Gymnázium, Praha 7, Nad Štolou 1

SOČ: kategorie 2, fyzika

Parametry rotace a tvaru asteroidů: limity inverzní metody

Autor: Marco Souza de Joode
Škola: Gymnázium, Praha 7, Nad Štolou
Vedoucí: Mgr. Petr Scheirich, Ph.D.
AsU AV ČR

Praha 2020

Poděkování

Nejvíce musím poděkovat Petrovi Scheirichovi, který je již roky vedoucím Astronomické Expedice, kde jsem se toho mnoho naučil, a který byl letos vedoucím mé práce. Dále bych chtěl poděkovat vedoucímu CCD skupiny pozorovatelů Filipovi Walterovi ze Štefánikovy hvězdárny na Petříně, za probdělé mrazivé noci a podporu. Také děkuji Martinu Maškovi za rady a jeho měření. Za pozorovací čas bych také chtěl poděkovat Zbyňovi Henzlovi a Stanislavu Daniši. Za 3D tisk modelu děkuji svému učiteli fyziky, Matěji Rystonovi.

Prohlášení

Prohlašuji, že jsem práci SOČ vypracoval samostatně, použil jsem pouze podklady uvedené v přiloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Praze dne 30. března 2020

Podpis: . . . .

Abstrakt

Tato práce se zabývá metodou zpětného určení tvaru asteroidů ze světelných křivek, a jejími omezeními. Jsou zde zpracovávány publikované světelné křivky, světelné křivky pořízenými pro tuto práci jinými pozorovateli a autorova měření ze Štefánikovy hvězdárny v Praze. Možnosti a omezení inverzní metody jsou zde ukazovány zejména na příkladu asteroidu hlavního pásu (511) Davida, ale i na řadě dalších objektech, aby výsledky byly zobecnitelné a užitečné.

Pomocí nových měření byla zpřesněna perioda rotace asteroidu (511) Davida.

V práci je zavedena veličina Ψ, kterou se následně charakterizuje kvalita pokrytí různých geometrií pro danou sadu světelných křivek. V závislosti na této veličině je zkoumána jednoznačnost určení souřadnic rotačního pólu planetky.

Klíčová slova

inverzní metoda, meziplanetární hmota, fotometrie

Abstract

In this work, we concern ourselves with a method of inverse determination of the shape of asteroids from photometric lightcurves. Published results and new observations made by the author at the Prague Observatory and new observations made by other observers were used for the analysis. The capabilities and limitations of this method were shown primarily on the main belt asteroid (511) Davida, however also on other objects, so that the results can be usefully generalised.

With new measurements, the rotational period of (511) Davida was determined with more precision than in any published work.

We define a parameter (Ψ) which describes the quality of the coverage of different geometries for a set of lightcurves. The ambiguity of the position of the rotational pole was calculated with respect to Ψ.

Keywords

the inverse method, interplanetary matter, photometry

Středoškolská odborná činnost – posudek konzultanta

Název práce:

Parametry rotace a tvaru asteroidů: limity inverzní metody

Autor: Marco Souza de Joode

Škola: Gymnázium, Praha 7, Nad Štolou

Rozsah práce: 54 stran včetně obrázků a seznamu literatury

Termín odevzdání: březen 2020

Konzultant: Mgr. Petr Scheirich, PhD., Astronomický ústav AVČR, v.v.i., Ondřejov

Předložená práce středoškolské odborné činnosti se věnuje tématu z oboru malých těles Sluneční soustavy – určování parametrů tvaru a rotace asteroidů z jejich fotometrického měření. Zatímco tato samotná úloha je již v oboru dobře etablovaná a zvládnutá (což neznamená, že její aplikace nevyžaduje důkladné pochopení principu, na němž je založena, a osvojení si příslušného matematického aparátu, což obojí autor výtečně dokázal), autor k ní přistupuje inovativním způsobem.

Nejde mu o pouhou aplikaci metody určení tvaru a rotace, ale snaží se ukázat, jaká má tato metoda omezení, zejména jak jsou kvalita a jednoznačnost získaných výsledků ovlivněny množstvím pořízených fotometrických dat. Rozhodujícím kritériem jednoznačnosti úlohy není samotný počet měření, ani jejich rozložení v čase, ale především míra pokrytí tzv. geometrií – různých poloh asteroidu vůči Zemi a vůči Slunci. Autor se míru tohoto pokrytí snaží kvantifikovat jediným parametrem (Ψ), což se mu úspěšně daří, a dále popisuje limity zmíněné metody v závislosti na tomto parametru a přináší řadu zajímavých závěrů. Pokud je mi známo, v odborné literatuře se tímto přístupem dosud nikdo nezabýval. Po rozšíření a doplnění by tato práce mohla být přijata i jako diplomová práce na VŠ či ve stručnější podobě opublikována v odborném časopise (což bych ostatně i doporučoval).

Kromě výše zmíněného, což je dle mého mínění nejzásadnější přínos práce, pořídil autor také vlastní měření pro jeden z analyzovaných asteroidů (511 Davida), což samo o sobě vyžadovalo seznámení se s novou problematikou.

Součástí práce je také rozsáhlý a srozumitelný úvod, v němž autor dobře popisuje zmiňovanou metodu a všechna její specifika, čímž opět prokazuje své hluboké porozumění oboru.

Závěr: Práce Marca de Joode splňuje všechna kritéria soutěže SOČ a má dle mého názoru šanci na vynikající umístění v dalších kolech.

V Ondřejově dne 30.3.2020

Petr Scheirich, AsÚ AVČR

Obsah

Kapitola 1

Teoretický úvod

1.1 Motivace

Prvního ledna 1801, Giuseppe Piazzi objevil novou „planetu“ s dráhou v oblasti mezi drahami Marsu a Jupitera. V následujících letech, astronomové objevili řadu dalších těles na podobných drahách, které se ale v dalekohledu nejevily jako rozlišitelné kotouče jako se jeví planety, ale jako body světla, stejně jako hvězdy. Pro tyto objekty se vytvořilo označení asteroidy, česky také planetky.

Stejně jako v okulárech prvních pozorovatelů, tak i dnes na CCD snímcích z největších dalekohledů se asteroidy jeví jako bodové zdroje světla.[1]

Do nedávné doby neexistoval způsob, jak určovat tvar asteroidů. Přímé radarové pozorování jsou možné jen při těsných přiblíženích se Zemí, které pro asteroidy hlavního pásu nenastávají nikdy. Vesmírné sondy jsou finančně a časově velice nákladné.

Tato práce se soustředí na jednu velmi atraktivní alternativu, a to inverzního určení tvaru čistě z fotometrických měření.

1.2 Konkrétější motivace

Určení tvaru planetek ze světelných křivek řeší inverzní úloha. Jedná se o minimalizační algoritmus, který hledá globální minimum odchylek v závislosti na řadě parametrů. Mezi tyto parametry patří rozmanitost geometrií daných pozorování, polohy rotační osy asteroidu, rotační perioda asteroidu a rozptylové parametry povrchu.

1.3 Rotace

1.3.1 Perioda rotace

Naprostá většina asteroidů nejsou pevnými monolity, ale spíše shluky suti, která vznikla v dřívějších dobách srážkami původních planetesimál a jiných těles. Proto existuje omezení na rychlost rotace: nad určitou úhlovou rychlost ω by části asteroidu poblíž rovníku odlétaly do volného prostoru. Tuto kritickou úhlovou rychlost si můžeme jednoduše odvodit.

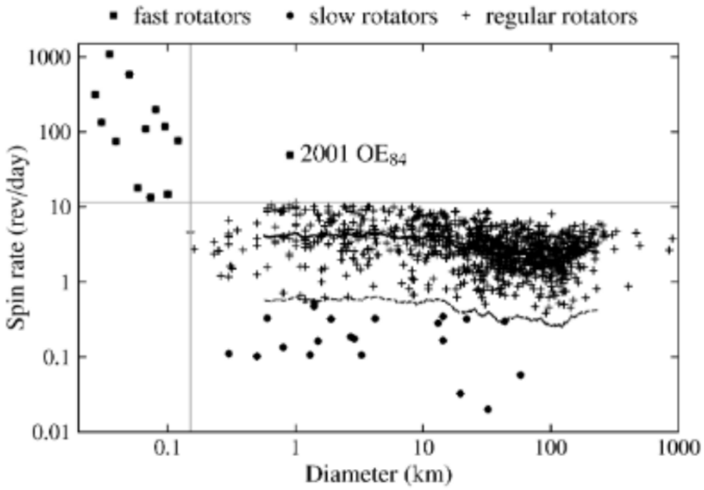
Pokud je celková energie testovacího tělesa na povrchu asteroidu větší než 0, není již k asteroidu gravitačně vázané.

$$E > 0\,, \tag{1.1}$$

$$E = -\frac{GMm}{R} + \frac{1}{2}mv^2\,, \tag{1.2}$$

kde M je hmotnost asteroidu a m je hmotnost testovacího tělesa, R je rovníkový poloměr asteroidu a v je obvodová rychlost testovacího tělesa.

[1] Výjimkou jsou dalekohledy vybavené adaptivní optikou, o kterých bude psáno později.



Obrázek 1.1: Závislost rotační frekvence na průměru planetky. Kromě běžně rotujících těles („haldy suti“ existují i malé rychlé rotátory a velké pomalé rotátory. Malé rychlé rotátory jsou jednolitá tělesa roztočena negravitačními jevy.

Pokud je tedy

$$v > \sqrt{\frac{2GM}{R}}\,, \tag{1.3}$$

testovací těleso přestane být k asteroidu gravitačně vázané. Této rychlosti říkáme úniková. Vyjádříme hmotnost asteroidu pomocí jeho hustoty a poloměru:

$$M = \frac{4}{3}\pi R^3 \tag{1.4}$$

a únikovou rychlost pomocí kritické úhlové frekvence:

$$v = \omega_c R\,, \tag{1.5}$$

z čehož vyplývá, že

$$\omega_c R = \sqrt{\frac{8}{3}G\rho R^2}\,, \tag{1.6}$$

$$\omega_c = \sqrt{\frac{8}{3}G\rho}\,, \tag{1.7}$$

což je velmi překvapivý výsledek. Kritická hodnota úhlové frekvence nezávisí na hmotnosti nebo poloměru asteroidu, ale jen na jeho hustotě.

1.4 Přímé pozorování tvaru

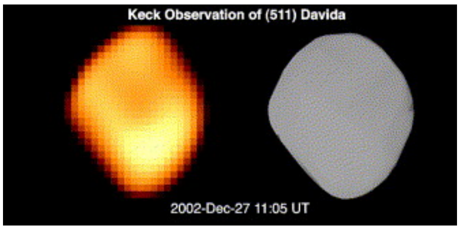
Existuje jen velmi malé množství asteroidů, jejichž tvar by byl znám z přímého pozorování. To proto, že se jedná o poměrně malá tělesa (jednotky až stovky kilometrů v průměru), a i při nejbližších opozicích jsou často velmi daleko. Schopnost je rozlišit závisí na jejich úhlové velikosti, která je dána

$$\theta_{max} \approx \frac{D}{\Delta_{min}}\,, \tag{1.8}$$

kde θ_{max} je maximální pozorovaná úhlová velikost, D je průměr a Δ_{min} je nejmenší geocentrická vzdálenost.

První omezení na minimální úhel, který jsme schopni rozlišit, je dán průměrem dalekohledu a vlnovou délkou, na které pozorujeme. Toto omezení se nazývá Rayleighovo kritérium a vychází z vlnové optiky.

$$\theta_{min} = \frac{1{,}220\lambda}{D_{tel}}\,, \tag{1.9}$$



Obrázek 1.2: Porovnání snímku z Keckových dalekohledů (vlevo) a výsledků inverzní metody (vpravo) z článku [10].

Obrázek 1.3: Keckovy desetimetrové dalekohledy na Havajských ostrovech, které mezi lety 2002 až 2007 pomocí adaptivní optiky pořídily snímky několika planetek, mezi nimi i (511) Davida.

kde θ_{min} nám udává nejmenší rozlišitelný úhel, λ je vlnová délka světla a D_{tel} je průměr teleskopu. Konstanta 1,220 není empirická a vychází z určitých matematických řad.

Bohužel, většina serióznějších astronomických přístrojů (s průměrem nad zhruba 25 cm) nejsou omezeny difrakcí, ale atmosférickými podmínkami. Minimální rozlišitelný úhel díky atmosférickému chvění se nazývá seeing, a v našich podmínkách se typicky pohybuje okolo

$$\theta_s = 3'' \,, \tag{1.10}$$

ale jedná se o funkci tloušťky vrstvy vzduchu, přes kterou se objekt pozoruje, rychlosti větru v různých výškách atmosféry, teplotního gradientu v troposféře a jiných vlivů. Za nejlepších podmínek na nejlepších pozorovacích stanovištích může dosáhnout až $\theta_s = 0{,}8''$, což je ale stále řádově horší, než by bylo potřeba pro pozorování detailů na asteroidech.

Přesto existují způsoby, jak se tomuto omezení vyhnout.

1.4.1 Adaptivní optika

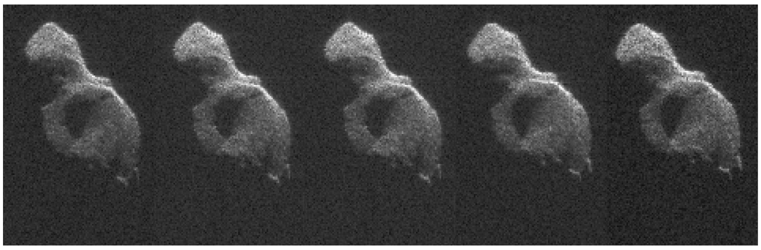
Pokřivení očekávané rovinné vlnoplochy atmosférou se dá aktivně korigovat pokřivováním malého zrcadla. Světlo z velkého primárního zrcadla je před příchodem na detektor soustředěno a odraženo od malého (např. 15 cm) a tenkého zrcadla, na jehož zadní straně je umístěna řada malých pístů, které do něj tlačí na různých místech a tím nepatrně mění jeho tvar. Síla, kterou jednotlivé písty na zrcadlo tlačí se aktualizuje několikrát za sekundu. V celém systému je zpětná vazba.

1.4.2 Radarové pozorování

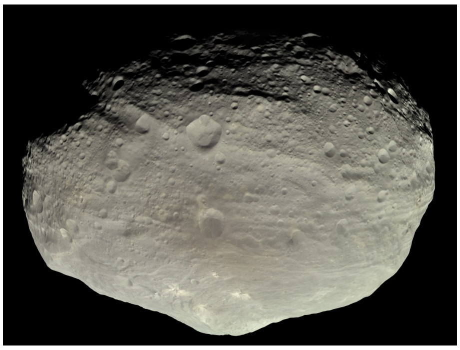
Pomocí velkých radarových vysílačů a přijímačů (Arecibo, Goldstone) lze vyslat rádiové záření na pozorovaný asteroid a sledovat rozdílné časy detekce a fázový posun vyslaných vln. Zpětně jde určit tvar asteroidu. Asteroid se však musí dostat do velmi těsné blízkosti Země.

1.4.3 Vesmírné sondy

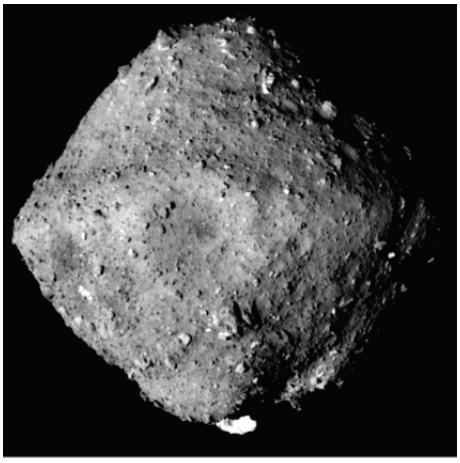
K dnešnímu dni vesmírné sondy navštívili již 16 planetek a 9 komet. V současné době je na planetce (101955) Bennu americká sonda OSIRIS-REx, japonská sonda Hayabusa2 se vrací se vzorky z uměle vytvořeného kráteru na planetce (162173) Ryugu.



Obrázek 1.4: Radarové snímky asteroidu 2014 HQ124 v době těsného průletu kolem Země. Jeho délka (370 m) je jen o trochu větší než průměr radiového přijímače (305 m).



Obrázek 1.5: Planetka (4) Vesta na snímku z mise Dawn z roku 2012. Její průměr je přibližně 530 km.



Obrázek 1.6: Planetka (162173) Ryugu na snímku z mise Hayabusa2 z roku 2019. Jedná se o velmi malé těleso, jehož průměr je přibližně 860 metrů.

1.5 Souřadnicové soustavy a čas

1.5.1 Obzorníková

Nejzákladnější soustavou souřadnic je obzorníková, která popisuje polohu tělesa na obloze pomocí azimutu A a výšky nad obzorem h. V této soustavě se pohybujeme při plánování pozorování. Například na Petřínské hvězdárně třeba platí, že nelze pozorovat objekty na severovýchodě pod výškou $h = 35°$, kvůli stromům, a obecně není dobré pozorovat pod $h = 20°$ kvůli světelnému znečištění a extinkci světla atmosférou. Na severní části oblohy se radši nepozoruje, kvůli technickému řešení kabelů.

1.5.2 Rovníková

Při pozorování ale typicky používáme rovníkové souřadnice 1. a 2. druhu, kde body na obloze popisujeme hodinovým úhlem t, rektazcenzí α (nebo RA), a deklinací δ (nebo DEC).

Převod na obzorníkové provádíme pomocí nautického sférického trojúhelníku, za znalosti zeměpisné šířky ϕ. První pomocí kosinové věty pro sférický trojúhelník:

$$\sin h = \sin\phi \sin\delta + \cos\phi \cos\delta \cos t\,, \tag{1.11}$$

a dále pomocí sinové věty

$$\frac{\sin A}{\cos\delta} = \frac{\sin t}{\cos h}\,. \tag{1.12}$$

Mezi rektazcenzí a hodinovým úhlem platí:

$$\alpha = \Theta + t\,, \tag{1.13}$$

kde Θ je místní hvězdný čas (čas, který uvažuje siderický den namísto Slunečního, tedy den má délku $23^h 56^{min} 4{,}095^s$). Hodinový úhel měříme v hodinách.

