goal would be to fly to its target, perform rendezvous maneuvers, dock with, and then attempt to conduct the primary mission while attached to the target (start deorbiting procedure, act as an external engine, perform the servicing operation, etc.).

NOAA 17 AVHRR Subsystem Summary

Component	Description	Status
BASEPLATE	AVHRR baseplate	Green
CHANNEL 1	Spectral channel #1, 0.58-0.68 micro meters.	Green
CHANNEL 2	Spectral channel #2, 0.725-1.0 micro meters.	Green
CHANNEL 3A	Spectral channel #3A, 1.58-1.64 micro meters.	Green
CHANNEL 3B	Spectral channel #3B, 3.55-3.93 micro meters.	Green
CHANNEL 4	Spectral channel #4, 10.3-11.3 micro meters.	Green
CHANNEL 5	Spectral channel #5, 11.5-12.5 micro meters.	Green
ELECTRONICS	AVHRR Electronics	Green
OPTICS	AVHRR Optical mirrors and telescopes	Green
PSU	Power Supply Unit	Green
RADIANT COOLER	AVHRR Radiant Cooler	Green
SCAN MOTOR	AVHRR Scan Motor	Red

Figure 36: The NOAA-17 satellite has all systems fully operational, except the scan motor, which makes all the other systems unusable. Image taken from [52]

The orbital parameters of the target can be retrieved from N2YO service [43]. Note the orbital parameters drift over time, so the following calculations are correct only for the period when the orbital parameters were retrieved (Oct. 2020). One of the most affordable launch services as of today seems to be RocketLab. While each launch deploys many CubeSats and each has slightly different orbital parameters, the launch provider advertises 500km SSO orbit as the reference, which seems popular among its customers. For those calculations, an ANDESITE satellite has been selected as a model. It is a CubeSat built by Boston University, launched on June 14th, 2020 by RocketLab. Its orbital parameters will be considered as an initial orbit. This CubeSat will serve as a mockup for a **GDASAT-1** mission proposal.

The TLE parameters of the initial orbit are as follows:

```
Initial orbit: GDASAT-1, based on ANDESITE (norad id 45726)
1 45726U 20037D 20278.45278018 .00000608 00000-0 65390-4 0 9991
2 45726 97.7132 96.2906 0012962 283.6573 76.3213 14.92011802 14275
```

The TLE parameters of the target orbit are as follows:

```
Rendezvous target: NOAA-17 (norad id 27453)
1 27453U 02032A 20263.80942421 -.00000011 +00000-0 +13583-4 0 9998
2 27453 098.5909 208.3215 0011096 327.5463 032.5033 14.25072668948324
```

The goal of this exercise is to plan a series of maneuvers that will put one satellite close enough to its target so the Rendezvous and Proximity Operations (RPO) would be possible. For an example of a CubeSat capable of performing RPO, see [8].

4.2.2.1. Mission analysis

The first step in mission assessment is to convert TLE data into more understandable Keplerian parameters. This can be done using *Perylune* software, using `load_tle` and `print_orb` commands (see Section 3.5). The initial orbit has the following parameters:

```
6961 x 6979 km x 97.7 deg (GCRS) orbit around Earth (♁) at epoch 2020-10-04T10:52:00.207552000 (UTC)
a( $\alpha$ )=6969.80km, b=6969.79km, e=0.00, i=97.71deg raan( $\Omega$ )=96.29deg argp( $\omega$ )=283.66deg nu( $\nu$ )=76.47deg
period=5790.84s periapsis=6961km(582.63km) apoapsis=6979km(600.70km)
```

and the target orbit:

```
7178 x 7194 km x 98.6 deg (GCRS) orbit around Earth (♁) at epoch 2020-09-19T19:25:34.251744000 (UTC)
a( $\alpha$ )=7186.39km, b=7186.38km, e=0.00, i=98.59deg raan( $\Omega$ )=208.32deg argp( $\omega$ )=327.55deg nu( $\nu$ )=32.57deg
period=6062.85s periapsis=7178km(800.27km) apoapsis=7194km(816.22km)
```

As seen, both are LEO orbits, but there are notable differences. First, they have the different inclination and thus are not co-planar. Second, they have not equal RAAN, which also makes them not co-planar. Third, the apoapsis and periapsis are different. In both cases, the orbits are roughly circular (with the eccentricity being lower than 0.005 and thus showed as 0.00 when shown with two digits precision).

The differences in mean anomaly (ν) can be ignored, as it shows the current position and was determined at the time of TLE data generation. The argument of periapsis can also be neglected to some degree. For roughly circular orbits, it determines the location of periapsis and apoapsis in the orbital trajectory. If periapsis and apoapsis altitude vary little, the importance of ω decreases. The initial and target orbits are presented in Fig. 37.

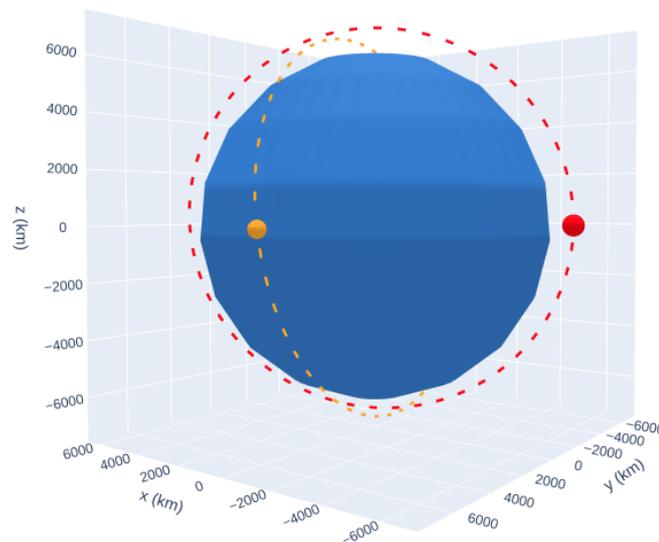


Figure 37: Visualization of the initial orbits: GDASAT-1 just after orbital insertion (orange), the target orbit of NOAA-17 (red). Image generated using *Perylune* software.

In a real-life scenario, an effort would have been made to time the rocket launch so that the orbital insertion done by the second stage would result in a RAAN close to that of the target's orbit. However, assuming a ride-share

launch, this may not always be possible. Nevertheless, the difference in RAAN is significant (112°).

Maneuver 1: RAAN synchronization. RAAN defines the angle between the intersection of the orbital and equatorial planes, as discussed in Section 1.4.1. A model available in [59] has been used to simulate the J2 perturbations. The implementation is based on [15]. In the first step, a time of flight of 24h was used to determine that the orbit precessed from RAAN= 96.29° to 97.27° in 24h, giving the rate of $0.9888^\circ/24\text{h}$. This was then used to simulate the period of 113 days, resulting in RAAN of 207.57° . The final propagation was done in a 10-minute loop that, after 110 iterations, reached the desired RAAN. The J2 perturbations are not completely free in the sense that they affect other orbital parameters as well. In particular, they degrade altitude slightly (7.3km for periapsis and 11.7km for apoapsis), but fortunately for this altitude, the degradation was small enough to be acceptable. This may not be viable for lower altitude orbits, however. After spending 113 days and 1100 minutes in space, the next orbit would be achieved:

```
6953 x 6967 km x 97.7 deg (GCRS) orbit around Earth (♁) at epoch 2021-01-26T05:12:00.207552000 (UTC)
a(α)=6960.24km, b=6960.24km, e=0.00, i=97.72deg raan(Ω)=208.32deg argp(ω)=282.47deg nu(ν)=-56.73deg
period=5778.93s periapsis=6953km(575.25km) apoapsis=6967km(588.96km)
```

Maneuver 2: Inclination change

The next step is to increase the inclination to the desired value of 98.6° . The Rocketlab usually launches their rideshare missions to roughly sun-synchronous orbits. Therefore the inclination corrections necessary is only 0.87° , which is a reasonably small value. Inclination change can be done only at the ascending or descending nodes (see discussion in Section 1.6.6.1). This was achieved by propagating the orbit to the nearest node, which happened to be the descending node. The orbital inclination change was calculated using `plane_change_manuever` function (see Section 3.5 for details). The maneuver required a modest Δv of 115.9 m/s. Expressed as vector, the burn required is [55.09, -100.80, 16.19] m/s. After the maneuver, the orbital parameters are as follows:

```
6955 x 6967 km x 98.6 deg (GCRS) orbit around Earth (♁) at epoch 2021-01-26T06:36:05.106362871 (UTC)
a(α)=6960.74km, b=6960.73km, e=0.00, i=98.60deg raan(Ω)=208.32deg argp(ω)=279.16deg nu(ν)=-99.16deg
period=5779.54s periapsis=6955km(576.41km) apoapsis=6967km(588.79km)
```

The major goal of syncing both RAAN and inclination has been achieved. This means that the orbits are co-planar, and it is now possible to use all co-planar maneuvers, in particular Hohmann transfer, apoapsis, and periapsis change. Another minor convenience is that it is now possible to visualize the orbits using 2D charts, as presented in Fig. 38.

Maneuver 3: Altitude synchronization As seen in Fig. 38, the spacecraft orbit has both periapsis and apoapsis lower than the target. As such, the Hohmann transfer orbit can be applied. This can be done using `hohmann` function from the *Perylune* software. Every Hohmann transfer consists of two burns. The first one being a departing burn, which puts the spacecraft on the elliptical trajectory. After the desired altitude has been reached, a second burn circularizes the orbit.

The Hohmann maneuvers calculated are as follows: maneuver 2 to be conducted after 1590 s with the burn vector to be [-49.52, -25.16, 8.85] m/s and maneuver 3 to be conducted after 2955 s with the burn vector [52.07, 26.46, -9.31] m/s.

After those two burns, the orbital parameters are:

```
7178 x 7178 km x 98.6 deg (GCRS) orbit around Earth (♁) at epoch 2021-01-26T07:51:51.156498878 (UTC)
a(α)=7178.00km, b=7178.00km, e=0.00, i=98.60deg raan(Ω)=208.32deg argp(ω)=0.00deg nu(ν)=99.16deg
period=6052.24s periapsis=7178km(799.86km) apoapsis=7178km(799.86km)
```

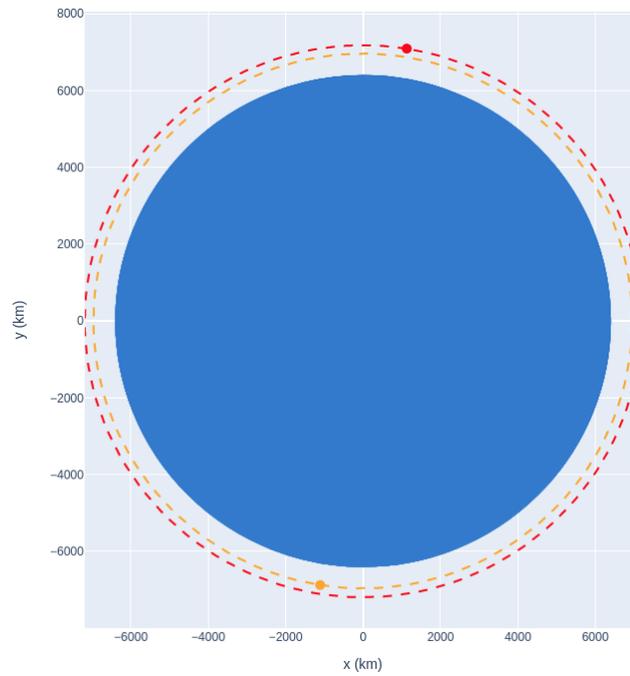


Figure 38: Visualization of the orbits after 2 maneuvers: J2 perturbations to sync RAAN and inclination change: GDASAT-1 (orange), the target orbit of NOAA-17 (red). Earth size and orbital altitude are to scale. Image generated using *Perylune* software.

No	Burn purpose	Δv [m/s]	Vector [m/s]
0	RAAN synchronization	0	varied (J2 perturbations)
1	Inclination change	115.9	[55.09, -100.80, 16.19]
2	Hohmann departure	56.3	[-49.52, -25.16, 8.85]
3	Hohmann arrival	59.1	[52.07, 26.46, -9.31]
4	Apoapsis raise	4.1	[-0.29, 0.55, 4.10]
Σ	Sum of all burns	235.5	

Table 4: Calculated burns needed to synchronize initial (GDASAT-1 launch) to target (NOAA-17) orbits.

Maneuver 4: Apoapsis synchronization

At this stage, the orbit is 800x800km, while the target is 800x816km. The remaining correction left is to raise the apoapsis by 16km. In general, this requires conducting a burn at the opposite apsis. However, since the orbit is circular, there is no designated apoapsis or periapsis of the departure orbit. When breaking the circular orbit, the moment of the burn initiation will be the point of decircularization and thus will set the argument of the periapsis. However, since the target orbit is nearly circular, this aspect has been neglected. Obtaining the argument of perigee consistent with the target orbit requires appropriate timing but does not require any extra Δv expenditure. The prograde maneuver has been calculated using `prograde_maneuver` function from *Perylune* software (see Section 3.5). The Δv required is 4.15 m/s and the burn vector is [-0.29, 0.55, 4.10] m/s.

4.2.2.2. Mission assessment

As discussed in previous sections, the four impulsive maneuvers needed (inclination change, two Hohmann transfers, and one apoapsis rise) with a total Δv necessary being 235.5 m/s. Burn details have been presented in table 4. This is a reasonably small requirement for an orbital propulsion system. To consider an actual rendezvous, additional small corrections will be needed for the approach and proximity operations.

The spacecraft orbit after conducting all maneuvers is as follows:

7178 x 7194 km x 98.6 deg (GCRS) orbit around Earth (δ) at epoch 2021-01-26T09:04:56.363403879 (UTC)
 $a(\alpha)=7186.00\text{km}$, $b=7186.00\text{km}$, $e=0.00$, $i=98.60\text{deg}$ $\text{raan}(\Omega)=208.32\text{deg}$ $\text{argp}(\omega)=360.00\text{deg}$ $\text{nu}(\nu)=0.00\text{deg}$
 $\text{period}=6062.36\text{s}$ $\text{periapsis}=7178\text{km}(799.86\text{km})$ $\text{apoapsis}=7194\text{km}(815.86\text{km})$

For comparison, the target orbit was:

7178 x 7194 km x 98.6 deg (GCRS) orbit around Earth (δ) at epoch 2020-09-19T19:25:34.251744000 (UTC)
 $a(\alpha)=7186.39\text{km}$, $b=7186.38\text{km}$, $e=0.00$, $i=98.59\text{deg}$ $\text{raan}(\Omega)=208.32\text{deg}$ $\text{argp}(\omega)=327.55\text{deg}$ $\text{nu}(\nu)=32.57\text{deg}$
 $\text{period}=6062.85\text{s}$ $\text{periapsis}=7178\text{km}(800.27\text{km})$ $\text{apoapsis}=7194\text{km}(816.22\text{km})$

Since the orbits are nearly identical within a few hundred meters, the orbit synchronization can be considered a success.

4.3. Problem 3: Are we all going to die on April 29, 2020? – Debunking fake news

Every once in a while, news media publish alarming articles about upcoming close flybys of asteroids, with varying levels of inaccuracy and fake sensationalism. The most recent example was a series of publications in March and April 2020 about the upcoming 1998 OR-2 asteroid and its possible capability to kill all humanity. The sample article title is presented in Fig. 39.



Figure 39: Sensationalist article about upcoming asteroid fly-by and a question whether it will kill all humanity. The title says “NASA warns: An asteroid speeding towards Earth. Will end of the world happen on April 29th, 2020?”. The publication date is March 23rd, 2020, or roughly a month before the closest approach.

The following sections will explain how to check the validity of such claims and assess the potential danger. While this skill, in theory, may confirm the future threat, it is most likely be used for debunking fake news.

4.3.1. Obtaining ephemerids

There are only two input parameters needed to assess the potential danger: the asteroid's name and the supposed collision date. Following the example shown in Fig. 39, the name would be "1998 OR2" and the target date would be 2020-04-29. Those two parameters are enough to use NASA's Horizons database to assess the potential threat of the incoming asteroid hitting Earth.

NASA's Horizons system can generate many different types of data. First, the ephemeris type should be set to vectors, which will cause the resulting data will be $[X, Y, Z]$ vectors of a body position in 3D space. The second item – target body – is the object of study. Third, the resulting position should be expressed as a distance from Earth. This is convenient to do by using a geocentric coordinate system. Timespan is pretty apparent. It specifies three parameters: the beginning of a period, the end of a period, and the step interval. Table settings should, in general, expressed in km and km/s values, although other units, such as meters, AU (astronomical units), or even miles are available. Since the result will be further processed, the output should be set to plain text.

The screenshot shows the NASA Horizons Web-Interface. At the top, it identifies the Jet Propulsion Laboratory (JPL) and the Center for Near-Earth Object Studies. The main navigation bar includes links for JPL HOME, EARTH, SOLAR SYSTEM, STARS & GALAXIES, and TECHNOLOGY. Below this is a section titled "Solar System Dynamics" with sub-links for BODIES, ORBITS, EPHEMERIDES, TOOLS, PHYSICAL DATA, DISCOVERY, FAQ, and SITE MAP. The "EPHEMERIDES" link is highlighted. The main content area is titled "HORIZONS Web-Interface" and provides a brief description of the tool. Below this, the "Current Settings" are displayed: Ephemeris Type: **VECTORS**; Target Body: **52768 (1998 OR2)**; Coordinate Origin: **Geocentric [500]**; Time Span: Start=**2020-04-20**, Stop=**2020-05-10**, Step=**1 d**; Table Settings: output units=**KM-S**; Display/Output: **plain text**. A "Generate Ephemeris" button is visible. At the bottom, there are "Special Options" and a footer with navigation links, a logo for FIRSTGOV, the server date/time (2020-Oct-27 20:37 UT), and contact information for the Site Manager (Ryan S. Park) and Webmaster (Alan B. Chamberlin).

Figure 40: Web interface of the Horizons database hosted at SSD (Solar System Dynamics) at JPL (Jet Propulsion Laboratory). The Horizons database provides ephemerids for over a million asteroids, 3600 comets, 209 moons and minor planets, 8 planets and many other objects, such as barycenters, lagrangian points and more.

An excerpt from ephemerids generated for asteroid 1998 OR2 for 10 days before and after its closest approach on April 29, 2020 is shown below. Full ephemeris is presented in Appendix A.

```
*****
JPL/HORIZONS                    52768 (1998 OR2)                    2020-Oct-29 16:25:01

Target body name: 52768 (1998 OR2)                    {source: JPL#306}
Center body name: Earth (399)                          {source: DE431}
Center-site name: BODY CENTER

*****
Start time      : A.D. 2020-Apr-20 00:00:00.0000 TDB
Stop time       : A.D. 2020-May-10 00:00:00.0000 TDB
```

```

Step-size      : 1440 minutes
*****
Output units   : KM-S
Output type    : GEOMETRIC cartesian states
Output format  : 3 (position, velocity, LT, range, range-rate)
Reference frame : Ecliptic of J2000.0
*****
$$SOE
2458968.500000000 = A.D. 2020-Apr-29 00:00:00.0000 TDB
X =-5.529321863440186E+06 Y = 1.050912388547180E+06 Z =-2.826230911612379E+06
VX= 9.222124272322638E-01 VY=-7.790822567699109E+00 VZ=-3.744609072444963E+00
LT= 2.100802409169358E+01 RG= 6.298047180172035E+06 RR=-4.292682001593992E-01
<more entries omitted for clarity>
$$EOE
*****
Coordinate system description:
<omitted for clarity>

Symbol meaning:

    JD TDB    Julian Day Number, Barycentric Dynamical Time
    X         X-component of position vector (km)
    Y         Y-component of position vector (km)
    Z         Z-component of position vector (km)
    VX        X-component of velocity vector (km/sec)
    VY        Y-component of velocity vector (km/sec)
    VZ        Z-component of velocity vector (km/sec)
    LT        One-way down-leg Newtonian light-time (sec)
    RG        Range; distance from coordinate center (km)
    RR        Range-rate; radial velocity wrt coord. center (km/sec)
*****

```

The actual position is presented in lines after the `$$SOE` prefix. The most critical parameters are X, Y, Z values. In the example above, those are expressed in kilometers. The actual distance can be trivially calculated using an equation for a vector length in 3D space. It is also specified explicitly as RG . In principle, the distance is shown between asteroid's and Earth's centers. That value should be decreased by Earth radius to calculate the distance to the Earth's surface. However, such subtraction can be neglected because the distance is over 6 million km, and subtracting 6317 km does not make any difference. Fig. 41 shows the Earth – asteroid distance over 20 days.

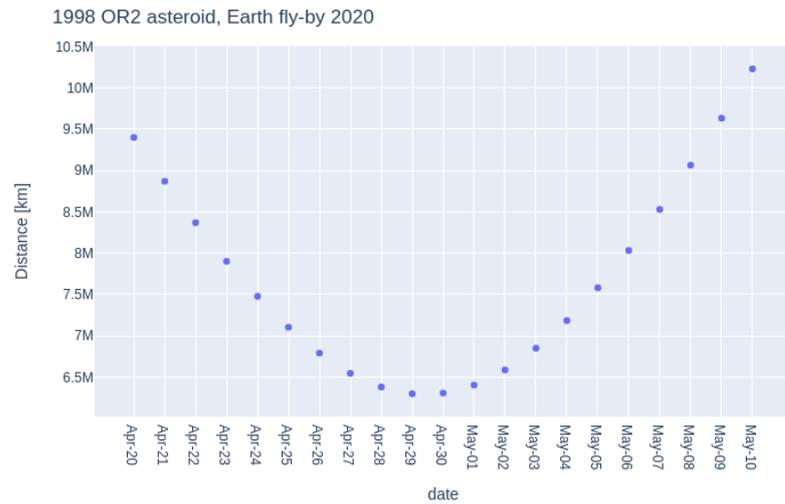


Figure 41: The distance between 1998 OR2 and Earth is 6.3 million kms from Earth at the closest approach. As such, it does not pose any danger.

Additionally, the situation can be visualized, as shown in Fig. 42. The 1998 OR2 asteroid position can be calculated using the ephemerids. For a better sense of scale, the Moon position is also shown. In this particular case, the simulation lasted 10 days, and as such, there is roughly 33% of the full Moon orbit. That is sufficient, however, to visually assess the distance. As expected, the asteroid passed in a safe distance and thus the article was debunked.

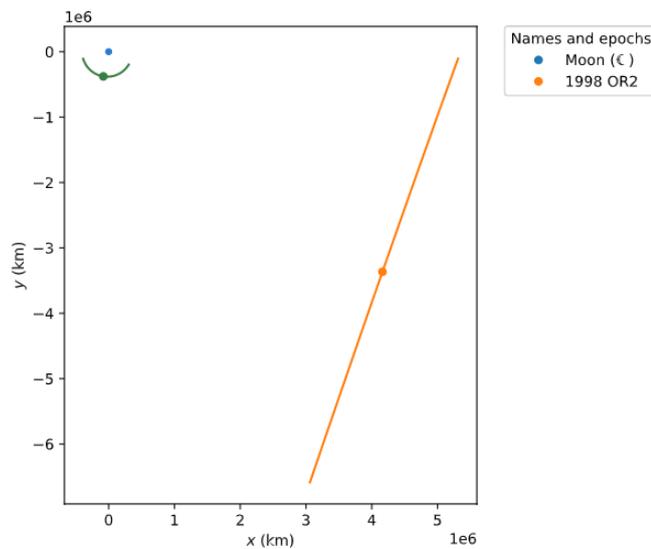


Figure 42: 1998 OR2 asteroid (orange) trajectory in the Earth vicinity (blue dot). The Moon's orbit (green line) is shown for scale. Asteroid's and Moon's trajectory plotted for period of 2020-04-20 to 2020-05-10.

