MASARYKOVA UNIVERZITA

PŘÍRODOVĚDECKÁ FAKULTA

Analýza komplexních světelných křivek planetek

Bakalářská práce

SAMUEL BURANSKÝ

Brno, 2023

MASARYKOVA UNIVERZITA

PŘÍRODOVĚDECKÁ FAKULTA

Analýza komplexních světelných křivek planetek

Bakalářská práce

SAMUEL BURANSKÝ

Vedoucí práce: Mgr. Tomáš Henych Ph.D.

Ústav teoretické fyziky a astrofyziky

Brno, 2023

MUNI
PŘÍRODOVĚDECKÁ
FAKULTA

Bibliografický záznam

Autor:	Samuel Buranský Přírodovědecká fakulta Masarykova univerzita Ústav teoretické fyziky a astrofyziky
Název práce:	Analýza komplexních světelných křivek planetek
Studijní program:	Fyzika
Obor:	Astrofyzika
Vedoucí práce:	Mgr. Tomáš Henych Ph.D.
Akademický rok:	2022/2023
Počet stran:	xvi + 52
Klíčová slova:	asteroid, tumbler, světelná křivka, Fourierova řada, genetický algoritmus

Bibliographic record

Author:	Samuel Buranský Faculty of Science Masaryk University Department of theoretical physics and astrophysics
Title of Thesis:	Analysis of complex asteroid lightcurves
Degree Programme:	Physics
Field of Study:	Astrophysics
Supervisor:	Mgr. Tomáš Henych Ph.D.
Academic Year:	2022/2023
Number of Pages:	xvi + 52
Keywords:	asteroid, tumbler, light curve, Fourier series, genetic algorithm

Abstrakt

Malá tělesa Sluneční soustavy a jejich studium jsou zásadním tématem moderního astronomického výzkumu. Díky moderní fotometrii můžeme přesně měřit světelný tok v závislosti na čase a získat světelné křivky planetek. Pomocí analýzy světelných křivek planetek můžeme studovat jejich rotaci a tvar. Většina planetek rotuje kolem nejkratší osy s nejnižší rotační energií. Planetky s vyšší rotační energií se nazývají tumbleři. Tumbleři mají dvouperiodické světelné křivky. Jedna perioda představuje rotaci a druhá precesi.

V této práci jsme implementovali genetický algoritmus. Genetické algoritmy jsou vyhledávací a optimalizační techniky inspirované přirozenou evolucí. Využívají pojmy jako mutace, křížení a selekce k iterativnímu zlepšování řešení složitých problémů. Tento algoritmus jsme testovali na syntetických datech a zjistili jsme, že genetický algoritmus je užitečná metoda. Při použití této metody jsme našli poměrně dobrý fit dat. Jednoznačné určení precesní a rotační periody by vyžadovalo další data a podrobnější analýzu.

Abstract

Small Solar System Bodies and their study is a crucial topic of modern astronomical research. Thanks to modern photometry, we can precisely measure light flux as a function of time, and produce asteroid light curves. By analyzing asteroid light curves we can study the rotation and shape of asteroids. Most asteroids rotate around the shortest axis with the lowest rotation energy. Asteroids with higher rotation energy are called tumblers. Tumblers have two-periodic light curves. One period represents rotation and the second represents precession.

In this thesis, we have implemented a genetic algorithm. Genetic algorithms are search and optimization techniques inspired by natural evolution. They use concepts such as mutation, crossover, and selection to improve solutions to complex problems iteratively. We have tested this algorithm on synthetic data and we found that the genetic algorithm was a useful method. Using this method, we found fairly good fits to the data. An unambiguous determination of the precession and rotation periods would require additional data and more detailed analysis.

MUNI
SCI

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
KOTLÁŘSKÁ 2, 611 37 BRNO
IČ: 00216224
DIČ: CZ00216224

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Akademický rok: 2022/2023

Ústav:	Ústav teoretické fyziky a astrofyziky
Student:	Samuel Buranský
Program:	Fyzika
Specializace:	Astrofyzika

Ředitel *ústavu* PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s názvem:

Název práce:	Analýza komplexních světelných křivek planetek
Název práce anglicky:	Analysis of complex asteroid lightcurves
Jazyk závěrečné práce:	angličtina

Oficiální zadání:
Většina planetek rotuje ve stavu s nejnižší možnou energií, tedy kolem své nejkratší osy. Část planetek, tzv. tumbleři, má ale rotaci excitovanou, což znamená, že volně precedují. To se projevuje poměrně komplikovanou světelnou křivkou, která obsahuje lineární kombinaci rotační a precesní periody. Student/ka využije genetické algoritmy k analýze takových světelných křivek a ověří spolehlivost nalezení period rotace a precese pro různé kombinace těchto period, různě kvalitní a různě vzorkovaná data. Tento postup simuluje dobrá a méně kvalitní pozorování i různou frekvenci pozorování. Protože u pozorovaných planetek skutečné periody neznáme, použijeme v tomto případě světelnou křivku uměle generovanou z dynamického modelu planetky a simulaci pozorování dalekohledem z jedné nebo několika observatoří.

Vedoucí práce:	Mgr. Tomáš Henych, Ph.D.
Datum zadání práce:	21. 9. 2022
V Brně dne:	13. 4. 2023

Zadání bylo schváleno prostřednictvím IS MU.

Samuel Buranský, 21. 11. 2022
Mgr. Tomáš Henych, Ph.D., 1. 12. 2022
RNDr. Luboš Poláček, 8. 12. 2022

Poděkování

V prvom rade by som chcel poďakovať vedúcemu práce, Tomášovi Henychovi za neustálu pomoc, podporu, cenné rady a venovaný čas. Ďakujem tiež všetkým vyučujúcim, ktorí ma na mojej ceste posunuli vpred a motivovali ísť ďalej. Za veľkú pomoc s angličtionou patrí poďakovanie Paťke a v neposlednom rade patrí poďakovanie mojej priateľke, rodine, spolužiakom a kamarátom za neustálu podporu.

Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracoval samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

V Brně dne 23. 5.2023 Samuel Buranský

Contents

Introduction

The study of the Small Solar System Bodies is relevant to modern astronomical research. Asteroids are the remnants of the early time of the Solar System, and their research can improve our knowledge of the origin and the evolution of the Solar system. All asteroids orbit the Sun and rotate around their axis. Some of them also show free precession. We call them Tumblers. Tumblers are in a more complex rotational state, and have a two-periodic light curve.

In computer science, a powerful method for solving optimization problems is a genetic algorithm. The genetic algorithm is a strong heuristic optimization method inspired by natural selection and genetic theory. The genetic algorithm can be applied to the modeling of the light curves of the tumbling asteroids. This method has the potential to improve the accuracy, and efficiency of asteroid data analysis.

1 Asteroids

1.1 Asteroid characteristics

According to the resolution of the IAU in 2006 in Prague, we divide the Solar System bodies into planets, dwarf planets, moons, and all other bodies orbiting the Sun, collectively called "Small Solar-System Bodies"("SSSB") (IAU, 2006). The "SSSB" category includes asteroids, comets, and meteoroids. Asteroids and comets are the only known residual planetesimals from the earliest era of the Solar System. During the formation of the Sun and the Solar System, the planetesimals formed in a disk of gas and dust surrounding the Sun. Due to a large number of collisions, not all of them were able to form into planets; this, however, enables us to research the history of the Solar System (Chapman et al., 1978).

An asteroid or a minor planet is a small rocky Solar System body orbiting the Sun. Asteroids mainly orbit the Sun in the asteroid belt [1] and comets mainly in the Kuiper belt[2]. We estimate the number of asteroids larger than 1 km to be around $1.3 \cdot 10^6$ (Bottke et al., 2005) but the total mass is negligible ($5 \cdot 10^{-4} M_\oplus$) to the mass of the Solar System (Carroll and Ostlie, 2007). The mean diameter range is between 1 meter and more than 500 km (the largest asteroids are 2 Pallas and 4 Vesta (Figure 1.2)). Asteroids have various orbital elements and orbit the Sun anywhere in the Solar System. However, most of the known asteroids orbit the Sun in the asteroid belt.

1. The Asteroid belt (or the main asteroid belt, or the main belt) is a part of the Solar System between 2.1 and 3.3 au (usually we say "between Mars and Jupiter") with the high number of the "Small Solar-System Bodies" (Figure 1.1). Most of the known asteroids orbit the Sun there (around 95 %), and the rest is a group of asteroids in Lagrange points of planets or Trans-Neptunian bodies. The real number of asteroids is likely much different (Brož, 2013).
2. The Kuiper belt is a part of the Solar System beyond Neptune's orbit (from 30 to 50 au from the Sun), (Stern and Colwell, 1997).