Na Petřínské hvězdárně jsou dalekohledy na poměrně dobrých, ale zato manuálně naváděných montážích, a znalost těchto souřadnic je naprosto zásadní. Mezi praktické poznatky například patří, že nelze dobře pozorovat objekty s nízkými deklinacemi (od -10 níže). Pozorování začíná (pochopitelně podle souřadnic objektu) na hodinových úhlech v okolí 18h-20h, kulminace nastává v 24h=0h, a konec pozorování nastává kolem úhlů 4h-6h.

1.5.3 Ekliptikální

Tato souřadnicová soustava je vztažena k Zemskému rovníku, a tedy pro popis těles v prostoru využíváme raději ekliptikální soustavu, která využívá roviny Zemské dráhy: používáme ekliptikální délku λ a šířku β.

Převod do této soustavy provedeme obdobným výpočtem pomocí sférických trojúhelníků, se znalostí sklonu ekliptiky $\epsilon \doteq 23{,}5°$.

$$\sin\beta = \sin\delta \sin\epsilon - \cos\epsilon \cos\delta \sin\alpha\,, \tag{1.14}$$

$$\frac{\cos\beta}{\cos\alpha} = \frac{\cos\delta}{\cos\lambda}\,. \tag{1.15}$$

1.5.4 Kartézské obdoby

Při práci s tělesy v trojrozměrném prostoru Sluneční soustavy pomocí počítače se ukazuje, že je velmi užitečné pracovat v kartézské soustavě souřadnic (předtím jsme zacházeli se sférickými systémy).

Souřadnice si převedeme jednoduchým způsobem, když známe geocentrickou vzdálenost objektu Δ:

$$X = \Delta \cos\beta \cos\lambda \tag{1.16}$$
$$Y = \Delta \cos\beta \sin\lambda \tag{1.17}$$
$$Z = \Delta \sin\beta \tag{1.18}$$

Avšak nyní je třeba brát ohled na to, že se jedná o geocentrické souřadnice, které je případně potřeba opravit o souřadnice Slunce nebo asteroidu, potřebujeme-li je mít centrované jinde.

Chceme-li zacházet s rovníkovými souřadnicemi, můžeme použít například transformační matici:

$$\begin{pmatrix} \xi \\ \zeta \\ \eta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon & \sin\varepsilon \\ 0 & -\sin\varepsilon & \cos\varepsilon \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{1.19}$$

1.5.5 Ko-rotující asteocentrická ekliptikální

Této souřadnicové soustavy využíváme při modelování tvaru. Hledáme v ní nějaký vektor $\boldsymbol{r_{ast}}$.

$$\boldsymbol{r_e} = R_z(\lambda) R_y(\pi - \beta) R_z\Big(\phi_0 + \frac{2\pi}{P}(t - t_0)\Big)\boldsymbol{r_{ast}}\,, \tag{1.20}$$

kde ϕ_0 je nějaké počáteční natočení asteroidu, t_0 je nějaký počáteční čas (např. okamžik prvního měření), P je perioda, $R_z(\theta)$, $R_y(\theta)$ jsou rotační matice kolem osy z, y o úhel θ, ve tvaru

$$R_z(x) = \begin{pmatrix} \cos x & -\sin x & 0 \\ \sin x & \cos x & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.21}$$

$$R_y(x) = \begin{pmatrix} \cos x & 0x & \sin x \\ 0 & 1 & 0 \\ -\sin x & 0 & \cos x \end{pmatrix} \tag{1.22}$$

V případě, že dochází k nějakým negravitačním jevům (YORP efekt), je transformace o tento jev upravena:

$$\boldsymbol{r_e} = R_z(\lambda) R_y(\pi - \beta) R_z\Big(\phi_0 + \frac{2\pi}{P}(t - t_0) + \frac{1}{2}\upsilon(t - t_0)^2\Big)\boldsymbol{r_{ast}}\,, \tag{1.23}$$

kde υ je nějaká konstantní úhlová rychlost odpovídající YORP efektu.

1.5.6 Časové standardy

Astronomie dala lidstvu přesný čas, přesto však bych rád pro čtenáře zpřehlednil vztahy mezi jednotlivými časovými standardy používanými v astronomii.

GMT (Greenwich mean time)

Tento termín označuje čas v časovém pásmu, ve kterém leží Londýn. Pro mnoho případů jej lze zaměňovat s UTC, avšak od UTC se může lišit až o 0,9s. Je ekvivalentní s UT1 a také značí střední sluneční čas na nultém poledníku.

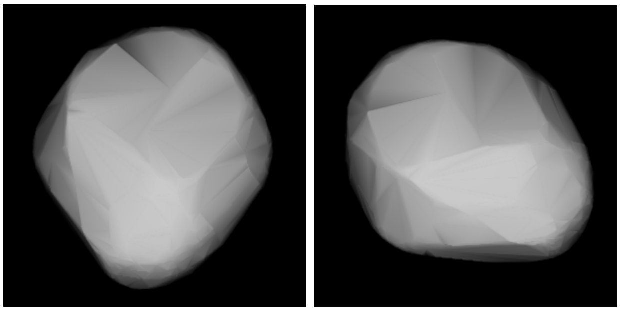
UT, UTC (universal time, coordinated)

Časy UT (kromě UTC) jsou vztažené k rotaci Země, seřizují se tedy vůči vzdáleným nebeským objektům (extragalaktické radiové zdroje, kvazary). Čas UTC je koordinovaný mezi atomovými hodinami. Lze jej vyžádat pomocí NTP (network time protocol) z různých serverů, pokud možno co nejbližších. V České republice je takových serverů několik. Toto lze provést pomocí programu Dimension 4. Opravuje se o tzv. přestupné sekundy.

JD (julian date)

Jedná se o desetinné číslo udávající počet dní uplynulých od poledne 1. ledna 4713 před naším letopočtem. Například, 1.1. 2020 v 12:00:00.000 UT mělo juliánské datum 2458850.0. Jedná se možná o nejpoužívanější formát v astronomii. Je možné jej vztahovat k

- Středu Země (GJD – geocentrické juliánské datum)
- Povrchu Země (TT – terestrický čas)



(a) Ve 20:30 UT (b) Ve 2:15 UT

Obrázek 1.7: Pohled na asteroid (511) Davida, v takovém osvětlení, v jakém byl při pohledu ze Země na začátku a na konci mého pozorování

- Středu Slunce (HJD – heliocentrické juliánské datum)
- Těžišti sluneční soustavy (BJD – barycentrické juliánské datum)
- Vztažené k pozorovanému objektu (zde bude označováno jako asteocentrické datum/čas)

Tyto časy se vyskytují v různých modifikacích, například se mohou lišit o 0,5 dne (vztažené k půlnoci a ne k poledni), nebo mohou být zkráceny o prvních několik cifer. Tento čas není opravován o přestupné sekundy, protože jeho účelem není, aby byla zachována denní doba, ale aby rozdíly časových intervalů byly v různých dobách stejně dlouhé.

Je velmi důležité vědět, se kterým časovým standardem zacházíme. Nepečlivosti vnášejí do výsledků systematické chyby.

1.6 Inverzní úloha

Světelná křivka asteroidu (závislost magnitudy, případně intenzity asteroidu) na čase je jednoznačně dána tvarem asteroidu, geometrií pozorování a vlastnostmi povrchu asteroidu.

1.6.1 Absolutní magnituda asteroidu H

Užitečnou veličinou pro popis světelné křivky asteroidu je jeho absolutní magnituda. Podobně jako ve stelární astronomii zavádíme pojem absolutní magnitudy hvězdy M, která odpovídá vizuální magnitudě hvězdy, kdyby se nacházela ve vzdálenosti 10 pc od pozorovatele, tak pro planetky zavádíme absolutní magnitudu H.

Jedná se o vizuální magnitudu, jak by se pozorovateli jevila planetka ve vzdálenosti 1 au od Slunce a 1 au od Země, v nulovém fázovém úhlu. Je dobré podotknout, že tato situace nemůže fyzikálně nastat. Vizuální magnitudu lze určit jako

$$m = H + 5\log\frac{\Delta_S\Delta_E}{1\mathrm{au}^2} - 2{,}5\log q(\alpha)\,, \tag{1.24}$$

kde d_S, d_E jsou vzdálenosti asteroidu a Slunce, resp. Země. α je fázový úhel a $q(\alpha)$ je fázový integrál definovaný jako

$$q(\alpha) = 2\int_0^\pi \frac{I(\alpha)}{I(0)}\sin\alpha d\alpha\,. \tag{1.25}$$

Tato hodnota nelze zjistit přímo, ale lze ji aproximovat různými modely (o různých modelech odrazivosti povrchů bude psáno níže). Pro difuzně odrážející dokonalou kouli platí, že

$$q(\alpha) = \frac{2}{3}\left(\left(1-\frac{\alpha}{\pi}\right)\cos\alpha + \frac{1}{\pi}\sin\alpha\right)\,. \tag{1.26}$$

1.6.2 Modely odrazu světla

V nejjednodušším případě, kdy bereme pouze jednu matnou (difuzní) odrazivou plochu, platí pro parametr $q(\alpha)$:

$$q(\alpha) = \cos \alpha \tag{1.27}$$

Když tento poznatek srovnáme s 1.26, zjistíme, že při rozptylu světla z matné koule zachytíme pouze 2/3 světla, než z matného disku.

Lambertův zákon

Označme si I, I_r, I_i intenzitu pozorovaného, odraženého a dopadajícího světla.

Platí, že

$$I_r = I_i \cos \theta \,, \tag{1.28}$$

kde θ je úhel mezi normálou povrchu a paprskem dopadajícího světla. Velikost pozorované intenzity I ještě závisí na úhlu κ, pod kterým se díváme na danou plošku (protože tím se mění její pozorovaná úhlová velikost). Tento úhel také odpovídá úhlu odrazu: od matné plošky se paprsky odrážejí ve všech směrech, ale jenom v jednom je pozorujeme.

$$I = I_r \cos \kappa = I_i \cos \theta \cos \kappa = I_i \mu \mu_0 = I_i S_L \,, \tag{1.29}$$

kde členy μ, μ_0 označují kosiny úhlů dopadu a odrazu.

Lambertův zákon rozptylu nám říká, že

$$S_L = \mu \mu_0 \,. \tag{1.30}$$

Lommel-Seelingerův zákon

Lommel-Seelingerův zákon popisuje rozptyl tak, že odražené světlo se od povrchu rozptyluje izotropně (tedy, ne podle kosinu jako v Lambertově zákoně). Jeho koeficient má tento tvar:

$$S_L = \frac{\mu \mu_0}{\mu + \mu_0} \,. \tag{1.31}$$

Oprava o fázi

Použitý model rozptylu světla v proceduře `convexinv` má následující podobu:

$$S = f(\alpha) \Big[S_{LS} + c S_L \Big] \,, \tag{1.32}$$

kde $f(\alpha)$ je člen odpovídající fázi, S_{LS} je člen podle Lommel-Seelingerova rozptylového zákona a S_L je člen podle Lambertova zákona. Fázový člen odpovídá

$$f(\alpha) = a \exp -\frac{\alpha}{d} + k\alpha + 1 \,, \tag{1.33}$$

tedy

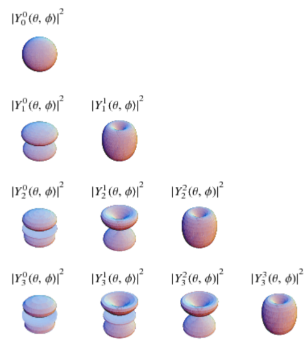
$$S = (a \exp -\frac{\alpha}{d} + k\alpha + 1) \Big[\frac{\mu \mu_0}{\mu + \mu_0} + c \mu \mu_0 \Big] \,, \tag{1.34}$$

kde a, c, d, k jsou rozptylové parametry, kterými je možno fitovat, máme-li přístup k velkému množství fotometrických dat. Ovšem, v našem případě, kdy máme přibližně 60 světelných křivek, tyto parametry jsou zafixovány na hodnotách $a = 0{,}5$; $c = 0{,}1$; $d = 0{,}1$; $k = -0{,}5$, které dobře popisují typický povrch planetky.

1.6.3 Přímá úloha a její inverze

Určení magnitudy nebo intenzity asteroidu probíhá na základě těchto parametrů:

- a, e, i, Ω, ω,M: dráhové parametry planetky a Země (tzv. Keplerovy elementy)
- Perioda planetky



Obrázek 1.8: Prvních několik sférických harmonik.

- λ, β: ekliptikální souřadnice rotačního pólu.
- počáteční čas, počáteční fáze
- Tvar planetky (popsán velkým množstvím normálových vektorů s danou velikostí).
- Rozptylové parametry povrchu.
- Rozměr, albedo (ty ovšem jenom posouvají hodnotu, bez toho, aniž by měnily tvar křivky). Proto je těžké (z tvaru světelné křivky nemožné) je určit.

V práci využívám procedury `convexinv`, kterou vytvořil Mikko Kaasalainnen, a která byla přepsána do jazyka C Josefem Ďurechem. Ukázalo se, že tato metoda je velmi užitečná, mezi její úspěchy patří například experimentální ověření YORP efektu.

Jedná se o algoritmus minimalizující rozdíly modelu a naměřených dat. Minimalizuje

$$\chi^2 = \sum_{i}^{N} \left(I_o^i - I_m^i \right)^2, \tag{1.35}$$

tedy rozdíl pozorovaných a modelovaných intenzit.

Funkce I, která popisuje pozorovanou intenzitu asteroidu v daný okamžik je závislá na výše vypsaných parametrech.

Funkce χ^2 nám definuje plochu nad tímto mnohorozměrným prostorem parametrů, která je hustě poseta lokálními minimy. Modelujeme-li však asteroid konvexním tělesem a máme-li dostatek pozorování, úloha má právě jedno globální minimum, které dobře aproximuje realitu.

1.6.4 Sférické harmonické funkce

Tvar asteroidu se při výpočtu intenzity modeluje sférickým harmonickým rozvojem, kdy výsledný tvar je dán

$$Y(\phi, \theta) = \sum^{m,l} k_{m,l} \, Y_l^m(\phi, \theta)\,, \tag{1.36}$$

kde ϕ,θ jsou sférické souřadnice v asteocentrické korotujicí soustavě a Y je délka průvodiče měřená od počátku soustavy souřadnic.

Při výpočtu je nutné vhodně zvolit nejvyšší hodnoty l a m, aby byl tvar dostatečně dobře popsán a zároveň časová náročnost nebyla příliš velká.

1.6.5 Gaussovský obraz

Procedura `convexinv` ale nezachází s $Y(\phi, \theta)$ jako se spojitou funkcí, ale s jejími hodnotami ve směru konkrétního vektoru.

Jeden sférický oktant (ekvivalent kvadrantu ve dvou rozměrech) je rozdělen na n řádků a n sloupců, a každému tomuto políčku přísluší jeden normálový vektor vedoucí ze středu soustavy souřadnic. Velikost tohoto vektoru odpovídá ploše jeho příslušného „políčka“. Při výpočtu tedy nepopisujeme tvar pomocí vrcholů a stěn, ale pomocí normálových vektorů na tyto stěny – jejichž přesný tvar v této fázi ještě není znám. Celkový počet těchto vektorů (a tedy i výsledných stěn) je

$$N = 8n^2 \,. \tag{1.37}$$

Tento popis se nazývá Gaussovský obraz daného tvaru.

1.6.6 Algoritmus `minkowski`

Algoritmus **`minkowski`** iterativně řeší tvar mnohostěnu na základě normálových vektorů k jeho stěnám. Tento algoritmus je poměrně pomalý velké $N = 8n^2$. Řešíme jej v případě, kdy nám nestačí fotometrické vlastnosti Gaussovského obrazu, ale potřebujeme znát samotný tvar. Jedná se například o srovnávání modelu s jinými metodami (zákryty, přímé pozorování), nebo když je třeba zjistit geometrické nebo dynamické vlastnosti samotného tvaru.

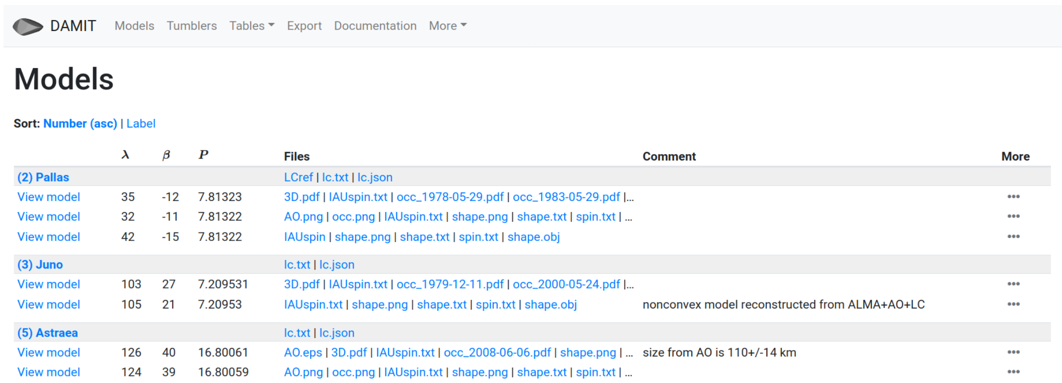
Pomocí procedury **`standardtri`** se vzniklé mnohoúhelníky o různém počtu stran rozdělí na trojúhelníky, se kterými se obecně jednodušeji zachází. Tento krok je již poměrně rychlý.

1.7 Database of Asteroid Models from Inversion Techniques: DAMIT

DAMIT je MySQL databáze 3D modelů asteroidů získaných na základě inverzní metody, kterou vyvinuli Kaasalainen a Torppa (2001) [8] a Kaasalainen et al. (2001) [9]. V současnosti obsahuje přes 4000 modelů pro více než 2000 asteroidů. Většina modelů jsou konvexní s nekalibrovanou velikostí, pro některé častěji měřené asteroidy existují i nekonvexní modely. Některé modely mají kalibrovanou velikost, která se dá určit z tětiv zákrytů, případně z výsledků adaptivní optiky, radioastronomie nebo i vesmírných dalekohledů (Hubble Space Telescope).

Pro planetky (99942) Apophis a 2008 TC3 existují modely s excitovanou (precedující) rotační osou. Samotnou databázi vytvořil Ďurech et al. (2010) [5].

Veškeré modely a zdrojové kódy programů v C a Fortranu jsou veřejně dostupné na `https://astro.troja.mff.cuni.cz/projects/damit/`.