Note: Reseach idea

This verification procedure can be reasonably easily turned into a web service that visualizes past or future asteroid fly-bys. With some effort, the user's input data would be minimal – just specify asteroid name and date of supposed close up.

4.4. Problem 4: Interplanetary Transfer Windows

The difficulty of reaching a place in a Solar system is not expressed as the distance but as the relative change of velocity. Due to the bodies being in constant movement, the difficulty changes over time. There are certain

Planet	Orbital period	Semi-major axis a [AU]	Eccentricity e	Inclination i [°]	RAAN Ω [°]	Argument of periapsis ω [°]
Mercury	88 d	0.3871	0.2056	7.005	48.332	29.19
Venus	243 d	0.7233	0.0068	3.39	76.681	55.12
Earth	365.2d	1.0004	0.0162	0.000	153.71	102.95
Mars	687d	1.5238	0.0934	1.848	49.50	335.04
Ceres	4.60y	2.7672	0.0777	10.588	80.28	73.72
Jupiter	11.87y	5.1967	0.0485	1.303	100.52	14.75
Saturn	29.65y	9.5788	0.0542	2.484	113.715	92.43
Uranus	84.08y	19.1926	0.0458	0.771	74.07	170.96
Neptune	166.28y	30.2379	0.0114	1.745	131.63	44.97
Pluto	248y	39.4739	0.2501	17.141	110.31	113.35
Haumea	283.6y	43.1654	0.1953	28.214	122.17	238.83
Makemake	306y	45.4128	0.1617	38.984	79.61	294.82
Eris	559.3y	67.8833	0.4357	44.022	35.96	191.50

Table 5: Known major and dwarf planets in the Solar System. Data calculated in Perylune, using ephemerids from NASA Horizons database. All parameters calculated for 2020-07-15.

planet positions where reaching one planet from another is easier. Such periods of favorable configurations are called a transfer window (*pol.* okno transferowe). The first step towards evaluating available windows is to look at the potential targets. The Solar system consists of 8 major planets, 5 known dwarf planets, well over a hundred moons, roughly a million known asteroids, and thousands of comets. They all can be potential destinations for a mission plan. For the sake of simplicity, the discussion will focus on planets, but the same principles can be used to investigate missions to other bodies.

The first step in determining potential destinations is to take a closer look at the targets' orbital elements. A list of major and dwarf planets with their main parameters are presented in Table 5.

As seen from the Table 5, the solar system is mostly flat with great majority of its bodies laying near the ecliptic (*pol.* ekliptyka) or more formally ecliptic plane (*pol.* płaszczyzna ekliptyki). The ecliptic is defined as a plane of the Earth's orbit. In this context, the inclination can be perceived as a metric of how close a planet is to the ecliptic. This means that the departure and arrival orbits will be mostly co-planar. Notable exceptions are Pluto, with its inclination of over 17°, and other dwarf planets. This, together with its odd, elongated elliptic orbit with an eccentricity of 0.25, was one reason why the International Astronomical Union decided to no longer consider it a planet [56] and define a new category of dwarf planets.

A similar or almost equal inclination means that the plane change burn will be minimal or neglected altogether from the mission analysis perspective. As such, the mission planning can use regular Hohmann transfer. In general, the high-level steps needed to reach an orbit around an interplanetary target are as follows:

Orbital insertion – this is the first step as with any other orbital launch. However, given the interplanetary nature of the mission, several additional factors should be taken into consideration. First, the alignment of planets should be favorable for Hohmann transfer. In principle, it is possible to achieve faster transfers than Hohmann. However, they are much more costly in terms of Δv and, in general, are not considered at the current state of propulsion development. The inclination should match that of the target body. Earth axis is currently tilted by 23°26' (the value is changing slowly between 22.1° and 24.5° over 41000 years). The desired equatorial inclination is greater than the elliptical inclination by the constant value of 23°26'.

Departure burn – This maneuver is conducted to achieve several goals. The first is to escape Earth's gravity. As such, the escape velocity has to be exceeded. Once the escape velocity is exceeded, the spacecraft will leave Earth's sphere of influence and can be considered flying on heliocentric orbit with parameters close to those of

Earth's. The second goal here is to initiate the Hohmann transfer between Earth and the target body. The third goal is to correct any inclination differences between the initial LEO orbit and the target. If the LEO orbit is elliptical, the burn should be done during periapsis, as the spacecraft has the highest velocity during its closest pass to the orbited body. Thus, the additional velocity needed to escape is smaller. Once the burn is complete, the spacecraft is in a curious state. From Earth's perspective, it is on the hyperbolic trajectory and will leave the Earth's system and its sphere of influence. From the Sun's perspective, it just entered a heliocentric orbit. Had the velocity matched and not exceeded escape velocity, the spacecraft's heliocentric orbit would be almost exactly equal to Earth's heliocentric velocity. However, since the goal is to reach a different planet, the additional component of the Hohmann departure burn must put the spacecraft on an elliptic orbit tangential to Earth and the target's orbit.

Insertion burn skip – Once the spacecraft reaches the target, the next step depends on the mission goals. If the goal is a fly-by only (such as New Horizons visiting Pluto), there is no need to conduct the second Hohmann maneuver. The obvious benefit of this approach is lower propulsion requirements. The drawback is that the spacecraft will spend minimal time in the vicinity of the target. Alternatively, it is possible to choose the Hohmann transfer orbit so that it will hit the planet surface or fly very low in the atmosphere conducting aerobreaking (*pol.* hamowanie aerodynamiczne). Such trajectories are called direct entry (*pol.* trajektoria bezpośredniego wejścia). This may be useful if the spacecraft has heat shields and the target planet has a sufficiently thick atmosphere (e.g., Mars 2020 mission) or the intention is to crash land the spacecraft (e.g., Deep Impact mission), either to excavate material or to retire a spacecraft without littering the orbit.

Insertion burn – Missions that intend to stay in the target body's system (e.g. many Mars orbiters) need to conduct a second Hohmann maneuver. Once the spacecraft reaches the apoapsis of its Hohmann transfer orbit, it will be in the target planet's vicinity. However, it will have its orbital velocity too low (when traveling outwards in the Solar system) or too high (when traveling inwards in the Solar system). The second Hohmann maneuver corrects that.

Table 6 shows the major characteristics of Hohmann transfers for major planets, dwarf planets and two selected asteroids.

4.4.1. Planet's Movement

Hohmann transfer ensures that the spacecraft will reach an orbit with certain parameters, presumably matching the targets. In particular, apoapsis and periapsis will match the target so that the spacecraft will be at the same distance from the Sun as the target. However, for the transfer to be successful, also the target has to be in the same place at the same time. In other words, it must have a very similar anomaly. This is ensured with the proper timing of the transfer. To understand how to choose the timing, a bit of an explanation is required. Kepler's third law dictates the orbital periods being strictly related to the semi-major axis of the orbit. In plain terms, the larger the orbit is, the slower the planets move, and it takes longer to complete the full circle. Depending on whether the intended travel is outwards (moving from a faster-moving planet – Earth – to another that is slower) or inwards (moving from a slower-moving planet), it is either the Earth "catching up" with the target or the target catches up with Earth. The moment of two planets being closest is called opposition (*pol.* opozycja) for outer planets, as they appear opposite (180° away) to the Sun when observed from Earth. A conjunction (*pol.* koniunkcja) is a period of particularly favorable conditions for astronomy: the planet is closest to Earth, visible almost all night and nearly completely sunlit (in full phase), but also provides advantages for space exploration. Due to the shortest physical distance, the radio transmission delay is smallest, the signal is stronger and thus often allows higher bandwidth rates.

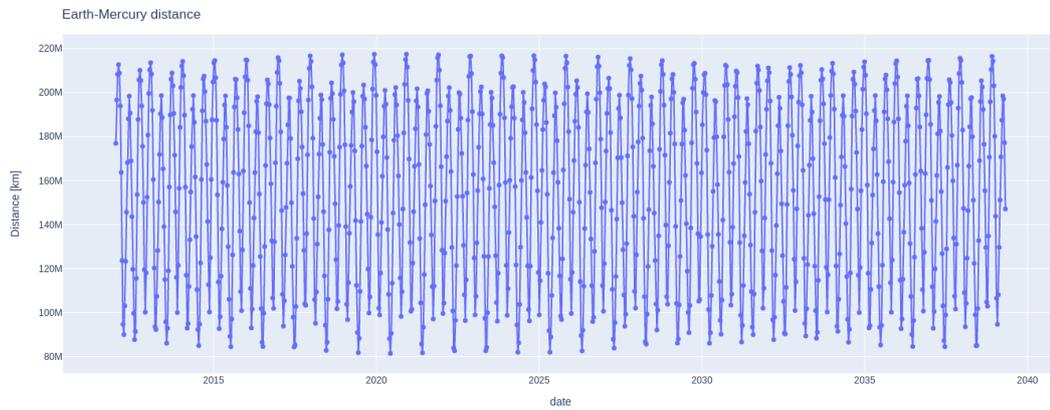
If planets were on perfectly circular and coplanar orbits, the conditions would repeat exactly on each opposition. However, due to the orbit's eccentricity and not being perfectly co-planar, the minimal distance between planets

Target	v_{h1} [km/s]	v_{hoh1} [km/s]	v_{h2} [km/s]	v_{hoh2} [km/s]	transfer time [days]	transfer time [years]	Δv_{hoh1} [km/s]	Δv_{hoh2} [km/s]
Mercury ☿	29.79	47.87	22.25	57.48	105.47	–	7.532	9.610
Venus ♀	29.79	35.02	27.29	37.73	146.06	–	2.494	2.705
Mars ♂	29.79	24.13	32.73	21.48	258.83	–	2.946	2.650
Jupiter ♃	29.79	13.06	38.58	7.41	–	2.732	8.794	5.644
Saturn ♄	29.79	9.62	40.09	4.18	–	6.083	10.298	5.439
Uranus ♅	29.79	6.80	41.07	2.14	–	16.055	11.283	4.658
Neptune ♆	29.79	5.42	41.45	1.37	–	30.867	11.659	4.046
Ceres	29.79	17.91	36.10	13.05	–	1.292	6.316	4.859
Eris	29.79	3.61	41.82	0.62	–	101.156	12.031	2.998
Pluto ♇	29.79	4.74	41.60	1.05	–	45.449	11.814	3.688
Makemake	29.79	4.42	41.67	0.92	–	55.836	11.882	3.503
Haumea	29.79	4.54	41.65	0.97	–	51.820	11.858	3.570
(4) Vesta	29.79	19.38	35.31	14.95	398.02	1.090	5.523	4.433
(5305) Berniolz	29.79	19.07	35.48	14.54	411.86	1.128	5.691	4.529

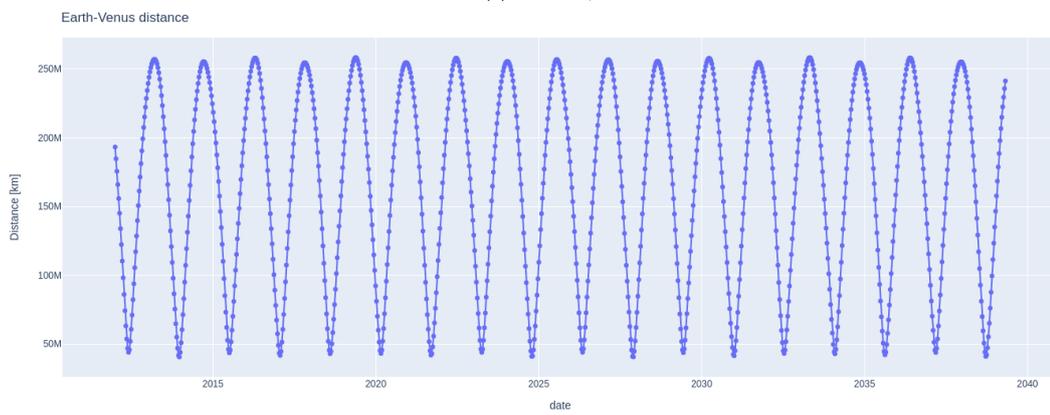
Table 6: Hohmann transfers to all major and dwarf planets and two example asteroids. All departures from Earth. v_{h1} is heliocentric velocity of the departing planet (Earth, before first burn). v_{hoh1} is a velocity required to enter transfer orbit (after first burn). v_{h2} is heliocentric velocity of the target body (velocity after second burn). v_{hoh2} is the arrival velocity of the transfer orbit (before second burn). Δv_{hoh1} and Δv_{hoh2} are Δv requirements for first and second Hohmann burns. All values calculated for average Sun distance of the target.

varies. Distance between Earth and Mercury, Venus, Mars, Jupiter, Saturn, Uranus, and Pluto are shown in Fig. 43 and 44. All charts were generated using Perylune software. Various phenomena can be observed on those charts. Mercury, being innermost and fastest planet on a inclined ($i = 7^\circ$) and elliptical ($e = 0.20$) orbit, has the closest and farthest maxima very uneven. Being the inner planet, Venus moves faster. Thus, the period of closest approaches is particularly short, which is shown as sharp minima. Also, Venus's orbital period is the shortest among the planets discussed here, clearly seen as the highest repeating period. Finally, both Venus and Earth orbits have low eccentricity, so oppositions are very similar.

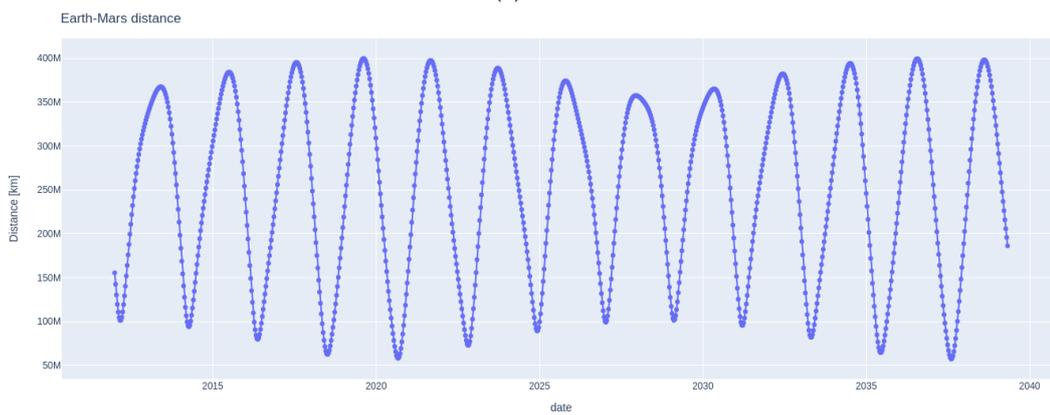
Mars has probably the most interesting properties. Its orbit is slightly inclined ($i = 1.8^\circ$), but more importantly, its orbit is much more elliptic than other planets (except Mercury and Pluto). Planets and other objects moving on elliptical orbits change their velocity depending on where they are in relation to apoapsis (slowest) and periapsis (fastest). This is clearly seen on the charts. Another observation is that the Mars opposition of 2020 was particularly close, and such good conditions will not repeat until 2035. Jupiter and Saturn have a similar characteristic. Being much farther from the Sun, they are moving slowly, and it is their distance from the Sun being a dominant factor. The orbital period of Jupiter (12 years) and Saturn (29 years) can be seen as a long time trend. Uranus has an even larger period and is currently moving towards the Sun. As mentioned earlier, Pluto is in a highly inclined ($i = 17.16^\circ$) and elliptical ($e = 0.29$) orbit with a huge period (247 years) and is currently moving away from the Sun and all inner planets quickly. This was one of the reasons for NASA to approve the New Horizons mission. One of the arguments in favor was that if NASA does not approve the Pluto mission, it will soon be too far away to be investigated. Several arguments were raised. First, the mission must be shorter than an average engineer career, so engineers that started it will still not retire when the mission reaches its target. Secondly, the long time in a harsh environment of space increases the likelihood of the spacecraft's catastrophic failure before its mission can be completed. Thirdly, with increasing distance, radio communication becomes more difficult. While the electronic equipment sensitivity improves over time as the technology advances, some fundamental physics laws (energy of EM wave decreases with the square of a distance) will not change.



(a) Mercury

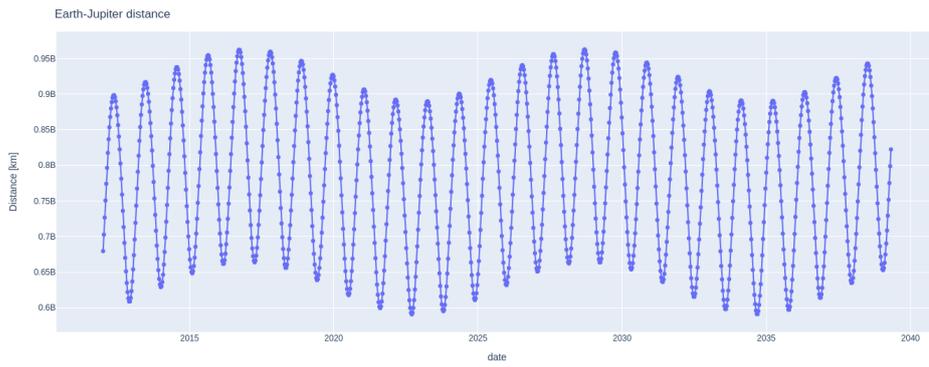


(b) Venus

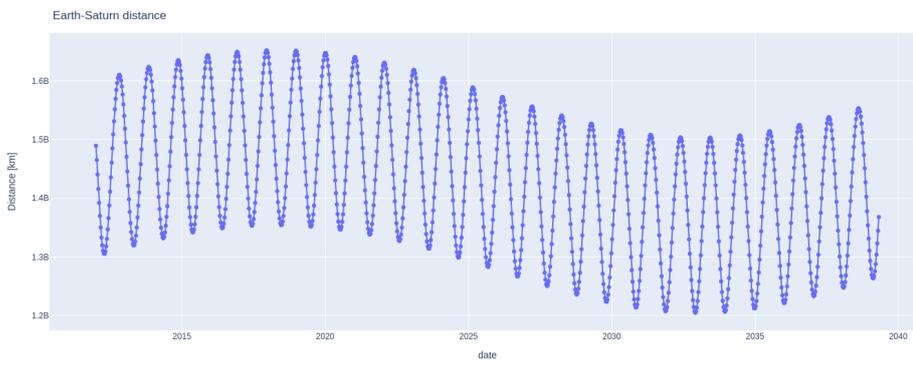


(c) Mars

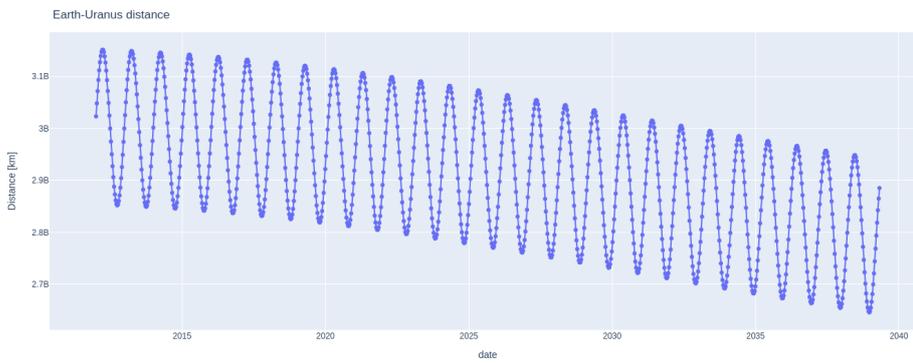
Figure 43: Mercury, Venus, and Mars distance from Earth for years 2012-2040



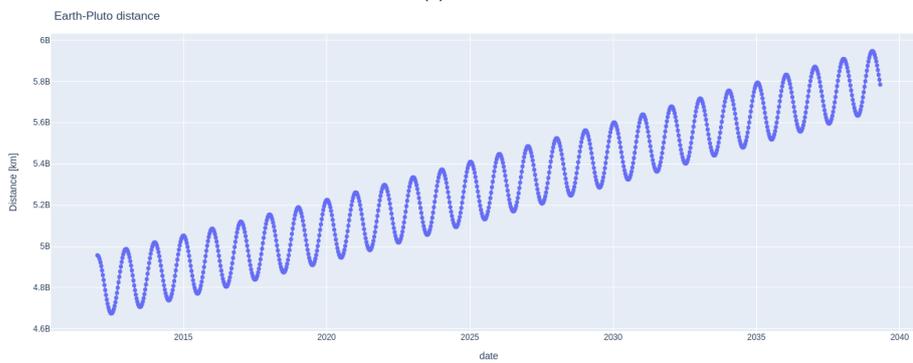
(a) Jupiter



(b) Saturn



(c) Uranus



(d) Pluto

Figure 44: Jupiter, Saturn, Uranus, Pluto distance from Earth for years 2012-2040

4.4.2. Launch Windows

Earlier sections discussed a high-level approach to the evaluation of transfers to various bodies. This is sufficient for the early stages of mission consideration. However, once a more specific target is chosen, a closer consideration is necessary to plan specific launch windows that use the much higher temporal resolution and considers both departure and arrival times. Such an analysis is often presented in the form of so-called porkchop plots, as it looks somewhat like a chopped leg of pork. The chart shows the relation between specific departure and arrival times and the Δv requirements. During the time of writing this thesis (2020), the optimal Mars transfer window just passed. Three spacecraft were launched during this window: American Mars 2020, China's Tianwen-1 and United Arab Emirates' Hope. The 2020 Mars launch window is presented in Fig. 45. A launch window to Venus has been presented in Fig. 46. Sadly, no spacecraft took advantage of it this time, but there are plans to send a private mission to Venus in 2023 by RocketLab.