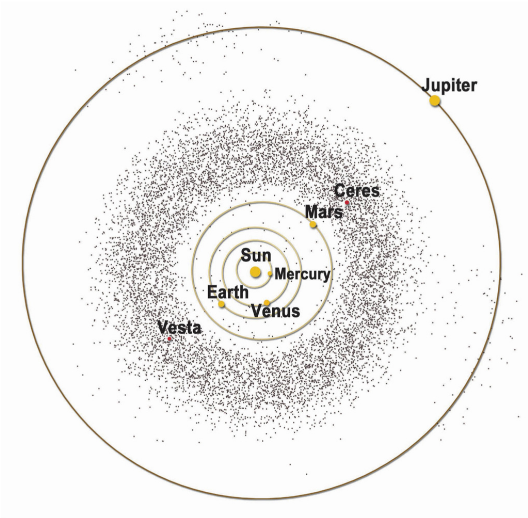

Figure 1.1: Scheme of the asteroid belt in the Solar System. Credit: NASA/McREL, 2007

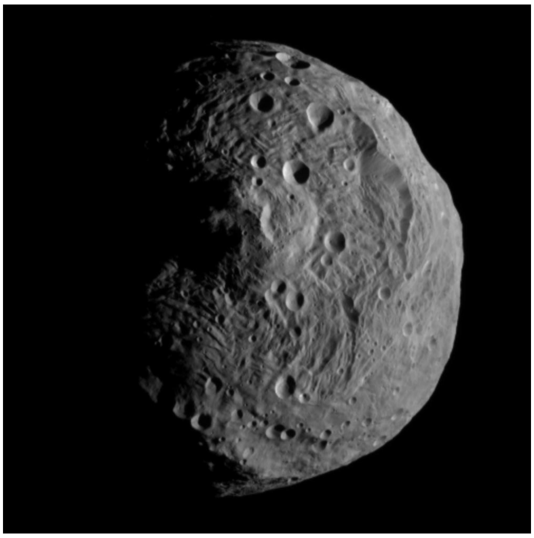

Figure 1.2: Image of the asteroid 4 Vesta. Image made by NASA spacecraft Dawn. Credit: NASA/JPL-Caltech/UCLA/MPS/DLR/IDA, 2011.

Smaller asteroids are not rigid bodies - they are called rubble piles. As the title suggests, an asteroid is usually not a one-piece rigid body, but rather a collection of smaller rock pieces held together by low gravitational force and geometric interlocking, as well as void space in between. There is a possibility of destroying or changing their motion via collision with another body [3].

1.2 A brief history of asteroid research

One of the first initiatives for asteroid discovery and research was the observation from Johann Daniel Titius in 1766. Titius stated his observation in the equation, which is nowadays known as the Titius-Bode law:

$$R_n = 0.4 + 0.3 \cdot 2^n, \tag{1.1}$$

where R_n is the radius of planetary orbits in astronomical units and n is $(-\infty, 0, 1, 2 \ldots)$ for Mercury, Venus, Earth, Mars, ... (Nieto, 1970) The actual distances and those predicted by equation 1.1 are compared in the Table 1.1 for the known planets at the time of Titius (Uranus was discovered in 1781, a few years after publishing Titius-Bode law).

3. Catastrophic collisions are not so frequent. Their likelihood and frequency depend on the orbit and the size of the asteroid.

Table 1.1: Comparison of the predicted and the real distances of the planets from the Sun (semi-major axis) by Titius-Bode law. There is a relatively good correlation with a potentially missing body between Mars and Jupiter. Real distances are taken from: Carroll and Ostlie, 2007.

n	Predicted distance [au]	Real distance [au]	Planet
$-\infty$	0.4	0.39	Mercury
0	0.7	0.72	Venus
1	1.0	1.00	Earth
2	1.6	1.52	Mars
3	2.8	-	-
4	5.2	5.20	Jupiter
5	10.0	9.58	Saturn
6	19.6	19.20	Uranus

Table 1.1 shows the high correlations between the predicted and the actual distances, as well as the gap between Mars and Jupiter in Titius-Bode law's predictions. Hungarian astronomer F. X. Zach believed the Titius-Bode law to be true. His suspicion rose due to the discovery was increased by the discovery of Uranus by William Herschel. Zach started the research and two years after meeting Herschel, he predicted the trajectory elements of the missing body (Cunningham, 2016).

On 1 January 1801, Italian astronomer G. Piazzi discovered the first asteroid, 1 Ceres at Palermo Observatory. He found an object which was not in the catalog, in the constellation Taurus. The next night, he saw the four arc minutes shift. He followed the object until 11 February. Astronomers were unsuccessful in finding Ceres during the following months (Cunningham, 2016). C. F. Gauss developed a different method to compute the trajectory elements from a few positions of the body. More importantly, he could predict the position in the future (Forbes, 1971). Finally, at the end of the year 1801 (on the night from 31 December 1801 to 1 January 1802), F. X. Zach observed 1 Ceres again near the Gauss's predicted position (Cunningham, 2016).

In the following years, astronomers started discovering more asteroids between the orbits of Mars and Jupiter. Their efforts revealed that there was no planet to be found, merely a large number of asteroids. We call this area the asteroid belt. Currently (11 April 2023) we know about 1 264 544 asteroids in the whole Solar System, 31 640 of which are Near-Earth asteroids ("Minor Planet Center", 2023).

1.3 Methods of research

We use plenty of methods for asteroid research, such as astrometry, photometry, spectroscopy, and research by radars or spacecrafts.

Astrometry measures the exact positions of sky objects. Astrometry is a common method in research of the Solar System. Measuring positions and their changes yields orbital parameters of the planets and other objects.

Photometry measures light intensity and counts the number of photons coming from celestial objects. From time-resolved photometry, we obtain the light curves. Analysis of the light curves is a strong method by which one can calculate the rotational period of the asteroid and some other characteristics, such as the rotational state or the overall shape of the asteroid. More details about asteroid light curves are in Chapter 2.

Spectroscopy measures and analyzes the spectrum of light coming from space objects. Using spectroscopy, we can mainly estimate the chemical properties of the celestial body or the medium between the body and the observer.

1.4 Classes of asteroids

With the help of spectroscopy, we can categorize asteroids into spectral classes based on the chemical properties of their surface (Carroll and Ostlie, 2007):

- S-type asteroids are dominated by iron or magnesium silicates and pure metallic iron-nickel. These asteroids have an albedo (or surface reflectance) between 0.1 and 0.2 and are located mainly in the inner asteroid belt,

- M-type are metal-rich asteroids. Locations and albedos are very similar to the S-type,
- C-type are rich in carbonaceous material and are very dark (small albedo in the range of 0.03 to 0.07). We can find them throughout the asteroid belt and they comprise some 3/4 of all asteroids in the belt,
- P-type are located in the outer asteroid belt and they are as dark as the C-type. They have a high percentage of organic compounds,
- D-type are similar to P-type but they orbit the Sun with a higher semi-major axis, they form the majority of Jupiter Trojans (orbiting the Sun in the Lagrange point 4 and 5 in the system Sun – Jupiter).

1.5 The rotational motion

The fundamental characteristic of all celestial objects is rotation which means spinning around its axis. The rotation is determined by initial conditions and conservation of the angular momentum. Rotation is studied by light curve analysis, thanks to the irregular shapes of the asteroids. We observe the maxima and minima of the light flux depending on the cross-sectional area facing the observer.

This equation describes the rotational motion of an asteroid:

$$\vec{L} = \hat{I}\vec{\omega}, \tag{1.2}$$

where $\vec{L}$ is the angular momentum of the asteroid, $\hat{I}$ is the inertia tensor, and ω is the angular velocity column vector (Pravec et al., 2005). Any inertia tensor can be rewritten into a diagonal matrix ($\hat{I_D}$):

$$\hat{I}_D = \hat{Q}^T \hat{I} \hat{Q}, \tag{1.3}$$

where $\hat{Q}$, $\hat{Q}^T$ are both orthogonal matrices (Henych, 2013). If the inertia tensor is a diagonal matrix (non-zero values are only on the diagonal) then the body rotates around a principal axis of that coordinate system. For bodies rotating around their principal axis, the following holds true: $I_1 \leqslant I_2 \leqslant I_3$ (equality is for the perfectly symmetric and homogeneous spherical or cubic body) (Pravec et al., 2005).

The kinetic energy of the rotational motion is given by the equation:

$$T = \frac{1}{2}\vec{\omega}^T \hat{I} \vec{\omega}. \tag{1.4}$$

For the principal axis rotation, where the inertia tensor is symmetric with only three components, we can rewrite kinetic energy as (Pravec et al., 2005):

$$T = \frac{1}{2}\sum_{i=1}^{3} I_i \omega_i^2, \tag{1.5}$$

or by the angular momentum (Henych, 2013):

$$T = \frac{L_x^2}{2I_1} + \frac{L_y^2}{2I_2} + \frac{L_z^2}{2I_3}. \tag{1.6}$$

The rotational energy of the asteroid has its minimum when the asteroid rotates around the shortest principal axis with the maximum moment of inertia (I_3) for angular momentum (L_z), (Henych, 2013). Minimum energy is given by the equation:

$$E_{\text{min}} = \frac{I_3 \omega_3^2}{2} = \frac{L^2}{2I_3}. \tag{1.7}$$

In the excited state of rotation, energy is given by the equation:

$$E_{\text{min}} < E \leqslant \frac{L^2}{2I_1}. \tag{1.8}$$

In the excited state, the asteroid has the complex rotational state. If the energy is maximum, the asteroid rotates around the principal axis with the lowest moment of inertia (Pravec et al., 2005).