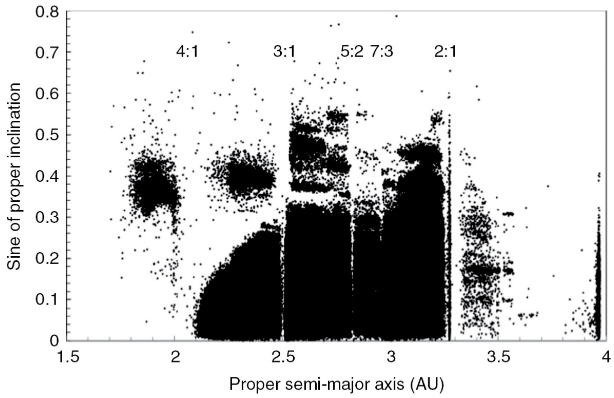


Obrázek 1.9: Nové webové stránky DAMITu.

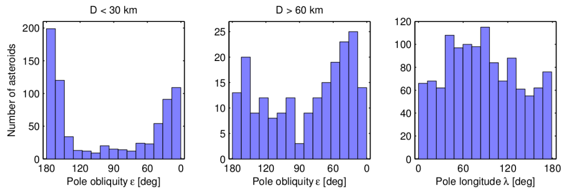
1.8 Negravitační jevy

Přestože tato práce si neklade za cíl se do hloubky zabývat negravitačními jevy, budou zde krátce vysvětleny, protože je zde o nich hovořeno. Jde zejména o Jarkovského efekt a YORP efekt.

Jarkovského efekt je síla působící na rotující planetku, která vzniká na základě anizotropního tepelného vyzařování (stejně jako YORP efekt). Protože planetky mají určitou tepelnou setrvačnost, nejteplejší místo povrchu není subsolární bod, ale nějaké místo, kde již je „odpoledne“, tedy Slunce již prošlo místním poledníkem. Vyzářené fotony předají hybnost nejen ve směru kolmém na oběh,



Obrázek 1.10: Graf znázorňující sinus inklinace na svislé ose a velkou poloosu na vodorovné ose. Každý bod značí jednu planetku. Lze vidět velmi výrazné mezery, pro velké poloosy odpovídající oblastem s orbitální rezonancí s Jupiterem. Dostane-li se těleso do této oblasti, v krátké době dojde k výrazné změně její dráhy.



Obrázek 1.11: Rozložení sklonů pólů vůči ekliptice. Je velmi zajímavé, že póly malých asteroidů směřují většinou buď k jižnímu nebo severnímu ekliptikálnímu pólu. Je to důsledek YORP efektu. Pro větší asteroidy to neplatí.

ale také částečně v tečném směru. Dochází tedy k změnám dráhy, v závislosti na tom, zda těleso rotuje prográdně nebo retrográdně.

Prográdní rotace vede k plynulému prodlužování velké poloosy dráhy, zatímco retrográdní rotace ke zkracování. Tento jev byl přímo pozorován na asteroidu (6489) Golevka, pomocí radarových měření, kdy se v průběhu několika let naměřila změna velké poloosy v řádu centimetrů.

Jedná se o hlavní mechanismus přenosu materiálu z hlavního pásu do okolí Země. Planetky pomalu migrují v rámci hlavního pásu, než se dostanou do oblasti rezonance (Kirkwoodovy mezery), kde dochází v krátké době k náhlé změně dráhy.

YORP efekt (*Yarkovsky–O'Keefe–Radzievskii–Paddack*) je jev, který urychluje nebo zpomaluje rotaci planetek a naklání sklon jejich rotační osy.

Kapitola 2

Pozorování

2.1 Vlastní fotometrická pozorování z Prahy: (511) Davida

Pořídit kvalitní fotometrická měření planetky je poměrně složitá činnost, v porovnání například s měřením proměnných hvězd. To proto, že amplitudy, které světelné křivky planetek vykazují jsou typicky mnohem menší.

Pro účely této práce byla pořízena vlastní fotometrická data na Štefánikově hvězdárně na Petříně (kód observatoře přidělený Centrem pro malá tělesa Mezinárodní astronomické unie: 541), pod dohledem Filipa Waltera, který je aktivním pozorovatelem a organizátorem skupiny CCD pozorovatelů na hvězdárně. Měření probíhala v tzv. domečku, což je nejzápadněji položené pozorovací stanoviště na hvězdárně. Je také v současnosti jediným plně funkčním stanovištěm na hvězdárně, které je pro veřejnost nepřístupné. Při pozorování ze západní kopule je třeba počkat, jestli se nekoná nějaká individuální prohlídka (ty končí ve 23:00). Pokud individuální prohlídka neproběhne, lze pozorovat i v západní kopuli již od 20:00.

Domeček je velmi specifickým pozorovacím stanovištěm, zejména proto, že nemá kopuli. Jako střecha slouží plechová konstrukce, kterou je třeba před pozorováním odsunout po kolejnicích a ty pečlivě zajistit svěrákem, aby se působením větru nezačala pohybovat a ohrožovat techniku i pozorovatele. Nepřítomnost kopule pro techniku a zejména pro pozorovatele znamená, že jsou vystaveni okolním povětrnostním podmínkám po celou dobu pozorování.

V domečku jsou v současnosti dva dalekohledy na původní rovníkové montáži firmy Zeiss. Tato montáž je oproti montážím typicky používanými amatérskými astronomy (EQ5, EQ6) velmi stabilní a spolehlivá, ale nemá elektronické ovládání.

2.1.1 Výběr cíle

Zdaleka ne každou planetku lze nám dostupným vybavením pozorovat fotometricky. Velké observatoře mají pochopitelně mnohem větší výběr cílů, ale získání času na takovýchto stanovištích je náročné, protože bývají velmi vytížené.

Obrázek 2.1: Pozorovací domeček na Štefánikově hvězdárně na Petříně.

Obrázek 2.2: Program Astrophotography tool.

Abychom mohli data efektivně využít pro další zpracování, musí být splněny tyto podmínky:

- Objekt je zařazen v databázi DAMIT.
- Databáze nemá měření tohoto objektu z posledních let.
- Objekt má relativně krátkou periodu rotace, aby byly fotometrické změny pozorovatelné v rámci jedné noci nebo části noci.
- Objekt je dostatečně jasný na to, aby byl pozorovatelný z Prahy 20cm dalekohledem (tedy, jeho magnituda se musí pohybovat v rozmezí 9 až 12 magnitudy).
- Předchozí měření naznačují, že objekt vykazuje fotometrické změny které jsou větší než dosažitelná přesnost měření. Přesnost měření s vybavením na Petříně dosahuje za dobrých podmínek kolem 0,01 magnitudy. Hledáme tedy objekty s amplitudami kolem 0,1 nebo 0,2 magnitudy.
- Objekt je pozorovatelný po co nejdelší část noci.

2.1.2 Ovládací software: APT

2.1.3 Pracovní postup

Před pozorováním je třeba vychladit snímač kamery na požadovanou teplotu (běžně kolem -30 stupňů Celsia) a namířit dalekohled na nějakou jasnou hvězdu a zjistit, zda je hvězdné pole na snímku orientováno severojižním směrem. Je třeba točit kamerou v okulárovém výtahu do té doby, dokud tato podmínka nebude splněna. Tímto zajistíme, aby při pohybu dalekohledu v osách deklinace a hodinového úhlu se objekt na snímku pohyboval kolmo resp. Rovnoběžně s okraji snímku. Poté je třeba dalekohled zaostřit.

Souřadnice pozorovaných objektů je tedy třeba vyhledat anebo spočítat a ručně nastavit na dělených kruzích. Prvně je třeba povolit aretaci v ose deklinace a hodinového úhlu a nastavit přibližnou polohu objektu pomocí vypočítaných souřadnic a hledáčku. Poté je třeba montáž zaaretovat a pomocí jemných pohybů najet na domnělé správné pole. Poté je třeba pořídit pár snímků a zjistit, jestli je dalekohled skutečně zamířen na objekt. Na první pokus se to běžně nepovede, a proto je třeba zjistit o kolik je třeba se posunout v příslušných osách. To se provede ručním srovnáním snímků s počítačovým planetáriem (např. zdarma nástroj Stellarium), případně je třeba nahrát snímky na službu astrometry.net, která snímek vyhodnotí a sdělí souřadnice, na které je dalekohled reálně namířen. Ruční srovnání bývá rychlejší.

Obrázek 2.3: CMOS kamera použita k mému pozorování. Jedná se o kameru ASI1600MM Pro.

2.1.4 Kamery: CCD a CMOS

Petřínská hvězdárna v době pozorování disponovala řadou kamer, z nichž většina jsou digitální zrcadlovky. Pro přesnější měření se však používaly dvě kamery, jedna s CMOS čipem a jedna s CCD čipem (CCD však v době psaní této práce vykazuje řadu závad).

V současné době existuje určitá kontroverze mezi CMOS a CCD snímači. Do nedávné doby byly CCD kamery jednoznačně lepší, avšak nyní jsou s CMOS kamerami srovnatelné. CMOS snímače mají menší pixely, které se rychle nasytí, nelze tedy správně fotometrovat jasné cíle. Na druhou stranu cíl zabírá na snímku více pixelů, které se dají binnovat a tím statisticky zvyšovat přesnost měření. CMOS snímače se vyčítají mnohem rychleji, a jsou tedy vhodné pro planetární fotografii a měření zákrytů.

2.1.5 Fotometrie: redukce dat v programu Muniwin

2.1.6 Kalibrační snímky

Neupravené snímky z kamery v sobě mají velké množství různých nežádoucích artefaktů, kterých je třeba se zbavit. Tyto artefakty vznikají průchodem světla optickou soustavou dalekohledu a následně na snímači.

Chyby vzniklé průchodu světla optickou soustavou většinou spadají do nějaké z následujících kategorií:

- vinětace
- prach na fotometrickém filtru
- nerovnosti na fotometrickém filtru
- prach na sklíčku před snímačem

Chyby vzniklé na snímači se projevují jako tzv. hot pixely (nějaký pixel je pokažený a při vyčítání dává nesprávnou hodnotu), nebo celé řádky nebo sloupce pixelů, jejichž hodnota je o nějakou konstantu posunutá vůči ostatním pixelům.

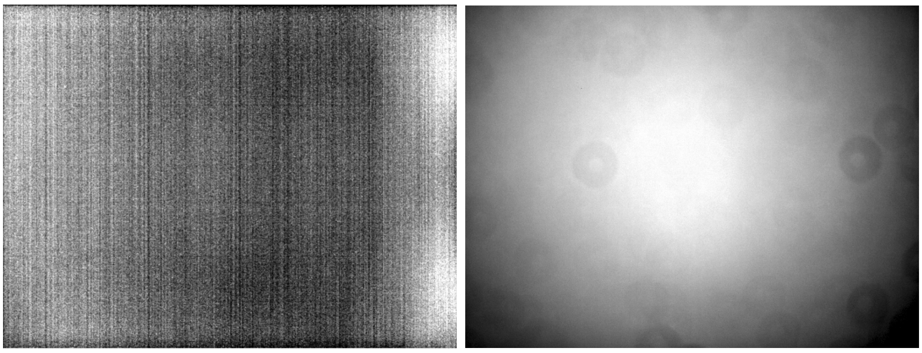
Je velmi důležité se těchto artefaktů zbavit. Při fotometrii jasných cílů s velkými amplitudami nepoužití kalibračních snímků znamená dramatické snížení přesnosti měření, při přesné fotometrii náročných cílů by vynechání tohoto kroku znemožnilo jakoukoli další práci s daty, protože by signál v šumu zcela zanikl.

Dark frame

Dark je snímek pořízený se stejnou expoziční dobou jako light, ovšem na snímač nesmí dopadat žádné světlo. Od lightu jej odečítáme. Tím se zbavujeme artefaktů vzniklých defekty na snímači.

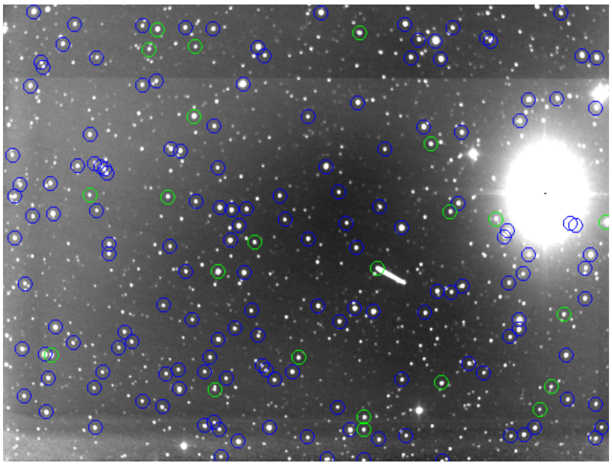
Je třeba pořídit velké množství dark snímků, a to při stejné teplotě snímače jako light snímky. Také je třeba dbát na to, aby byl použit stejný binning pixelů.

Jednotlivé darky průměrujeme do jednoho masterdarku.



(a) Master dark (b) Master flat

Obrázek 2.4: Kalibrační snímky mého pozorování. I samotný flat field je třeba opravit o dark frame, tzv. darkflat.



Obrázek 2.5: Snímky planetky (511) pořízené z Petřína složené tak, aby hvězdy v pozadí byly nehybné. Jasná hvězda vpravo je κ Geminorum, kterou lze vidět i pouhým okem. Planetka (511) vytvořila úsečku, protože v rámci pozorování změnila svou polohu na obloze.

Flat frame

Flat je snímek, který je pořízen při vhodném nasycení pixelů, aby vynikly nerovnoměrnosti způsobené nedokonalostmi optické soustavy. Je třeba, aby nasvícení pozadí bylo rovnoměrné, aby nebyla do dat vnesena nějaká systematická chyba.

Vhodné nasycení nastává například při soumraku nebo svítání, na jasné obloze.

Flat frame se pozorování od pozorování mění, podle drobných změn natočení kamery, podle použitého barevného filtru, podle rozdílného zaostření. Je třeba tedy flat frame pořizovat co nejdříve před nebo po pozorování. Špatné použití flatu může pozorování zcela znehodnotit.

Také lze použít flatovací desky, tj. ploše svítící desky. Ovšem výsledky z nich jsou smíšené.

Každý flat se musí opravit o jeho vlastní dark frame, případně se průměrný master flat musí opravit o jeho průměrný flat dark.

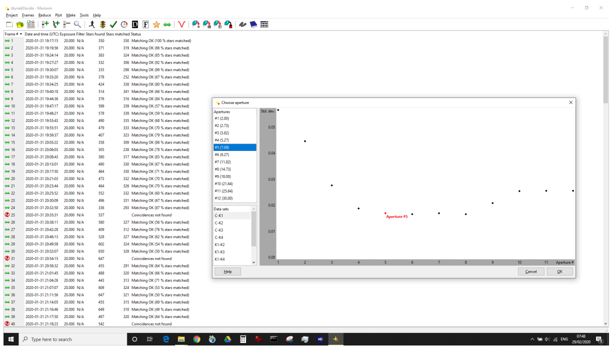
Flat framem je třeba light snímek vydělit. Jednotlivých flat framů se typicky pořizuje lichý počet, protože výsledný master flat je jejich mediánem.

Problematika správné kalibrace je velmi důležitá a je velmi diskutovaná.

Převedení měření do asteocentrické soustavy

Aby byla naše měření zařaditelná mezi ostatní měření dostupná v databázi DAMIT, musíme je převést do příslušného formátu. To znamená, že je třeba provést následující kroky:

- Přepočítat magnitudy na jednotky intenzity světla pomocí Pogsonovy rovnice.



Obrázek 2.6: Prostředí programu Muniwin, který se běžně používá pro zpracování fotometrických dat z proměnných hvězd, lze jej ale použít i pro fotometrii pohyblivého cíle.

- Převést geocentrický čas na asteocentrický čas (čas vztažený k soustavě asteroidu). Rozdíl těchto časů je způsoben nenulovou vzdáleností mezi asteroidem a Zemí. Nejedná se o heliocentrickou korekci, jako při pozorování proměnných hvězd! Při heliocentrické korekci uvažujeme pouze dráhový rozdíl způsobený pohybem Země, zde je třeba vzít v úvahu i polohu asteroidu.
- Doplnit každý změřený bod o informace o poloze Země a Slunce vůči asteroidu. Konkrétně, je třeba uvést souřadnice Země a Slunce v Kartézské ekliptikální asteocentrické soustavě souřadnic. V této soustavě, osa z míří k severnímu ekliptikálnímu pólu, osa x k jarnímu bodu.

Prakticky to provedeme následujícím způsobem. Napsal jsem program v pythonu využívající knihovnu rebound, která umí vyžádat orbitální elementy a efemeridy ze služby HORIZONS od JPL. Pro každý bod pozorování se efemeridy zpětně integrují, a získávají se Kartézské ekliptikální souřadnice Země a asteroidu a Slunce, které jsou centrované na těžiště Sluneční soustavy.

Máme tedy dostupné vektory

$$\boldsymbol{r_s},\, \boldsymbol{r_z},\, \boldsymbol{r_a}$$

v soustavě těžiště Sluneční soustavy a chceme je převést do soustavy asteroidu (můžeme tyto vektory označovat $\boldsymbol{R}$). Platí

$$\begin{aligned} \boldsymbol{r_a} + \boldsymbol{R_z} &= \boldsymbol{r_z}\,, \\ \boldsymbol{R_z} &= \boldsymbol{r_z} - \boldsymbol{r_a}\,, \\ \boldsymbol{r_a} + \boldsymbol{R_s} &= \boldsymbol{r_s}\,, \\ \boldsymbol{R_s} &= \boldsymbol{r_s} - \boldsymbol{r_a}\,. \end{aligned} \tag{2.1}$$

Protože vektory $\boldsymbol{r_s}, \boldsymbol{r_z}, \boldsymbol{r_a}$ žádáme v programu po složkách, také následně po složkách zacházíme se všemi zmíněnými vektory.