4.4.3. Hohmann transfer example

Once departure and arrival dates are known, it is possible to plot the actual trajectory. During the time of writing this thesis, the Mars 2020 mission was en route to Mars. NASA published departure time as 2020-07-20 11:50 UTC and expected the arrival time to be on 18 Feb. 2020 [47]. The exact time is not specified, so 12:00 was assumed. With the departure and arrival dates being known, the actual trajectory can be determined by solving Lambert's problem (see Section 1.9.2). The calculated trajectory is presented in Fig. 47. This calculation does not take into consideration the minimal correction burns that are conducted during the flight. Orbital parameters for the transfer trajectory are as follows:

1.33 x 1.65 AU x 24.0 deg (HCRS) orbit around Sun (\odot) at epoch 2020-07-15 11:51:09.184 (TDB)
 $a(\alpha)=1.3303\text{AU}$, $b=1.2906\text{AU}$, $e=0.24$, $i=24.01\text{deg}$ $\text{raan}(\Omega)=356.87\text{deg}$ $\text{argp}(\omega)=298.85\text{deg}$ $\text{nu}(\nu)=-2.94\text{deg}$
 $\text{period}=560.45\text{d}$ $\text{periapsis}=1.0078\text{AU}$ $\text{apoapsis}=1.6529\text{AU}$

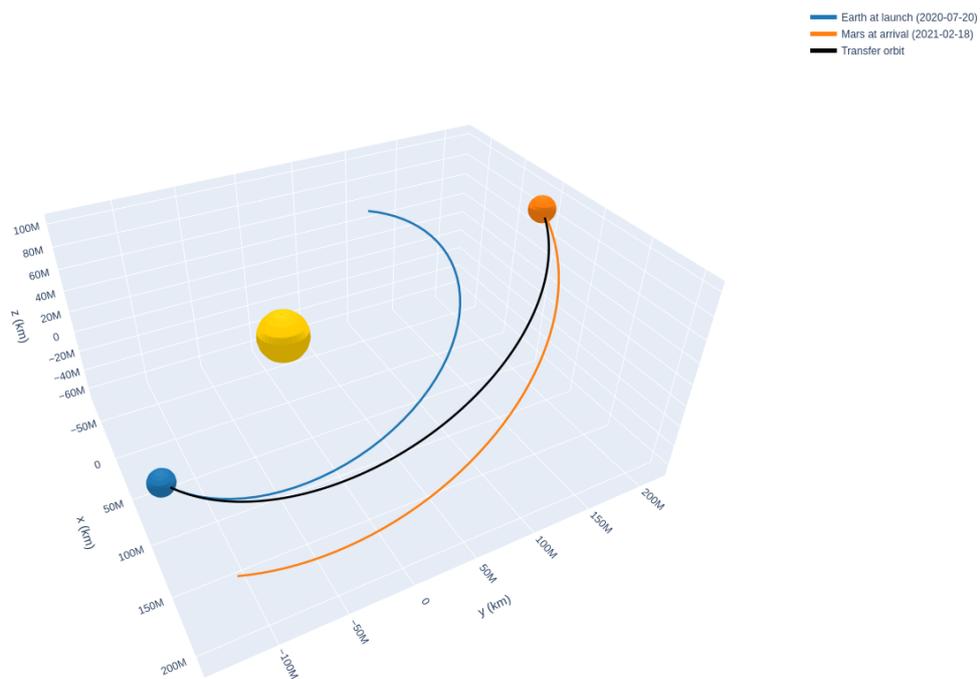
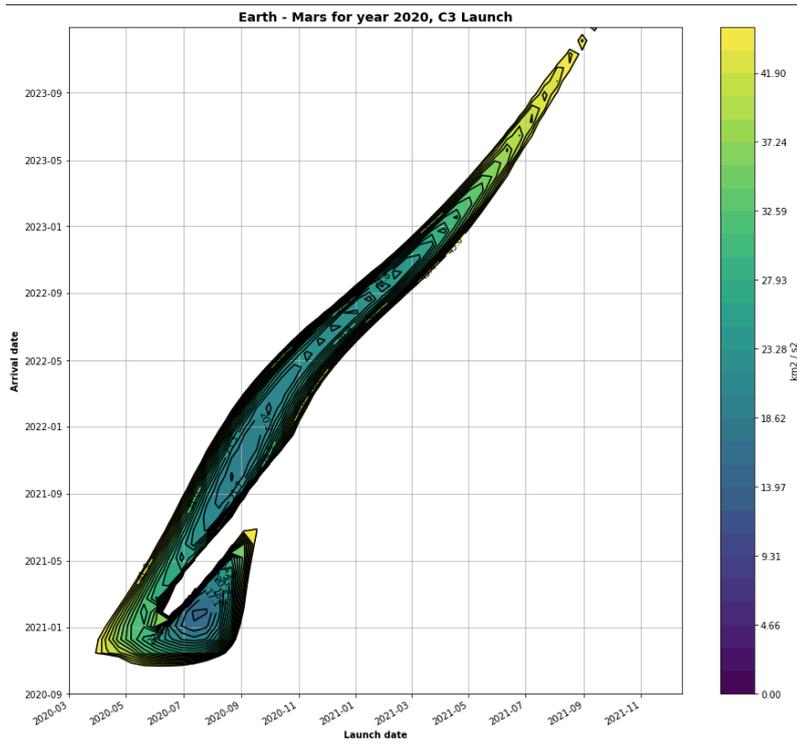
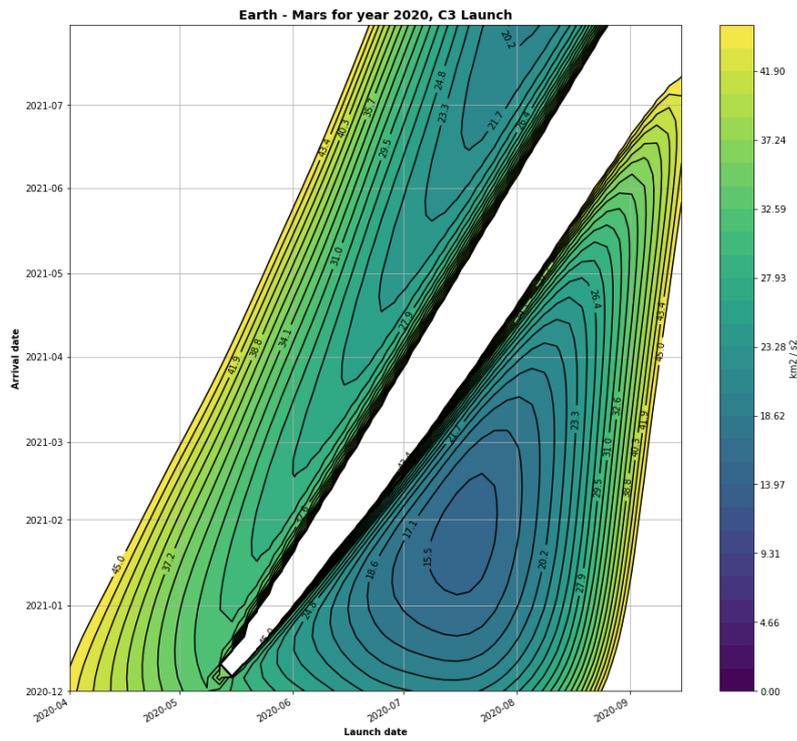


Figure 47: Mars 2020 Hohmann transfer trajectory recreated using Perylune software.

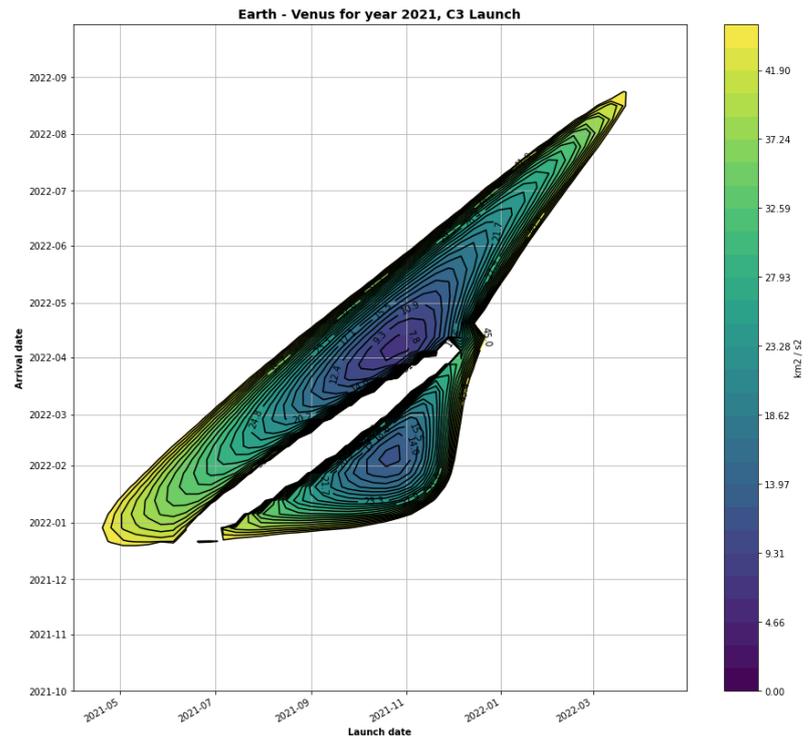


(a) This diagram shows the full Mars transfer window, for departure in 2020-2021 and arrival in 2021-2022.

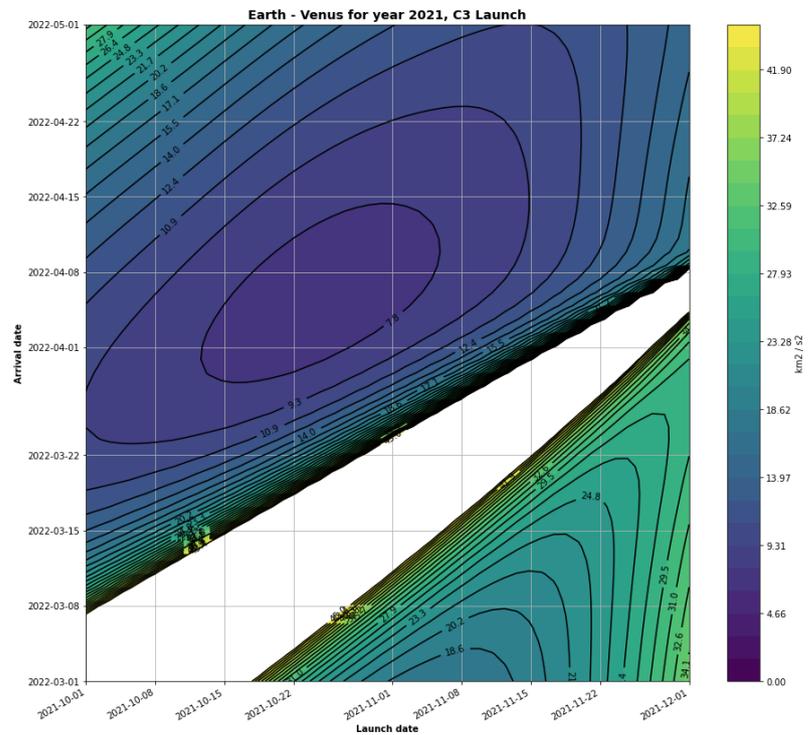


(b) A close-up of the optimal Mars trajectories region.

Figure 45: Mars transfer window, taking advantage of the 2020 opposition.



(a) This diagram shows the full Venus transfer window, for departure in 2021-2022 and arrival in 2022. A close-up of the most optimal trajectories is shown in the next Figure.



(b) A close-up of the optimal Venus trajectories.

Figure 46: Venus transfer window, taking advantage of the 2021 opposition.

4.5. Problem 5: Asteroid Survey

As of today, there are over one million asteroids known in the Solar System. Many of them belong to a NEA (Near Earth Asteroid) class. This problem aims to review all existing known asteroids, pick several as target candidates, and then assess the difficulty of mission leaving LEO and intercepting the target. This problem brings in the additional complexity of reaching Earth escape velocity, changing the frame of reference to heliocentric, changing inclination, and other orbital parameters to match those of the target.

4.5.1. Asteroid Discovery Process

An asteroid (*pol.* *asteroida*) is a general term that describes all-natural bodies smaller than planets that do not have a visible tail. Although with the introduction of the dwarf planet (*pol.* *planeta karłowata*) category, some asteroids are now considered planets, they still retain their status and designations as planetoids. Historically, the first discovered asteroid was Ceres, discovered in 1801 by Giuseppe Piazzi, followed by Juno in 1804, Pallas and Vesta in 1821 and Metis in 1822. The process of asteroid discovery is not straightforward. Several observations are necessary to determine the asteroid's orbit. In theory, this requires only two observations to solve the Gauss problem (see Section 1.9.1). However, to achieve reasonable accuracy that is sufficient for follow-up observations, in practice, at least three observations are needed. Nowadays, there are observatories available around the world in pristine locations, such as the Atacama desert, Mauna Kea summit in Hawai'i, or La Palma, all of which provide almost continuous observation capabilities. However, back in the 19th century, observations were conducted mostly independently, with local weather being a significant factor. It was a widespread occurrence to spot a candidate object only to lose it, sometimes for many years. As such, the exact chronology of the asteroid's discovery is somewhat convoluted.

Each asteroid is given a unique number when reported for the first time. Historically, the initial asteroids were given names. For example, **(1) Ceres** means the first asteroid with an assigned proper name of Ceres. Nowadays, asteroids are discovered by the hundreds each month, and they no longer bear unique names, although it is a privilege of the discoverer to name its asteroid. There are rules regarding name choice (up to 16 characters, pronounceable, non-offensive, non-commercial, if named after political or military persona, it must be over 100 years since the person died). The naming process is described in [34].

Once at least one observation is reported, the new object is given a provisional designation that consists of year, followed by a letter describing half-months (A-T), followed by a single-letter designating the chronological discovery (A-Z, with the letter I not being used). However, since modern techniques can discover more than 25 asteroids per half-month, the next designation after Z is A with an additional index. For example, 1981 EF₁₈ means the object was discovered in the year 1981, in the first part of March (E). F is the sixth letter. However, there were $18 \cdot 25 = 450$ asteroids discovered before it. Therefore it is the 456th asteroid discovered in the first part of March 1981. Once enough observations are recorded to determine an orbit, the asteroid is assigned a sequential number, specified in parentheses, e.g. (8254) 1981 EF₁₈.

4.5.2. MPC Asteroids Database

The International Astronomical Union (IAU) maintains Minor Planet Center (MPC) [39], an organization dedicated to the search and monitoring of asteroids in the Solar System. It is a robust service with a lively community of professional and advanced amateur astronomers that search for new asteroids, help with follow-up observations of those already discovered to improve their determined orbits' precision and provide open access to the data. MPC defined its own data format that is tuned for the asteroids. In particular, the format used can handle reasonably efficiently a large number of entries. As of November 2020, the number of known asteroids passed one million. The data format provides additional information, such as the year of first discovery, number of

observations, the period used to determine orbit, and more. The format uses one object per line. Each line uses fixed column positions for name, dates, epoch, orbital elements, the number of observations, and other parameters. The format is described in Section 2.5.5. The full specification of the format is available in [42].

4.5.3. Asteroids Overview

An interface for the MPC database was implemented in *Perylune* software. It allows retrieving, updating, and processing all currently known asteroids. As of 2020-11-14, the database contained information about 1024421, or a bit over a million objects. The *Perylune* software was used to perform an initial assessment of the database contents. It is not practical to visualize orbits for all asteroids, so a statistical approach was used. The most important parameter describing an orbit is the semi-major axis. A histogram of major semi-axes (a) was generated and is presented in Fig. 48. The known asteroids have a in the range between 0.555 AU and 3549.257 AU. There are asteroids with smaller a , however, when observed from Earth, they always appear near the Sun and thus are difficult to detect. On the other hand, the Solar System's outer ridges contain a ring or a belt of rocky material called Kuiper Belt (*pol.* pas Kuipera), which is located mostly around the ecliptic. Even further away from the Sun is Oort cloud (*pol.* obłok Oorta), which is a source of aperiodic comets and is believed to enshroud the Solar System from all directions. Given the significant range (5 orders of magnitude difference between closes and furthest) and very uneven distribution, it is somewhat difficult to present the histogram in a readable manner. The typical way of presenting the data on the logarithmic scale would not work as the great majority of asteroids are cluttered in the main asteroid belt (a between 2 AU and 3.5 AU). An attempt has been made to show the histogram with a range truncated to 6 AU.

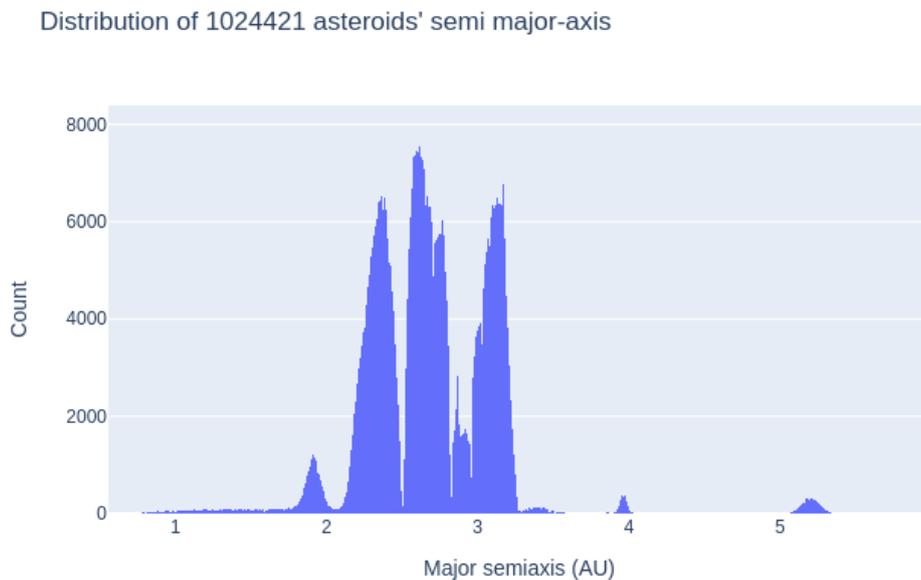


Figure 48: Histogram of semi-major axis values of orbits of all currently known asteroids (1024421 as of 2020-11-14). The histogram has been truncated at 6 AU.

Another important parameter is eccentricity. Here the distribution is much more manageable. While technically the asteroids could have an eccentricity of over one and thus travel on a hyperbolic trajectory, all known asteroids move along closed elliptical orbits, with two known exceptions of I1/'Oumuamua and I2/Borisov. As such, $e < 1$ holds true. The eccentricity diagram for is presented in Fig. 49. The MPC database analysis shows that the eccentricity of the asteroid varies between 0.000000 and 0.998747, with the great majority having $e < 0.3$.

However, the most interesting way to present the data visually seems to be a relation between the semi-major

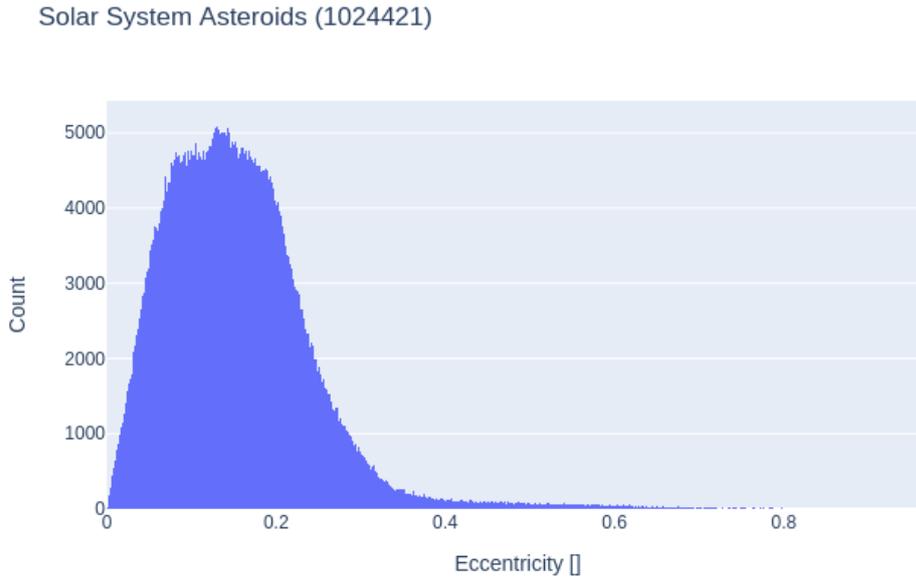


Figure 49: Histogram of eccentricity of all currently known asteroids (1024421 as of 2020-11-14).

axis and eccentricity. This diagram is presented in Fig. 50. Several conclusions can be made. First, it is increasingly difficult to detect asteroids the closer they get to the Sun. The detection is easier as long as they are in a highly eccentric orbit. This is reasonable, as objects on eccentric orbits have apoapsis farther away from the Sun, thus providing more favorable observation conditions. Secondly, there are several asteroid belts between Mars (1.52 AU) and Jupiter (5.2 AU). The tear-like shape around 5.2 AU is two groups of asteroids called Trojans. They are orbiting around Jupiter's L_4 (nicknamed Greeks) and L_5 (nicknamed Trojans) Lagrangian points of the Sun-Jupiter system.

Another interesting observation is notable gaps in the main asteroid belts around 2 AU, 2.5 AU, 2.8 AU, 2.95 AU, and 3.2 AU. Daniel Kirkwood first noted those gaps in 1957. After closer study, he discovered that the orbits are in resonance with Jupiter. For example, the $a = 2.5$ AU orbits have an orbital period of 3.95 years and would make three orbits for one orbit of Jupiter. During the closest approach to Jupiter, the massive planet perturbs the orbit slightly. However, since the process is repetitive, it pushes the asteroids away from those orbits over a more extended period. Those gaps are now known as Kirkwood gaps (*pol. przerwy Kirkwooda*). This topic is discussed in detail in [71].

4.5.4. Osculating and Perturbed Orbits

In the context of asteroids, it is essential to introduce the concept of osculating orbit (*pol. orbita oskulacyjna*). Classical Keplerian elements are a solution to the two-body problem, which is a simplified model that assumes the attraction of the primary body (attractor) and the orbiting body and the complete absence of other bodies in the system. This is a reasonably good approximation for heavy planets. However, the smaller the object, the less precise such model is. In practice, the smaller objects move along perturbed orbit, which differs from the ideal shape of the osculating orbit. The osculating orbit can be perceived as an instantaneous value or constantly changing orbital elements.

4.5.5. Asteroid Target Evaluation

Making a selection from a list of over a million potential targets is not a trivial task. During actual mission planning, many different criteria have to be taken into consideration. First and foremost, the difficulty of reaching

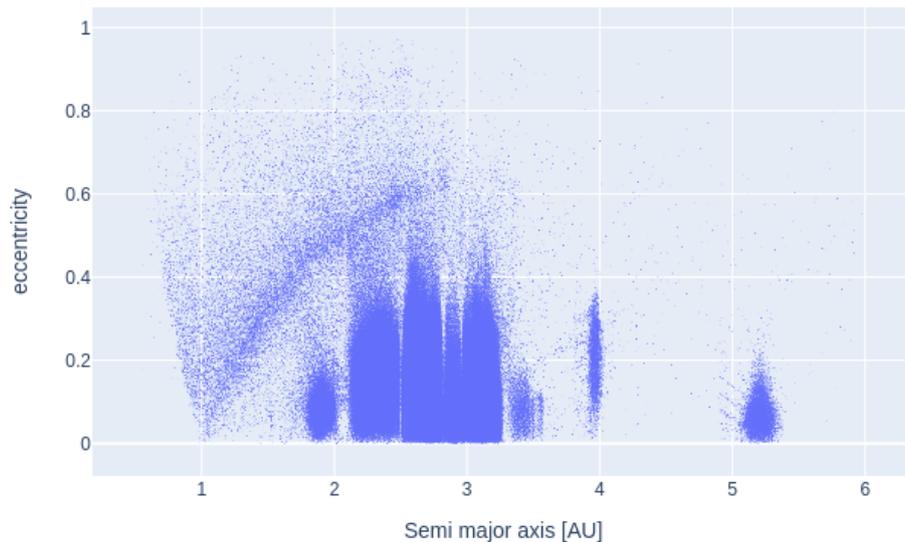


Figure 50: A scatter plot of eccentricity-semi major axis of all currently known asteroids (1024421 as of 2020-11-14).

the potential target is of utmost importance. Secondary criteria depend on mission goals. If the goal is simply a visit, then a fly-by mission is easier to do, as there is no breaking maneuver needed. The same is true for the impact class of missions, for example, when studying asteroid content by using an impactor (such as the Deep Impact mission conducted by NASA). For missions that expect to pick up samples, land on or some time in the future also conduct resources mining, a rendezvous is necessary, which adds another burn that has to be accounted for in the Δv budget of the mission. Another two crucial aspects neglected in this analysis were asteroid size and asteroid composition. Missions whose goal is mining and related activities, such as prospector class missions, would be interested in larger asteroids, especially those rich in metals and ice. Missions to alter the trajectory would be interested in the smallest asteroids.