1.6 Tumbling asteroids

Our knowledge about asteroid rotation normally comes from photometric measurements (light curves) (more in chapter 2). Almost all asteroids have principal axis rotation (around an axis with the lowest energy, shortest axis), but some asteroids do not. These asteroids which have excited rotation (and sometimes more complex rotational states) are called tumblers or tumbling asteroids. The light curve of this asteroid is quasi-periodic with two fundamental frequencies which are not exactly constant (Harris, 1994).

Due to energy dissipation, the tumbler with excited rotation damps to constant period rotation (principal axis rotation). This timescale τ (damping scale) is given by the equation:

$$\tau \sim \frac{\mu Q}{\rho K_3^2 r^2 \omega^3}, \tag{1.9}$$

where μ is the rigidity of the material, Q is the quality factor (ratio of lost energy per cycle to the total rotational energy), ρ is the bulk density, K_2^3 is the numerical description of the irregularity (0.01 for the spherical shape to 0.1 for the irregular shape), r is the mean radius and ω is the angular rotational frequency. The standard damping scale is in the range 10^5 to 10^8 years. Factor $r^2\omega^3$ in the denominator means that small slowly-rotating asteroids can have long damping scales. This equation can be rewritten into another form supposing that one knows certain parameters:

$$P \approx 17 D^{2/3} \tau^{1/3}. \tag{1.10}$$

Uncertainty of the constant in the equation 1.10 is factor 2.5 (Harris, 1994).

While observing and measuring a tumbling asteroid, it exhibits a two-periodic light curve instead of a single periodic (another case of a two-periodic light curve is a binary asteroid [4]). One period represents rotational motion, while the other represents precession motion (Kaasalainen, 2001).

There are various possible causes of tumbling asteroids. One of the events that might cause the free precession is a collision with another asteroid. These collisions change the inertia tensor of the asteroid

4. A binary asteroid is a system of two asteroids orbiting one barycenter (Margot et al., 2015). Usually, we see one body, but its light curve is a superposition of the single-periodic light curve of the larger asteroid and the orbital motion of the smaller one, whereas, in tumblers, one body is the source of the two periods.

and its angular momentum. Another possible cause is the "YORP" effect (Yarkovsky-O'Keefe-Radzievskii-Paddack effect), the rotational equivalent of the Yarkovsky effect[5], though the "YORP" effect affects the rotational motion (Vokrouhlický et al., 2015).

1.6.1 Examples

The best-known and the first discovered tumbler is 4179 Toutatis. Tab.1.2 shows a few examples with relatively well-determined periods.

The asteroid 4179 Toutatis is a Near-Earth asteroid in orbital resonance 4:1 with Earth. It was discovered in 1989. Toutatis has been observed intensively and there was a flyby of Toutatis. The main reason for its research was the close approach to Earth (0.0104 au) in the year 2004. Toutatis has dimensions of $4.6 \times 2.3 \times 1.9$ km. Its rotational period is 5.41 days, while its precession period is longer (7.35 days), (Busch et al., 2011).

Table 1.2: A few examples of tumbling asteroids (Pravec et al., 2005) P_1 and P_2 are precession and rotational periods, PAR is the characterization of the quality of description of the rotation state, and A is the amplitude of the light curve of the asteroid

Object	PAR	P_1[hod]	P_2[hod]	A [mag]
4179 Toutatis	−4	176	130	1.2
2002 TD_{60}	−3/ − 4	2.851	6.783	1.4
2000 WL_{107}	−3	0.1609	0.2188	1.1
253 Mathilde	−2/ − 3	418	250	0.5

5. The Yarkovsky effect is the process of the absorption and thermal emission of sunlight, which has a long-term effect on the orbital motion of the body (Vokrouhlický et al., 2015).

2 Light curves of asteroids

A light curve is a graph showing the brightness of a celestial object as a function of time. On the y-axis we plot brightness, flux, or magnitude, and on the x-axis we plot time, usually in Julian (or Modified Julian) days. Light curves are used in all areas of astronomy, especially in the study of variable stars, double stars, nova research, asteroids and comets, etc.

Analysis of the shape, periodicity, and amplitude of a light curve can tell us a great deal about the physical properties of a celestial body. Only a small number of light curves are non-periodic; these are the light curves of novae or other cataclysmic events. An interesting area of research is the analysis of periodic light curves, for example for rotating asteroids.

2.1 Light curves of asteroids and tumblers

Our knowledge of asteroid shapes, rotational periods, and rotation states comes mainly from their light curves. Light flux coming from asteroids changes periodically because of changes in the cross-section or because of the variation in the albedo[6]. The typical variation is a few tenths of a magnitude, and in extreme cases over one magnitude. By the asteroid's rotational period, we understand the time between two minima and maxima for geometrical reasons (Chapman et al., 1978).

Most asteroids have a quasi-periodic light curve. The strict periodic function is disrupted by the changing of the solar phase and distances from the Earth and from the Sun. Most asteroids have a single-periodic light curve. Such an asteroid rotates around the axis of maximum moment of inertia. There are two categories of exceptions: binary asteroids and tumblers. Tumblers are asteroids in a complex rotational state (more details are in Section 1.5) and their light curves contain a linear combination of two periods. Binary asteroids are a pair of gravitationally bound asteroids where usually one dominates gravitationally and also in the light curves. The light curve is two-periodic but it

6. Albedo measures the reflectivity of the celestial body. It is the ratio of reflected radiation and incoming radiation. The standard albedo for asteroids is in the range of 0.02 and 0.50.

is composed of two single-periodic components (Pravec and Hahn, 1997).

By analyzing light curve data we can categorize them into PAR categories. The PAR category is connected with the principal axis (PA) and non-principal axis (NPA) rotation and the quality of its detection. We use categories from -4 to +4. Negative categories indicate NPA rotation (-4 for constructed NPA model), positive ones indicate PA rotation (+4 for certain PA rotation), and 0 is for not enough data to categorize (Pravec et al., 2005).

2.2 Modeling asteroid light curves

The light curve of the asteroid is by default a periodic function, which means, that we can model it by the Fourier series. The Fourier series for an asteroid in the principal-axis rotation is used in the form:

$$R(t) = C_0 + \sum_{n=1}^{m} C_n \cos \frac{2\pi n}{P}(t - t_0) + \sum_{n=1}^{m} S_n \sin \frac{2\pi n}{P}(t - t_0), \quad (2.1)$$

where $R(t)$ is the computed magnitude in the time t, C_0 is the mean magnitude, C_n and S_n are the Fourier coefficients, P is the rotational period, t_0 is the shift in the time axis (epoch) and m is the order of the Fourier series. From the Fourier coefficients we can compute the amplitudes (A_n) and arguments (Φ), which is done by equations:

$$A_n = \sqrt{C_n^2 + S_n^2} \quad (2.2)$$

$$\cos \Phi = \frac{C_n}{A_n}, \ \sin \Phi = \frac{S_n}{A_n}. \quad (2.3)$$

Usually, the accuracy of a model is higher with the higher order of the Fourier series. On the other hand, with higher order, we have a higher number of free parameters, which makes computation much more difficult and longer. Usually, we use the Fourier series with $m = 5$ or lower (Pravec et al., 1996).

Finding the free parameters (period, epoch, Fourier coefficients, and mean magnitude) is usually done by minimizing the sum of the squares of residuals of the computed and measured function value (Pravec et al., 1996).

Sometimes, instead of the simple sum of the squared residuals we use χ^2 fitting:

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - y(x_i)}{\sigma_{y_i}} \right)^2, \tag{2.4}$$

where y_i is the measured value, $y(x_i)$ is the computed (fitted) value, σ_i is the uncertainty of the measurements and N is the size of the dataset (Press et al., 1986). It is the simple sum of the squared residuals divided by an uncertainty. It means that the points with higher uncertainty have a smaller influence on the resulting model.

2.2.1 Modeling tumbler light curves

Tumbler light curves are more complex than those of an asteroid in principal-axis rotation. The model function for the tumbler is also a Fourier series but two-dimensional:

$$\begin{aligned} F(\psi(t), \phi(t)) \doteq F^m(t) = C_0 + \sum_{j=1}^{m} \left[C_{j0} \cos \frac{2\pi j}{P_\psi} t + S_{j0} \sin \frac{2\pi j}{P_\phi} t \right] \\ + \sum_{k=1}^{m} \sum_{j=-m}^{m} \left[C_{jk} \cos \left(\frac{2\pi j}{P_\psi} + \frac{2\pi k}{P_\phi} \right) t \right. \\ \left. + S_{jk} \sin \left(\frac{2\pi j}{P_\psi} + \frac{2\pi k}{P_\phi} \right) t \right], \end{aligned} \tag{2.5}$$

where C_0 is the mean value of the flux, C, S are matrices of the Fourier coefficients and P_ϕ, P_ψ are the periods. C_{jk} or S_{jk} are the Fourier coefficients for the linear combination of the frequencies (frequencies are defined as the $f_\psi = P_\psi^{-1}$ and $f_\phi = P_\phi^{-1}$. Usually, we can't say which period corresponds to rotation or precession so we use the notation $(1, 2)$ rather than (ψ, ϕ) (Pravec et al., 2005).