Opravu času provedeme jednoduše jako

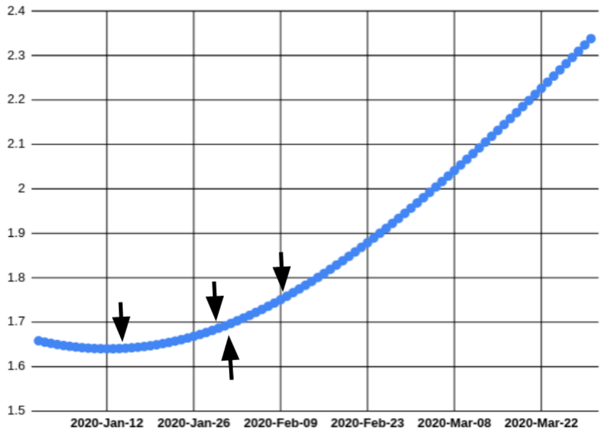
$$\Delta t = \frac{||\boldsymbol{R_z}||}{c}\,. \tag{2.2}$$

Převedením magnitud na intenzitu provedeme vyjádřením z Pogsonovy rovnice:

$$\Delta m = 2{,}5 \log\left(\frac{\Delta I}{I_0}\right), \tag{2.3}$$

$$\Delta I = Q \cdot 10^{\left(\frac{\Delta m}{2{,}5}\right)}, \tag{2.4}$$

kde Q je vhodně zvolená konstanta, aby výsledná intenzita měla číselnou hodnotu blízkou jedničce. To proto, aby ji program `convexinv` mohl jednoduše sesadit k ostatním měřením.



Obrázek 2.7: Vzdálenost planetky (511) v průběhu prvních měsíců 2020. Šipkami jsou označeny noci, kdy probíhala pozorování.

Obrázek 2.8: Dalekohled FRAM na ostrově La Palma (na Roque de los Muchachos, kde sídlí největší světové dalekohledy.) Dalekohled FRAM, operovaný Martinem Maškem, pro ně dělá podpůrná měření atmosferických podmínek.

2.2 Pozorování od jiných pozorovatelů

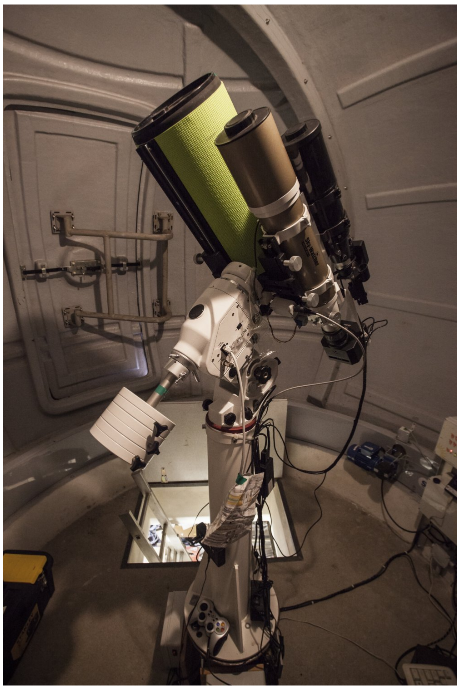
Poprosil jsem řadu přátel a známých, kteří mají k dispozici sestavy schopné pořizovat kvalitní fotometrická měření, aby mi jednu noc měřili planetku (511) Davida. Také touto cestou bych jim rád poděkoval.

Pozorovatel	datum (večer)	Δt	$\Delta t/P$	dalekohled
Souza de Joode, M., Walter, F.	17. 1. 2020	4h 48 min	0.94	Newton 200/1200
Henzl, Z.	1. 2. 2020	2h 47 min	0.54	Newton 203/1000
Mašek, M.	2. 2. 2020	6h 56 min	1.35	Schmidt-Cassegrain 300/3000
Daniš, S.	8. 9. 2020	8 h 45 min	1.71	Cassegrain 200/1254

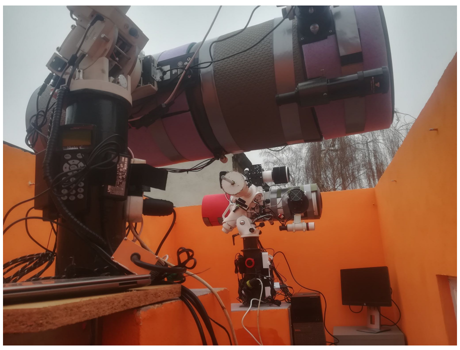
2.3 Archivní pozorování

Jako historickou zajímavost bych rád uvedl, že první odhad periody rotace planetky (511) provedl v roce 1954 Gerard Kuiper [7] na základě svých měření. Určil ji jako

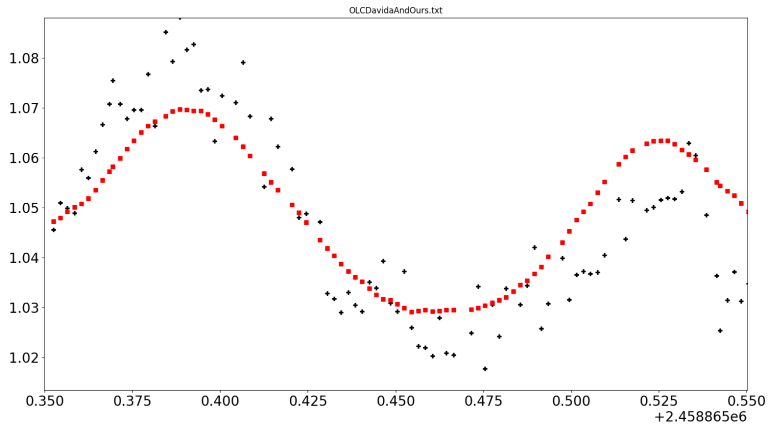
$$P = 5{,}13 \pm 0{,}05\mathrm{h}\,. \tag{2.5}$$



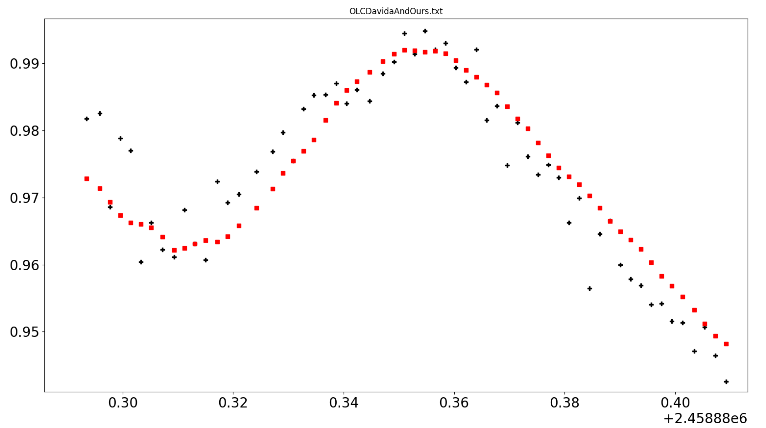
Obrázek 2.9: Pozorovací sestava doc. S. Daniše.



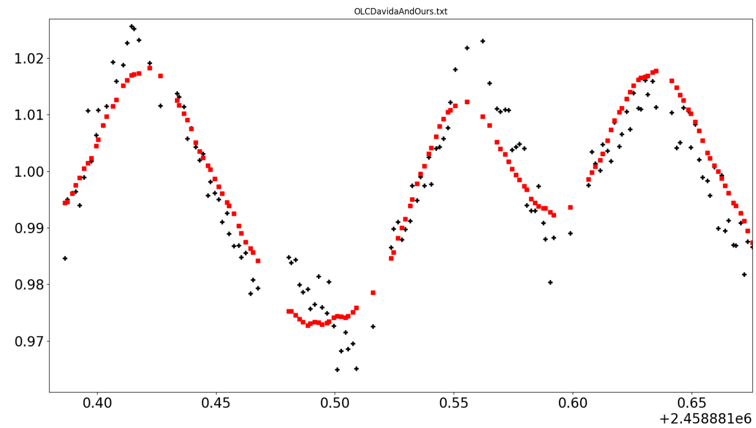
Obrázek 2.10: Pozorovací sestava Zbyňka Henzla. Použit byl bílý 20cm dalekohled v pozadí.



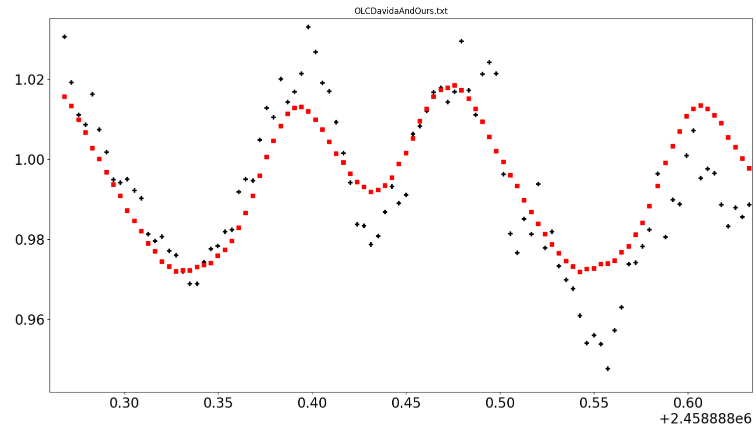
(a) Autorova světelná křivka z Petřína. Černé body jsou měření, červené model.



(b) Pozorování Zbyňka Henzla.



(a) Pozorovací Martina Maška z FRAMu.



(b) Pozorování doc. S. Daniše.

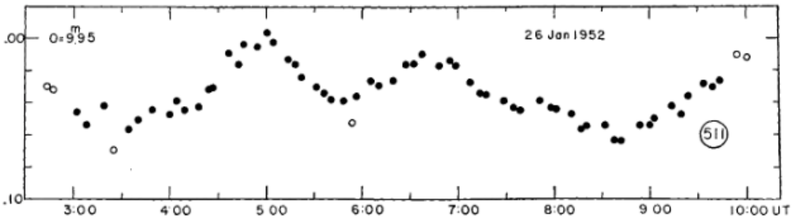


Fig. 20.—Light-curve of 511 Davida

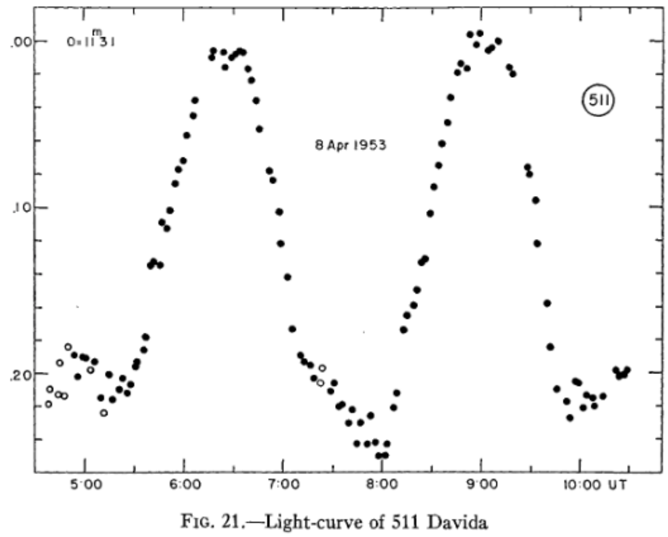
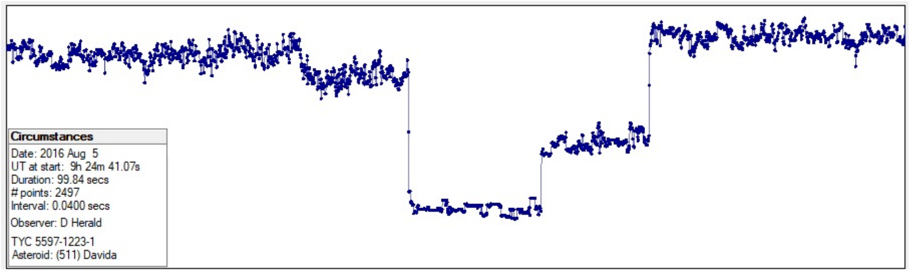
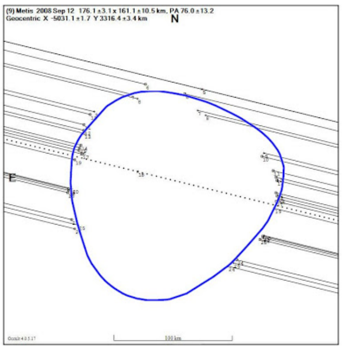


Fig. 21.—Light-curve of 511 Davida

Obrázek 2.13: Pozorování Gerarda Kuipera z roku 1954.



Obrázek 2.14: Zákryt hvězdy planetkou (511) Davida jak ji pozoroval Dave Herald v roce 2016.



Obrázek 2.15: Najde-li se v dráze stínu asteroidu po Zemi větší množství pozorovatelů, lze určovat tvar asteroidu i tímto způsobem. Zde jsou pozorování zákrytu (9) Metis, modrý obrys odpovídá očekávanému tvaru z inverze světelných křivek.

2.4 Pozorování zákrytu

Čas od času se stane, že mezi pozorovatelem na Zemi a nějakou vzdálenou hvězdou proletí planetka. Z daného stanoviště lze pozorovat přibližně jeden zákryt dostatečně jasné hvězdy planetkou týdně. Ovšem k zákrytu hvězdy viditelného z daného stanoviště nějakou konkrétní planetkou dochází jenom velmi vzácně. Zákryt hvězdy planetkou (511) Davida byl pozorován pouze 9krát, a to z různých stanovišť na světě.

Takováto pozorování mohou být velmi cenná. Jedná se možná o nejpřesnější astrometrickou metodu, protože souřadnice hvězd jsou známy nesmírně přesně a střed zákrytu jde často zjistit s přesností na $\sigma_t = 0{,}1$ s.

Při běžném pozorování tohoto nelze dosáhnout, protože přesnost určení polohy je omezena seeingem, velikostí pixelů a difrakcí, a přesnost určení času je omezena délkou expozice. Expoziční doba pro snímání slabších cílů může být velmi dlouhá, většina planetek je pro amatérské dalekohledy pod 30 cm nepozorovatelná.

2.4.1 Pozorování zákrytu planetkou (1017) Jacqueline

Pozorování bylo provedeno na Štefánikově hvězdárně v Praze, v západní kopuli na dalekohledu 350 mm Maksutov-Cassegrain CMOS kamerou ASI1600MM Pro. Pozorování zákrytů již není snímání snímků s dlouhou expozicí, ale natáčení nekomprimovaného videa s přesnou časovou známkou. Formát používaný pro tyto účely jsou soubory `.ser`, a protože data nejsou komprimovaná, již několikaminutové sekvence mají stovky MB.

Kapitola 3

Inverzní metoda a její limity

3.1 Logistika

V rámci této práce bylo vytvořeno přibliže 200 000 souborů s různými vstupními parametry, výstupními parametry, Gaussovskými obrazy, soubory s mnohoúhelníkovými sítěmi popisujícími tvar, vstupními a výstupními světelnými křivkami a řadou dalších.

Bylo třeba vytvořit poměrně přehledný systém složek a podsložek, se kterým se jednoduše strojově zachází a usnadňuje pipelining mezi jednotlivými programy.

Pro tuto práci byly napsány desítky menších programů v Pythonu, které zpracovávají pozorování, charakterizují pozorovací sady, generují systémy vstupních souborů, porovnávají vstupní a výstupní světelné křivky, graficky znázorňují vypočítané hodnoty, charakterizují výstupní tvary a hlavně programy, co převádějí soubory do různých formátů podle potřeby.

Také bylo třeba efektivně nakládat s výpočetním časem. Byla velká snaha omezit čas na jeden výpočet měněním vstupních parametrů a volbou časového rozsahu vstupních dat, přesto celá řada výpočtů trvala desítky CPU hodin.

Tento problém se zčásti řešil tak, že výpočty probíhaly na různých intervalech vstupních parametrů paralelně na osmi jádrech počítače, v různých složkách.

Celý pracovní postup je znázorněn na obr 3.1.

3.2 Stanovení charakterizačního parametru

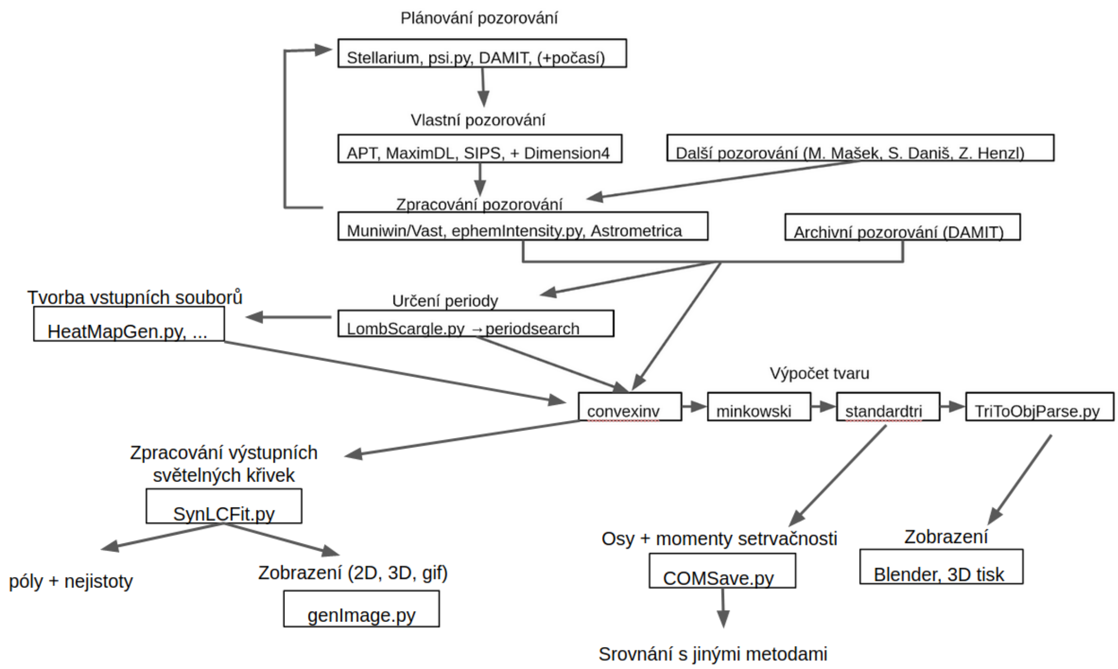
Je třeba stanovit nějaký parametr pro charakterizaci kvality datové sady. Pro určení geometrických a dynamických vlastností planetky (tvar, polohu pólů, sféričnost, momenty setrvačnosti), je třeba provádět měření v různých geometriích.