A capability in the *Perylune* software has been developed that offers an assessment of all known asteroids. The software downloaded over a million orbit definitions from the MPC database and calculated the necessary cost of Hohmann transfer between Earth and potential targets. The calculations do not account for the escape velocity necessary to leave Earth gravity, but they account for the inclination change necessary. The list of top 20 asteroids that are easiest to reach is presented in Table 7. The full list (over million entries) is available in `perylune-data` project [67] in the `mpc/asteroids-2020-11-13.txt` directory.

The best 10 out of those top 20 were selected for visualization. The orbits are presented in Fig. 51. The 2020 CD3 asteroid is particularly interesting. It is a near-Earth asteroid that orbits the Sun most of the time, but it sometimes passes close to the Earth-Moon system and temporarily becomes Earth's moon. According to [7] and [36], this small asteroid was captured around 2016 or 2017 and likely escaped Earth's sphere of influence around May 2020. It is expected to remain on an orbit similar to Earth's and may be temporarily captured again. According to [7], such a temporary second mini-moon of Earth may not be such an uncommon occurrence. A model is discussed that proposes a population of 10^9 NEO asteroids of a size larger than 1 m, with a small fraction 10^{-7} being temporarily captured by Earth-Moon every year.

Post Scriptum: After the survey was completed, additional data became available on 2020-12-02 regarding 2020 CD. It is now identified to be a spent upper stage of a Centaur II rocket that launched Surveyor 2 mission back in 1966 [48].

Target	total Δv [m/s]	time of flight [days]	Hohmann burn 1 [m/s]	Hohmann burn 2 [m/s]
2010 FN	178.39	181.2	100.74	77.65
2018 UC	189.48	181.6	132.27	57.20
2016 JA	218.83	184.6	111.16	107.67
2020 SO	224.53	180.8	123.50	101.03
2019 YB ₄	308.83	182.1	280.12	28.71
2019 AU ₆	342.60	180.4	218.93	123.67
2000 SG ₃₄₄	351.41	179.5	180.04	171.37
2010 JK ₁	446.56	186.7	232.52	214.03
2016 WQ ₃	466.28	184.8	348.10	118.17
2013 BS ₄₅	468.61	181.5	406.01	62.60
2015 XF ₂₆₁	490.89	181.3	418.18	72.71
2001 XX ₄	492.14	183.5	444.06	48.07
2012 VS ₇₆	492.14	181.4	423.79	68.35
2019 JH ₇	509.94	181.4	443.90	66.04
2008 UA ₂₀₂	517.11	187.2	277.64	239.47
2019 OV ₃	529.29	180.2	394.59	134.70
2014 WE ₆	532.06	178.4	294.17	237.88
2011 OJ ₄₅	535.35	185.0	410.02	125.33
2020 CD ₃	538.68	185.7	373.04	165.64
2011 DS	546.15	187.4	293.28	252.87

Table 7: Top 20 asteroids that are easiest to reach from Earth. The Δv in relation to spacecraft at $C3=0$ (escape velocity from Earth system). The list is ordered by increasing total Δv , which assumes rendezvous and two Hohmann burns. For fly-by missions, only the first Hohmann burn is necessary.

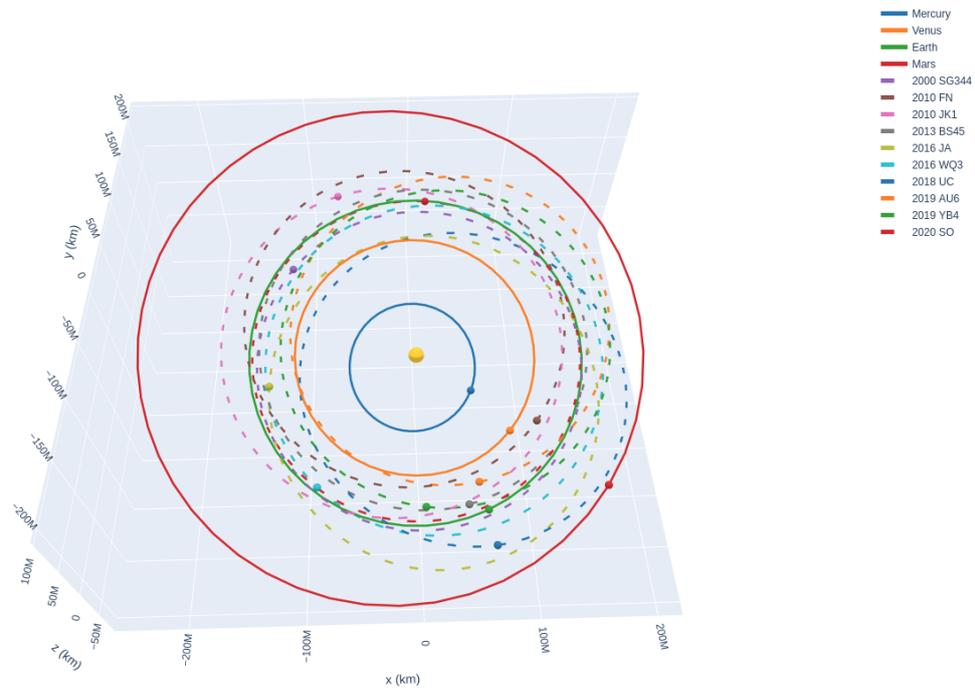


Figure 51: Inner Solar System with four innermost planets and top 10 asteroids with lowest Δv requirements with regards to accessibility from Earth.

Note: Research Idea

Asteroids study is an attractive research area. The author plans to extend his study in this area. Nowadays, asteroids are detected at a rate of over thousands per month. Some of the asteroids have orbits close to Earth and can be reached reasonably easily. This makes them an attractive potential target for a space mission. The survey can be refined in many ways. The Δv assessment was basic and was based on viv-viva equation and inclination. A much more precise ephemerids based approach can be used for selected top candidates, as this method is many orders of magnitude slower. A visualization of all (over 1 million) asteroids with major and dwarf planets marked could be created. The asteroids are highly perturbed. With such a large data set, many methods can be applied, hopefully yielding interesting results. Statistical methods can be applied to find different patterns. Artificial Intelligence methods could be used to pick outliers. The orbital dataset could be extended with additional information about size and composition, although this type of information is only available to a selected subset of all known asteroids. Finally, the database is being updated daily. It makes sense to prepare an automated method that would repeat the analysis periodically, perhaps monthly.

4.6. Problem 6: Navigating with low power engine or propellantless cubesat

The economic reality implies that Poland is currently incapable of launching any satellites larger than CubeSats. This form factor is too small to have any substantial conventional chemical propulsion useful as a primary engine. However, several possible alternative propulsion mechanisms can be taken into consideration. One of them is an ion engine (*pol.* silnik jonowy), which is a blanket name for many different solutions that do not rely on chemical reactions to expel exhaust, but instead use electromagnetic (EM) field to accelerate ionized plasma. The primary benefit of all ion engines is a much higher specific impulse, roughly interpreted as fuel efficiency in layman terms. The primary flaw is very low thrust.

Another potential solution is a solar sail, a solution that is rarely discussed. The following sections provide an analysis of the potential use of a solar sail for effective navigation.

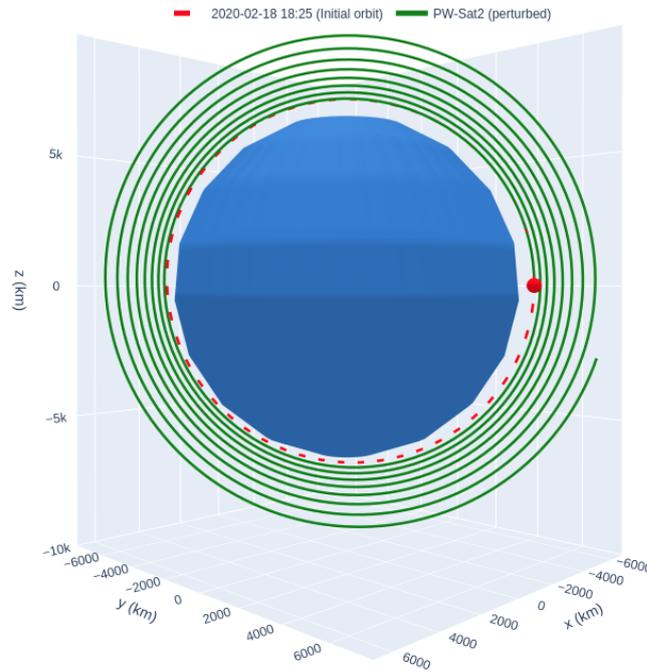


Figure 52: Constant acceleration perturber. This type of perturber is a good model for ion engine. In this particular example, it was configured to provide constant forward thrust. The initial osculating orbit has been marked with dashed red line. The perturbed trajectory is marked in green and shows a period of almost 16 hours, which is roughly 10 orbits.

4.6.1. Simulating Ion Thrusters

The family of ion engines provides low and long-lasting thrust. The engine cannot conduct typical burns as it does not have sufficient thrust. Maneuvers performed by ion engines can be perceived as a slow evolution of an orbit towards the intended shape. For example, consider a scenario of PW-Sat2 satellite being equipped with an ion engine that provides a constant thrust of $2 \cdot 10^{-5} \text{ m/s}^2$. This is an unreasonably high acceleration for an ion engine, but this exaggeration lets showcase the phenomena more easily.

To escape Earth gravity, a reasonable strategy is to take advantage of the current orbital velocity. This means the optimal burn direction is prograde. A result of a simulation of 10 orbits has been presented in Fig. 52. Initially, the spacecraft moves along its osculating orbit (*pol. orbita oskulacyjna*), marked in dashed red. Once the burn is commenced, the spacecraft slowly increases its velocity and moves to a higher altitude orbit. This trajectory is a perturbed orbit (*pol. orbita perturbowana*) and is marked with green. Since the burn is continuous, there is no clearly visible apoapsis or periapsis. The initial constant thrust model (see `constant_accel` perturber implementation in Section 3.5) assumes the burn is continuous. This provides the fastest change, which often is very long nevertheless. However, it is not the most optimal as it does not take advantage of the Oberth effect (*pol. efekt Oberth'a*), which implies that the most efficient place to conduct a burn is around periapsis.

4.6.2. Solar Sail

The PW-Sat2 [64] and PW-Sat3 [66] satellites are or will be equipped with a solar sail. The PW-Sat2 is currently in orbit with the sail deployed, while PW-Sat3 is under development. The primary purpose of the sail as envisioned by the PW-Sat designers were to provide faster deorbiting for old satellites by increasing the atmospheric drag. However, this is not the only possible application of a large surface in space. Sun provides a constant stream of photons and other particles that exert force on any illuminated object. The solar radiation pressure depends on the reflectivity of the surface. According to [99] and [15] eq. 12.95, the pressure in Earth vicinity varies between $4.5\mu\text{Pa}$ for absorbing surface and $9.08\mu\text{Pa}$ for perfectly reflecting surface. Since the sail's reflectivity in PW-Sat2

is not publicly known, the worst case of $4.5\mu\text{Pa}$ was used in simulations. The PW-Sat2 mass is 2.5 kg and the sail has an area of 4 m^2 . As such, the sail set at the most efficient configuration (perpendicular to solar illumination), generates 0.000018N of force and provides a thrust of $7.2 \cdot 10^{-9}\text{ m/s}$. Those values may seem very small, but the huge advantage is that this force does not require any fuel. As such, this can be considered propellantless propulsion.

To investigate whether the solar sail is a viable concept, a simulation environment has been developed. The first experiment conducted assumes realistic criteria: the worst-case value of solar pressure, a mass of 2.5 kg, and the sail area of 4 m^2 . The orbital data is based on the oldest TLE data available to the author (15 March 2020), as published by [9]. The only modification was to change the RAAN by 60° . This is deemed acceptable, as J2 perturbations can be used to rotate the orbital plane, and the purpose-built mission designed to take advantage of a solar sail would be launched to an orbit that has its orbital plane more favorably inclined towards the Sun.

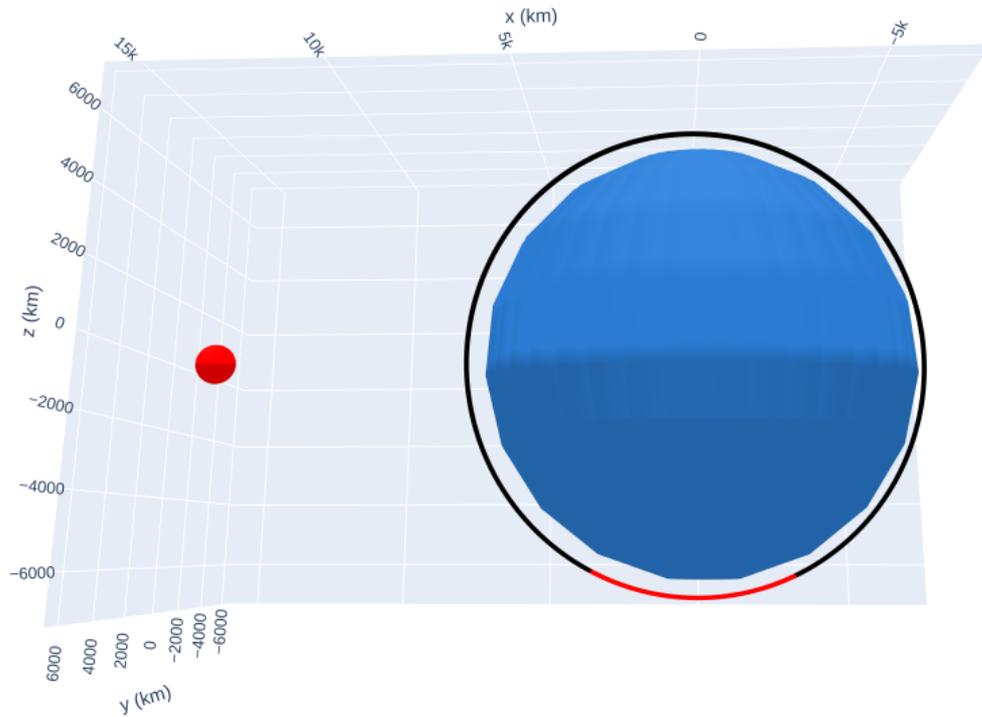
The solar sail propulsion was implemented as a perturber model. The perturbed trajectory was calculated using Cowell method (*pol. metoda Cowell'a*) (see Section 3.6 for method description and implementation details). The Cowell method conducts numeric integration. Without any perturbations, the solution is a perfect Keplerian trajectory. With every integration step, the perturber introduces a small acceleration resulting in slightly altered trajectory. The implemented solar perturber model considers the following parameters:

1. solar pressure – if not specified, the value of $4.5\mu\text{Pa}$ (absorbing surface in Earth vicinity) is used, expressed in Pa;
2. eclipse conditions (whether the spacecraft is lit or in Earth's shadow);
3. spacecraft mass in kg;
4. sail area – expressed in m^2 ;
5. λ – angle between current velocity and solar pressure direction;
6. λ_{min} – minimal angle – the sail is set to “running” configuration only if the angle between current velocity and solar pressure direction is greater than the minimal angle;
7. orbital state vectors – r (position) and v (velocity).

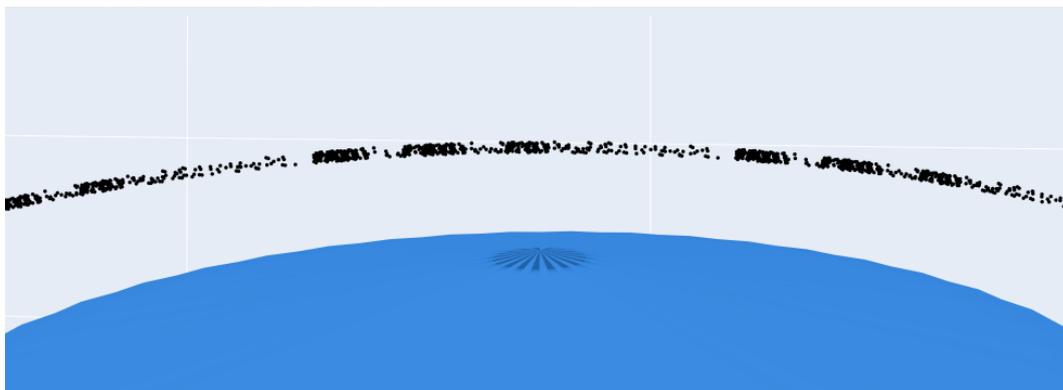
The perturber model currently has the following values hard coded and simplifications. If necessary, those can be easily modified into additional parameters:

1. Sun position – assumed to be at $[149600000, 0, 0]$ km.
2. Earth radius – assumed to be 6378.137 km. This is used to calculate the shadow cone.
3. Simplified shading model – the light rays are assumed completely parallel and Earth's shadow is modeled as a cylinder. In practice, the Earth's shadow is a cone. The actual Earth shadow cone extends over Moon's orbit (see [15], page 565).
4. The difference between penumbra (*pol. półcień*) and umbra (*pol. cień całkowity*) is not modelled. The hard shadow is used – either the spacecraft is either fully lit or completely in shadow.

A result of the simulation of 1000 orbits is presented in Fig. 53a. With the $\lambda_{min} = 140^\circ$, the parts of the trajectory where the sail is active ($\lambda > 140^\circ$, not in shadow) are marked in red. The apoapsis raises slowly, which is barely visible at this scale. To better demonstrate this effect, the simulation has been repeated for 40 m^2 sail. The result is presented in Fig. 53b.



(a) Visualization of the solar sail perturbation. The part of the trajectory where the sail is turned on is marked with red. The Sun position (Sun size and distance not to scale) is presented with a red sphere. The apoapsis is raising very slowly. This effect is barely visible.



(b) A close-up of the apoapsis with the integration points shown. The consecutive orbits do not perfectly repeat, but rather slowly raise. To better show this effect, the simulation was repeated with a 40 m^2 sail.

Figure 53: Solar sail perturbation for PW-Sat2 orbit.

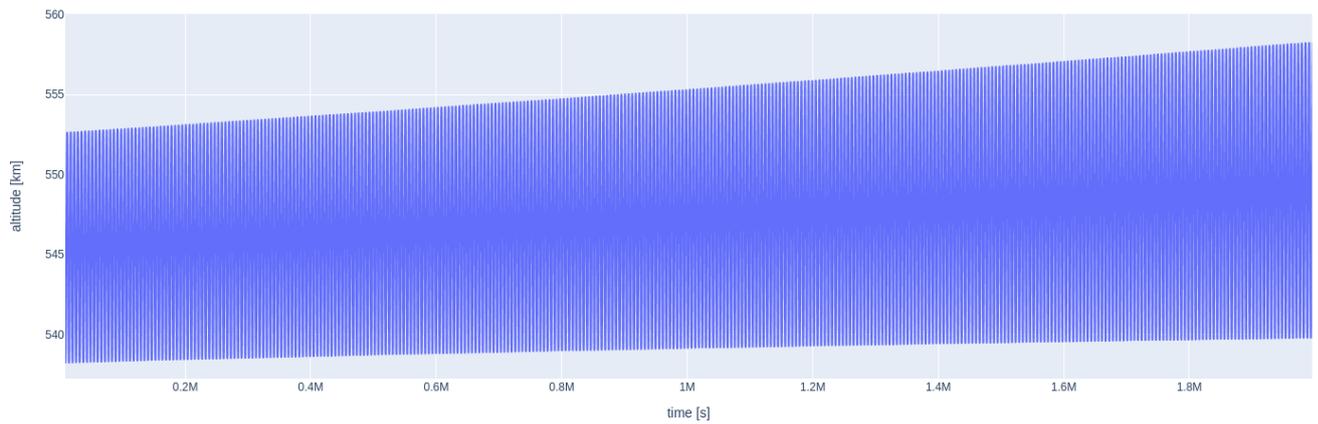


Figure 54: A change of the altitude over a period of $2 \cdot 10^6$ s (roughly 23 days) due to solar radiation pressure exerted on a 4 m^2 sail on a 2,5 kg satellite.

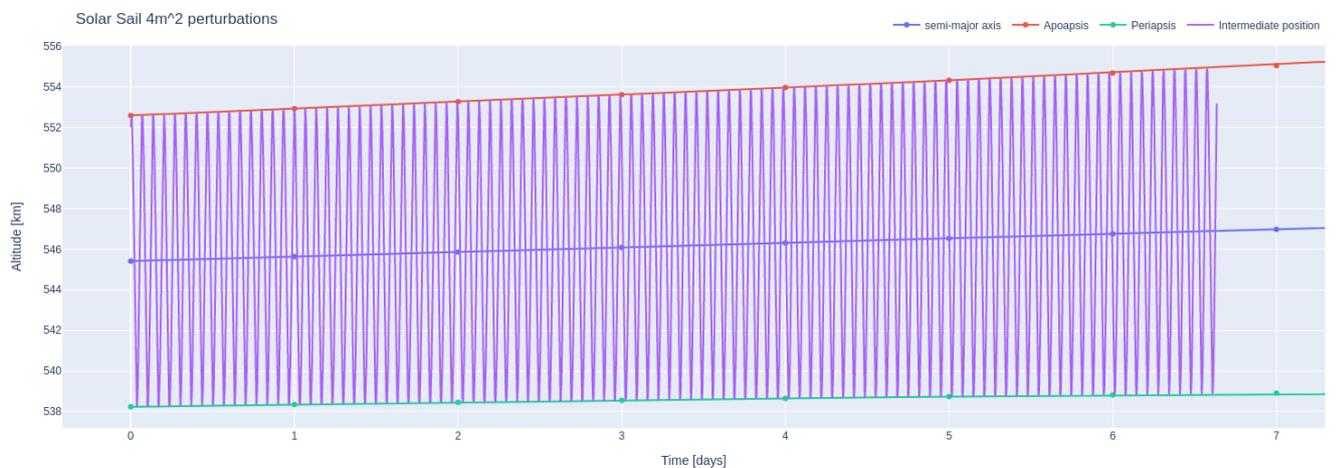


Figure 55: Zoomed in first several days of simulated operation. Each orbital cycle has slightly higher apoapsis (red line) and barely measurable increase in periapsis (green line). The average altitude grows slowly.

Longer simulations over a period of $2 \cdot 10^6$ s (roughly 23 days) show that with this configuration, the sail can be used to raise apoapsis of the orbit, with only a minimal impact on periapsis. After 23 days, the orbit became more elongated. A chart presenting the altitude change over time is presented in Fig. 54, with a close-up and apses and semi-major axis marked is presented in Fig. 55.

4.6.3. Solar Sail Performance Summary

As shown in the earlier sections, the solar sail has several interesting properties:

1. It provides thrust roughly in the direction away from the Sun. This directional nature of this force limits the applicability of this propulsion method.
2. The sail can be roughly set to running mode (by rotating it in a position perpendicular to spacecraft – Sun light vector) or turned off mode (by rotating to a position that is parallel to Sun light vector).
3. The orbited body geometry imposes its own restrictions by casting a shadow. The eclipses happen to be longer if the orbit is lower.
4. One maneuver that is particularly well suited for the solar sail is raising or lowering apoapsis.