Similarly to the principal-axis rotating asteroids light curves (Section 2.2), we can calculate an average normalized amplitude, defined as:

$$A_{jk}^{\text{norm}} = \frac{\sqrt{C_{jk}^2 + S_{jk}^2}}{C_0}. \tag{2.6}$$

This is the amplitude of the light curve that contains the linear combination of frequencies $(jf_1 + kf_2)$. This normalization can be used only with light flux or similar linear units (not magnitudes) (Pravec et al., 2005).

3 Genetic algorithm

Optimization is one of the most frequent types of problems in physics. We search for the extreme (maximum or minimum) of the function. It is possible to do so analytically or numerically. Well-known functions can be solved by analytical methods. For most problems in physics, we use some sort of numerical method.

One numerical method is a genetic algorithm (GA). The genetic algorithm is a heuristic method for solving an optimization problem. It is inspired by natural selection and reproduction in nature. It is mainly used for the numerical solution of problems with a higher number of free parameters.

3.1 A brief history

The history of the genetic algorithm starts in the 1950s and 1960s, when computer scientists thought about the usage of evolutionary systems in optimization problems (Mitchell, 1996).

The first real genetic algorithm was invented and developed by John Holland and his students and colleagues at the University of Michigan in 1967. Independently in Berlin, Germany, three students (Bienert, Rechenberg, and Schwefel) developed evolutional strategies too (De Jong et al., 1997). It was first developed primarily for the research of evolution in nature. Holland's algorithm is a method used for moving from one generation to another by some kind of natural selection. He used the encoding of chromosomes by zeros and ones (today we also use encoding by real numbers) (Mitchell, 1996).

3.2 Basic terminology

Before further analyzing the genetic algorithm, some important terms need to be explained. A gene is a functional block of DNA, and in the context of the genetic algorithm, it is the string that encodes a free parameter. A genome is a complete set of genes.

A crossover (or recombination in some papers) is the changing of a part of the genome between two parents to create two new offspring. A mutation is an operation performed on the offspring, which changes some elementary bit of the genome (Mitchell, 1996). Selection is a phase that determines which genomes are chosen for mating (reproduction). The probability of selection directly corresponds to a fitness function (Shukla et al., 2015). The fitness function takes the whole genome and returns the real number (typically positive), which defines its probability to live to reproduce. A generation is each iteration of the genetic algorithm when the population is changed (Grefenstette, 1986).

3.3 How does the genetic algorithm work

Genetic algorithms use methods inspired by natural selection and reproduction. The genetic algorithm works with a string. It can be a string of numbers (physical problems), or a string of letters (typically in biology or chemistry). The genetic algorithm consists of these steps (Charbonneau, 2002):

1. creating the initial population and calculating the fitness of each individual,
2. selection of parents for crossover and mutation,
3. crossover and mutation,
4. replacing parents with offspring,
5. repeat steps 2 – 5 again until we stop the algorithm.

Each iteration is called a generation. All of the generations are called a run. In a standard case, there are at least a couple of tens to a couple of thousands of generations. The GA employs pseudo-random numbers, therefore each run might finish with slightly different results. To achieve better outcomes it is good to evaluate the results of several runs statistically (Mitchell, 1996).

3.3.1 Initial population and encoding of solutions

Genetic algorithms can only work with strings, which is to say that we put all genes (free parameters) into one string (genome). First, we need to remove the decimal point. When using real numbers, we simply divide the real numbers by a constant to remove the decimal point (the same constant for the same gene). Conversely, when we translate a gene to a real number, we just add a decimal point and prepend a zero and multiply it by the scale factor. For example, we are fitting a linear function (we are searching for a and b), and for simplicity, we have only four candidates, see Tab. 3.1.

Table 3.1: Example of encoding in the genetic algorithm used in fitting linear functions. Searching solutions in the interval $(0, s)$.

Genome	Gene a	Gene b	Scaling factors	a	b
5879414572	58794	14572	1, 10	0.58794	1.4572
1278504569	12785	04569	1, 10	0.12785	0.4569
9873251478	98732	51478	1, 10	0.98732	5.1478
6548338564	65483	38564	1, 10	0.65483	3.8564

This works only if we search for positive numbers. If we search for numbers in interval $(-s, s)$, we simply use the equation:

$$G = -s + 2 \cdot s \cdot g, \quad (3.1)$$

where g is the gene (in the interval $(0, 1)$), s is the scaling factor, and G is the gene translated into a real number in the interval $(-s, s)$. An example is in Tab. 3.2

Table 3.2: Example of encoding in the genetic algorithm used in fitting linear functions. Searching solutions in the interval $(-s, s)$ using the equation 3.1.

Genome	Gene a	Gene b	s	a	b
5879414572	58794	14572	1, 10	0.17588	-7.0856
1278504569	12785	04569	1, 10	-0.7443	-9.0862
9873251478	98732	51478	1, 10	0.97464	0.2956
6548338564	65483	38564	1, 10	0.30966	-2.2872

The initial population is the first set of solution candidates. It is generated randomly in the defined intervals of the chromosomes.

The important parameters of the zeroth population are l, the length of the string (depends on the required accuracy of each gene), and n, the size of the population (number of solution candidates).

3.3.2 Fitness function

The fitness function is an important part of every problem solved by genetic algorithms. It is the function that defines the problem and defines which solution is more probable than the other. In general, the fitness function is composed of two functions:

$$u(x) = g(f(x)), \tag{3.2}$$

where f is the objective function and g transforms the value of f to positive numbers and for a better candidate for a solution, it increases. The function g is necessary if the value of f could be negative or if you search for the minimum of the function (the genetic algorithm usually finds the maximum of the function) (Grefenstette and Baker, 1989). The function g can only add some constant to make f positive or it can be an inverse function. The inverse function in some cases can change a decreasing function to an increasing function which turns a minimum to a maximum.

3.3.3 Operators

Three main processes in the GA are selection, crossover, and mutation, which we call operators. These operators change the whole population and try to randomly change the genomes, in a similar manner to natural selection, which means that better individuals are mating with a higher probability.

Selection is the operator that selects the genomes from the population for mating. The probability of being selected for the concrete candidate depends exactly on its fitness value (Mitchell, 1996). From the population of size n, we select n parents (one can be selected more than one time). There exist plenty of selection methods, e.g. tournament selection, proportional roulette wheel selection, and rank-based roulette wheel selection (Razali and Geraghty, 2011). A way to improve the selection algorithm is elitism. Elitism keeps few candidates from previous generations. It causes the fitness function to never decrease with proceeding evolution.

Roulette wheel selection is the method that assigns the probability of selection to every genome. This probability is simply solved by this equation:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}, \tag{3.3}$$

where f_i is the fitness function of the i-th individual in the population and j iterates over whole population. In equation 3.3 p_i is also normalized fitness function. Another variant of Roulette wheel selection is Roulette wheel selection using cumulative normalized fitness. In this case, every genome is calculated with cumulative normalized fitness value, which is a sum of the fitness of the previous genome (including the actual) (Wiangtong et al., 2002).

Crossover takes two genomes, randomly chooses a position (locus), and exchanges parts before and after the position between the genomes and creates two new individuals (Mitchell, 1996). The probability of the crossover, p_c, is usually a parameter of the algorithm. If the crossover does not occur, the results of the crossover are the exact copies of the parents. In a standard case, p_c is 0.7 or higher. There are more ways how to crossover strings. The most common is a single-point crossover. The less common is a two-point crossover, where one randomly chooses two positions to cut the strings.

Mutation randomly changes some numbers in a string (Mitchell, 1996). In binary encoding, the number is changed to another (0 to 1 or 1 to 0). In number encoding, (described in Section 3.3.1) we can simply change the number to another random number. The mutation is done with probability p_m, which is usually 0.01 and sometimes much lower. The mutation is used only for small changes, the main impact should still produce a crossover. A useful technique is an adaptive mutation, which increases the probability of mutation when the algorithm is stuck in a local extreme of the fitness function. We simply increase p_m to some higher value.

3.3.4 Stopping genetic algorithm run

A difficult question in genetic algorithm theory is when should we stop the genetic algorithm run. The simplest way to stop the run is after reaching a defined number of generations, but we do not know when the solution is good enough. In some problems, we know the expected value of the fitness function, so we can stop the run when the fitness of the solutions reaches some defined fitness threshold.

There are many other sophisticated ways of stopping the GA run. One of them is the number of generations when the fitness of the best candidate does not change. Another one is comparing progress in the first and second halves of the run. Progress in the second half should be significantly lower than in the first half. It means that we can stop the run if the ratio of the second and first half is smaller than some small number:

$$\frac{f(n) - f(n/2)}{f(n/2) - f(0)} < 10^{-r}, \tag{3.4}$$

where $f(n)$ means the fitness of the best candidate in generation number n and r is some positive number (the higher r means the longer calculation but the higher precision (Eiben and Smith, 2015).

4 Implementation and tests

4.1 Implementation

I have implemented the genetic algorithm project in the programming language Python [7] to be used for modeling light curves of asteroids. (Appendix A) The project has several separate layers.