Nejlepší měření jsou taková (pro zjištění zmíněných vlastností),

- kterých bylo pořízeno mnoho.
- která byla pořizována s dostatečnou frekvencí,
- při velkém rozsahu fázových úhlů
- v různých geometriích.

Nejspolehlivější sady měření jsou takové, které různé geometrie pokrývají co nejrovnoměrněji. Stanovení jednoho parametru, který by popsal velkou řadu různě dlouhých pozorováních, při různých polohách Země, Slunce a Planetky není triviální. Co je autorovi známo, podobnou charakterizací se nikdo ještě nezabýval. Myslím, že by mohlo být pro případného čtenáře přínosné, jaké parametry byly zamítnuty a z jakého důvodu.

- Celkový čas pozorování. Toto by byl intuitivní parametr, avšak pro charakterizaci není vhodný. Je možné například intenzivně měřit objekt pouze při jedné opozici, případně jenom v opozicích. Toto se v datech vyskytuje poměrně často například u slabších cílů, které se dají pozorovat nejlépe v opozici, kdy jsou nejjasnější.



Obrázek 3.1: Diagram znázorňující nejdůležitější kroky, programy a vztahy mezi nimi, pro tuto práci.

- Počet světelných křivek. Narážíme na stejný problém, lze mít velké množství světelných křivek v podobných geometriích.
- Celkový čas od prvního do posledního pozorování. Tento parametr bude mít vliv například na přesnost určení periody. Není ovšem vhodné, když mezi dvěma po sobě následujícíma měřeními je dlouhý časový úsek, protože pak není jednoznačné, jestli uplynulo n nebo $n+1$ period.
- Poměr pozorovacího a celkového času. Toto představuje v určitém smyslu kompromis mezi předchozíma dvěma přístupy, a tedy se může ukázat jako vhodný parametr k popisu například přesnosti určení periody, avšak nijak nezohledňuje geometrii pozorování a nelze jej tedy k popisu geometrie asteroidu použít.
- Těžiště špiček normovaného PAB vektoru. Ukazuje se, že je velmi užitečné zavádět koncept **PAB vektoru**. PAB vektor (phase angle bisector), tedy vektor osy fázového úhlu, je vektor v kartézské asteocentrické ekliptikální soustavě souřadnic, který míří z asteroidu mezi Slunce a Zemi. Konkrétně, jsou-li r_S a r_E vektory z asteroidu k Zemi a Slunci v řečené soustavě, tak

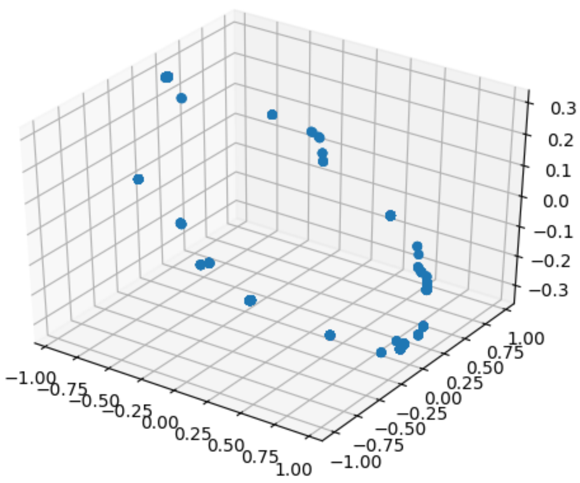
$$P\vec{A}B = \frac{r_S + r_E}{||r_S + r_E||}\,.$$

Tyto vektory leží přibližně v jedné rovině určené rovinou ekliptiky a rovinou oběhu asteroidu (inklinací jeho dráhy).

Máme-li tedy rovinu, ve které vektory přibližně leží (jejich špičky se nacházejí velmi blízko této ideální roviny), je třeba promítnout PAB vektory do této roviny, čímž dostáváme množinu bodů. Mohlo by se například zdát, že čím blíže je těžiště těchto bodů počátku soustavy souřadnic (tedy asteroidu), tím rovnoměrněji byla pozorování pořízena.

Ovšem, narážíme na problém, že například nějaká pozorování byla pořízena přesně na opačných stranách kružnice tvořené množinou bodů. Takovéto sady by se jevily jako dokonalé, přestože by obsahovaly třeba jen dvě měření.

Řečený příklad není nutně okrajový, jsou-li asteroidy poblíž nějaké orbitální rezonance se Zemí, takovéto případy nastávají.



Obrázek 3.2: PAB vektory v 3D prostoru pro všechny dostupné pozorování planetky (511) Davida, jak jsou zobrazeny mým programem. Každý bod je ve skutečnosti celá řada bodů, každý z nich odpovídá bodu na světelné křivce.

Velmi rozumný charakterizační parametr bychom mohli definovat takto. Špičky PAB vektorů promítneme do roviny, ve které všechny přibližně leží (tedy, najdeme nějakou střední rovinu, a do ní je promítneme). Vzniklá množina bodů leží na kružnici, a vyznačují mnohoúhelník. Čím je mnohoúhelník podobnější kružnici, tím je sada pozorování kvalitnější. Poměr ploch mnohoúhelníku a kružnice nazveme Ψ.

3.3 Funkce Ψ

Prakticky tento parametr vypočítáme pomocí níže popsaného programu.

3.3.1 Určení PAB vektoru

Data ve formátu, který je srozumitelný pro procedury `convexinv` a jiné mají následující formát: jedná se o soubor, na jehož prvním řádku je uveden počet světelných křivek jako celé číslo. Počet světelných křivek se pohybuje v rozmezí od 20 do 200, ovšem typicky bývá zhruba v rozmezí 40 až 90.

Následuje řádek obsahující dvě celá čísla, jedno určuje počet řádků (změřených bodů) světelné křivky a druhé určuje, zda je pozorování relativní nebo absolutní. (0 je relativní, 1 je absolutní).

Poté jsou už samotné body světelné křivky, kde na řádku máme 8 desetinných čísel (floatů). První z nich je čas v asteocentrickém juliánském datu. Druhá je intenzita asteroidu. Nakonec následují dvě trojice čísel, souřadnice Slunce a Země v asteocentrické kartézské ekliptikální soustavě souřadnic, v astronomických jednotkách.

Tyto trojice určují vektory r_s, r_z, které určují PAB vektor.

3.3.2 Určení roviny PAB vektorů

Jako jednoduchý a účinný způsob určení roviny se ukázalo vycházet přímo z dat a nikoli z dráhových parametrů asteroidu. Program vezme dva náhodné vektory a těmi určí rovinu. Vezme-li vektory $\boldsymbol{A}$, $\boldsymbol{B}$, které leží přibližně v hledané rovině, která je dána ortonormálními vektory

$$\boldsymbol{I},\ \boldsymbol{J},\ \boldsymbol{K}\,,$$

kde první dva vektory leží v rovině PAB vektorů a vektor $\boldsymbol{K}$ je na ni kolmý.

Tedy, z náhodně vybraných vektorů $\boldsymbol{A}$, $\boldsymbol{B}$ určíme vektory $\boldsymbol{I}$, $\boldsymbol{J}$, $\boldsymbol{K}$ následujícím způsobem.

$$\boldsymbol{I} = \frac{\boldsymbol{A}}{||\boldsymbol{A}||} \tag{3.1}$$

$$\boldsymbol{K} = \frac{\boldsymbol{A} \times \boldsymbol{B}}{||\boldsymbol{A} \times \boldsymbol{B}||} \tag{3.2}$$

$$\boldsymbol{J} = \frac{\boldsymbol{J} \times \boldsymbol{K}}{||\boldsymbol{J} \times \boldsymbol{K}||} \tag{3.3}$$

3.3.3 Projekce

Využijeme vlastnosti skalárního součinu. Pro dva obecné vektory $\boldsymbol{v}$, $\boldsymbol{w}$ platí, že

$$\boldsymbol{v} \cdot \boldsymbol{w} = ||\boldsymbol{v}||||\boldsymbol{w}|| \cos\alpha\,, \tag{3.4}$$

kde α je úhel mezi nimi sevřený. Také platí, že výsledný skalár je délka projece jednoho vektoru do směru druhého. Hledáme souřadnice x, y, do kterých se promítne vektor $\boldsymbol{PAB}$

$$x = \boldsymbol{PAB} \cdot \boldsymbol{I} \tag{3.5}$$

$$y = \boldsymbol{PAB} \cdot \boldsymbol{J}\,. \tag{3.6}$$

3.3.4 Kruhové reziduum

Ne každá náhodně zvolená dvojice vektorů $\boldsymbol{A}$, $\boldsymbol{B}$ ovšem vhodně popisuje rovinu. Proto v programu zavádím proměnnou CircRes, která může být matematicky vyjádřena takto:

$$\mathrm{CircRes} = \sum_{1}^{N} 1 - (x^2 + y^2) \tag{3.7}$$

Nepřesáhne-li hodnota CircRes v počtu zhruba 3500 změřených bodů hodnoty okolo 3, můžeme projekci považovat volbu náhodných vektorů za velmi dobrou. Přesahuje-li hodnota např. 20, je vidět, že dva zvolené PAB vektory například míří podobným směrem nebo jsou jiným způsobem anomální.

3.3.5 Určení plochy mnohoúhelníku

Je třeba rozlišit dva příklady. V jednom z nich je množina bodů dostatečně dobře rozmístěna nato, aby střed kružnice ležel uvnitř mnohoúhelníku, v druhém případě střed kružnice leží mimo konvexní obal samotných měření.

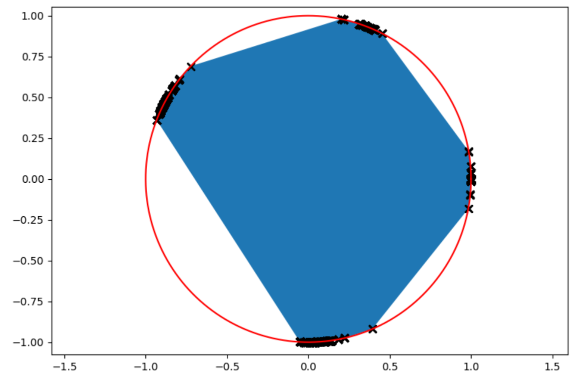
3.4 Porovnání modelů: χ^2, RMS

Jediný způsob, jak určit, zda nějaký konkrétní model dobře popisuje fyzikální realitu, je srovnáním předpovídané světelné křivky (I_m – modelová) a reálných naměřených dat (I_o – observed).

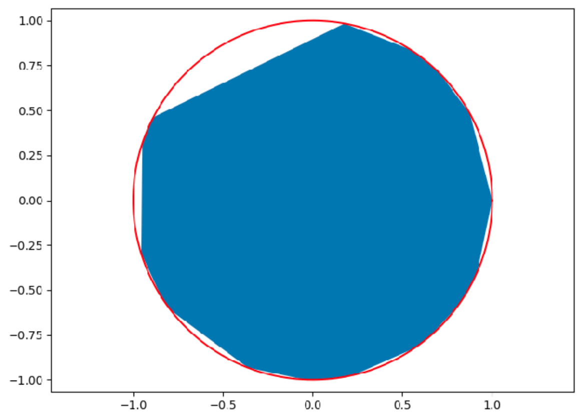
$$\chi^2 = \sum^{N} (I_o - I_m)^2\,, \tag{3.8}$$

kde N je celkový počet měření (ne světelných křivek, ale jednotlivých bodů v nich). Hodnoty I už jsou normované, aby se pohybovaly kolem jedničky. Tato veličina však závisí na počtu naměřených bodů, a proto se také používá veličina RMS (root mean square), kterou určujeme jako

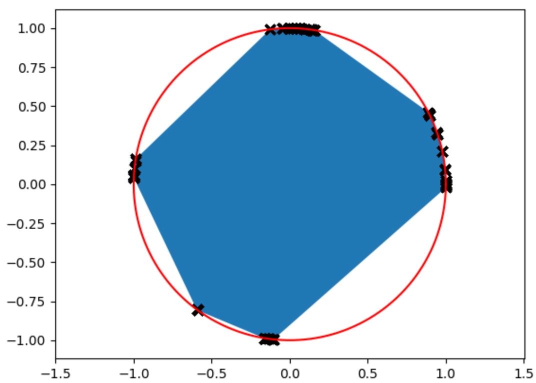
$$RMS = \sqrt{\frac{1}{N} \sum^{N} (I_o - I_m)^2} = \sqrt{\frac{\chi^2}{N}}\,. \tag{3.9}$$



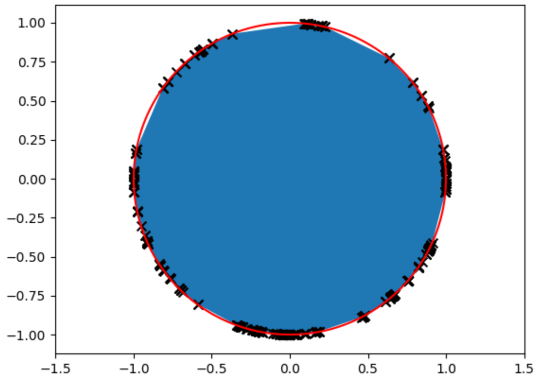
Obrázek 3.3: Křížky značí projekci PAB vektoru do jeho střední roviny. Plocha modrého mnohoúhelníku proti celkové ploše kružnice nám dávají představu o tom, jak dobře jsou zastoupeny různé geometrie v sadě dat. Poměr ploch mnohoúhelníku a kruhu označuji Ψ. Tento konkrétní obrázek je pro měření planetky (16) Psyche z databáze DAMIT. Lze vidět, že přestože bylo pořízeno mnoho světelných křivek (206), byly pořízeny v podobných geometriích.



Obrázek 3.4: Přestože pro planetku (511) Davida najdeme méně světelných křivek (58 z databáze DAMIT + 4 nové pořízené v rámci této práce) než pro (16) Psyche, měření byla pořízena v širším rozsahu geometrií. Pro celou sadu světelných křivek vychází $\Psi \doteq 0{,}9295 \pm 0{,}0003$. Odchylka zde odráží skutečnost, že se jedná o průměr několika výpočtů.



Obrázek 3.5: Znázornění $\Psi = 0{,}797$ u planetky (6070) Rheinland.



Obrázek 3.6: Velmi dobré pokrytí $\Psi \doteq 1$ u planetky (51915) Andry. Přestože má tato planetka vysoké Ψ, nebyla pozorována příliš dlouho.

Protože je lépe představitelné pracovat v magnitudách, než v abstraktních jednotkách intenzity, můžeme si poměry intenzit převést Pogsonovou rovnicí:

$$\Delta m = 2{,}5 \log \left(\frac{I_o}{I_m} \right), \tag{3.10}$$

a definovat si RMS pomocí magnitud:

$$RMS = \sqrt{\frac{1}{N} \sum^{N} (\Delta m)^2}\,. \tag{3.11}$$

Pak platí pro jednotky RMS

$$[RMS] = \text{mag}\,.$$

Je třeba si představovat tuto skalární funkci jako plochu nad mnohorozměrným prostorem, protože χ^2 a RMS závisí na I_o, které je funkcí

$$I_o = I_o(t,\lambda,\ \beta,\ P,\ TVAR,\ \text{geometrie pozorování, rozptylové parametry}) \tag{3.12}$$

Geometrii pozorování v této práci popisujeme veličinou Ψ. Procedura `convexinv` provádí jednotlivé kroky ve směru $\nabla\chi^2$, avšak my se podíváme na řez funkcí χ^2 v různých směrech.

Řez ve směru

$$\frac{\partial \chi^2}{\partial P}$$

provedeme pomocí `periodsearch`, ve směrech β a λ se na řez podíváme pomocí 2D-heatmap.

3.5 Určení periody: Lombův-Scargleův periodogram

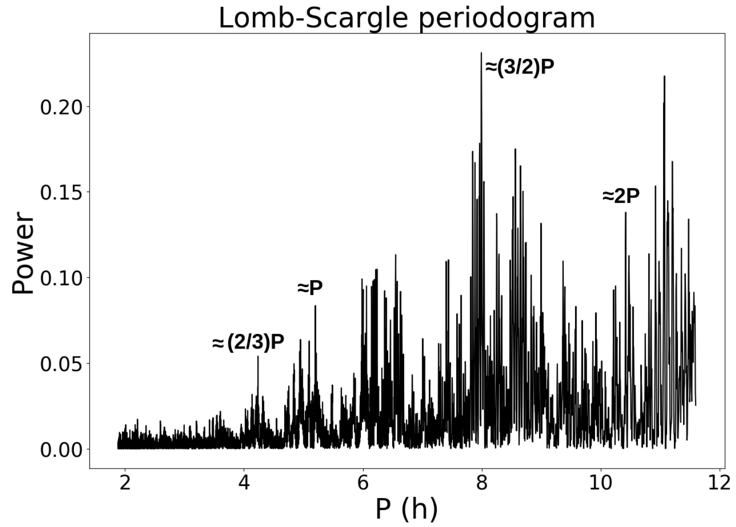
Máme-li data v podobě dvojic času a intenzity, a chceme-li najít odpovídající periodu rotace, můžeme použít něco jako Fourierovu transformaci, kterou se dostaneme do reciprokého prostoru frekvencí.

Tohoto lze často prakticky docílit pomocí algoritmů jako DFT (discrete fourier transform) nebo FFT (fast fourier transform). Podmínkou pro jejich použití je ale rovnoměrné časové rozmístění vstupních dat.

Tato podmínka v našem případě rozhodně splněna není. Délka jednoho pozorování je typicky v řádu jedné periody (hodiny), ale mezery mezi pozorováními jsou v řádu týdnů, měsíců či roků.

Pro tyto účely se používá Lombův-Scargleův periodogram, zde byla použita jeho implementace v knihovně `scipy` pro Python.

Ukazuje se, že tato metoda občas nefunguje pro fotometrická data planetek, která je v mnohém specifická, a selhala i v našem případě. Na grafu 3.5 lze vidět periodu $P = 5{,}2$ h podobnou hledané periodě $P = 5{,}13$ h, a nějaké její harmonické periody.



Obrázek 3.7: Lombův-Scargleův periodogram. Tato metoda určení periody se ale ukazuje jako problematická.

3.6 Určení periody: `periodsearch`

Určení periody není úplně jednoduchou úlohou. Na rozdíl od jiných objektů (zákrytových nebo pulzujících proměnných hvězd) se v různých geometriích výrazně liší amplituda a celkový tvar světelné křivky. Je-li rotační osa planetky namířena naším směrem, fotometrické změny amplitudy téměř vymizí.