	Model name	Body	Date	Altitude [km]	Parameters
[H]	COESA62	Earth	1962	0-700	Temp, Density, Pressure
	COESA76	Earth	1976	0-1000	Temp, Density, Pressure
	Jacchia	Earth	1977	90-2500	Temp, Density, Pressure

Table 8: List of popular atmosphere models

5. Orbits with a plane that is parallel to the Sun light are well suited for the solar sail.
6. Orbits with a perpendicular plane to the Sun light are poorly suited for the solar sail.

The directional nature of the solar pressure in particular limits the practical applications of the sail. Nevertheless, a solar sail could be used to perform some maneuvers, such as raising periapsis and apoapsis of the orbit. One complication is that the force vector always points directly away from the Sun. This would imply the sail would have to change orientation in various sections of its orbit around Earth. However, that should be doable with magnetorquer, an innovative mechanism that generates magnetic dipole that interacts with Earth’s magnetic field, thus providing torque and eventually rotating the spacecraft.

4.6.4. Atmospheric drag

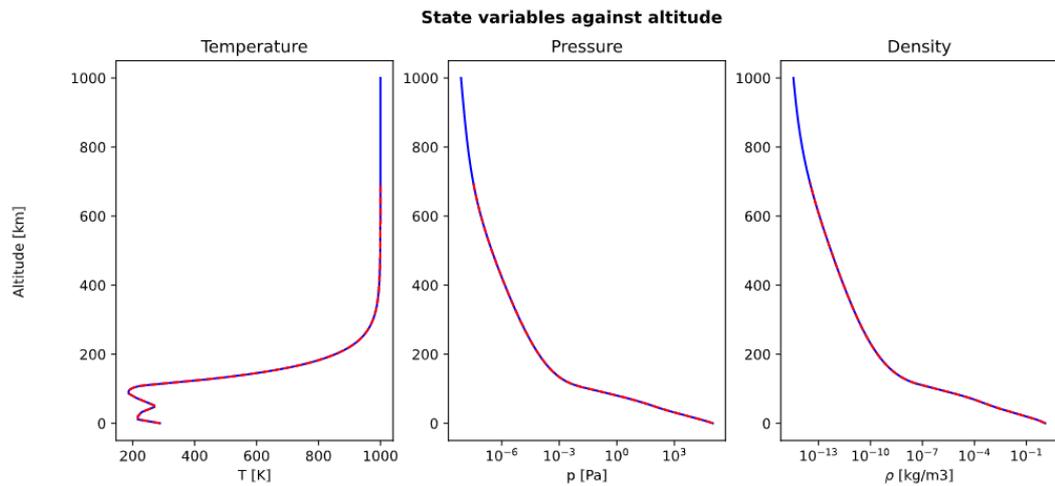
Earth’s atmosphere extends far beyond the Kármán line. The most obvious implication is that satellites and other objects are being slowed down due to atmospheric drag. Depending on the altitude, this effect can be anything from almost negligible (for altitudes of 500 km and higher) to destructive. Most meteors (*pol.* meteor) that enter Earth atmosphere burn up completely and only a small fraction reach surface as meteorites (*pol.* meteoryt). Many models of the atmosphere are available. The two most popular are COESA 62, and COESA 76 [14], more commonly known as U.S. Standard Atmosphere (*pol.* atmosfera wzorcowa). The second is an updated version, published in 1976 and seems to be the most popular one currently in use. The name is sometimes supplemented with the publication year (U.S. Standard Atmosphere 1976) or the “U.S.” prefix is dropped (“Standard Atmosphere”). A supplemental high altitude model was published by [28] in 1977, which covers the range between 90 km and 2500 km. It requires other models to cover the full altitude range, thus the “supplemental” name. A list of common atmospheric models is presented in Table 8. Major properties of the Standard Atmosphere are presented in Fig. 56. A rich list of atmospheric models for Earth, Venus, Mars, Titan, and other planets is available in [60].

The *Perylune* software uses the atmospheric drag model implemented in the *Poliastro* package. The model is documented in [60]. The `atmospheric_drag_model` perturber takes the following parameters: R_E (Earth radius, expressed in km), C_D - Coefficient drag (dimensionless), A_{over_m} (drag area in m^2 over mass, expressed in kg) and model (which is a pointer to atmosphere model). The *Poliastro* software package provides another perturber `atmospheric_drag_exponential`, but it is not recommended for use, as its atmosphere model is oversimplified.

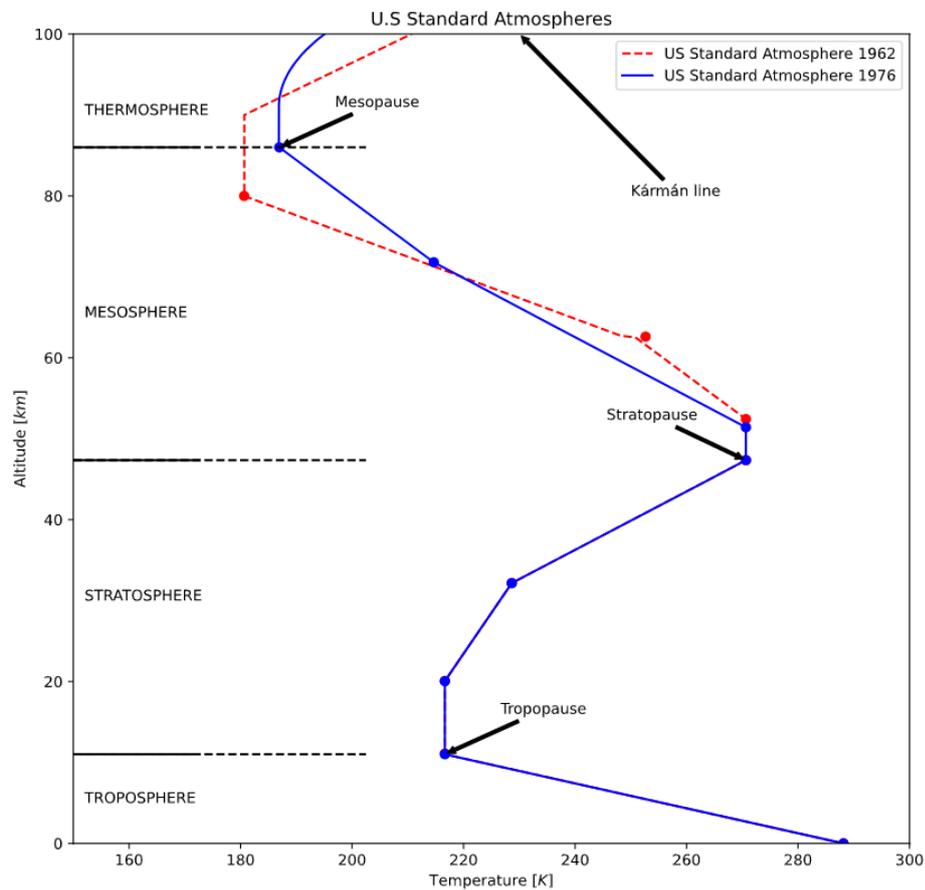
The atmospheric drag becomes the major factor for altitudes below 250 km. It is possible to maintain short-lived orbits at an altitude 185 km, but the trajectory degrades quickly. A model was defined as an exercise assuming a 100 kg satellite on a 250 km circular orbit. The simulation result is presented in Fig. 57. The reentry event happens after 7 days and 3 hours.

4.6.5. PW-Sat2 Sail

Another atmospheric model was created for PW-Sat2. There is limited data available regarding the orbital decay on PW-Sat2 Sail page [65]. The data in the numeric format is not available. The only accessible format is a PNG file with thick predicted and actual altitude being presented. It is possible to read specific points from the chart, but the data will have significant uncertainty. Another data type available is orbital parameters in TLE format, published by Celestrak service [9]. The oldest data available to the author are from 15 March 2020. The TLE data



(a) The changes of temperature [K], pressure [Pa] and density [kg/m³] in function of altitude. Full range up to 1000km.



(b) Lower atmosphere up to 100km.

Figure 56: Two popular atmospheric models are COESA 62 and COESA 76, often referred to as U.S. Standard Atmosphere. The earlier (COESA62) model covered altitudes from 0 km to 700 km (dashed red), while the update (COESA76) covers range of 0 km to 1000 km (solid blue), with some extra coverage for negative altitudes, which are useful for simulating phenomena in depressions. This diagram is roughly based on diagram from [59].

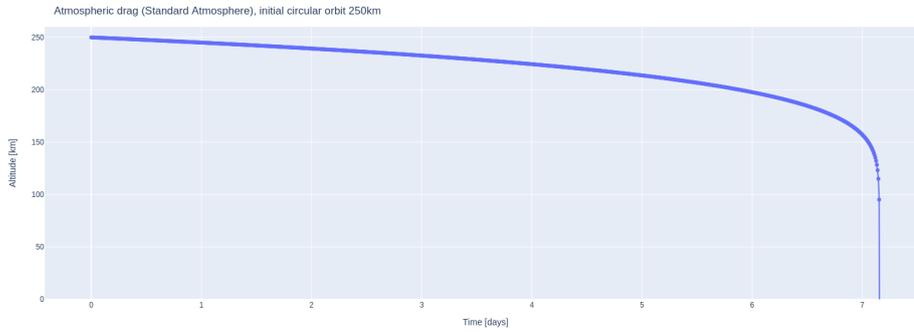


Figure 57: Atmospheric drag impact on 100 kg satellite moving on 250 km equatorial orbit.

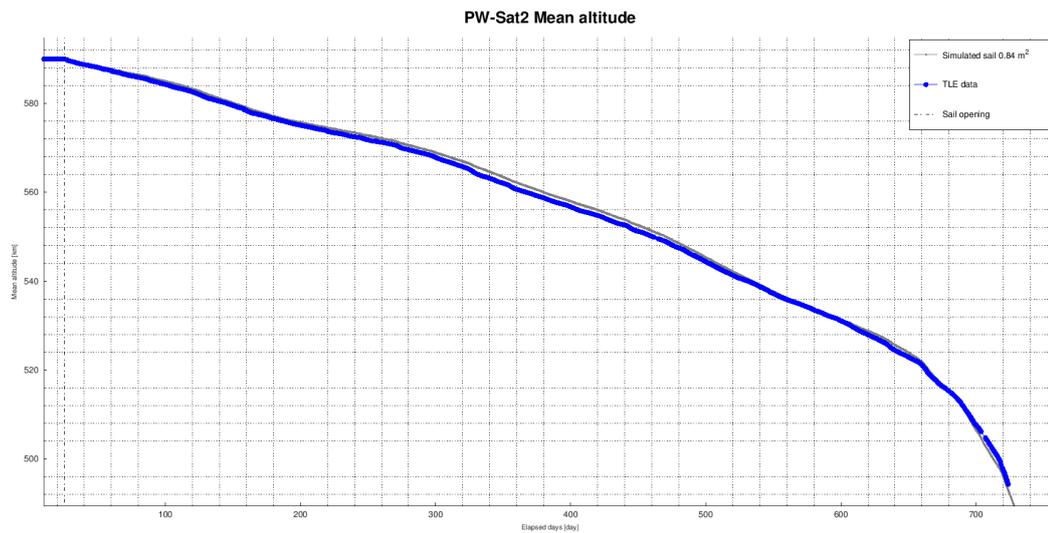
indicates an average altitude of 545 km. This roughly corresponds to the altitude reported by the PW-Sat2 team on day 500. The simulation covered 100 days. The parameters used are $m = 2.5\text{kg}$ and sail area $A = 0.84\text{m}^2$. The dimensionless drag coefficient is unknown, and a value of 0.3 was chosen after several experiments. The original data from the PW-Sat2 team and results simulated with Perylune are presented in Fig. 58a and 58b, respectively.

The atmospheric drag simulation obtained roughly fits the experimental observations and the model developed by the PW-Sat2 team. However, given the uncertainty in the PW-Sat2 data, it is premature to make any specific conclusions regarding the model fitting. Nevertheless, it can be considered anecdotal evidence that the available atmospheric drag model can predict satellite behavior at low (250 km) and medium altitudes (540 km). The drag model works properly but is susceptible to several parameters. With extra tuning, it should be well suited for producing high-quality simulations.

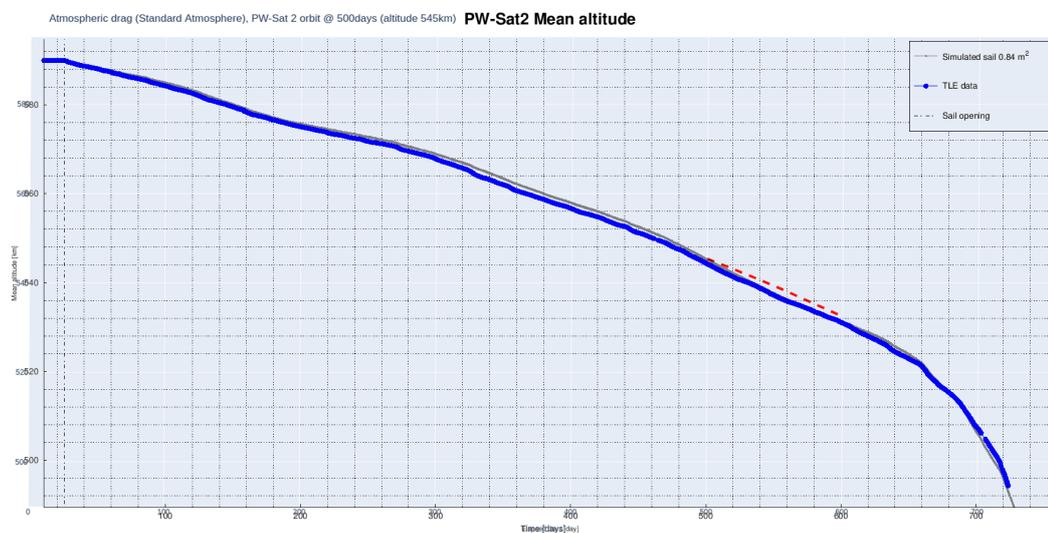
Note: Research Ideas

The idea of using sail and solar pressure can be developed further. The atmospheric drag is a dominant factor below a certain altitude, and the sail can only be used to slow down the satellite. However, above a certain altitude, the solar pressure can take over as the primary factor. In essence, this mechanism can be used to slowly increase apoapsis until the spacecraft can escape Earth gravity. Once heliocentric orbit is reached, the sail can be used to move away from the Sun to travel to outer planets or targets further away from Sun than Earth. The inner planets could also be visited, but this would require gravity assist on one of the outer planets. Whether such a concept is viable is an interesting research area. Also, the PW-Sat team could be reached, and perhaps more detailed data could become available. This would allow better tuning of the model to provide a better fit for experimental data.

Even in the LEO environment, it seems that the solar sail can raise apoapsis, but not periapsis. Although this may make the solar sail somewhat impractical on its own, it can be paired with a small ion thruster that would be used during the section of the orbit where the spacecraft travels roughly towards the Sun. This could make such maneuvers up to 50% cheaper in terms of fuel efficiency than pure ion engine solution.



(a) The original altitude chart, provided by PW-Sat2 team. Source: [65]



(b) The original PW-Sat2 data with overlaid Perylune simulation results.

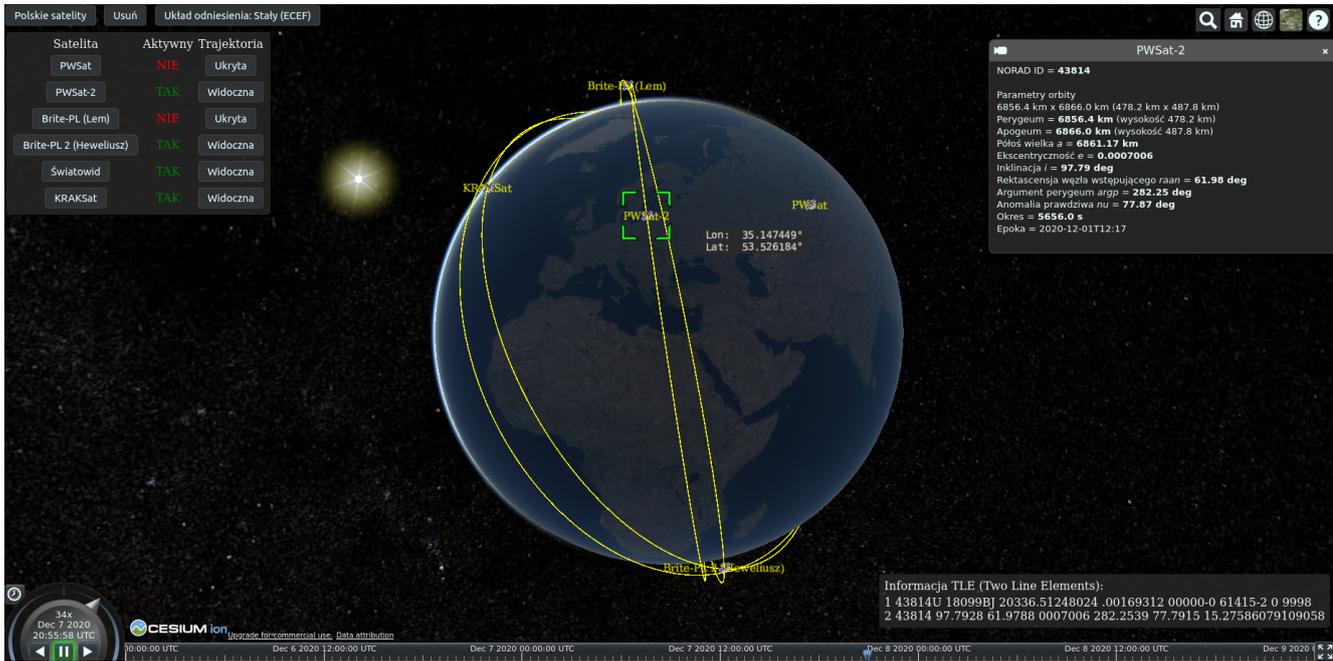
Figure 58: Comparison of actual and simulated PW-Sat2 orbit deterioration due to atmospheric drag. The results of *Perylune* simulation was marked in dashed red.

4.7. Problem 7: Website for tracking Polish Satellites

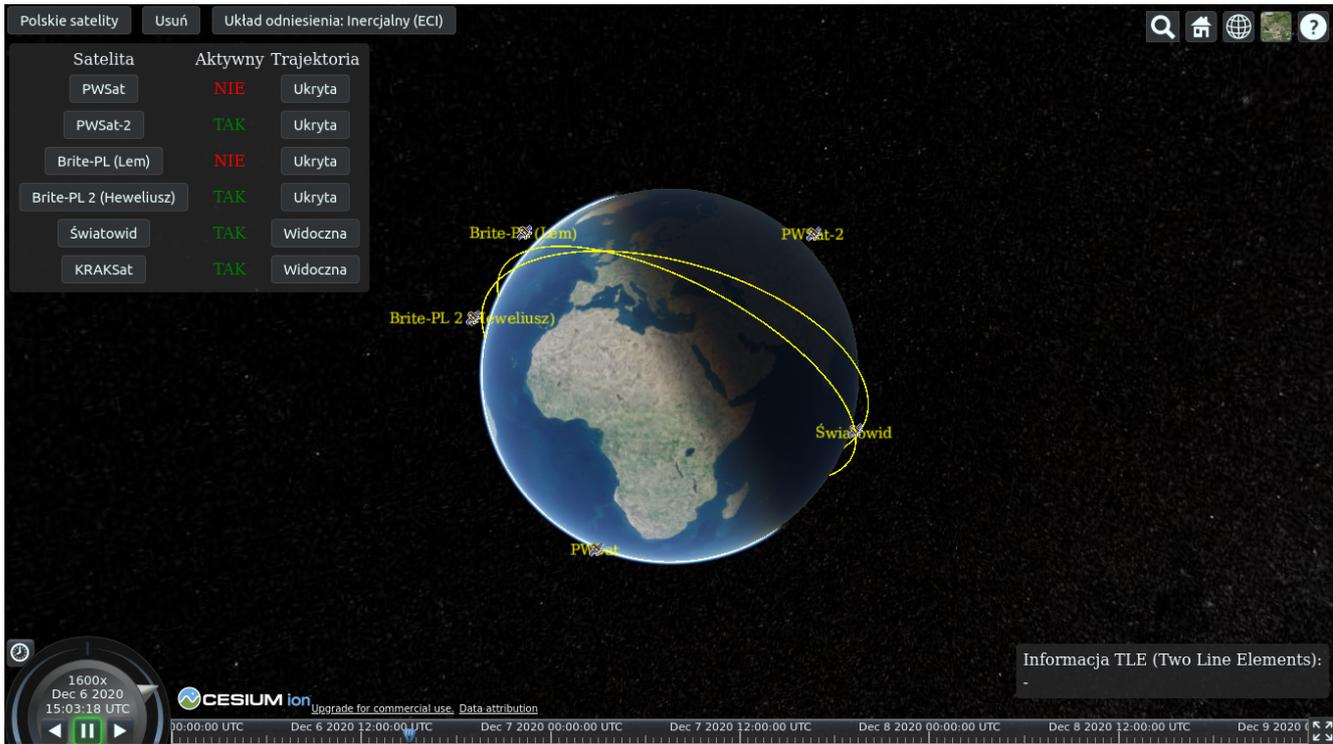
Poland has launched six satellites: LEM, Heweliusz, Światowid, PW-Sat, PW-Sat2, and KRAKSat, with two more known to be in development Pw-Sat3 and WroSat. As of late 2020, four of them are still in orbit. A website that predicts fly-overs of Polish satellites was developed. It uses graphically rich 3D visualization to demonstrate trajectories. The goal of this site is mostly educational and hopes to increase public interest in Polish space activities.

4.7.1. Web Interface Capabilities

An initial view after loading is presented in Fig. 59a. A list, the satellites themselves and their upcoming trajectories are presented. The server-side backend downloads the available orbital TLE information from Celestrak service [9] and recalculates orbital trajectories as needed. From the user's perspective, the site presents a simulation for the next 72 hours. Currently, this period is hardcoded but can be easily changed. The time slider at the bottom of the page allows controlling time flow – increasing and decreasing speed, moving forward and backward in time.



(a) Default view of the Perylune web interface. Six Polish satellites are loaded and their trajectories displayed. Satellites can be selected and their orbital parameters are shown.



(b) Perylune uses realistic shading model. Sun position is calculated using Cesium library and the shading model represents it.

Figure 59: Several views of the website developed.

Satellites can be selected, and their detailed orbital parameters (orbital elements, period, epoch, and more) are presented. The corresponding data in the TLE format is also available. Each satellite, as well as its trajectory, can be shown or hidden. A piece of basic information on whether the satellite is active (currently in space or not) is available. For satellites that are deorbited now (PW-Sat and Lem), the last available historic TLE information and their orbits before deorbiting is used.

The site uses a realistic light and shading model. The Sun location (as shown in Fig. 59a) and light/shadow conditions (as shown in Fig. 59b) are calculated on the fly. This information is also available in a flat projection model to be discussed later in this chapter.

Two reference systems are available: ECEF and ECI (both discussed in Section 1.5.2). ECEF model is fixed to Earth, so the Earth appears static and the stars, Sun, Moon, satellites, and their trajectories appear to rotate slowly. This is an excellent view to explain why predicted satellite ground tracks seem to move West. The alternate ECI model is inertial. In this view, the Sun, Moon, and satellites and their trajectories appear stationary and the Earth rotates beneath them. The actual Sun and Moon movement is simulated, although it is not very visible due to relative slowness (Sun's angular velocity on the sky is only 0.983° per day). Nevertheless, it can clearly be observed by stopping the flow of time and then moving the slider across the ruler.