The structure of the project follows:

1. **basics** implements the basic structures and functions needed in our implementation of the genetic algorithm:

    - **scaling.py** scales the gene into the suitable interval. For positive intervals $(0, s)$ it just multiplies the gene by s. For interval $(-s, s)$ it uses equation 3.1.
    - **solution_to_tuple.py** takes a string genome and makes the list of scaled chromosomes.
    - **initial_population.py** returns the zeroth (initial) population. For a better start of the run in our implementation we have set the initial population to be twenty times bigger than the populations in the run.

2. **processes** processes implement the three genetic algorithm processes (described in Section 3.3.2):

    - **selection.py** selects the genomes for mating (crossover and mutation). We use the Roulette wheel selection with cumulative normalized fitness (described in Section 3.3.3).
    - **crossover.py** makes crossover operator. It takes two strings and a probability of crossover. With this probability, it makes a crossover of the two strings.
    - **mutation.py** make mutation on one string. It takes a string and changes every digit to any other digit with a probability of mutation.

7. Python is a modern programming language highly used in science. It is especially useful in science because of the various libraries such as numpy, pandas, or scipy for science calculations and working with data.

3. **one_cycle** implements one generation of the genetic algorithm run:

   - **one_cycle.py** combines operators. It takes the previous generation and returns the next after executing the selection, crossover, and mutation.

4. **genetic_algorithm** implements the whole genetic algorithm as it is theoretically described in this section:

   - **genetic_algorithm.py** makes the whole genetic algorithm run in a cycle. It includes adaptive mutation (if the best solution does not change for many generations, the probability of the mutation is rising) and stopping function as it is described in Section 3.3.3.

4.1.1 Usage

For the usage of the genetic algorithm in this project the user should create an empty Python file. One should create a fitness function for a specific problem. Next, the user runs the function genetic_algorithm from `genetic_algorithm.py`.

The user should fill in these mandatory parameters to run the code:

- number_of_solutions is the number of individuals in each generation. The optimal number of individuals depends on the specific problem. A higher number of individuals means a more accurate final solution but on the other hand, it means a long processing time,

- number_of_genes is the number of free parameters which the user searches,

- number_of_digits is the number of digits in every free parameter. Number of digits has a similar effect on the run as the number of solutions. Higher values increase the accuracy of each parameter but also the time of the calculation.

- fitness_function is the function that takes the set of the free parameters (solution) and returns a positive number. (detail in Section 3.3.2).

The user can experiment with these non-binding parameters to find better solutions:

- r_crossover (= 0.95) is the probability of the two selected individuals producing new offspring by crossover. The probability of crossover affects the rate of convergence to the maximum of the fitness function,
- r_mutation (= 0.01) is the probability of changing the concrete bit of a string. The probability of the mutation affects the escaping from the local extreme. High probability causes the total randomization of the genome during the run. But a slightly higher probability helps to escape a local extreme when a majority of genomes are stuck in there,
- scales (= None) are the factors that set free parameters to another range than $(0, 1)$,
- maximal_generations (= 10 000) is the maximum number of the calculated generations,
- elitism (= 0) is the number of the best individuals that are just copied to the next generation. Elitism is by default set to zero, but it is good to set it to at least two. These settings cause the fitness function not to decrease,
- stop (= 10) is the parameter r from the equation 3.4,
- threshold (= None) is the threshold of the fitness function when the run is stopped. If it is None, it is not used. Useful parameter when the user expects fitness function and its values in the run.

4.2 Application and tests

First, I tried our implementation of a genetic algorithm for modeling something simple and familiar; linear functions (`lines_test.py`). An example of the run and the evolution of the fitness can be seen in Figure 4.1 and 4.2. We can see a good fit of data produced by the genetic algorithm model (model from scipy too). We can also see the raising (or non-decreasing) fitness function over the generations.

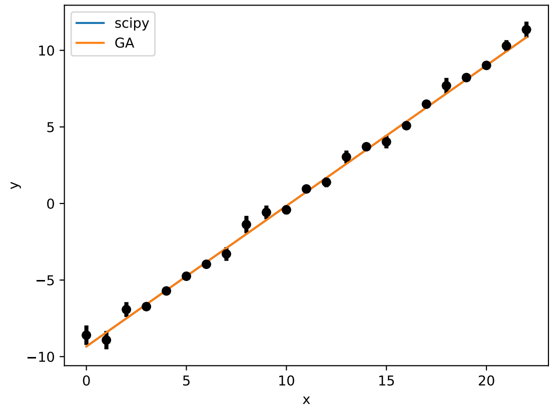

Figure 4.1: Plot of the linear function model by the genetic algorithm.

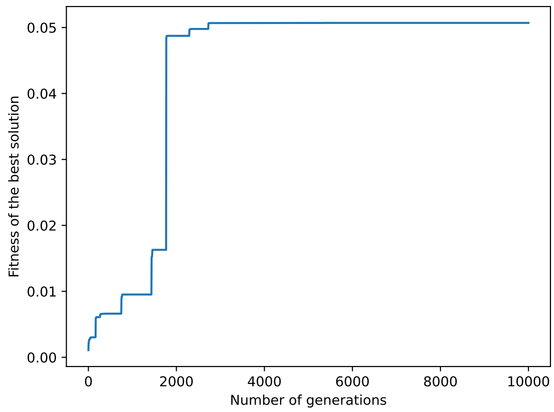

Figure 4.2: Evolution of the fitness function over a generation in the fitting fitness function.

5 Results

Finally, we applied the implemented genetic algorithm to the tumbling asteroid. The data [8] is in three columns (Julian day, Normalized light flux, and Uncertainty of the light flux). The data is synthetic for better control of the data analysis method. A Python script that calls the genetic algorithm and contains the fitness function is in the file `tumbler.py`. During testing, we observed difficulties in using light flux errors. Therefore, we did not use the errors in modeling (we set them equal to one for all the data points).

As the model function, we use the two-dimensional Fourier series from the equation 2.5. The model is represented by the set of free parameters which are the periods of the rotational and precessional motion (P_ψ, P_ϕ), Fourier coefficients of the two-dimensional Fourier series for cosine (C_{jk}) and for sine (S_{jk}) and the constant term C_0.

First, we modeled the light curve using the Fourier series for the PA asteroid. We got a rough estimate of 1.5 days for one of the periods. This meant that we could use the specific scales in Tab. 5.1. We set the hyperparameters in Tab 5.1, and the light flux errors were all set to one.

Table 5.1: Table of the hyperparameters for the genetic algorithm modeling tumbler light curve. The first part of the scales stands for the Fourier coefficients, and the last three for C_0, and two periods.

Parameter	Value
number_of_solutions	500
number_of_digits	5
r_crossover	0.95
r_mutation	0.01
elitism	2
scales	$(-1, \ldots, -1, 1.5, 5, 10)$
stop	30
maximal_generations	15 000

8. Data is synthetic for better comparison of the result and the actual values of periods. Data was generated by the supervisor of this thesis.

5.1 The Fourier series (order $m = 1$)

In this Section, we compare two models by the first-order Fourier series. The first one contained the free parameter t_0 (a shift on the x-axis) and the second did not. More models with Fourier series of the first order are in Appendix B.

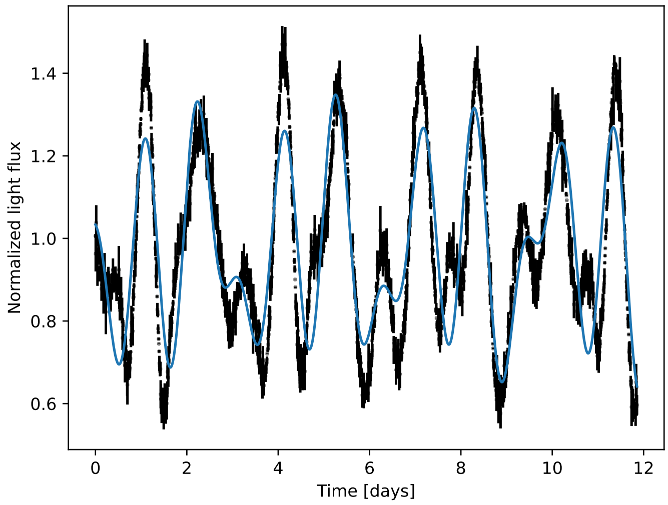

Figure 5.1: Tumbling asteroid light curve model by the genetic algorithm. Modeled with the genetic algorithm, by Fourier series with $m = 1$ and with a free parameter t_0.

Table 5.2: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 1$ and with a free parameter t_0.