Jediným spolehlivým řešením určení periody je provést inverzní úlohu pro celý interval možných period, v poměrně malých krocích.

3.6.1 Přesnost určení

Obecně řečeno, periodické jevy ve vesmíru spojené s rotací bývají velmi stabilní, protože zacházíme s velkými předměty v prostředí, které klade minimální odpor. [1]

Jev, který může měnit periodu rotace je například YORP efekt, ten se ovšem projeví jen u velmi malých těles, a jen v malé míře.

Podobně jako bychom určovali periodu kyvadla, můžeme určovat periodu rotace asteroidu. Došlo-li za dobu Δt k n rotacím (tj. uběhlo n period), periodu jednoduše určíme jako

$$P = \frac{\Delta t}{n}\,. \tag{3.13}$$

Odchylka této periody bude úměrná počtu uplynulých period. Určíme-li na jedné světelné křivce periodu s průměrnou odchylkou σ_{P_1}, tak odchylku po n periodách určíme jako

$$\sigma_P = \frac{\sigma_{P_1}}{n}\,. \tag{3.14}$$

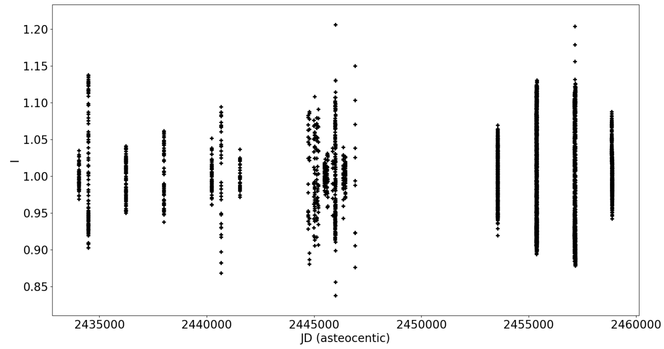
Toto je banální poznatek se kterým máme zkušenost z běžného života (např. kyvadla), který ovšem implikuje, že

$$\sigma_P = \frac{\sigma_{P_1} P}{\Delta t}\,, \tag{3.15}$$

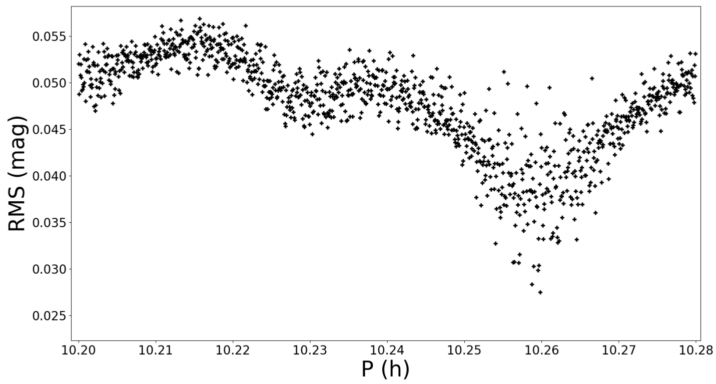
kde σ_{P_1} se pohybuje v řádu P_1. Odchylka určení periody z jednoho měření je nutně kratší než toto měření: je-li perioda v řádu hodin, z jedné světelné křivky ji můžeme určit s přesností v řádu minut (pokud je ovšem jednoznačné, že určujeme P a ne například $2P$ nebo $P/2$, což často nebývá). Pak odchylka určení periody je v řádu

$$\sigma_P = \frac{P^2}{\Delta t}\,. \tag{3.16}$$

[1]Například délka dne na Zemi se měří spolehlivě v řádu desetin milisekund.



Obrázek 3.8: Všechny světelné křivky pro planetku (511) použité pro výpočet, na ose x je časový interval 70 let.



Obrázek 3.9: Periodogram vytvoření pomocí `periodsearch`, kolem periody odpovídající $2P \approx$ 10,3h.

Perioda planetky je v řádu jednotek hodin (desetiny dne) a první dostupné pozorování, které zpracováváme, je z roku 1952: Δt je tedy 68 let, což zhruba odpovídá 25 000 dní. Odchylky se tedy budou pohybovat v řádu zlomků sekund.

3.6.2 Maximální délka kroku

Našim cílem je vytvořit závislost kvality fitu (jako RMS nebo χ^2) na periodě. Výpočet probíhá tak, že se zvolí osm bodů na kouli, které určují počáteční polohu rotační osy. Pro danou periodu (která může také konvergovat k lokálnímu minimu RMS/χ^2) se určí až osm tvarů, z nichž se vybere ten, který dává nejmenší odchylku.

N-rozměrná „plocha" určená závislostí χ^2 na všech geometrických parametrech, všech parametrech tvaru, rozptylových parametrech povrchu a periodě je hustě poseta lokálními minimy. Při krokování počáteční periody nesmíme volit krok, který je větší než tato vzdálenost minim.

Povaha dat je taková, že mezi měřeními trvajícími přibližně jednu noc jsou prázdné prodlevy trvající v řádu měsíců až let. Zásadní otázkou tedy je, zda mezi dvěma minimy nebo maximy světelných křivek uplynulo n nebo $n+1$ (nebo $n-1$) period. Toto lze ošetřit určením maximální délky kroku ΔP. Nevíme tedy, jestli má hledaná perioda délku P nebo $P + \Delta P$.

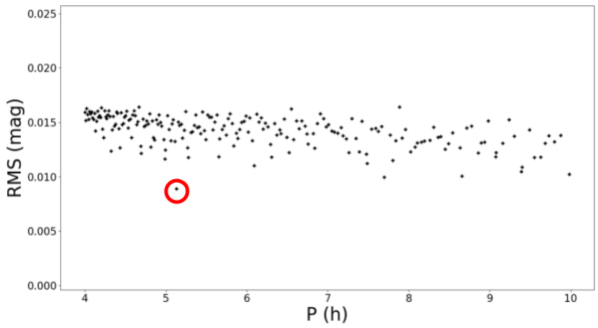
Pomůžeme si soustavou rovnic:

$$n(P + \Delta P) = \Delta t \tag{3.17}$$

$$(n+1)P = \Delta t\,, \tag{3.18}$$

tedy si vyjádřeme

$$\Delta P = \frac{\Delta t - nP}{n} \tag{3.19}$$



Obrázek 3.10: Period scan na základě „našich“ dat, tedy pořízených za rok 2020. Červeně vyznačena hodnota nejlepšího fitu, která je velmi blízká skutečné periodě. Výpočet byl proveden ve velmi širokém rozsahu, od 4 do 10 hodin.

a také

$$n = \frac{\Delta t - P}{P} \approx \frac{\Delta t}{P} \tag{3.20}$$

tedy

$$\Delta P \approx \frac{P^2}{\Delta t}\,. \tag{3.21}$$

Toto by však byl přesně limit toho, co si můžeme dovolit, a na něm by také docházelo k nejednoznačnostem. Je tedy třeba volit

$$\Delta P \approx 0{,}5\frac{P^2}{\Delta t}\,. \tag{3.22}$$

Tato skutečnost je poměrně nepříjemná, protože je třeba provést nesmírné množství výpočtů pro přibližný odhad periody. Výpočty odhadu periody jsou nejnáročnější na čas (desítky CPU hodin).

3.6.3 Minimální charakteristika tvaru

Je třeba, aby výpočet probíhal co nejrychleji. Ukázalo se, že při hledání periody stačí tvar popsat jednodušším způsobem.

Procedura `convexinv` popisuje tvar pomocí součtu sférických harmonických funkcí. Výsledný tvar (funkce) je tedy podobně jako u Taylorova nebo Fourierova rozvoje dán

$$Y(\phi,\,\theta) = \sum^{m,l} k_{m,l}\; Y_l^m(\phi,\,\theta)\,, \tag{3.23}$$

tedy nějakým váženým součtem sférických harmonik. Jako vstupní parametr do procedur `convexinv` a `periodscan` jsou právě hodnoty l a m.

Vyšší řády sférických harmonik odpovídají delšímu času výpočtu. Je třeba najít kompromis mezi přesností a rychlostí výpočtu.

Po několika cvičných výpočtech jsem došel k závěru, že pro menší vstupy lze volit

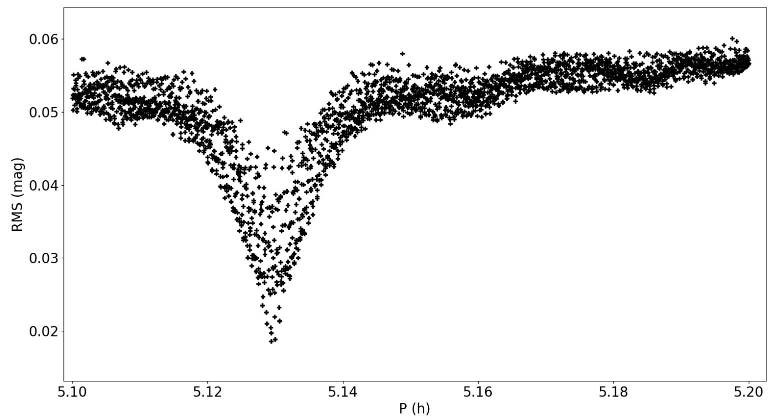
$$l = m = 1\,, \tag{3.24}$$

avšak pro větší je bohužel nutno volit

$$l = m = 2\,. \tag{3.25}$$

Tvar ale není popsán spojitou funkcí $Y(\phi,\,\theta)$, ale její diskretizovanou podobou. Jednotková koule je rozdělena na oktanty (osm stejně velkých ploch/prostorových úhlů), které jsou rozděleny na řádky a sloupce. Je-li oktant rozdělen na n řádků, vede k němu z počátku soustavy souřadnic n^2 vektorů. Celkem tedy $8n^2$ vektorů, které popisují model.

Nejmenší volbou n, která vede k přijatelným výsledkům, je $n = 2$.



Obrázek 3.11: Period scan na základě dat od roku 1982, výpočet v malém rozsahu, 5,1h až 5,2 h. Přesto trval v přepočtu 12 CPU hodin.

3.7 Pohled na RMS ve směrech β, λ: heatmapy

Jedním ze způsobů, jakým lze prozkoumat a znázornit řez funkcí RMS, je pomocí „map“. Při provádění výpočtu `convexinv` je možné změnit, kterými parametry bude program fitovat, a které zůstanou fixní. Při výpočtech těchto map byly tedy fixovány (mimo jiné) i souřadnice rotačního pólu.

Tyto souřadnice jsou v ekliptikální soustavě, tedy $\beta = 90°$ znamená, že rotační pól planetky míří k severnímu ekliptikálnímu pólu. [2]

Výpočty jsem prováděl po krocích o velikosti 5° v obou směrech. Jeho barvou/odstínem šedi znázorňuji, jaké je RMS pozorované a modelované světelné křivky.

Dříve než se pokusíme určovat odchylky u polohy nějakého konkrétního pólu, je třeba vyřešit nejednoznačnost jeho určení.

V databázi DAMIT najdeme přibližně 4200 modelů pro 2400 asteroidů. Tedy, pro velkou část asteroidů v databázi existuje více než jeden model. U dobře proměřených asteroidů (to jsou často ty s nižšími čísly, (3) Juno, (5) Astraea a podobně) najdeme konvexní i nekonvexní model. Avšak u velkého množství hůře proměřených asteroidů najdeme více modelů odpovídající více polohám rotačního pólu.

Tyto nejednoznačnosti vznikají díky nedostatečné různorodosti geometrií pozorování (tedy, pro malé Ψ). Povaha tohoto parametru je taková, že její hodnota se může s jedním dalším pozorováním skokově navýšit: stejným způsobem se může stát, že s jedním dalším pozorováním se skokově vyřadí jeden potenciální pól.

Běžný způsob, jakým se hledají polohy víceznačných pólů, je rozmístění počátečních poloh pólů do středů oktantů koule. Tedy, na souřadnicích

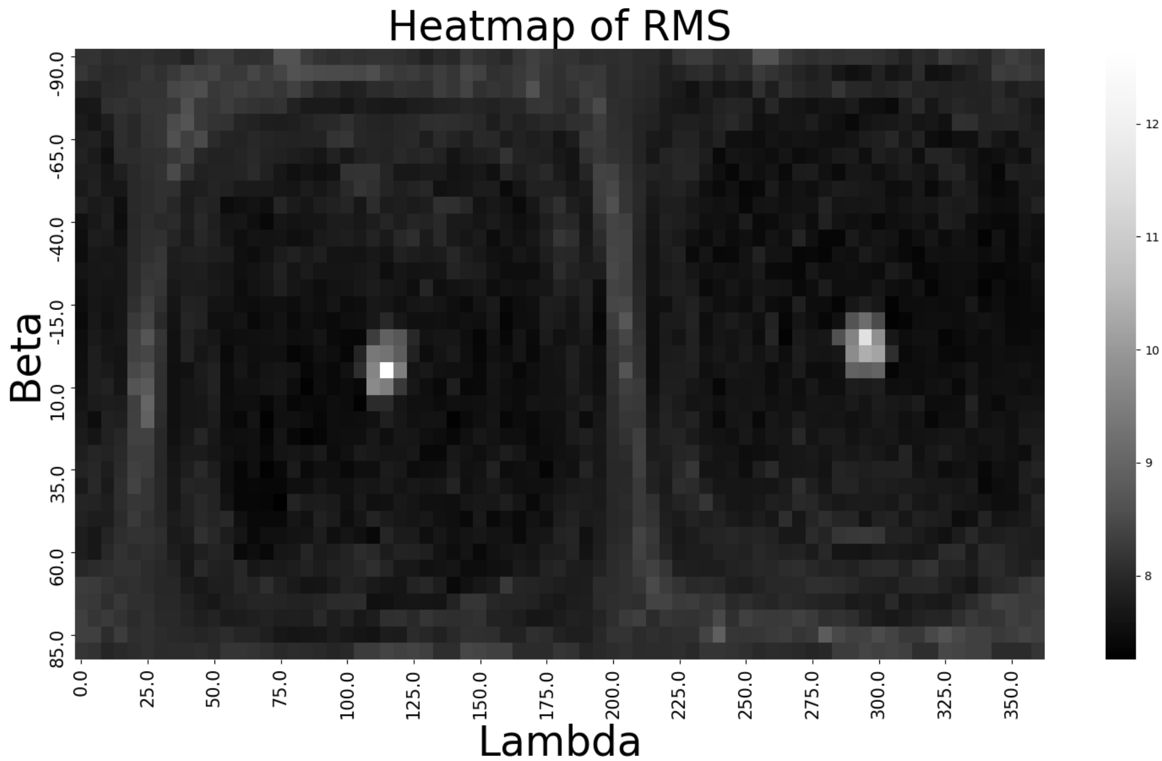
$$\beta \in \{45°;\ -45°\}$$
$$\lambda \in \{0°;\ 90°;\ 180°;\ 270°\}\,.$$

Ovšem, obzvlášť pro menší sady dat, může tato volba počátečních pólů být nedostatečná, a nemusí konvergovat k globálnímu minimu RMS.

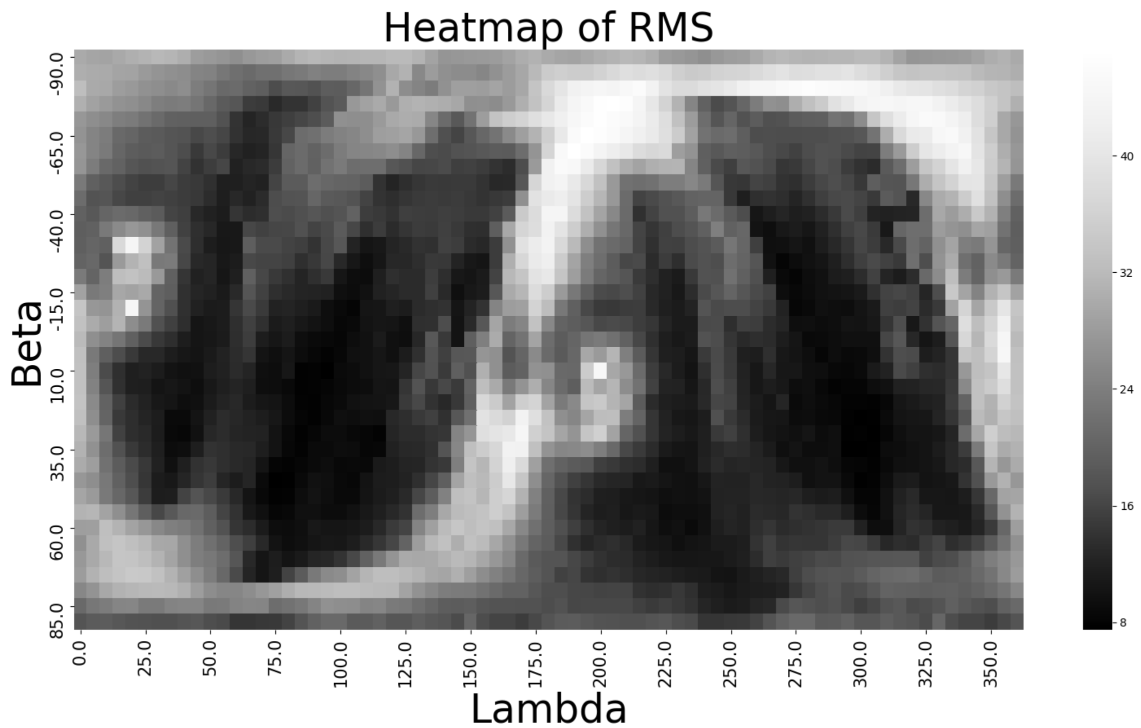
3.8 Histogramy

Pro každý pixel heatmapy bylo určeno odpovídající RMS. Pro každou heatmapu byl vytvořen histogram odpovídající četnosti RMS v řadě binů. Ukázalo se, že se jedná o poměrně dobrý způsob, jakým charakterizovat kvalitu řešení a lokalizace pólu.