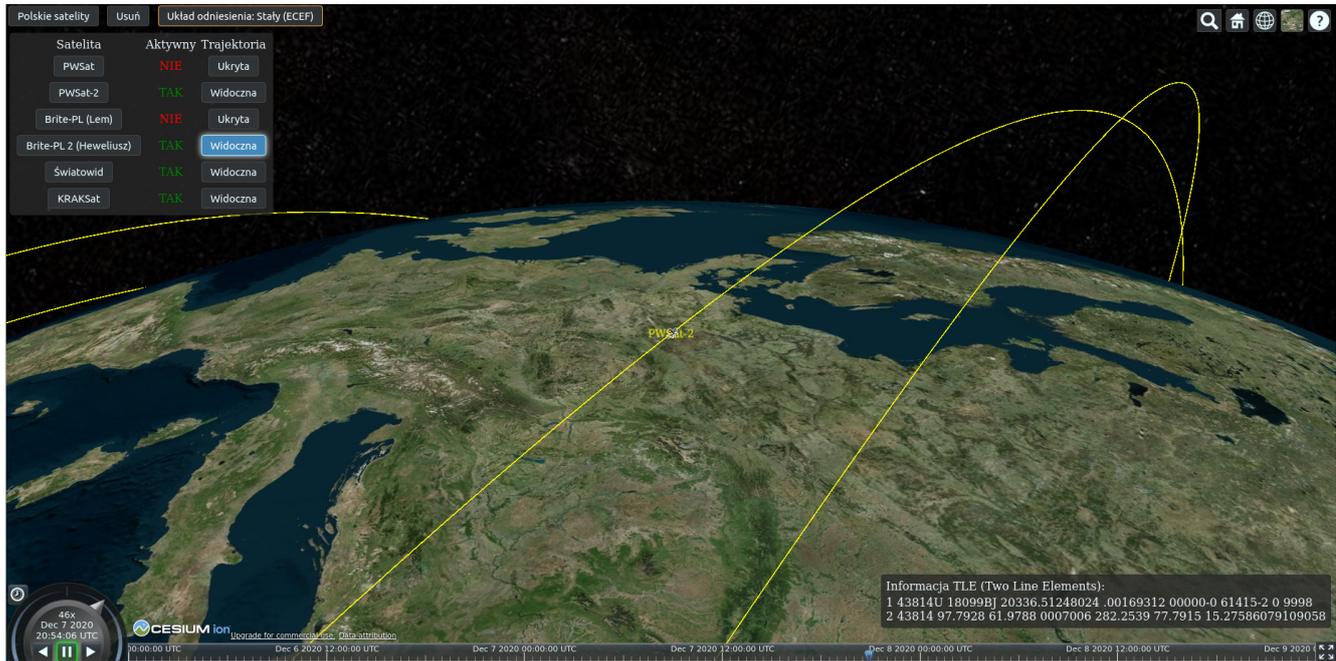
The default 3D view is appealing visually, but it does have a major flaw. The Earth frequently occludes the satellites. To avoid this difficulty, an alternate flat 2D map is also available. The map uses a Mercator projection. An example 2D view is presented in Fig. 60b.

The website is currently available at <https://perylune.space/cesium/>, with the source code available in [62]. The website was written in TypeScript language and used Cesium [12] library for graphical presentation. The backend and all orbital calculations are conducted using Perylune software. The backend and front-end components exchange data using CZML, an open JSON-based standard, defined in [16]. The data can be exported and processed further in any CZML compatible software. See Section 3.1 for a detailed description of the website technical details.

4.7.2. Potential Future Improvements

Note: Research Idea

The website in its current phase of development can be considered a proof-of-concept prototype. It demonstrated that the orbital parameters could be kept up to date and presented in a visually appealing manner. The prototype can be expanded in many ways. The prediction period of 72 hours can be changed or made configurable. The list of satellites can be expanded, e.g., to show all satellites that have components produced in Poland. The satellite descriptions can be expanded with a descriptive text and perhaps links to additional resources. The satellites are now visualized as simple icons, but the Cesium library allows displaying full 3D models. This would require some expertise in 3D modeling, but it is doable. An ability to pinpoint an observer's location could be added. This, along with available Earth terrain models, could make observation planning more appealing. With the observer location available, the website could be extended to produce a list of upcoming fly-overs. Another potential possibility would be to make some predictions regarding satellites' brightness as observed from Earth. Unfortunately, this requires a good reflectivity model and orientation for each satellite. Such models exist and are successfully used for larger satellites. Amateur astronomers used to observe Iridium flares (*pol.* flary Iridium), a Sun's reflection off the solar panels of Iridium satellites. Some websites, e.g., [22], had good satellite orientation models and were able to make precise predictions of the satellite brightness, expressed as stellar magnitude.



(a) Satellites and other points of interest can be zoomed in. One of the PW-Sat2 flights over Poland on Dec 6, 2020 is visualized.



(b) A 2D view that uses Mercator projection. This view is particularly useful for high level view, as it shows all the satellites all the time.

Figure 60: Comparison of available and simulated PW-Sat2 orbit deterioration due to atmospheric drag.

5. Solution evaluation

This chapter describes the process used to validate the developed software. Three verification methods are taken into consideration. The first one compares the calculations with the reference data or results generated by existing, well-known software packages, such as STK, GMAT, or SGP4 models. Second validation is based on the concept of unit and system tests, a software development regime developing tests and software as two interdependent elements. The third validation to be considered is to conduct actual observations and compare measurements with the models. The author has access to astronomical equipment, and an attempt could be made to measure some parameters empirically.

5.1. Software Validation

Several techniques can be used to validate the obtained computational results. Tests can be developed that compare calculation results produced by the software with some well-known results. Another possible verification method is to repeat the computations using a different implementation and compare results. The results can be compared to other data available.

Depending on the specifics of the calculations, some methods are better suited than others. For example, when calculating the orbits of planets, the data is readily available in multiple sources. However, when solving the same problem for asteroids, there are no references to compare to.

The following sections discuss various approaches that were used.

5.2. Test Driven Development

One of the essential programming paradigms is Test Driven Development (*There is no accepted Polish equivalent. English name is used in Polish industry.*), or TDD. The concept assumes that automated tests are an integral part of the developed software. The tests are developed first before the production code. Without the code, all the tests fail. This step protects against false positives. Afterward, the actual production code development commences. Tests are run and frequently updated along with the production code. Once all the tests pass, the software is considered ready.

Another important practice in modern software development is Continuous Integration. This practice implies that the software is being developed in small incremental steps and is tested as frequently as possible, preferably after each commit to the source code repository.

The TDD and CI paradigms have been used to develop the *Perylune* software throughout the research conducted for this thesis. The developed software uses the Travis-CI platform to run the developed tests automatically. The test results are available at <https://travis-ci.com/github/tomaszmrugalski/perylune>. The environment is configured to run tests in python 3.6, 3.7, and 3.8. An example result of the CI testing for all three Python software environments is presented in Fig. 61a. An excerpt from one execution, including a list of tests executed in presented in Fig. 61.

Travis CI Dashboard Changelog Documentation Help

Search all repositories

tomaszmrugalski / perylune build passing

My Repositories Running (0/0) +

Current Branches Build History Pull Requests More options

✓ tomaszmrugalski/perylune # 106
Duration: 3 min 5 sec
Finished: 2 hours ago

✓ tomaszmrugalski/dibbler # 108
Duration: 3 min 35 sec
Finished: 5 months ago

✓ master added upload_cesium target to makefile #106 passed
Commit ddb8344
Compare 809ab12...ddb8344
Branch master
Tomek Mrugalski
Ran for 1 min 4 sec
Total time 3 min 5 sec
2 hours ago
Restart build

Build jobs View config

Job #	Platform	OS	Language	Environment	Duration
✓ # 106.1	AMD64	Xenial	Python: 3.6	no environment variables set	1 min 2 sec
✓ # 106.2	AMD64	Xenial	Python: 3.7	no environment variables set	1 min
✓ # 106.3	AMD64	Xenial	Python: 3.8	no environment variables set	1 min 3 sec

(a) The overview of the test results presented on Travis-CI, a Continuous Integration test platform. The *Perylune* software is tested on 3 Python environments: 3.6, 3.7 and 3.8.

```

182 $ python --version
183 Python 3.8.6
184 $ pip --version
185 pip 20.3.1 from /home/travis/virtualenv/python3.8.6/lib/python3.8/site-packages/p
▶ 186 $ python --version
▶ 188 $ cp .travis-perylune.ini perylune.ini
▶ 189 $ pip install -r requirements.txt
353 $ make check
354 PYTHONPATH=.pollastro/src pytest -v --ignore=pollastro
355 ===== test session starts =====
356 platform linux -- Python 3.8.6, pytest-6.1.1, py-1.9.0, pluggy-0.13.1 -- /home/tr
/python3.8/bin/python
357 cachedir: .pytest_cache
358 rootdir: /home/travis/build/tomaszmrugalski/perylune, configfile: setup.cfg
359 collected 29 items
360
361 tests/almanac_yuma_test.py::test_loadYuma PASSED [ 3%]
362 tests/almanac_yuma_test.py::test_recreateSpaceMissionsLab1 PASSED [ 6%]
363 tests/almanac_yuma_test.py::test_yumaAppend PASSED [ 10%]
364 tests/almanac_yuma_test.py::test_yuma2orbit PASSED [ 13%]
365 tests/geom_test.py::test_solar_angle PASSED [ 17%]
366 tests/gpsdop_test.py::test_getObserverECEP PASSED [ 20%]
367 tests/gpsdop_test.py::test_getObserverLLA PASSED [ 24%]
368 tests/gpsdop_test.py::test_DopCalculation PASSED [ 27%]
369 tests/gpsdop_test.py::test_DopCalculation_many PASSED [ 31%]
370 tests/horizons_test.py::test_name_to_horizons_id PASSED [ 34%]
371 tests/interplanetary_test.py::test_escape_velocity PASSED [ 37%]
372 tests/interplanetary_test.py::test_transfer_vel PASSED [ 41%]
373 tests/orbcalc_test.py::test_deg2rad_case PASSED [ 44%]
374 tests/orbcalc_test.py::test_constant_rho PASSED [ 48%]
375 tests/orbcalc_test.py::test_constant_earth_radius PASSED [ 51%]
376 tests/orbit_tools_test.py::test_norm_zpi PASSED [ 55%]
377 tests/orbit_tools_test.py::test_norm_pipi PASSED [ 58%]
378 tests/orbitdb_test.py::test_db_stats PASSED [ 62%]
379 tests/orbitdb_test.py::test_load_db PASSED [ 65%]
380 tests/time_test.py::test_leap_year PASSED [ 68%]
381 tests/time_test.py::test_leap_years PASSED [ 72%]
382 tests/time_test.py::test_days_0 PASSED [ 75%]
383 tests/time_test.py::test_days_2019_09_22 PASSED [ 79%]
384 tests/time_test.py::test_days_2019_09_22_15h56m PASSED [ 82%]
385 tests/tle_test.py::test_tle_bad_init PASSED [ 86%]
386 tests/tle_test.py::test_tle2lines PASSED [ 89%]
387 tests/tle_test.py::test_tle3lines PASSED [ 93%]
388 tests/tle_test.py::test_tle_parse PASSED [ 96%]
389 tests/utls_test.py::test_url_to_filename PASSED [100%]
390
391 ===== warnings summary =====
392 perylune/orbit_tools.py:286

```

(b) A detailed view of the tests run on a single python version. List of tests is visible.

Figure 61: The *Perylune* software uses Travis-CI platform to run tests automatically. The tests are written using standard pytest framework, the default recommended solution for all Python software.

5.2.1. Test Warnings

The tests discovered great variety of problems throughout the development process. All major issues are already fixed. However, there's a small number of 3 remaining issues. When executed on the latest version as of 26 Dec. 2020, the tests report the following warnings:

```

===== warnings summary =====
tests/interplanetary_test.py::test_transfer_vel
  /home/thomson/devel/perylune/venv/lib/python3.8/site-packages/perylune/interplanetary.py:167: DeprecationWarning:

    Orbit.from_horizons is deprecated and will be removed in a future release, use Ephem.from_horizons instead

tests/interplanetary_test.py::test_transfer_vel
  /home/thomson/devel/perylune/venv/lib/python3.8/site-packages/astropy/units/decorators.py:234: UserWarning:

    Wrapping true anomaly to  $-\pi \leq \nu < \pi$ 

tests/interplanetary_test.py::test_transfer_vel
  /home/thomson/devel/perylune/venv/lib/python3.8/site-packages/perylune/interplanetary.py:168: DeprecationWarning:

    Orbit.from_horizons is deprecated and will be removed in a future release, use Ephem.from_horizons instead

-- Docs: https://docs.pytest.org/en/stable/warnings.html
===== 29 passed, 3 warnings in 5.38s =====

```

This can be traced back to two remaining problems. The first one is coming from the fact of using older API from the *Poliastro* package. The code obtains the orbital data directly, instead of obtaining the ephemerids first and then fitting an orbit to the ephemerid. The older API currently used is still working but may be removed in future versions of the dependent library. This is a potential forward compatibility problem but does not impact the correctness of the produced calculations.

The second problem is minor aesthetics. There is a general agreement to specify true anomaly ν in the range of between $-\pi$ and π . The *Perylune* software neglects to do that in one particular place where transfer velocities for interplanetary transfers are calculated. This is a minor problem that does not have any impact on the calculations. Nevertheless, this flaw will be fixed in the future.

5.2.2. Testing Dependencies – Poliastro

The *Poliastro* software [59] is a library used to conduct many operations. Some of them are critical for valid *Perylune* operations. As such, the author tested this dependency as well. Being much more mature software, it provides a substantially larger number of tests. When ran on the latest version as of Dec. 26, 2020, there were 45 failed and 854 passing tests. Several tests were marked as expected fail (10) or skipped for various reasons (20). The tests are failing in mainly two areas: plotting module (which is responsible for generating plots and charts) and hyperbolic orbits. The hyperbolic tests failures are a concern. However, hyperbolic trajectories were only a minuscule part of the thesis (when discussing escaping from Earth's sphere of influence), so any potential problems should not cast any significant doubt into the correctness of the results.

5.3. Reporting and Fixing Bugs

During the course of development, the author discovered several problems. The first one was a bug in CZML exporter used in the *Poliastro* software. The exporter code did not account for timezones correctly. The software

worked correctly only when running in the UTC timezone. The problem has been reported (see <https://github.com/poliastro/poliastro/pull/872>) and after some discussion with the Poliastro team, the author developed a patch that was contributed, accepted and included in the release 0.14 of Poliastro library.

Another problem was reported for the tle-tools software. The problem was the incompatibility with a new API of software dependency. This time the fix was developed, and tests were extended to test that the issue is indeed fixed. The report, together with the author's fix, is available (<https://github.com/FedericoStra/tletools/pull/7>). The fix was accepted and was the reason to publish release 0.2.3 of the tle-tools library.

5.4. Comparison with well-established sources

One of the most respected book in astrodynamics is [5] and its recently published second edition [6]. It is considered the best introductory book used by NASA, US Naval Academy, and countless universities during astrodynamics courses. It contains many numerical data, such as orbital details for planets, many example problems with solutions provided, etc. The author used the orbital data available in [6] to validate some of the results discussed in Section 4.4. The calculations were conducted for major planets, dwarf planets and asteroids using the same code. The results obtained for major planets were compared with [6] and proved to be consistent to a high degree. There were minor epsilon discrepancies. Those can be explained with two reasons. The first is a computational rounding error inherent to the floating-point notation. The second reason is more profound. With the exception of the Sun, the planets are the most massive objects in the Solar System and thus have very stable orbits. Nevertheless, they do perturb each other to some small degree. While the effect is small, it is measurable over longer periods. There is also the aspect of uncertainty and improved measurement precision over time. Reputable sources acknowledge those changes over time. For example, when comparing the first (1971) and second (2020) editions of the BMW book, there are small but notable differences in planets' orbital elements.

5.5. Appeal to the authority

Some data have been obtained from the NASA Horizons database. Being the most experienced space agency on this planet, NASA can be considered an authority. For some data provided, there is no alternative source of information that is more trustworthy.

5.6. Comparison with Geographical Data

Some calculations can be verified empirically. In particular, the Section 4.1 overlaid political borders over received NOAA satellite images. Some of the borders align with geographic features, such as peninsulas or seashores. The resulting image shows both the geographic features photographed as well as the calculated borders. The results are presented in Fig. 35. As can be inspected visually, the alignment is not perfect, but it is reasonably good. Those small discrepancies can probably be explained with the orbital perturbations over time (the calculations did not account for J2 perturbations, and the TLE data was several days old). Also, the period for calculating the azimuth was chosen arbitrarily. A different value may have produced different results. The conversion of TEME (inertial) coordinates to ECEF coordinates (fixed, rotating with Earth) can be calculated using several methods that vary in complexity (especially with regards to taking into consideration more subtle movements, such as precession and nutation) and precision. Since one reference system rotates and the other does not, the coordinates conversion is also highly dependant on time. There may be small measurement errors related to clock skew on the hardware used to record the transmission. Also, the AOS was calculated with the assumption of a perfect, unobstructed horizon. In practice, the transmission was received in an urban environment with buildings obstructing the line

of sight, especially low above the horizon. This means that the actual transmission likely started several seconds later than the theoretical calculated AOS.

5.7. Comparison with PW-Sat2 Data

The PW-Sat2 team published the results of the PW-Sat2 satellite orbital decay. The data is available as an image. Sadly, there is no numeric data available, at least not in an easily accessible format. In principle, some sources, such as Celestrak [9], publish updated TLE data for known active satellites based on radar and other measurements. At any given time, only the latest data is available. However, it is possible to develop a web crawler, which would periodically retrieve the data and store it in a way that would allow retrieving historical data. Such data could be used to calculate positions at specific intervals or, better yet, at the times of TLE data being published. This could have been used to obtain numerical reference data. However, such an endeavor would be a side project on its own.

The data obtained with the proposed atmospheric drag perturber have been plotted and overlaid over the plot published by PW-Sat2 team. The results are presented in Fig. 58b. The obtained decay matches reasonably closely with the actual degradation. The results could be further improved by simulating for a period longer than the 100 days simulated. However, since this was one of many computations and the simulation time was already taking hours, the author chose not to pursue this direction any further.

5.8. Additional Verification

The author determined that the validation methods described in earlier sections are good enough for a master thesis. However, several potential methods can be used to further increase the confidence in the obtained results. Some of those methods may be employed if higher requirements regarding data validation are necessary, such as for publications in scientific journals.

The calculations can be repeated in different software. This is problematic for several reasons. First and foremost, this thesis covered various types of calculations in vastly different areas – from satellite ground tracks and optical geometry to orbital maneuvers to perturbations to long term orbital positions spanning decades. Many of those tasks would require using different types of software. This may be a lesser problem for a more focused paper. However, for a broad thesis covering multiple problems, it would require a substantial amount of effort.

Some of the results obtained could be verified by comparing them with well-known information. For example, there was a great Jupiter and Saturn conjunction on Dec. 21, 2020. The position of Earth, Jupiter, and Saturn could be calculated for a period of perhaps a month before and after that date. If the calculations are correct, the angular distance between Jupiter and Saturn as observed from Earth should be smallest on the date of actual conjunction. While this would not prove the correctness, it would be attractive anecdotal evidence supporting the results' correctness.

Finally, it is possible to verify some results empirically. The satellite can be photographed using a camera connected to a telescope. The author, being an avid amateur astronomer, intends to conduct such experiments in the near future. One potential problem is that satellites are small and thus rather dark. The standard way to handle dark objects is to increase exposure. However, with the satellites moving fast, they will be recorded as trails rather than point objects. There are at least two potential solutions to that problem. First would be to observe the biggest satellite currently available – the International Space Station. When still in Sun light, it is bright enough to be photographed with single-digit milliseconds exposure. This is short enough to register as a static object rather than a smudged line. The second approach would be to record a line and use precise timing to determine the beginning and end of the recorded line to determine the satellite position. Both seem to be an interesting side project worth future investigation.

6. Achievements and Conclusions

This final, short chapter provides a summary of achieved goals and discusses major conclusions. A survey of expected directions of the future work concludes this dissertation.

6.1. Achieved Goals

During the course of research and development activities related to this thesis, the following goals were achieved:

1. The developed *Perylune* software provides capabilities for managing orbital information, a database of known satellites and asteroids, propagate orbits, calculating various orbital maneuvers, and applying them to orbits, producing various types of graphs and charts, and export data in various formats, such as CSV or CZML. The CZML data can be visualized using the visually attractive 3D web interface.
2. The developed *Perylune* software provides a platform that integrates many existing libraries. It takes advantage of existing projects, most notably Poliastro (orbit definitions, propagation, atmospheric models), tle-tools (conversion of TLE data to Keplerian elements), plotly (graphical library for plots, charts, and data visualization), and others.
3. The georeferencing for NOAA satellite images was implemented.
4. The **developed *Perylune* software was released under MIT license**, which has permissive open-source terms that impose very few restrictions and allows code reuse without any fees for any purpose, including educational, personal and commercial. The source code is available on GitHub, a leading open-source software development platform.
5. A **survey of over a million currently known asteroids** was conducted. Many aspects were evaluated.
6. Many scenarios were simulated for orbital adjustments, hypothetical missions spanning areas between LEO to interplanetary.
7. **Atmospheric models were implemented** based on U.S. Standard Atmosphere 1976 that modeled atmospheric drag of a PW-Sat2 satellite.
8. The developed *Perylune* software now can interact with and automatically import data from three on-line databases: NASA Horizons for planets and asteroids, Minor Planet Center database for latest asteroid data, and Celestrak for latest satellite information.
9. **Transfer windows for all major planets were studied, focusing on Mars and Venus.** Several optimal transfer windows were plotted in detail in the form of so-called pork-chop charts for Mars and Venus transfers in 2021. Distances for inner (Mercury, Venus, Mars) and a selection of outer major and dwarf planets (Jupiter, Saturn, Uranus, Pluto) for years 2012-2040 were plotted and studied.
10. The developed *Perylune* software is written in Python that offers several advantages. It is portable so that it can be used on Windows machines during development and then run on embedded platforms, potentially as

on-board software for future space missions. It can also run on Linux systems on the servers, thus providing back-end services for websites and other on-line services.