Parameter	Value
Number of generations	1380
χ^2	24.72
P_1 [days]	1.484
P_2 [days]	3.281

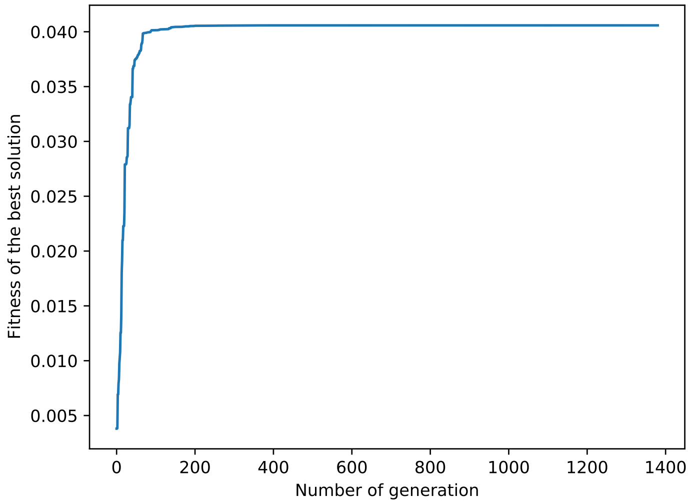

Figure 5.2: Evolution of the fitness function over generations in the modeling of the non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 1$ and with a free parameter t_0.

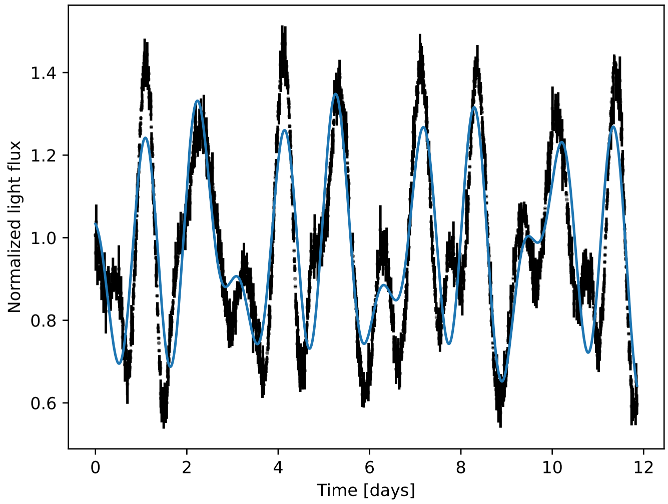

Figure 5.3: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 1$.

Table 5.3: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 1$.

Parameter	Value
Number of generations	8890
χ^2	24.64
P_1 [days]	1.487
P_2 [days]	3.266

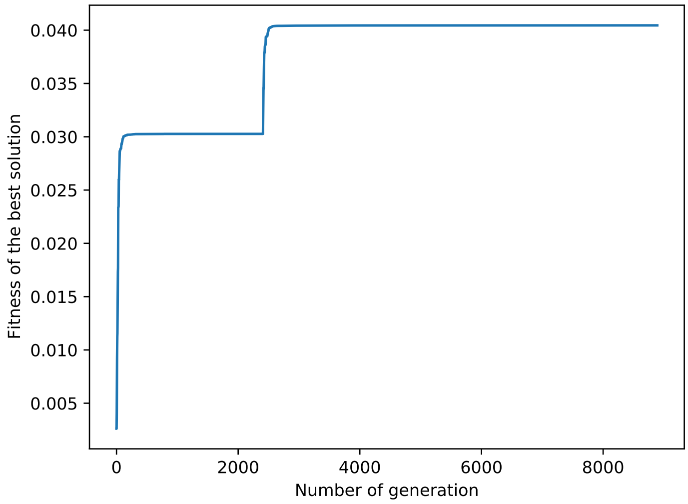

Figure 5.4: Evolution of the fitness function over generations in the modeling of the non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 1$.

5.2 The Fourier series (order $m > 1$)

In this Section, we model the light curve by the Fourier series of the higher order. More models are in Appendix B.

5.2.1 $m = 2$

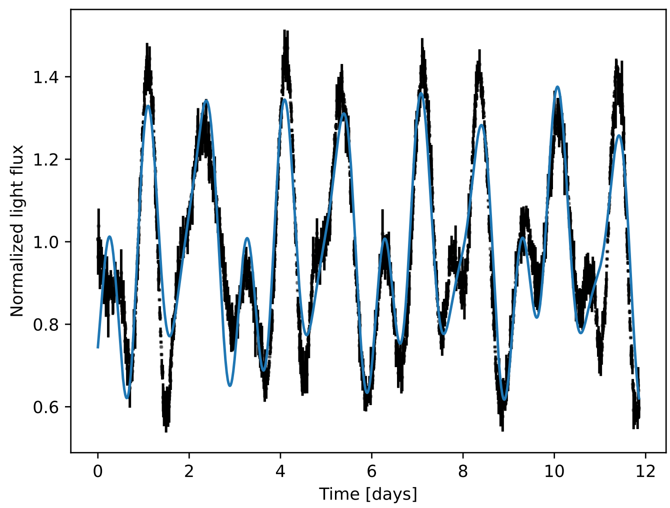

Figure 5.5: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 2$.

Table 5.4: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 2$.

Parameter	Value
Number of generations	4 400
χ^2	13.90
P_1 [days]	3.004
P_2 [days]	2.995

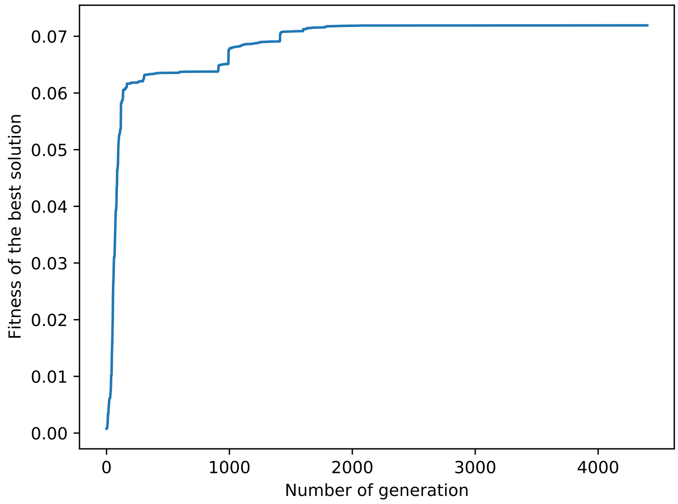

Figure 5.6: Evolution of the fitness function over generations in the modeling non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 2$.

5.2.2 $m = 3$

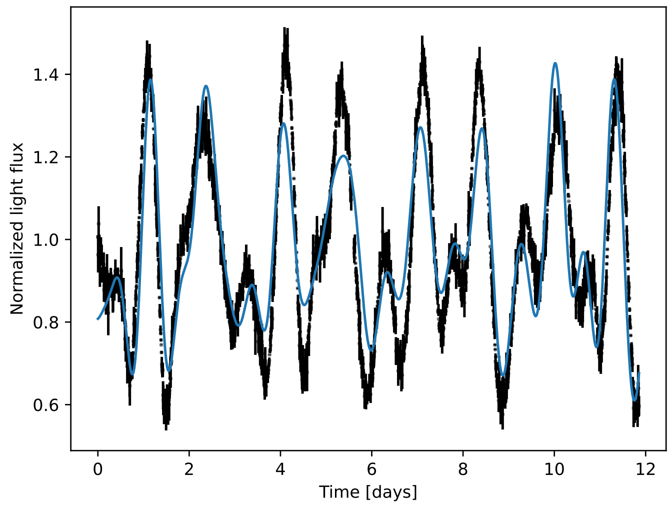

Figure 5.7: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 3$.

Table 5.5: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 3$.

Parameter	Value
Number of generations	15 000
χ^2	20.45
P_1 [days]	2.88
P_2 [days]	2.86

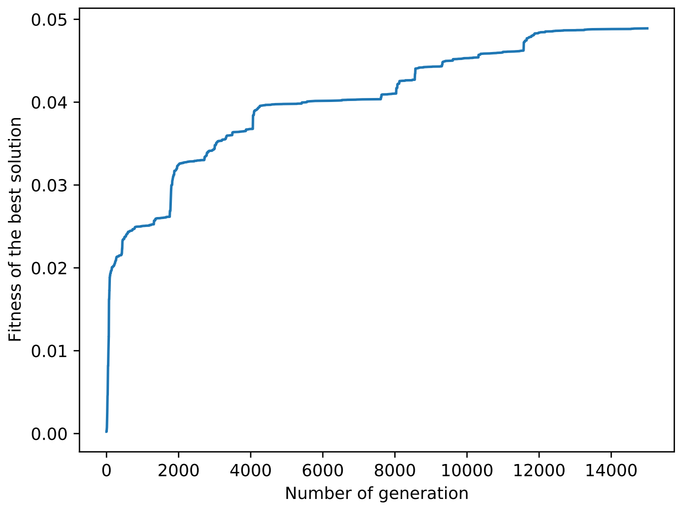

Figure 5.8: Evolution of the fitness function over generations in the modeling of the non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 3$.

5.3 Summary of the results

Table 5.6: Summary of the results of all models.

m	P_1 [days]	P_2 [days]	χ^2
1	1.484	3.281	24.72
1	1.487	3.266	24.64
1	1.486	3.250	25.58
1	1.200	6.536	39.96
2	3.004	2.995	13.90
2	2.989	3.000	12.07
3	2.880	2.858	20.45

6 Discussion

We ran the genetic algorithm with 500 individuals in each generation and 5 digits for all free parameters in the encoded string. The probability of the crossover we set high, 95 %, and the probability of mutation was low, 1 %. The range for Fourier coefficients was (-1, 1), the range for the constant term C_0 was (0, 1.5), for the first period it was (0, 5) and for the second period it was (0, 10). For the non-decreasing fitness function, we set elitism to two and the parameter r was 30 for a long enough calculation.