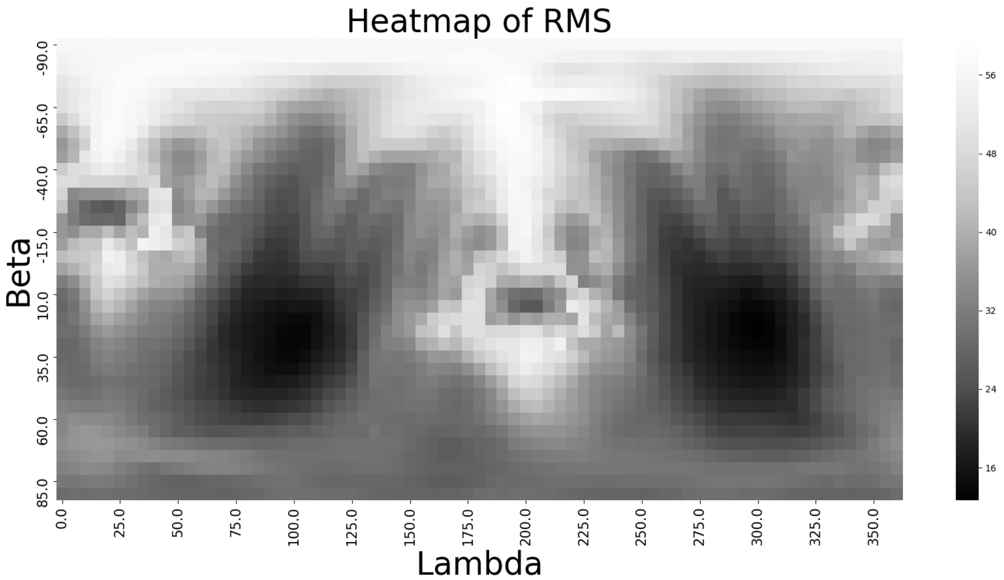
[2]Shodou okolností se severní ekliptikální pól nachází ve směru planetární mlhoviny kočičí oko.



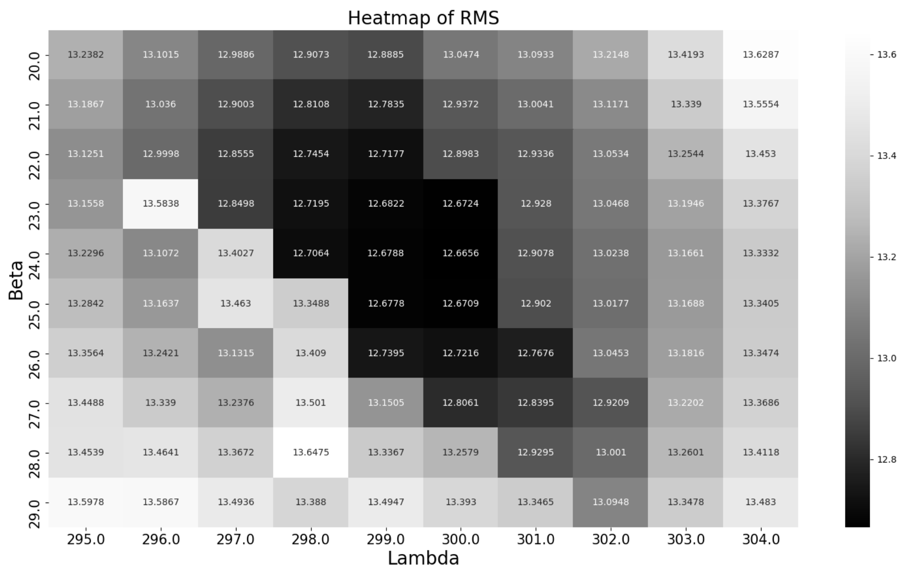
Obrázek 3.12: Mapa znázorňující RMS pro naše měření pořízená v roce 2020. Protože v průběhu měření došlo jen k velmi nevýrazné změně geometrie ($\Psi = 0{,}01$), nelze o poloze pólu tvrdit vůbec nic. Výrazné jsou oblasti, kde je RMS vyšší, a tedy bychom si mohli myslet, že již nyní lze tvrdit, že se tam póly nenacházejí. Ovšem, při tak malém pokrytí Ψ se jedná pravděpodobně o nějaké artefakty nesouvisející s tvarem tělesa.



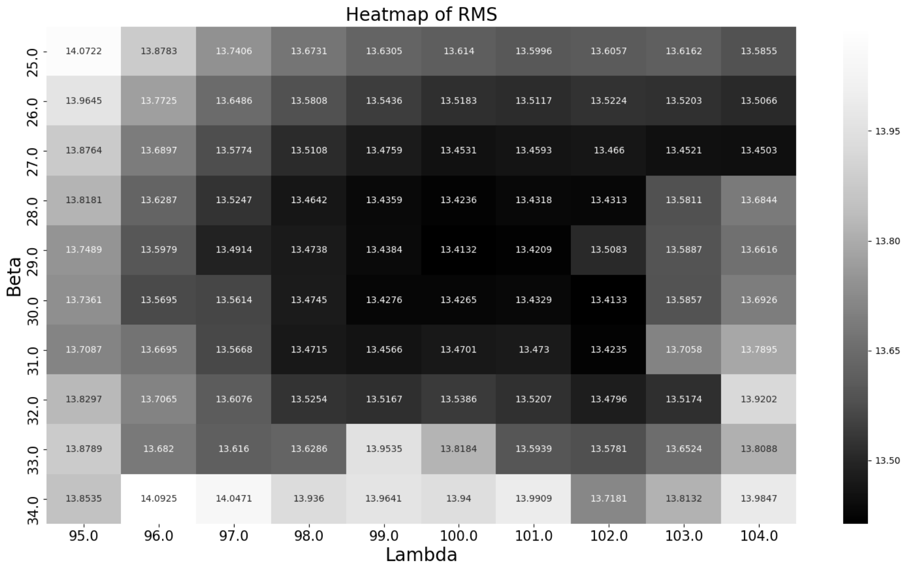
Obrázek 3.13: Mapa znázorňující RMS pro výpočet z malého množství světelných křivek, pokrývajících $\Psi = 0{,}21$. Určení polohy pólu není jednoznačně možné.



Obrázek 3.14: Mapa znázorňující RMS pro výpočet s fixovanými póly na daných souřadnicích. Barevná škála RMS je v milimagnitudách, kde černá je přibližně 13 mmag a bíla asi 60 mmag. Výpočet byl proveden na všech dostupných měřeních, tedy $\Psi = 0{,}93$.



Obrázek 3.15: Mapa znázorňující RMS v okolí jednoho z možných pólů, na souřadnicích $\lambda = 300°$ a $\beta = 24°$. RMS uvedeno v každém binu, v mmag.



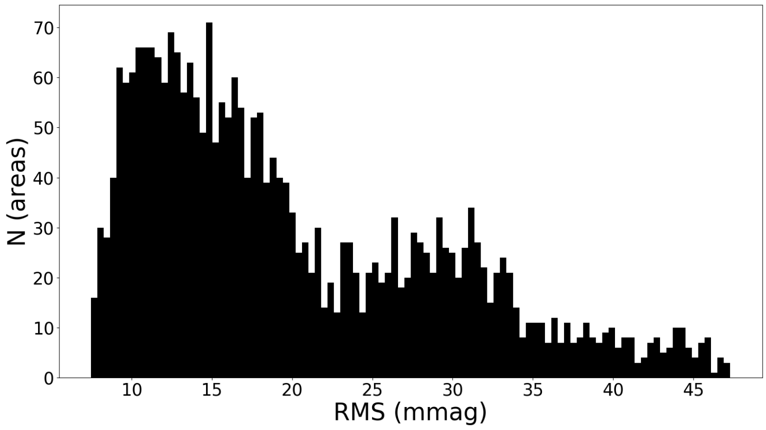
Obrázek 3.16: Mapa znázorňující RMS v okolí jednoho z možných pólů, na souřadnicích $\lambda = 100°$ a $\beta = 29°$. RMS uvedeno v každém binu, v mmag.

Nedostatečné sady dat produkují histogramy, kde většina pixelů dává podobné řešení jako nejlepší řešení. Nejde tedy diskriminovat mezi dobrými a špatnými výsledky, a poloha pólu je tedy nejednoznačná. To je příklad pro malé sady dat u asteroidu Davida (obrázky 3.15, 3.21, 3.22).

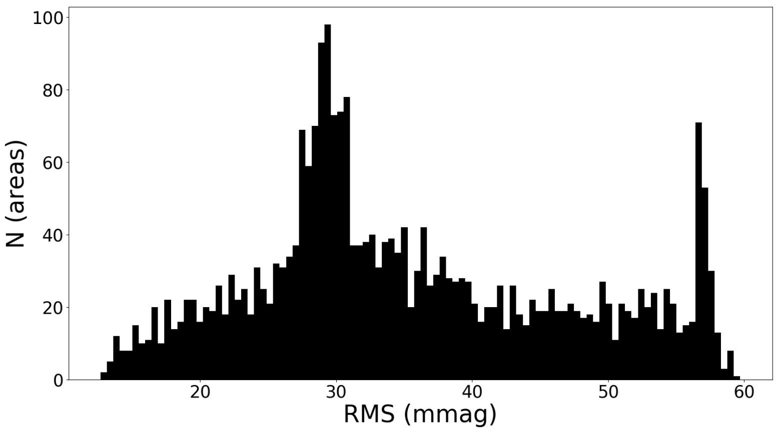
Naopak, velké a geometricky rozlišné sady produkují takové histogramy, kde v binech kolem nejlepšího řešení (levá strana osy x) je jenom málo pixelů.

Porovnávat asteroidy mezi sebou je v tuto chvíli trochu problematické, protože do lokalizace pólu vstupuje velká řada proměnných, a ne všechny jsme schopni regulovat - například albedové útvary a jiné.

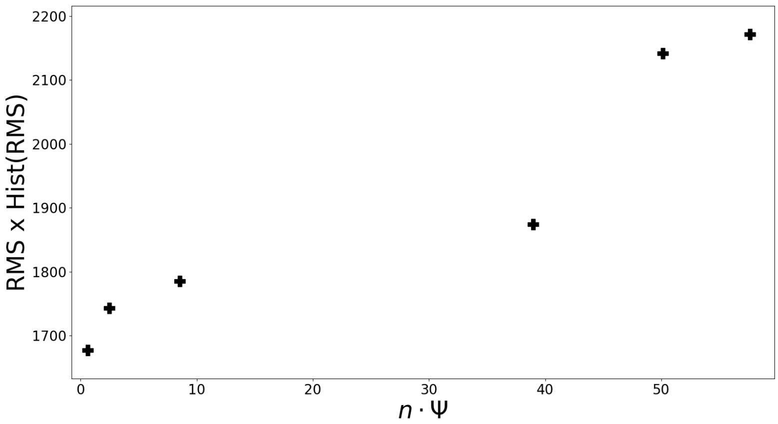
Obrázek 3.17



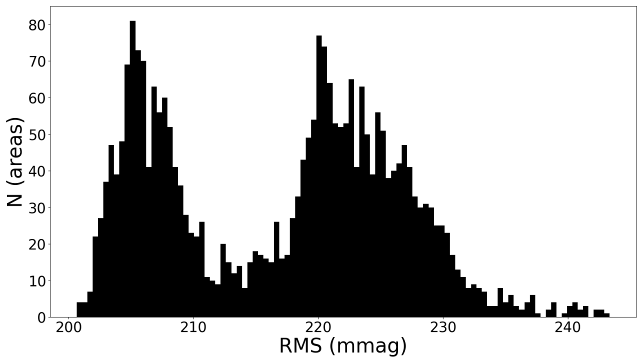
Obrázek 3.17: Histogram znázorňující různé počty „pixelů,“ tedy výsečí prostorového úhlu, pro které bylo zjištěno RMS v daném binu. Zde pro malou sadu světelných křivek (Ψ = 0,21). Lze vidět, že většina světelných křivek (odpovídajících jednomu pixelu heatmapy) nemá RMS příliš odlišné od nejlepšího řešení. Pro asteroid (511) Davida.



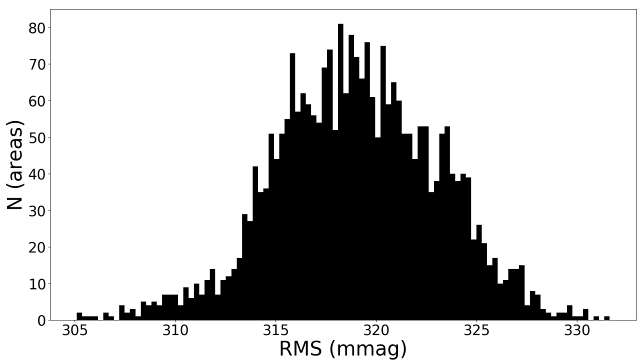
Obrázek 3.18: Histogram znázorňující různé počty „pixelů,“ tedy výsečí prostorového úhlu, pro které bylo zjištěno RMS v daném binu. Výpočet proveden pro všechny dostupné světelné křivky (Ψ = 0,930). Na straně nejlepšího řešení (nalevo) je již výrazně méně světelných křivek. Tedy, větší počet měření pomůže rozlišit mezi dobrými a špatnými počátečními polohami rotační osy. Z histogramu vyčteme, zda odpovídá dobré heatmapě tak, že „těžiště“ histogramu je posunuto z levé části ke středu. Levá část, odpovídající dobrým počátečním polohám osy, je tedy méně zastoupená a řešení jsou jednoznačnější. Pro asteroid (511) Davida.



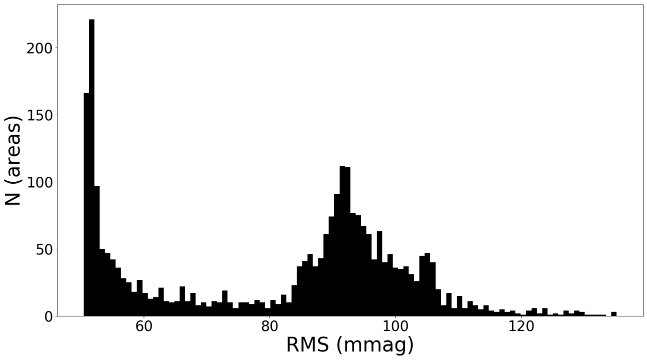
Obrázek 3.19: Závislost „vyváženosti“ histogramu (popisovanou součinem počtu pixelů v binu a RMS binu) a $n\Psi$, kde n udává počet světelných křivek, použitých k výpočtu mapy. Jenom poslední dva body odpovídají skutečně jednoznačnému řešení.



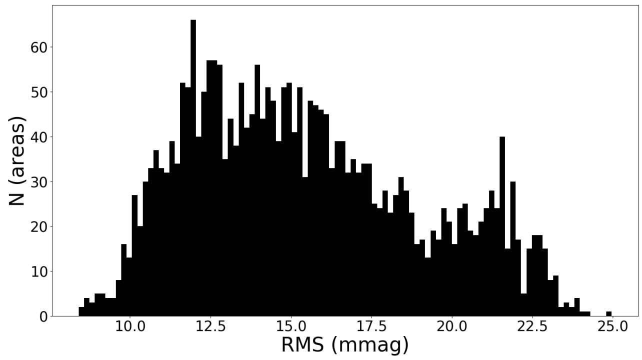
Obrázek 3.20: Tentýž výpočet pro (1388) Aphrodite. Zde se zdá, že řešení spadají do dvou „tříd“.



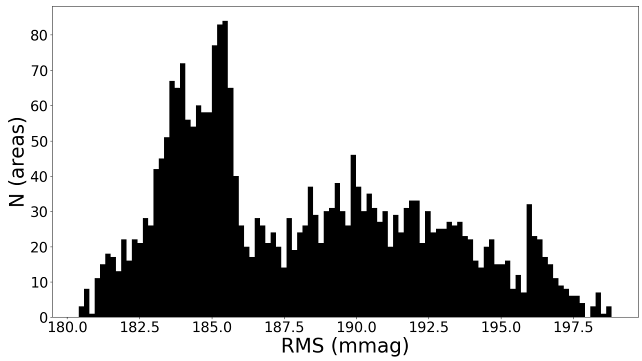
Obrázek 3.21: Tentýž výpočet pro (51915) Andry. Zde je mezi řešeními pólu dvojznačnost, ale zato jsou obě polohy velmi přesně lokalizovány. Lze to poznat z „pomalého začátku“ histogramu.



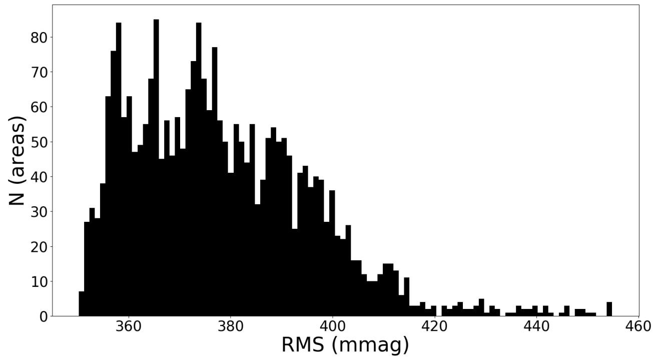
Obrázek 3.22: Tentýž výpočet pro (6070) Rheinland. Vidíme zase dvě maxima, první je velmi silné. Je to dáno tím, že se rotační pól nachází v blízkosti ekliptikálního pólu, a nikdy jej tedy nepozorujeme přímo. Také je to ale dáno způsobem výpočtu, kde polární oblasti jsou převzorkované. $\Psi = 0{,}797$.



Obrázek 3.23: Tentýž výpočet pro (227) Philosophia. Podobně jako u asteroidu (51915) Andry je poloha sice nejednoznačná, ale obě řešení jsou dobře lokalizované.



Obrázek 3.24: Tentýž výpočet pro (281) Lucretia. Podobný tvar jako histogram (511) Davida. $\Psi = 0{,}987$.



Obrázek 3.25: Tentýž výpočet pro (7233) Majella. Zde má histogram „těžiště" posunuté velmi dopředu. To proto, že (7233) má zatím poměrně malé $\Psi = 0{,}317$.

3.8.1 Provedené výpočty

(511) Davida

Pro (511) Davidu bylo provedeno 10 výpočtů s různými sadami světelných křivek odpovídajících různým hodnotám Ψ, přes různá časová období.

Kroky v λ a β mají stejnou velikost, a to 5°. Počet pixelů na jedné mapě je tedy

$$N_{px} = (72 + 1)(36 + 1) = 2701 \text{ px} \tag{3.26}$$

kde +1 značí duplicitní řádky, odpovídající překryvu na ekliptikálních pólech a na nultém poledníku. Jsou zdánlivě převzorkovávány oblasti pólů, ale ty jsou právě problematické tím, že při pohledu ze Země (ležící v rovině ekliptiky odpovídající $\beta = 0°$) jsou vždy pozorovány v tečném směru.