6.2. Conclusions

During numerous experiments and research conducted by the author, several conclusions have emerged. The results obtained in this dissertation clearly prove that:

1. Python is an attractive environment for research and scientific activities. It covers the language itself and a rich set of libraries, including many related to astrodynamics, orbital predictions and more. There is no need to reinvent the wheel by reimplementing many well-known algorithms. However, the huge variety of available libraries, many aims to address similar goals, may be overwhelming at first.
2. The map projection problem is surprisingly complex when dealing with large-scale imaging covering the whole continent, such as the NOAA satellite images. The developed gnomonic projection provides reasonably good results that are validated with overlaid country boundaries. The close fit of political borders and actual physical objects, such as coastlines and notable peninsulas, improves image readability.
3. There are many ways how TEME coordinates, produced by SGP4 models commonly used in the space industry, can be converted to geographical coordinates. The conversion to geographical coordinates is complex and involves aspects such as Julian date calculation, Greenwich Mean Sidereal Time calculation, considers precession (and in some algorithms also nutation), may use different reference systems, such as WGS-84, to model Earth oblateness. The author implemented several alternative algorithms for those conversions. **The impact of the choice of the algorithms on the final precision may be an interesting area for future study with a high chance of producing research papers.** The final precision can be assessed by selecting some well known geographical landmarks, such as Hel peninsula of Gibraltar that's recognizable on the NOAA satellite images.
4. Author reviewed existing Polish satellites, the launch services used and compared them with services currently available now or expected to be available in the near future. The price of launching a CubeSat in 2020 went down considerably. With the lowest **price for launching a 1U satellite as low as 50.000 USD**, the prospect of initiating a successful **cubesat project in Gdańsk in the near future seems viable.**
5. A series of simulations showed that it is possible to perform rendezvous with existing satellites, such as defunct NOAA-17, from an orbit offered by low-cost launch services, such as RocketLab. The proposed orbital maneuvers have only reasonably small fuel requirements. This exercise can be useful for designing a low-cost CubeSat that can either revive or deorbit existing high-value satellites.
6. It is possible to use natural phenomena, such as Earth oblateness, Moon gravity, and others, that are commonly modeled as J2 perturbations to conduct certain orbital maneuvers without using any propellant. The trade-off between time (the change of RAAN orbital parameter in the simulated environment took 113 days) and fuel expenditure is often very favorable.
7. Alarmist media often scare readers with impending 'doom' of incoming close fly-by of asteroids. The only necessary information needed to perform the analysis is the asteroid designation. It is easy to use the NASA Horizons database to obtain ephemerids and assess whether the asteroid is a threat or not, effectively debunking alarmist press.
8. **An asteroid survey for over a million of currently known asteroids was conducted.** The research is based on the Minor Planet Center database, with all data available in Nov. 2020. The process of asteroids discovery

was discussed. The energy requirements for reaching all asteroids were investigated, and the top 20 asteroids that are easiest to reach from Earth were selected. A complete list of Δv requirements for all asteroids is available on the *Perylune* project site.

9. The asteroid survey was implemented in a mostly automated way, although some manual steps are necessary. With some additional effort, it is possible to fully automate the survey generation. Given the dynamic changes caused by the rapid discovery of new asteroids and natural perturbations causing the asteroid orbits to change, it may be useful to publish such a survey periodically.
10. The asteroid survey may be useful for several goals. The asteroids that are easy to reach are **potentially attractive targets for missions**. While CubeSat mission that leaves Earth sphere of influence is currently out of reach for Polish universities, there are several demonstrated or upcoming examples of such deployments. MarCO-A and MarCO-B were two 6U CubeSats developed by NASA that completed Mars fly-by. NASA has contracted RocketLab to launch a CubeSat towards the Moon to validate the concept of NRHO orbit. RocketLab plans to launch a Venus CubeSat mission in 2023. With the price of launch going down significantly and expected radical drops in the next 5 years due to expected new technologies (SpaceX's Starship is expected to have over 100 tons lifting capability and a price tag of single flight expected to be in single-digit million USD due to full reusability), it can be speculated that interplanetary CubeSat mission may be within financial capabilities of Polish universities within a decade.
11. The fact that some asteroids have orbits so close to Earth they can be temporarily captured as Earth's mini-moons is not commonly known. This information spurs general audience interest in asteroids and makes it an attractive research area.
12. A study of the perturbations phenomena used PW-Sat2 (a CubeSat developed by Warsaw University of Technology that demonstrated a deployable deorbit sail) as an example and demonstrated several aspects. **An atmospheric drag model was developed that matches experimental data provided by PW-Sat2 team**, although the data format proved to be challenging to work with.
13. A concept of **repurposing the sail as a solar sail to use solar radiation pressure** was proposed and simulated. Although the obtained force is small, it is perpetually available and does not consume any fuel. The simulations demonstrated that this could be used to raise the apogee of the orbit. While the studies conducted so far suggest it cannot be used as sole propulsion, it may be used together with other solutions, such as ion engines, to substantially decrease fuel consumption, thus increasing mission lifetime and overall capabilities.
14. A demonstration site with 3D visualizations of all six Polish satellites was developed. This website can be used to predict fly-overs over Poland or any other point of interest on Earth, compare orbits, investigate orbital parameters, and more. The realistic model calculates the positions of the Sun and the Moon as well as proper shading. This can be used to plan visual observations of the satellites in favorable conditions (the observer on the ground is experiencing after sunset conditions, while the satellite is still fully in Sun light).

In light of the presented achievements, it can be stated that astrodynamics is an attractive, rich area that should be a field of further studies.

6.3. Organizational Suggestions

The Space and Satellite Technologies (*pol.* Technologie Kosmiczne i Satelitarne) or TKiS is a young and attractive new study field held jointly by Gdańsk University of Technology, Gdynia Maritime University, and Polish Naval Academy (with University of Gdańsk joining the initiative in 2020). The following observations and

suggestions are made from the point of a graduate and an amateur space enthusiast that strongly supports the development of space-related activities in Poland. They should not be considered a critique but rather suggestions for potential future improvements.

1. The topic of astrodynamics was covered only briefly during the lectures and classes (Space Missions lecture). This rich field merits its own class. The course could study on-going and planned space missions, attempt to recreate trajectories of historic missions, and encourage students to explore what-if scenarios, such as to conduct speculations regarding upcoming launchers' capabilities, backed by some calculations, design space missions of varying complexity and more.
2. The information about TKiS studies is difficult to find on the university's websites. As an appealing future-looking field, this should be made much more prominent, leading to more ambitious projects in the future.
3. The English information about TKiS studies is practically non-existent.
4. Students create many software projects. There is no shared place to publish them. One potential idea would be to organize a group on existing platforms, such as Github or GitLab, to facilitate TKiS developed projects. The author of this thesis, together with Slawomir Figiel, another TKiS graduate, initiated such a space on the GitHub platform and made several developed projects available. This is especially important for more complex, long term projects that may span several years of student classes.
5. Students should be encouraged to create profiles on platforms that are less known among students, such as ResearchGate, and encouraged to publish their research reports there. Although most student reports will have limited scientific value, follow-up studies will be based on earlier works, eventually leading to more ambitious projects.

6.4. Future work

As this research proved the validity of several proposals, the next logical step is to refine and expand the research in the most promising directions. In particular:

1. One of the author's goals was to increase the accessibility of the field of astrodynamics. The literature available in Polish is limited. Even the most popular source of information among students – Wikipedia – has fewer Polish articles than its English counterpart. Even then, the articles that are there are substantially shorter. As such, the author, a Wikipedia editor since 2008, started a process of extending existing and adding missing Polish articles.
2. The asteroid survey is particularly interesting for the author. He intends to pursue this research, especially in the direction of automated period survey and better-selecting asteroids of interest, from the potential rendezvous targets for near-future missions and potential mining targets for the longer term. Being an enthusiastic and well equipped amateur astronomer, the author intends to augment this survey with observational data.
3. The asteroid survey is also useful for the author for another reason. He hopes to start a project searching for unknown asteroids. While this topic is challenging, many amateurs routinely discover new asteroids.
4. Developing a website that takes the name of an asteroid and produces easy to understand charts may be an interesting student project. If sufficiently developed, this may also be an educational asset that may promote interest in space in the general population.

5. The geomapping algorithm implemented will be used in the satellite ground stations project that the author started with another TKiS graduate. The software will require some clean-up and will be published as a python module, available for others for easy installation. The sources are available on the GitHub platform, but they are a bit difficult to use.

A. Ephemerides For Asteroid 1998 OR₂

The following text is an example ephemerides for a near Earth asteroid 1998 OR₂ for the period of 10 days before and after its closest approach to Earth in April 2020. The ephemerides were generated using NASA's Horizons database.

```
*****
JPL/HORIZONS                52768 (1998 OR2)                2020-Oct-29 16:25:01
Rec #: 52768 (+COV) Soln.date: 2020-Oct-14_07:45:57 # obs: 5235 (1987-2020)

IAU76/J2000 helio. ecliptic osc. elements (au, days, deg., period=Julian yrs):

EPOCH= 2458104.5 ! 2017-Dec-17.00 (TDB)                Residual RMS= .22755
EC= .5727252702158157   QR= 1.016880935119138   TP= 2457613.8493807735
OM= 27.06517785351406   W= 174.4506086196864   IN= 5.879436792982998
A= 2.379922949416556   MA= 131.7140427235783   ADIST= 3.742964963713973
PER= 3.67158           N= .268447718           ANGMOM= .02175419
DAN= 3.7196           DDN= 1.01862           L= 201.5447974
B= .5675753           MOID= .0133135           TP= 2016-Aug-13.3493807735

Asteroid physical parameters (km, seconds, rotational period in hours):
GM= n.a.                RAD= .875                ROTPER= 4.112
H= 15.8                 G= .150                 B-V= n.a.
                        ALBEDO= n.a.            STYP= n.a.

ASTEROID comments:
1: soln ref.= JPL#306, PHA OCC=0          radar( 3 delay, 0 Dop.)
2: source=ORB
*****

*****
Ephemeris / WWW_USER Thu Oct 29 16:25:01 2020 Pasadena, USA / Horizons
*****
Target body name: 52768 (1998 OR2)           {source: JPL#306}
Center body name: Earth (399)                {source: DE431}
Center-site name: BODY CENTER
*****
Start time      : A.D. 2020-Apr-20 00:00:00.0000 TDB
Stop time       : A.D. 2020-May-10 00:00:00.0000 TDB
Step-size       : 1440 minutes
*****
Center geodetic : 0.00000000,0.00000000,0.00000000 {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.00000000,0.00000000,0.00000000 {E-lon(deg),Dxy(km),Dz(km)}
Center radii    : 6378.1 x 6378.1 x 6356.8 km      {Equator, meridian, pole}
Small perturbers: Yes                            {source: SB431-N16}
```

Output units : KM-S
 Output type : GEOMETRIC cartesian states
 Output format : 3 (position, velocity, LT, range, range-rate)
 EOP file : eop.201029.p210120
 EOP coverage : DATA-BASED 1962-JAN-20 TO 2020-OCT-29. PREDICTS-> 2021-JAN-19
 Reference frame : Ecliptic of J2000.0

Initial IAU76/J2000 heliocentric ecliptic osculating elements (au, days, deg.):

EPOCH= 2458104.5 ! 2017-Dec-17.00 (TDB) Residual RMS= .22755
 EC= .5727252702158157 QR= 1.016880935119138 TP= 2457613.8493807735
 OM= 27.06517785351406 W= 174.4506086196864 IN= 5.879436792982998

Equivalent ICRF heliocentric equatorial cartesian coordinates (au, au/d):

X= 3.528821141519463E+00 Y= 3.393369612823992E-01 Z=-2.057891332160555E-03
 VX= 1.701391064757778E-03 VY= 5.566191998686528E-03 VZ= 2.95422242504058E-03

Asteroid physical parameters (km, seconds, rotational period in hours):

GM= n.a. RAD= .875 ROTPER= 4.112
 H= 15.8 G= .150 B-V= n.a.
 ALBEDO= n.a. STYP= n.a.

JDTDB

X Y Z
 VX VY VZ
 LT RG RR

\$\$\$SOE

2458959.500000000 = A.D. 2020-Apr-20 00:00:00.0000 TDB

X = -6.266593046617419E+06 Y = 7.005017946413181E+06 Z = 1.070403098959085E+05
 VX= 9.614999132126698E-01 VY=-7.515970464349345E+00 VZ=-3.785887554472560E+00
 LT= 3.135359173811720E+01 RG= 9.399570334298644E+06 RR=-6.285402112415858E+00

2458960.500000000 = A.D. 2020-Apr-21 00:00:00.0000 TDB

X = -6.183556136587858E+06 Y = 6.354201578219132E+06 Z = -2.200503117879541E+05
 VX= 9.605422489656963E-01 VY=-7.549068608213721E+00 VZ=-3.785476219443566E+00
 LT= 2.958408268005982E+01 RG= 8.869084864330359E+06 RR=-5.984256117961132E+00

2458961.500000000 = A.D. 2020-Apr-22 00:00:00.0000 TDB

X = -6.100631284396052E+06 Y = 5.700563900126768E+06 Z = -5.470592783782808E+05
 VX= 9.588926294286075E-01 VY=-7.581298196730101E+00 VZ=-3.783998836220364E+00
 LT= 2.791064647509702E+01 RG= 8.367401311138370E+06 RR=-5.616732336544743E+00

2458962.500000000 = A.D. 2020-Apr-23 00:00:00.0000 TDB

X = -6.017881590776831E+06 Y = 5.044174958884024E+06 Z = -8.738948606472174E+05
 VX= 9.564759198120250E-01 VY=-7.612773705121025E+00 VZ=-3.781464159745512E+00
 LT= 2.635415845481632E+01 RG= 7.900777941690866E+06 RR=-5.170569218303377E+00

2458963.500000000 = A.D. 2020-Apr-24 00:00:00.0000 TDB

X = -5.935376261474863E+06 Y = 4.385095265416775E+06 Z = -1.200466088055260E+06
 VX= 9.532254956536086E-01 VY=-7.643600847622652E+00 VZ=-3.777880972002776E+00
 LT= 2.493910214826060E+01 RG= 7.476554733340125E+06 RR=-4.633210497384630E+00

2458964.500000000 = A.D. 2020-Apr-25 00:00:00.0000 TDB
X = -5.853190008735687E+06 Y = 3.723376686108320E+06 Z = -1.526682749987233E+06
VX = 9.490798215677003E-01 VY = -7.673873576304459E+00 VZ = -3.773258130700895E+00
LT = 2.369341421763527E+01 RG = 7.103106886717025E+06 RR = -3.993646561746774E+00

2458965.500000000 = A.D. 2020-Apr-26 00:00:00.0000 TDB
X = -5.771402809208870E+06 Y = 3.059063556037452E+06 Z = -1.852455425160171E+06
VX = 9.439776931140074E-01 VY = -7.703672051315840E+00 VZ = -3.767605226056477E+00
LT = 2.264763910860051E+01 RG = 6.789591396264276E+06 RR = -3.245376032638274E+00

2458966.500000000 = A.D. 2020-Apr-27 00:00:00.0000 TDB
X = -5.690100110735595E+06 Y = 2.392193902745474E+06 Z = -2.177695603706154E+06
VX = 9.378527316785004E-01 VY = -7.733062213915629E+00 VZ = -3.760934090748902E+00
LT = 2.183308592076555E+01 RG = 6.545394493911498E+06 RR = -2.390274656379933E+00

2458967.500000000 = A.D. 2020-Apr-28 00:00:00.0000 TDB
X = -5.609373514145851E+06 Y = 1.722800618123550E+06 Z = -2.502315974412711E+06
VX = 9.306280794100807E-01 VY = -7.762097475410634E+00 VZ = -3.753261043690411E+00
LT = 2.127885299673693E+01 RG = 6.379239643312430E+06 RR = -1.442325775081082E+00

2458968.500000000 = A.D. 2020-Apr-29 00:00:00.0000 TDB
X = -5.529321863440186E+06 Y = 1.050912388547180E+06 Z = -2.826230911612379E+06
VX = 9.222124272322638E-01 VY = -7.790822567699109E+00 VZ = -3.744609072444963E+00
LT = 2.100802409169358E+01 RG = 6.298047180172035E+06 RR = -4.292682001593992E-01

2458969.500000000 = A.D. 2020-Apr-30 00:00:00.0000 TDB
X = -5.450052594986692E+06 Y = 3.765542247636928E+05 Z = -3.149357098529342E+06
VX = 9.124979973362741E-01 VY = -7.819278752283148E+00 VZ = -3.735008558564601E+00
LT = 2.103393837104350E+01 RG = 6.305816085675646E+06 RR = 6.098070634566155E-01

2458970.500000000 = A.D. 2020-May-01 00:00:00.0000 TDB
X = -5.371683199937150E+06 Y = -3.002524687287466E+05 Z = -3.471614103979710E+06
VX = 9.013596580272640E-01 VY = -7.847509075664858E+00 VZ = -3.724495354849281E+00
LT = 2.135780358898607E+01 RG = 6.402908435423356E+06 RR = 1.631200281973823E+00

2458971.500000000 = A.D. 2020-May-02 00:00:00.0000 TDB
X = -5.294342791116834E+06 Y = -9.794902605078408E+05 Z = -3.792924676573067E+06
VX = 8.886529971539723E-01 VY = -7.875563331890834E+00 VZ = -3.713106333142711E+00
LT = 2.196861952791604E+01 RG = 6.586026447140750E+06 RR = 2.595303369219021E+00

2458972.500000000 = A.D. 2020-May-03 00:00:00.0000 TDB
X = -5.218173943671092E+06 Y = -1.661146539392374E+06 Z = -4.113214601556786E+06
VX = 8.742098855447651E-01 VY = -7.903504754196440E+00 VZ = -3.700874080723334E+00
LT = 2.284543485674925E+01 RG = 6.848889069783736E+06 RR = 3.473496123300573E+00

2458973.500000000 = A.D. 2020-May-04 00:00:00.0000 TDB
X = -5.143334972359717E+06 Y = -2.345215142118899E+06 Z = -4.432412143175505E+06
VX = 8.578335098152188E-01 VY = -7.931421937175241E+00 VZ = -3.687822940798885E+00
LT = 2.396100419809273E+01 RG = 7.183328344694537E+06 RR = 4.250775608960323E+00

2458974.500000000 = A.D. 2020-May-05 00:00:00.0000 TDB
X = -5.070002454062477E+06 Y = -3.031699307301667E+06 Z = -4.750447258760888E+06
VX = 8.392988924094844E-01 VY = -7.959447474789613E+00 VZ = -3.673967822760738E+00
LT = 2.528558300533416E+01 RG = 7.580427081132155E+06 RR = 4.924312828882201E+00

2458975.500000000 = A.D. 2020-May-06 00:00:00.0000 TDB

```

X =-4.998373209895656E+06 Y =-3.720616406421098E+06 Z =-5.067250841177939E+06
VX= 8.183658259410080E-01 VY=-7.987779229341541E+00 VZ=-3.659315990606120E+00
LT= 2.678994281184138E+01 RG= 8.031422805241358E+06 RR= 5.499851985626780E+00
2458976.500000000 = A.D. 2020-May-07 00:00:00.0000 TDB
X =-4.928664522703320E+06 Y =-4.412004288037807E+06 Z =-5.382754219626738E+06
VX= 7.948064936318886E-01 VY=-8.016694517136663E+00 VZ=-3.643871065376768E+00
LT= 2.844727983130501E+01 RG= 8.528279944040757E+06 RR= 5.987891896833698E+00
2458977.500000000 = A.D. 2020-May-08 00:00:00.0000 TDB
X =-4.861111534228548E+06 Y =-5.105928158477476E+06 Z =-5.696889053199851E+06
VX= 7.684413121611477E-01 VY=-8.046547025279764E+00 VZ=-3.627637847488085E+00
LT= 3.023413119833846E+01 RG= 9.063964507444372E+06 RR= 6.400715978493062E+00
2458978.500000000 = A.D. 2020-May-09 00:00:00.0000 TDB
X =-4.795961660936981E+06 Y =-5.802486302622564E+06 Z =-6.009587617898123E+06
VX= 7.391696861331987E-01 VY=-8.077743449150306E+00 VZ=-3.610626347218347E+00
LT= 3.213059309187363E+01 RG= 9.632509480010612E+06 RR= 6.750508968427262E+00
2458979.500000000 = A.D. 2020-May-10 00:00:00.0000 TDB
X =-4.733467033373579E+06 Y =-6.501813152774006E+06 Z =-6.320783361957833E+06
VX= 7.069835869692049E-01 VY=-8.110707492903781E+00 VZ=-3.592853839276641E+00
LT= 3.412013201168554E+01 RG= 1.022895824306769E+07 RR= 7.048368974593980E+00
$$$EOE

```

Coordinate system description:

Ecliptic at the standard reference epoch

Reference epoch: J2000.0

X-Y plane: adopted Earth orbital plane at the reference epoch

Note: obliquity of 84381.448 arcseconds (IAU76) wrt ICRF equator

X-axis : ICRF

Z-axis : perpendicular to the X-Y plane in the directional (+ or -) sense of Earth's north pole at the reference epoch.

Symbol meaning:

- JDTDB Julian Day Number, Barycentric Dynamical Time
- X X-component of position vector (km)
- Y Y-component of position vector (km)
- Z Z-component of position vector (km)
- VX X-component of velocity vector (km/sec)
- VY Y-component of velocity vector (km/sec)
- VZ Z-component of velocity vector (km/sec)
- LT One-way down-leg Newtonian light-time (sec)
- RG Range; distance from coordinate center (km)
- RR Range-rate; radial velocity wrt coord. center (km/sec)

Geometric states/elements have no aberrations applied.

Computations by ...