In Section 5.1 we see the two models of the same data of the Fourier sequence of the first order. The first one contains the free parameter t_0 and it fits the synthetic data reasonably well. We used the free parameter t_0 as the test despite the fact that the model Fourier series does not contain it. The second model also fits the data well, even without the t_0 parameter. We can conclude that the free parameter t_0 does not influence the result.

In Section 5.2 we modeled the same data with the Fourier series of the higher order. Visually we see a much better fit of the data. Using the higher order, we reach a higher value of the fitness functions, and lower χ^2, respectively. It is caused by a closer correspondence of the model and the data.

It is important to know that our results do not have to be direct periods of the asteroid. It is possible that some results are the multiples of the original periods or others can be linear combinations of the original periods (Kaasalainen, 2001).

Uncertainties of our results can be estimated as statistics over several runs. It is possible to make statistics only from runs with similar χ^2 and similar results. For example from the first three runs with $m = 1$ we get the values $P_1 = (1.486 \pm 0.001)$ days and $P_2 = (3.27 \pm 0.01)$ days.

The actual rotation period was $P_1 = 34.417$ h (1.434 days) and the precession period was $P_2 = 80.559$ h (3.357 days). Comparing that to our results we see a relatively good agreement between the actual periods and the periods from the first order of the Fourier series. In the last first-order fit, we see that the second period is the double precession period. In the third order fit the second period is double the rotation period.

6.1 Critical points

The correct fitness function is the most important thing when using genetic algorithms. It should be non-negative and monotonically increase for better solutions. From a technical point of view, the calculation of the fitness takes the longest time to compute. Therefore, it is important to write the fitness function optimally.

Another problem is stopping the genetic algorithm run. Various methods proved to be imperfect. Stopping by the number of generations was almost useless because it depends on the randomness and the size of the problem. The combination of a threshold and the ratio between changes in the second and the first part were the most useful. The threshold is the minimum fitness function value the run has to reach.

The next point, which can make the run even faster and more precise, is the accurate range of each free parameter. The large range causes long computation time and lesser precision or sometimes incorrect results.

7 Conclusions

In summary, free precession is an interesting physical phenomenon of asteroid rotation. Most asteroids rotate around the principal axis, which is the shortest axis with the lowest rotation energy. Asteroids with higher rotation energy show a free precession. Rotational motion is usually studied by analyzing light curves. The asteroid spinning around the shortest principal axis shows a single-periodic light curve while the freely precessing asteroid shows a two-periodic light curve. One period represents rotation and the other period represents precession.

Asteroid light curves are modeled using the Fourier series. For modeling the asteroid light curve we use a simple Fourier series, but for the tumbling asteroid, we use a two-dimensional one. To model the light curves, we minimize the sum of the squared residuals.

In this thesis, we focused on using a genetic algorithm to model tumbler light curves. Genetic algorithms offer a powerful approach to optimization tasks inspired by natural evolution. Through iterative processes involving mutation, crossover, and selection, these algorithms excel in improving solutions for complex problems with a high number of free parameters.

In the practical part of the thesis, we implemented the genetic algorithm and applied it to the tumbling asteroid light curve. For the modeling of the synthetic data, we used the two-dimensional Fourier series. Firstly, we found that the shift on the x-axis does not affect the result. Finally, we found genetic algorithm modeling to be a useful method for light curve analysis. The method works with pseudo-random numbers and for better results, it is necessary to run it several times with more generations due to every run generating a slightly different result. However, if the fit is reasonable for a few of those runs, it enables us to derive uncertainties of the results. Sometimes we do not get the direct periods but the multiples of the actual periods or their linear combination. The way to solve this is to get more data.

7.1 Future plans

Testing in this thesis was done on synthetic data without the usage of uncertainties with a high sampling rate. Normally, the data has a lower sampling frequency and breaks between datasets. The first plan is to test and potentially optimize the algorithm for more realistic data.

Another possible improvement is automation. We would like to automate running multiple runs and obtain statistics of all results. That would also enable us to estimate the uncertainties of the results.

Bibliography

Bottke, W. F., Durda, D. D., Nesvorný, D., Jedicke, R., Morbidelli, A., Vokrouhlický, D., & Levison, H. (2005). The fossilized size distribution of the main asteroid belt. *175*(1), 111–140. https://doi.org/10.1016/j.icarus.2004.10.026

Brož, M. (2013). *Fyzika sluneční soustavy* (Vyd. 1.). Matfyzpress.

Busch, M. W., Benner, L. A. M., Scheeres, D. J., Margot, J. .-., Magri, C., Nolan, M. C., & Giorgini, J. D. (2011). Twenty Years of Toutatis. *EPSC-DPS Joint Meeting 2011*, *2011*, 297.

Carroll, B. W., & Ostlie, D. A. (2007). *An Introduction to Modern Astrophysics* (S. F. P. Addison-Wesley, Ed.; 2nd (International)).

Cunningham, C. J. (2016). *Discovery of the first asteroid, Ceres: Historical studies in asteroid research*. Springer.

De Jong, K., Fogel, D., & Schwefel, H.-P. (1997). A history of evolutionary computation.

Eiben, A., & Smith, J. (2015). *Introduction to Evolutionary Computing*. https://doi.org/10.1007/978-3-662-44874-8

Forbes, E. G. (1971). Gauss and the Discovery of Ceres. *Journal for the History of Astronomy*, *2*, 195. https://doi.org/10.1177/002182867100200305

Grefenstette, J. J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, *16*(1), 122–128. https://doi.org/10.1109/TSMC.1986.289288

Grefenstette, J. J., & Baker, J. E. (1989). How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. *International Conference on Genetic Algorithms*.

Harris, A. W. (1994). Tumbling Asteroids. *Icarus*, *107*(1), 209–211. https://doi.org/10.1006/icar.1994.1017

Henych, T. (2013). *Excitation of asteroid rotations through impacts* (Disertační práce). Masarykova univerzita, Přírodovědecká fakulta. Brno. https://is.muni.cz/th/gvhco/

Chapman, C. R., Williams, J. G., & Hartmann, W. K. (1978). The asteroids. *16*, 33–75. https://doi.org/10.1146/annurev.aa.16.090178.000341

Charbonneau, P. (2002). *An introduction to Genetic Algorithms for Numerical Optimization*. https://doi.org/10.5065/D608638S

Kaasalainen, M. (2001). Interpretation of lightcurves of precessing asteroids. *376*, 302–309. https://doi.org/10.1051/0004-6361:20010935

Margot, J.-L., Pravec, P., Taylor, P., Carry, B., & Jacobson, S. (2015). Asteroid Systems: Binaries, Triples, and Pairs. In *Asteroids IV*. University of Arizona Press. https://doi.org/10.2458/azu_uapress_9780816532131-ch019

Mitchell, M. (1996). *An introduction to genetic algorithms*. Bradford Book.

Nieto, M. M. (1970). Conclusions about the Titius Bode Law of Planetary Distances. *8*, 105.

Pravec, P., & Hahn, G. (1997). Two-Period Lightcurve of 1994 aw $_1$: Indication of a Binary Asteroid? *127*(2), 431–440. https://doi.org/10.1006/icar.1997.5703

Pravec, P., Harris, A., Scheirich, P., Kušnirák, P., Šarounová, L., Hergenrother, C., Mottola, S., Hicks, M., Masi, G., Krugly, Y., Shevchenko, V., Nolan, M., Howell, E., Kaasalainen, M., Galád, A., Brown, P., DeGraff, D., Lambert, J., Cooney, W., & Foglia, S. (2005). Tumbling asteroids [Hapke Symposium]. *Icarus*, *173*(1), 108–131. https://doi.org/https://doi.org/10.1016/j.icarus.2004.07.021

Pravec, P., Šarounová, L., & Wolf, M. (1996). Lightcurves of 7 Near-Earth Asteroids. *Icarus*, *124*(2), 471–482. https://doi.org/10.1006/icar.1996.0223

Press, W. H., Flannery, B. P., & Teukolsky, S. A. (1986). *Numerical recipes. The art of scientific computing*.

Razali, N., & Geraghty, J. (2011). Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. *2*.

Shukla, A., Pandey, H. M., & Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, 515–519. https://doi.org/10.1109/ABLAZE.2015.7154916

Stern, S. A., & Colwell, J. E. (1997). Collisional Erosion in the Primordial Edgeworth-Kuiper Belt and the Generation of the 30-50 au Kuiper Gap. *The Astrophysical Journal*, *490*(2), 879. https://doi.org/10.1086/304912

Vokrouhlický, D., Bottke, W. F., Chesley, S. R., Scheeres, D. J., & Statler, T. S. (2015). The Yarkovsky and YORP Effects. In *Asteroids IV* (pp. 509–531). https://doi.org/10.2458/azu_uapress_9780816532131-ch027

Wiangtong, T., Cheung, P. Y. K., & Luk, W. (2002). Comparing Three Heuristic Search Methods for Functional Partitioning in Hardware–Software Codesign. *Design Automation for Embedded Systems*, *6*(4), 425–449. https://doi.org/10.1023/A:1016567828852

Online Resources

IAU. (2006). *IAU 2006 General Assembly: Result of the IAU Resolution votes*. https://www.iau.org/news/pressreleases/detail/iau0603/

Minor Planet Center [Accesed on 04 11,2023]. (2023). https://minorplanetcenter.net/

NASA/JPL-Caltech/UCLA/MPS/DLR/IDA. (2011). *Latest Image of Vesta Captured by Dawn on july 17, 2011*. Retrieved April 25, 2023, from https://solarsystem.nasa.gov/resources/2119/latest-image-of-vesta-captured-by-dawn-on-july-17-2011/

NASA/McREL. (2007). *Artist's graphic of the asteroid belt, part of Dawn's Mission Art series.* Retrieved May 7, 2023, from https://solarsystem.nasa.gov/resources/2156/asteroid-belt/

A Appendix

The genetic algorithm Python project with all tests is in `bachelor_thesis.zip`. It is available on the internet address:
https://is.muni.cz/auth/th/dwrj3/bachelor_thesis.zip

B Appendix

B.1 $m = 1$

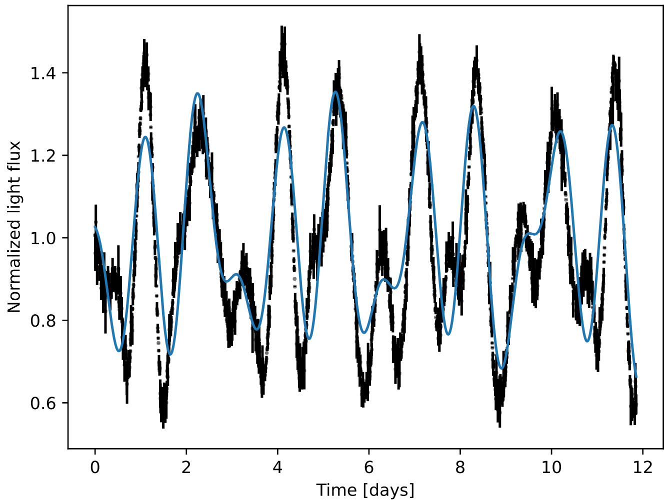

Figure B.1: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 1$.

Table B.1: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 2$.

Parameter	Value
Number of generations	1810
χ^2	25.58
P_1 [days]	1.486
P_2 [days]	3.250

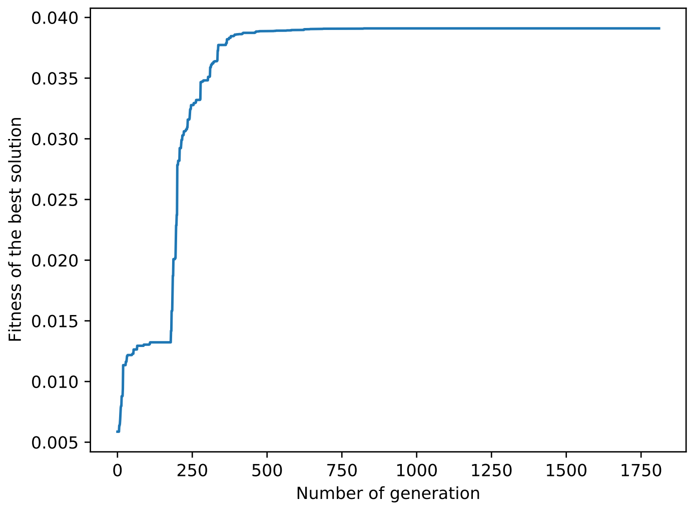

Figure B.2: Evolution of the fitness function over generations in the modeling non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 1$.

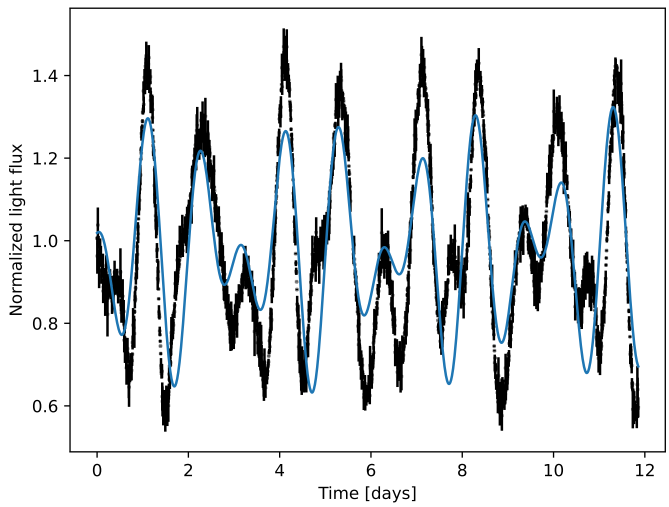

Figure B.3: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 1$.

Table B.2: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 2$.

Parameter	Value
Number of generations	3720
χ^2	39.96
P_1 [days]	1.200
P_2 [days]	6.536

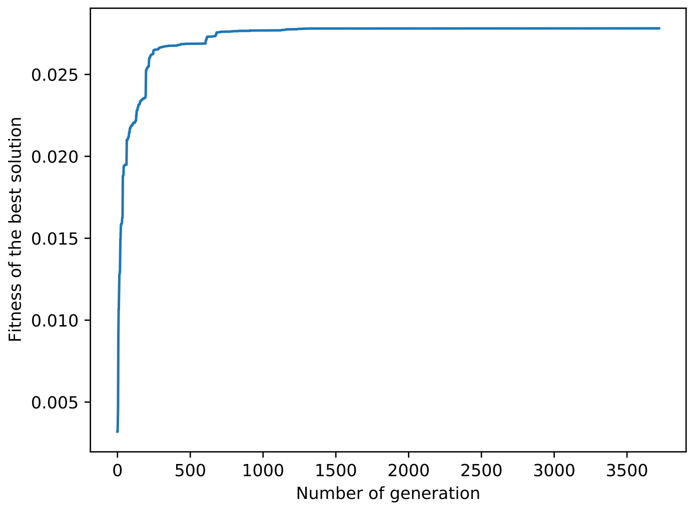

Figure B.4: Evolution of the fitness function over generations in the modeling non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 1$.

B.2 $m = 2$

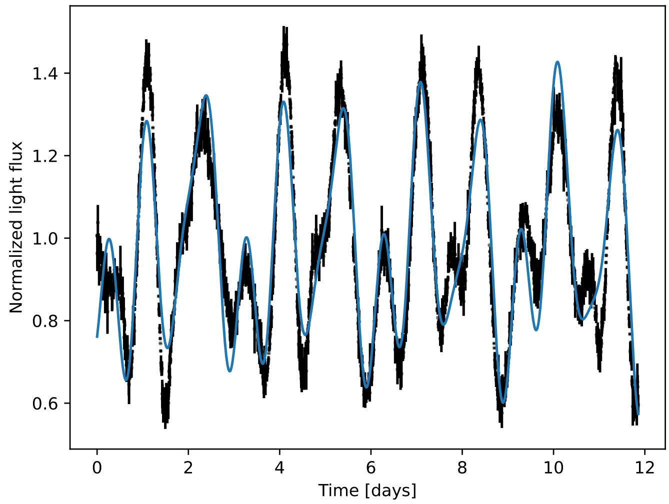

Figure B.5: Tumbler asteroid light curve model by the genetic algorithm. Modeled by Fourier series with $m = 2$.

Table B.3: Table of the Results for the genetic algorithm run for modeling the tumbler light curve. Modeled by the Fourier series with $m = 2$.

Parameter	Value
Number of generations	15 000
χ^2	12.07
P_1 [days]	3.00
P_2 [days]	3.00

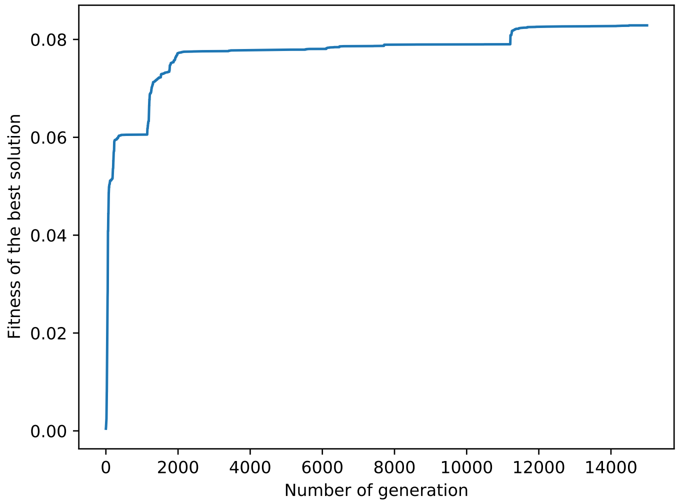

Figure B.6: Evolution of the fitness function over generations in the modeling non-principal axis rotating asteroid light curve. Modeled by Fourier series with $m = 2$.