Solar System Dynamics Group, Horizons On-Line Ephemeris System

4800 Oak Grove Drive, Jet Propulsion Laboratory

Pasadena, CA 91109 USA

Information : <https://ssd.jpl.nasa.gov/>

Documentation: https://ssd.jpl.nasa.gov/?horizons_doc

Connect : <https://ssd.jpl.nasa.gov/?horizons> (browser)

telnet ssd.jpl.nasa.gov 6775 (command-line)

e-mail command interface available

Script and CGI interfaces available

Author : Jon.D.Giorgini@jpl.nasa.gov

Bibliography

- [1] Analytical Graphics, Inc., “*Systems Tool Kit (STK)*”, <https://www.agi.com/products/stk>, retrieved Dec. 2020
- [2] D. Vallado, P. Crawford, R.Hujsak, T.S.Kelso, “*Revisiting Spacetrack Report #3*”, AIAA Astrodynamics Specialist Conference, 2006
- [3] R. Makówka, “*NASA ostrzega: W stronę Ziemi pędzi asteroida. Koniec świata może nastąpić 29 kwietnia 2020?*”, <https://www.antyradio.pl/News/NASA-ostzega-W-strone-Ziemi-pedzi-asteroida-Koniec-swiata-moze-nastapic-29-kwietnia-2020-39951>, AntyRadio, Mar. 2020
- [4] E. Aldrin, “*Line-of-sight guidance techniques for manned orbital rendezvous*”, PhD thesis, MIT press, 1963
- [5] R. Bate, D. Mueller, J. White, “*Fundamentals of astrodynamics*”, Dover Publications, 1971
- [6] R. Bate, D. Mueller, J. White, W. Saylor “*Fundamentals of astrodynamics*”, Second Edition, Dover Publications, 2020
- [7] B. Bolin, et. al., “*Characterization of Temporarily-Captured Minimoons 2020 CD3 by Keck Time-resolved Spectrophotometry*”, <https://arxiv.org/abs/2008.05384>, preprint, accepted for Astrophysical Journal Letters, Aug 2020
- [8] J. Bowen, M. Villa, A. Williams, “*CubeSat based Rendezvous, Proximity Operations, and Docking in the CPOD Mission*”, 29th Conference on Small Satellites, AIAA, Utah State University, 2015
- [9] T.S. Kelso, “*Celestrak service*”, <https://celestrak.com/>, an on-line service, retrieved Nov. 2020
- [10] Celestrak, “*Yuma format*”, <https://celestrak.com/GPS/almanac/Yuma/definition.php>, retrieved Dec. 2020
- [11] Celestrak, “*Definition of a SEM Almanac*”, <https://celestrak.com/GPS/almanac/SEM/definition.php>, retrieved on May 2019
- [12] AnalyticalGraphics, Inc., “*Cesium*”, on-line service, <https://cesium.com/cesiumjs/>, retrieved Dec. 2020
- [13] K. Ciołkowski, “*Exploring world spaces with rocket devices*”, 1903, “*Selected works of Konstantin Tsiolkovsky*”, Univ. of Hawaii, 2004
- [14] NASA & NOAA, “*U.S. Standard Atmosphere*”, <https://ntrs.nasa.gov/api/citations/19770009539/downloads/19770009539.pdf?attachment=true>, retrieved Dec. 2020
- [15] H. Curtis, “*Orbital Mechanics for Engineering Students*”, Elsevier, 2010
- [16] Analytical Graphics, Inc, “*Cesium Language (CZML) Guide*”, <https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Guide>, retrieved Dec. 2020
- [17] Consultative Committee for Space Data Systems, “*XML specification for Navigation Data Messages*”, <https://public.ccsds.org/Pubs/505x0b1.pdf>, retrieved Dec. 2020 505.0-B-1,
- [18] R.W. Farquhar, “*The control and use of libration-point satellites*”, PhD thesis, Stanford University, 1968
- [19] P. Fortescue, G. Swinerd, J. Stark, “*Spacecraft Systems Engineering*”, 4th Ed, Wiley, 2011

- [20] NASA, “*General Mission Analysis Tool*”, <https://sourceforge.net/projects/gmat>, retrieved Dec. 2020
- [21] P. Kwan, “*Interface Control Document ICD-GPS-870C*”, <https://www.gps.gov/technical/icwg/ICD-GPS-870C.pdf>, retrieved May 2019
- [22] C. Peat, “*Heavens Above*”, <https://heavens-above.com/>, retrieved Dec. 2020
- [23] A. Hein, N. Perakis, M. Eubanks, A. Hibberd et al., “*Project Lyra: Sending a Spacecraft to 1I/’Oumuamua (former A/2017 U1), the Interstellar Asteroid*”, preprint, Oct. 2018
- [24] A. Hibberd, N. Perakis, A. Hein, “*Sending a Spacecraft to Interstellar Comet C/2019 Q4 (Borisov)*”, preprint, Sep. 2019
- [25] A. Hibberd, A. Hein, M. Banks, “*Project Lyra: Catching 1I/’Oumuamua – Mission Opportunities After 2024*”, *Acta Astronautica*, May 2020
- [26] D. Izzo, “*Revisiting Lambert’s problem*”, <https://www.esa.int/gsp/ACT/doc/MAD/pub/ACT-RPR-MAD-2014-RevisitingLambertProblem.pdf>, retrieved Apr. 2020
- [27] G. Petit, B. Luzum, “*IERS Conventions (2010)*”, International Earth Rotation and Reference Systems Service, IERS Technical Note 36, 2010
- [28] L.G.Jacchia, “*Standard Jacchia Reference Atmosphere 1977*”, <https://ccmc.gsfc.nasa.gov/modelweb/atmos/jacchia.html>, 1977
- [29] T.S. Kelso, “*Orbital Coordinate Systems, Part III*”, <https://celestrak.com/columns/v02n03/>, *Satellite Times*, Jan. 1996
- [30] J. Kepler, “*The Harmonies of the World*”, 1619, *Encyclopaedia Britannica*, published 1952
- [31] M. Kopernik, “*De revolutionibus orbium coelestium*” (pol. “*O obrotach sfer niebieskich*”), 1543, published 1953
- [32] J. Kovalevsky, “*Comparison of Old and New Concepts: Reference Systems*”, *Proceedings of the IERS Workshop on the Implementation of the New IAU Resolutions*, IERS Technical Note. 2002
- [33] J. Kryński, “*Major Concepts of Recent Celestial and Terrestrial Reference Systems*”, *Annual of Navigation*, 2004
- [34] Lowell Observatory, “*Naming asteroids*”, <https://lowell.edu/sga/naming-asteroids>, retrieved Nov. 2020
- [35] M. Bloch, Mapshape service, <https://github.com/mbloch/mapshaper>, retrieved Aug. 2020
- [36] C. Marcos, R. Marcos, “*On the orbital evolution of meteoroid 2020 CD3, a temporarily captured orbiter of the Earth–Moon system*”, *Monthly Notices of the Royal Astronomical Society*, May 2020
- [37] J. Meeus, “*Astronomical Algorithms*”, Willmann-Bell, 2nd Ed, 2000
- [38] L. Modarski, “*Pierwszy polski satelita niemal gotowy do lotu. Powstał we Wrocławiu*”, *Radio Wrocław*, Sep. 2018
- [39] International Astronomical Union, “*Minor Planet Center*”, <https://minorplanetcenter.net/>, retrieved Nov. 2020
- [40] Minor Planet Center, “*Date format*”, <https://minorplanetcenter.net/iau/info/PackedDates.html>, retrieved May 2019

- [41] Minor Planet Center, “*Residuals*”, <https://www.minorplanetcenter.net/iau/info/Residuals.html>, retrieved May 2019
- [42] International Astronomical Union, “*Export Format for Minor-Planet Orbits*”, <https://www.minorplanetcenter.net/iau/info/MPOrbitFormat.html>, retrieved Nov. 2020
- [43] Web service, N2YO, <https://n2yo.com>, retrieved Oct. 2020
- [44] J. J. Retyk, “*Narratio Prima*”, Fundacja Nicolaus Copernicus, 2015
- [45] E. Mahoney, “*NASA Selects Blue Origin, Dynetics, SpaceX for Artemis Human Landers*”, <https://www.nasa.gov/feature/nasa-selects-blue-origin-dynetics-spacex-for-artemis-human-landers>, NASA, April 2020
- [46] NASA, “*Cubesat 101: Basic concepts and Processes for First-Time CubeSat Developers*”, NASA CubeSat Launch Initiative, 2017
- [47] NASA, “*NASA Science: Mars 2020 Mission*”, <https://mars.nasa.gov/mars2020/>, retrieved Dec. 2020
- [48] G. Hautaluoma, J. Handal, “*New Data Confirm 2020 SO to be the Upper Centaur Rocket Booster from the 1960’s*”, <https://www.nasa.gov/feature/new-data-confirm-2020-so-to-be-the-upper-centaur-rocket-booster-from-the-1960-s>, Dec. 2020
- [49] NASA, “*The SPICE Toolkit*”, <https://naif.jpl.nasa.gov/naif/toolkit.html>, retrieved Dec. 2020
- [50] N. Kelso, T. Patterson, “*Natural Earth*”, <https://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-0-countries/>, retrieved on July 2020
- [51] I. Newton, “*Philosophiæ Naturalis Principia Mathematica*”, 1687, polish edition, Copernicus Center Press, 2017
- [52] National Oceanic And Atmospheric Administration, “*NOAA 17 AVHRR Subsystem Summary*”, <https://www.ospo.noaa.gov/Operations/POES/NOAA17/avhrr.html>, retrieved Oct. 2020
- [53] M. Bernardi, “*The NOAA-APT project: How it work?*”, <https://noaa-apt.mbernardi.com.ar/>, retrieved Aug. 2020
- [54] C. Bailer-Jones, D. Farnocchia, Q. Ye, K. Meech, M. Micheli, “*A search for the origin of the interstellar comet 2I/Borisov*”, Astronomy & Astrophysics, ESO 2020
- [55] A. Hibberd, “*Optimum Interplanetary Trajectory*”, https://github.com/AdamHibberd/Optimum_Interplanetary_Trajectory, retrieved Dec. 2020
- [56] International Astronomical Union, “*IAU 2006 General Assembly: Result of the IAU Resolution votes*”, <https://www.iau.org/news/pressreleases/detail/iau0603/>, Prague, 2006
- [57] Claudius Ptolemy, “*Almagest*”, 140CE, Greece, published by University of Chicago 1952
- [58] RocketLab, “*Launch: Payload User’s Guide*”, <https://www.rocketlabusa.com/assets/Uploads/Rocket-Lab-Launch-Payload-Users-Guide-6.5.pdf>, Aug 2020
- [59] J.L. Cano Rodríguez et al., “*Poliastro*”, software package, <https://docs.poliastro.space>, retrieved Dec. 2020
- [60] J.L. Cano Rodríguez et al., “*Atmosphere module*”, https://docs.poliastro.space/en/stable/api/safe/atmosphere/atmosphere_index.html, retrieved Dec. 2020

- [61] J.L. Cano Rodríguez et al., “*Poliastro documentation: poliastro.twobody.propagation*”, <https://docs.poliastro.space/en/latest/autoapi/poliastro/twobody/propagation/index.html>, retrieved Dec. 2020
- [62] T. Mrugalski, “*Perylune project*”, <https://perylune.space>, retrieved Nov. 2020
- [63] Warsaw University of Technology, “*PW-Sat*”, <https://pw-sat.pl/pw-sat/>, retrieved Dec. 2020
- [64] Warsaw University of Technology, “*PW-Sat2 Mission Status*”, <https://pw-sat.pl/obecny-status-misji-current-mission-status/>, retrieved Dec. 2020
- [65] Warsaw University of Technology, “*PW-Sat2 Deorbit Sail Monitoring*”, <https://sail.pw-sat.pl/>, retrieved on Nov. 2020
- [66] Kosmonauta.net, “*PW-Sat3 – status prac (kwiecień 2020)*”, <https://kosmonauta.net/2020/04/pw-sat-3-status-prac-kwiecien-2020/>, retrieved Dec. 2020
- [67] T. Mrugalski, “*Data results repository for Perylune project*”, <https://github.com/tomaszmrugalski/perylune-data>, retrieved Nov. 2020
- [68] Python Software Foundation, “*Installing packages using pip and virtual environments*”, <https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/>, retrieved Dec. 2020
- [69] T.Mrugalski, S.Figiel, “*Svarog project*”, <https://svarog.space>, retrieved Aug 2020
- [70] J. Wertz, W. Larson,, “*Space Mission Analysis and Design*”, 3rd Ed, Microcosm Press, 2010
- [71] D. Sivolella, “*Space mining and manufacturing*”, Praxis Springer, 2019
- [72] Spaceflight.com web service, <https://spaceflight.com>, retrieved Oct. 2020
- [73] US Space Force, “*USSF announces initial operational capability and operational acceptance of Space Fence*”, <https://www.spaceforce.mil/News/Article/2129325/ussf-announces-initial-operational-capability-and-operational-acceptance-of-spa/>, retrieved on Dec. 2020
- [74] C. Specht, “*System GPS*”, Pelplin, 2007
- [75] C. Specht, M. Skóra, M.Mania, M. Specht, “*Analiza porównawcza formatów danych satelitarnych (almanac) w systemie GPS*”, Zeszyty Naukowe Akademii Morskiej, 2014
- [76] Space Stack Exchange, “*Cubesat Launch Costs*”, <https://space.stackexchange.com/questions/20009/cubesat-launch-costs>, retrieved Oct. 2020
- [77] T.S. Kelso, “*Socrates: Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space*”, <https://celestrak.com/SOCRATES/>, retrieved Dec. 2020
- [78] NASA, “*Definition of Two-line Element Set Coordinate System*”, https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html, retrieved Apr. 2020
- [79] D. Vallado, W. McClain, “*Fundamentals of Astrodynamics and Applications*”, 4th Ed, Microcosm, 2013
- [80] D. Vallado, P. Cefola, “*Two-Line Element Sets – Practice and Use*”, International Astronautical Congress, IAC-12-A6.6.11, 2012
- [81] H. Kramer, “*Earth Observation Portal: VELOX-1*”, <https://directory.eoportal.org/web/eoportal/satellite-missions/v-w-x-y-z/velox-1>, eoPortal, ESA, retrieved Oct. 2020

- [82] W. von Braun, *“Project Mars, a technical tale”*, 1949, published by Ishi Press, 2017
- [83] J. Wertz, R. Bell, *“Autonomous Rendezvous and Docking technologies – Status and Prospects”*, Space Systems Technology and Operations Conference, Orlando, 2003
- [84] J. Wertz, *“Mission Geometry: Orbit and Constellation Design and Management”*, Microcosm inc, 2001
- [85] Wikipedia, *“Absolute Magnitude”*, https://en.wikipedia.org/wiki/Absolute_magnitude#Asteroids, retrieved on May 2019
- [86] Wikipedia, *“Orbital Mechanics”*, https://en.wikipedia.org/wiki/Orbital_mechanics, retrieved Aug. 2020
- [87] Wikipedia, *“Orbit”*, <https://en.wikipedia.org/wiki/Orbit>, retrieved Jan. 2021
- [88] Wikipedia, *“Prędkość ucieczki”*, (pol. Escape velocity), https://pl.wikipedia.org/wiki/Pr%C4%99dko%C5%9B%C4%87_ucieczki, retrieved Sep. 2020
- [89] Wikipedia, *“The United States Department of Defense World Geodetic System 1972”*, Wikipedia, https://en.wikipedia.org/wiki/World_Geodetic_System#The_United_States_Department_of_Defense_World_Geodetic_System_1972, retrieved on Aug. 2020
- [90] Wikipedia, *“Lissajous orbit”*, https://en.wikipedia.org/wiki/Lissajous_orbit, retrieved Sep. 2020
- [91] Wikipedia, *“Yarkovsky effect”*, https://en.wikipedia.org/wiki/Yarkovsky_effect, retrieved Dec. 2020
- [92] Wikipedia, *“Lambert’s problem”*, https://en.wikipedia.org/wiki/Lambert%27s_problem, retrieved Dec. 2020
- [93] Wikipedia, *“Gauss’s method”*, https://en.wikipedia.org/wiki/Gauss%27s_method, retrieved Dec. 2020
- [94] Wikipedia, *“Kepler orbit”*, https://en.wikipedia.org/wiki/Kepler_orbit, retrieved Dec. 2020
- [95] Wikipedia, *“Bi-elliptic transfer”*, https://en.wikipedia.org/wiki/Bi-elliptic_transfer, retrieved Dec. 2020
- [96] Wikipedia, *“Yarkovsky–O’Keefe–Radzievskii–Paddack effect”*, https://en.wikipedia.org/wiki/Yarkovsky%E2%80%93O%27Keefe%E2%80%93Radzievskii%E2%80%93Paddack_effect, retrieved Dec. 2020
- [97] Wikipedia, *“Space rendezvous, First attempt failed”*, Wikipedia, https://en.wikipedia.org/wiki/Space_rendezvous#First_attempt_failed, retrieved on Sep. 2020
- [98] Wikipedia, *“Sun-synchronous orbit”*, https://en.wikipedia.org/wiki/Sun-synchronous_orbit, retrieved Sep. 2020
- [99] Wikipedia, *“Radiation pressure”*, https://en.wikipedia.org/wiki/Radiation_pressure, retrieved Dec. 2020
- [100] E. Zimovan, K. Howell, D. Davis, *“Near Rectilinear Halo Orbits and their application in cis-lunar space”*, 3rd IAA Conference on Dynamics and Control of Space Systems, 2017

Index (EN)

- aerobreaking, 21, 71
- almanac, 38
- AOS, 53
- apoapsis, 13
- apse line rotation, 26
- argument of periapsis, 11
- Aries constellation, 11
- Aries point, 10
- ascending node, 11
- asteroid, 78
- astrodynamics, 3
- atmospheric drag, 27
- auxiliary circle, 12

- Babylon, 4
- bi-elliptic transfer, 23
- burn, 21
 - prograde, 22

- chase maneuver, 22
- comet
 - I2/Borisov, 14
- conic sections, 29
- conjunction, 71
- Continuous Integration, 95
- Cowell method, 85
- critical inclination, 15
- CZML, 35

- delta-v, 21
- deorbit burn, 22
- descending node, 11
- direct entry, 71
- dwarf planet, 78

- eccentric anomaly, 12
- eccentricity, 7
- ECEF, 20
- ECI, 20
- ecliptic, 10, 70
- ecliptic plane, 70
- ENU, 20
- equinox, 11
- escape velocity, 4

- Gauss's method, 28
- Gauss's problem, 28
- GCRF, 20
- GEO, 14
- Geocentric Celestial Reference Frame, 20
- geocentric model, 5
- gravity assist, 21
- GSO, 14

- HCRS, 20
- heliocentric model, 5
- HEO, 15
- history of astrodynamics, 4

- ICRF, 20
- inclination, 9
- inclination change maneuver, 25
- injection burn, 26
- ion engine, 83
- Iridium flares, 93

- J2000, 19, 20

- keplerian elements, 7
- Kessler syndrome, 29
- Kirkwood gaps, 80
- Kuiper Belt, 79
- Kármán line, 6, 14

- lagranian point, 16
- law of cosines, 25
- LEO, 14
- libration point, 16
- line of nodes, 26
- LLO, 15
- longitude of the ascending node, 10
- LOS, 53

- magnitudo scale, 5
- major semi-axis, 8
- mean anomaly, 12
- MEO, 14
- meteor, 88
- meteorite, 88

- MPC, 78
- Oberth effect, 84
- Oort cloud, 79
- opposition, 71
- orbit
- capture, 8, 13
 - circular, 13
 - elliptical, 13
 - equatorial, 15
 - escape, 8, 13
 - geostationary, 14
 - geosynchronous, 14
 - halo, 16
 - high Earth, 15
 - hyperbolic, 8, 14
 - Lissajous, 17
 - low Earth, 14
 - low lunar, 15
 - medium Earth, 14
 - Molniya, 16
 - near-rectilinear halo, 17
 - osculating, 80, 84
 - parabolic, 13
 - perturbed, 80, 84
 - polar, 9, 15
 - prograde, 15
 - retrograde, 15
 - Sun-synchronous, 16
 - Tundra, 16
- orbital elements, 7
- orbital maneuver, 21
- orbital mechanics, 3
- orbital resonance, 28
- 'Oumuamua, 8, 14
- out of plane maneuvers, 25
- patched conics approximation, 29
- penumbra, 85
- periapsis, 13
- perturbers, 27
- Pisces, 11
- plane change maneuvers, 25
- plane rotation, 26
- plane rotation maneuver, 25
- Polaris, 19
- Poynting–Robertson effect, 27
- prograde, 9
- propulsive maneuver, 21
- RAAN, 10
- radial trajectory, 14
- rendezvous, 24
- retrograde, 9
- retrograde burn, 22
- right ascension of ascending node, 10
- rocket, 6
- rocket equation, 6, 21
- SAR, 13
- SEM, 39
- sexagesimal system, 4
- Space and Satellite Technologies, 103
- sphere of influence, 16, 29
- sub-satellite point, 56
- synthetic aperture radar, 13
- TEME, 20
- Test Driven Development, 95
- time
- GMST, 56
 - terrestrial, 19
- TLE, 40
- transfer window, 70
- True anomaly, 12
- Tsiolkovsky's equation, 21
- Two Line Element, 40
- U.S. Standard Atmosphere, 88
- umbra, 85
- V-2 rocket, 6
- vis viva equation, 21
- Yarkovsky effect, 27
- Yarkovsky–O'Keefe–Radzievskii–Paddack effect, 27
- YORP effect, 27
- Yuma, 39

Indeks (PL)

- aerobreaking, 21
- almanach, 38
- anomalía ekscentryczna, 12
- anomalía prawdziwa, 12
- anomalía średnia, 12
- AOS, 53
- apoapsis, 13
- aproksymacja krzywymi stożkowymi, 29
- argument perygeum, 11
- asteroida, 78
- astrodynamika, 3
- asysta grawitacyjna, 21
- atmosfera wzorcowa, 88

- Babilon, 4
- Baran, 11

- cień całkowity, 85
- czas
 - ziemski, 19
 - średni gwiazdowy Greenwich, 56
- CZML, 35

- delta-v, 21
- długość węzła wstępującego, 10

- ECEF, 20
- ECI, 20
- efekt Jarkowskiego, 27
- efekt Oberth'a, 84
- efekt Poyntinga–Robertsona, 27
- efekt YORP, 27
- ekliptyka, 10, 70
- elementy keplerowskie, 7
- elementy orbitalne, 7
- ENU, 20

- flary Iridium, 93

- GCRF, 20
- GEO, 14
- Geocentric Celestial Reference Frame, 20
- GSO, 14
- gwiazda Polarna, 19

- hamowanie aerodynamiczne, 71
- HCRS, 20
- historia astrodynamiki, 4

- ICRF, 20
- inklinacja, 9
- inklinacja krytyczna, 15

- J2000, 19, 20

- kometa
 - I2/Borisov, 14
- koniunkcja, 71
- koło opisane, 12
- krzywe stożkowe, 29

- LEO, 14
- linia Kármána, 6, 14
- LOS, 53

- magnitudo, 5
- manewr deorbitacyjny, 22
- manewr dwueliptyczny, 23
- manewr napędowy, 21
- manewr orbitalny, 21
- manewr pościgowy, 22
- manewr prograde, 22
- manewr wsteczny, 22
- manewr zmiany inklinacji, 25
- manewr zmiany płaszczyzny orbity, 25
- mechanika orbitalna, 3
- MEO, 14
- meteor, 88
- meteoryt, 88
- metoda Cowell'a, 85
- metoda Gauss'a, 28
- mimośród, 7
- model geocentryczny, 5
- model heliocentryczny, 5
- MPC, 78

- obłok Oorta, 79
- okno transferowe, 70
- opozycja, 71

- orbita
 — eliptyczna, 13
 — geostacjonarna, 14
 — geosynchroniczna, 14
 — halo, 16
 — heliosynchroniczna, 16
 — hiperboliczna, 8, 14
 — kołowa, 13
 — Lissajous, 17
 — Molniya, 16
 — Mołnia, 16
 — niska okołozemska, 14
 — niska wokółksiężycowa, 15
 — oskulacyjna, 80, 84
 — paraboliczna, 13
 — perturbowana, 80, 84
 — polarna, 9, 15
 — prawie prostoliniowa halo, 17
 — prograde, 15
 — przechwytyjąca, 8, 13
 — równikowa, 15
 — uciezkowa, 8, 13
 — wsteczna, 15
 — wysoka okołozemska, 15
 — średnia okołozemska, 14
 'Oumuamua, 8, 14
- pas Kuipera, 79
 perturbacje, 27
 peryapsis, 13
 planeta karłowata, 78
 problem Gauss'a, 28
 przerwy Kirkwooda, 80
 prędkość uciezki, 4
 punkt Barana, 10
 punkt Lagrange'a, 16
 punkt libracyjny, 16
 punkt podsatelitarny, 56
 półcień, 85
 półoś wielka, 8
- RAAN, 10
 radar z syntetyczną aperturą, 13
 rakieta, 6
 rektascensja węzła wstępującego, 10
 rendez-vous, 24
 rezonans orbitalny, 28
 rotacja linii apsyd, 26
 rotacja płaszczyzny orbity, 26
 Ryby, 11
 równanie raketowe, 6, 21
 równanie vis-viva, 21
 równonoc wiosenna, 11
- SAR, 13
 SEM, 39
 sfera wpływu, 16, 29
 silnik jonowy, 83
 syndrom Kesslera, 29
 system sześćdziesiątkowy, 4
- tarcie atmosferyczne, 27
 Technologie Kosmiczne i Satelitarne, 103
 TEME, 20
 TLE, 40
 trajektoria bezpośredniego wejścia, 71
 trajektoria radialna, 14
 twierdzenie cosinusów, 25
- V-2, rakieta, 6
- wzór Ciołkowskiego, 21
 węzeł wstępujący, 11
 węzeł zstępujący, 11
- Yuma, 39