Výpočet jednoho pixelu (odpovídá řešení inverzní úlohy, kde jediným volným parametrem je tvar) trvá na jednom jádru přibližně 20 až 40 sekund, pokud není řešen `minkowski`. Celkově na jeden výpočet tedy připadá přibližně 15 až 30 CPU hodin, rozděleno mezi 8 jader 2 až 4 hodiny.

Také byl proveden detailnější výpočet dvou oblastí, které se při úvodním hrubším výpočtu jevily jako pravděpodobné oblasti výskytu pólu, v kroku 1° a s vyššími hodnotami počtu řádků na jeden oktant.

(13) Egeria

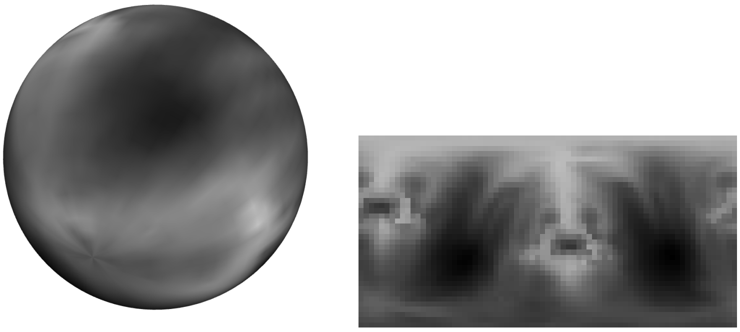
Jako další byla vybrána měření planetky (13) Egeria. Jedná se o velký a dlouho známý asteroid (objeven roku 1850). Z toho důvodu pro něj existuje velké množství světelných křivek přes dlouhé časové období.

O to zajímavější je skutečnost, že poloha jeho rotačí osy je velmi špatně určená (tab. 3.1), což byl důvod, proč byl proveden výpočet.

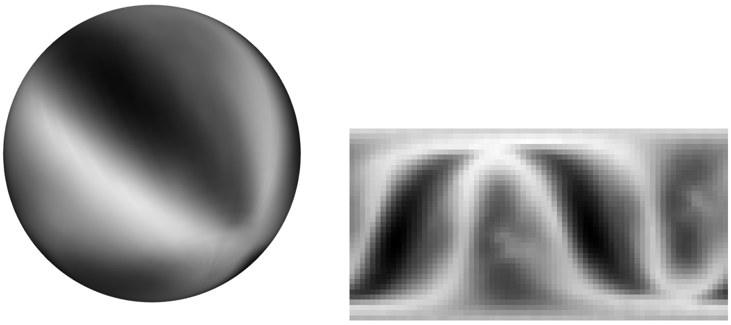
Na obr. 3.25 lze vidět, že existuje hlavní kružnice (poledník), na které se pól pravděpodobně nachází. Důvod této neurčitosti jsou pravděpodobně albedové útvary na povrchu.

model	λ (°)	β (°)
kovexní A	44	21
konvexní B	238	11
nekonvexní A	54	34
nekonvexní B	233	6

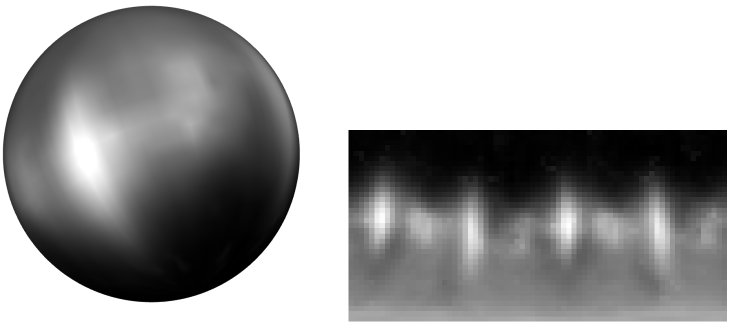
Tabulka 3.1: Polohy pólu pro (13) Egeria z databáze DAMIT



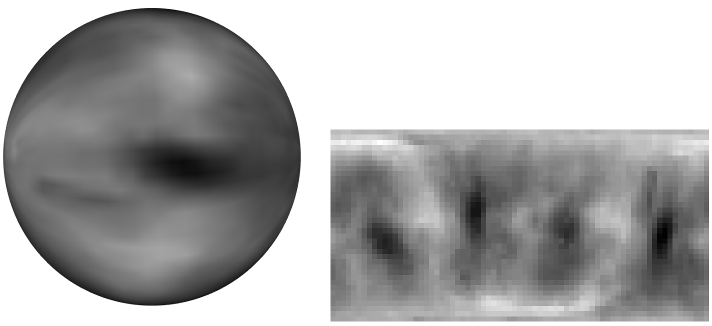
Obrázek 3.26: (511) Davida. $\lambda \approx 299°$, $\beta \approx 24°$. $P = 5{,}129644$h, $\Psi = 0{,}93$. Reference na světelné křivky jsou v tab. 4.1.



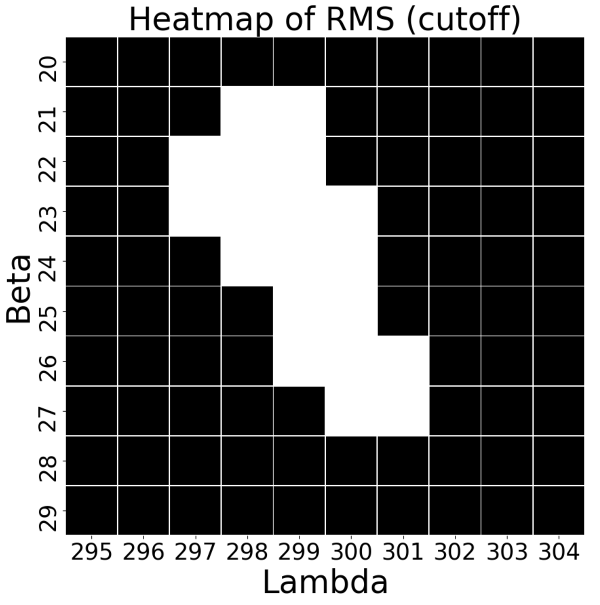
Obrázek 3.27: (13) Egeria. λ ?, β ?. $P = 7{,}04667$h, $\Psi = 0{,}988$. Světelné křivky z [6].



Obrázek 3.28: (6070) Rheinland. $\lambda \approx 124°$, $\beta \approx -87°$. $P = 4{,}27371$h, $\Psi = 0{,}797$. Křivky z [12].



Obrázek 3.29: (51915) Andry. $\lambda \approx 124°$, $\beta \approx -20°$ nebo $\lambda \approx 305°$, $\beta \approx -1°$. $P = 14{,}8956$h, $\Psi = 0{,}987$.



Obrázek 3.30: Mapa znázorňující pro každý pixel velikosti $1° \times 1°$, zda by se v něm v rámci 1 směrodatné odchylky mohl nacházet rotační pól. Jedná se o stejnou oblast jako na obrázku 3.13.

(6070) Rheinland

Tento asteroid byl zvolen, protože jeho rotační pól míří velmi blízko k jižnímu ekliptikálnímu pólu. Jedná se mimochodem o binární těleso.

3.9 Určení nejistoty polohy pólu

Platí přibližně, že pokud je

$$\mathrm{RMS} < \mathrm{RMS}_{min}\left(1 + \frac{1}{f}\right) \tag{3.27}$$

tak se hodnota nachází v rámci jedné směrodatné odchylky σ. f značí počet stupňů volnosti a odpovídá počtu světelných křivek. To proto, že použitá data jsou relativní s každou světelnou křivkou lze posouvat o konstantí hodnotu intenzity nahoru nebo dolů.

3.9.1 Zánik zrcadlové dvojznačnosti

Přestože by se mohlo zdát, že například mapa 3.12 je zrcadlově symetrická podle nějakého ekliptikálního poledníku, není tomu tak. Při překročení určité hranice hodnoty Ψ tato dvojznačnost zaniká. Bylo zjištěno, že v našem případě dochází ke kolapsu této dvojznačnosti v intervalu

$$0{,}78 < \Psi < 0{,}91\,, \tag{3.28}$$

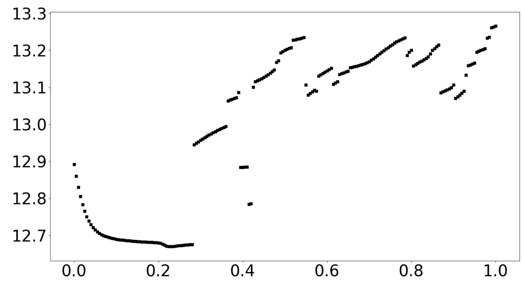
ale v obecnosti, pro různé hodnoty inklinace dráhy, nemusí k tomuto kolapsu dojít vůbec.

Je-li již zřejmé, který z pólů prokazatelně lépe odpovídá fyzikální realitě, je možné určit odchylky polohy tohoto pólu.

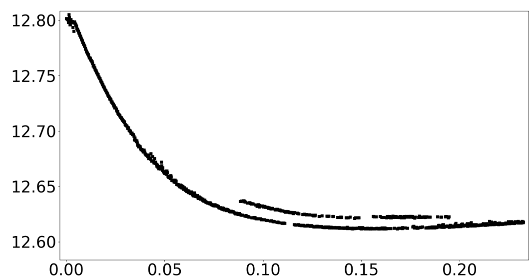
$$\lambda = (299{,}5 \pm 3{,}5)°$$
$$\beta = (24 \pm 2{,}5)°$$

Pro představu, odpovídající místo na obloze této poloze pólu je v souhvězdí Orla zhruba dva stupně jižně od hvězdy Altair.[3]

[3]To také znamená, že Altair je pro (511) Davidu „polárkou“.



Obrázek 3.31: Výpočet RMS pro různé hodnoty rozptylového parametru $c \in (0,\ 1)$.



Obrázek 3.32: Výpočet RMS pro různé hodnoty rozptylového parametru $c \in (0;\ 0{,}3)$.

Hodnoty odchylek ve směru ekliptikální délky a šířky nejistotu trochu nadhodnocují, protože oblast nejistoty pólu je určena elipsou kolmou na oba tyto směry. Její poloosy jsou

$$\sigma_a \doteq 3{,}5^\circ$$
$$\sigma_b \doteq 1^\circ\,,$$

což v prostoru vytyčuje prostorový úhel

$$\Omega_\sigma \doteq 5 \cdot 10^{-3}\ \mathrm{sr}\,. \tag{3.29}$$

3.10 Rozptylové vlastnosti povrchu

Jak již bylo zmiňováno v úvodu, funkce použitá k modelování rozptylu světla má podobu

$$S = (a \exp -\frac{\alpha}{d} + k\alpha + 1)\Big[\frac{\mu\mu_0}{\mu + \mu_0} + c\mu\mu_0\Big]\,, \tag{3.30}$$

kde a, d, k, c jsou empirické parametry popisující rozptyl. Člen v kulatých závorkách je funkcí fáze a člen v hranatých závorkách rozptyl. Parametr c popisuje příspěvek Lambertova zákona

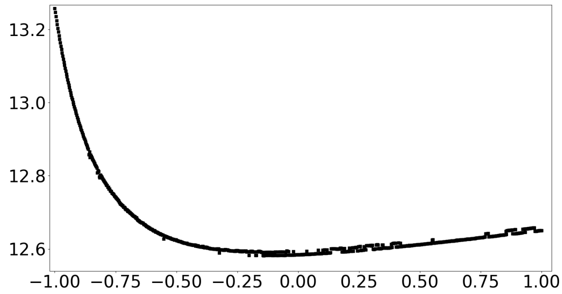
$$S_L = \mu\mu_0\,. \tag{3.31}$$

Jeho hodnota se typicky pohybuje kolem $c \approx 0{,}1$, ovšem může nabývat i jiných hodnot pro různé povrchy.

První výpočet byl proveden pro 200 bodů v rozsahu $c \in (0{,}1)$, v případě že by se jednalo o nějaký nestandardní případ. Výpočet byl proveden tak, že ostatním parametrům byla dána pevná hodnota, aby výpočet probíhal rychleji. Jeho výsledky jsou vyneseny na grafu 3.29.

Lze vidět, že pro $c \in (0{,}05;\ 0{,}3)$ jsou odchylky malé a skoro stejně velké, zatímco pro $c > 0{,}3$ jsou výrazně větší a chaotické. To proto, že už neodpovídají fyzikální realitě situace.

Následně byl proveden výpočet pro 600 bodů v rozsahu $c \in (0;\ 0{,}3)$, tentokrát s volným fitováním pólů. Výsledky jsou na grafu 3.31.



Obrázek 3.33: Výpočet RMS pro různé hodnoty rozptylového parametru $a \in (-1;\ 1)$.

Dále jsem se zabýval parametrem a, z rovnice 3.30. Jedná se o trochu problematický parametr, stejně jako ostatní parametry fázové funkce, protože se tedy vztahuje na změnu fáze v průběhu jedné světelné křivky. Těmito parametry by se nemělo fitovat, protože u relativních dat to často vede na nesprávné řešení.

3.11 Srovnání tvarů: sféričnost, moment setrvačnosti, délky hlavních os

Jak lze o dvojici tvarů říct, že jsou si podobné nebo nepodobné? Jak lze složitému tvaru přiřadit malou sadu čísel, která jej popíše?

Rád bych zdůraznil, že pokud vycházíme pouze z relativní fotometrie, nejsme schopni určovat velikost modelu, pouze jeho tvar.

Tvar tělesa můžeme brát jako výstup z procedury `standardtri`, která převádí síť mnohoúhelníků na síť trojúhelníků, případně už převedený .obj formát. Jedná se o formát, kde se první pomocí souřadnic (vektoru) definují polohy vrcholů (na řádcích označených "v") a následně se definují stěny zápisem `v1 v2 v3` kde `f` značí „face“, neboli stěnu a dále jsou uvedené čísla vrcholů, které stěnu tvoří.

Jedna trojúhelníková stěna je tedy jednoznačně popsána maticí 3 krát 3.

Implementaci níže popsaného postupu, který z části vychází z (cite!) jsem provedl v Pythonu.

3.11.1 Výpočet povrchu tělesa

Mějme trojúhelníkovou stěnu (není-li uvedeno jinak, všechny stěny, se kterými budeme zacházet budou trojúhelníkové) danou polohovými vektory vrcholů $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{s}$. Vektory mají počátek v bodě (0,0,0), který není v obecném případě pro těleso nijak významný.

Zaveďme si vektory $\boldsymbol{a}$, $\boldsymbol{b}$, které popisují dvě hrany trojúhelníku:

$$\boldsymbol{a} = \boldsymbol{q} - \boldsymbol{r} \tag{3.32}$$

$$\boldsymbol{b} = \boldsymbol{s} - \boldsymbol{r}\,, \tag{3.33}$$

pak určíme plochu rovnoběžníku určeného těmito vektory jako

$$2s = ||\boldsymbol{a} \times \boldsymbol{b}|| \tag{3.34}$$

a plochu stěny tedy jako

$$s = \frac{||\boldsymbol{a} \times \boldsymbol{b}||}{2}\,. \tag{3.35}$$

Pokud si označíme počet stěn F, tak celková plocha tělesa je

$$S = \sum_{i=1}^{F} s_i\,. \tag{3.36}$$

Zajímavé je, že průměrná plocha průmětu do náhodného směru je rovna povrchu promítaného tělesa. Toto lze intuitivně vidět např. na kouli.

Plochu nelze počítat jako

$$2s = \det(\boldsymbol{a}, \boldsymbol{b}), \tag{3.37}$$

protože $(\boldsymbol{a}, \boldsymbol{b})$ není matice $n \times n$.

Samotná plocha tělesa není užitečná, protože není kalibrovaná o velikost tělesa. Můžeme ji ale uvádět do poměru například s objemem.

3.11.2 Výpočet objemu tělesa

Zaveďme kromě vektorů $\boldsymbol{a}$, $\boldsymbol{b}$, jenž byly definovány v předchozí části, vektor

$$\boldsymbol{c} = -\boldsymbol{r}. \tag{3.38}$$

Pak smíšený součin vektorů $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ určuje objem rovnoběžnostěnu V_p vytyčeného těmito vektory.

$$V_p = (a \times b) \cdot c = \det(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}). \tag{3.39}$$

Vztah s použitím determinantu matice $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ je obecný a platí v libovolném n-rozměrném prostoru pro n vektorů. Objem čtyřstěnu vytyčeného mezi $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{s}$ je roven objemu mezi vektory $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$: jedná se o totéž místo v prostoru. Objem tohoto čtyřstěnu je 1/6 z objemu řečeného rovnoběžnostěnu.

$$V_i = \frac{1}{6} V_p = \frac{1}{6}(a \times b) \cdot c. \tag{3.40}$$

Celkový objem tělesa je pak

$$V = \frac{1}{6} \sum_{i=1}^{F} V_i. \tag{3.41}$$

Vzhledem k tomu, že neznáme hustotu tělesa, můžeme brát

$$\varrho = 1 \tag{3.42}$$

a tedy

$$\{M\} = \{V\}. \tag{3.43}$$

Složenými závorkami zde označuji číselnou hodnotu.

Také bereme, že těleso je homogenní. Zdůrazňuji to zde proto, že budu zacházet s pojmy jako těžiště a moment setrvačnosti, které jsou v běžném kontextu vztahovány k hmotnosti. My však nyní nezacházíme s fyzickým asteroidem, ale s geometrickým modelem, který nemá velikost ani hustotu, a proto tyto pojmy můžeme brát jako ekvivalentní.

3.11.3 Sféričnost

Sféričnost je veličina používaná k popisu toho, jak je daný tvar kulatý. Existuje řada způsobů, jak takovou veličinu zavést, běžně se však používá této definice:

$$\text{Sph.} = \frac{\sqrt[3]{\pi (6V)^2}}{S}, \tag{3.44}$$

kde V je objem a S je povrch tělesa. Sféričnost koule je z definice 1 a pro každé jiné těleso nabývá

$$\text{Sph.} \in (0{,}1). \tag{3.45}$$

Jedná se o bezrozměrnou veličinu, která je pro naše účely užitečná, protože je nezávislá na velikosti tvaru.

Z praktického hlediska jsem zavedl tuto veličinu také z důvodu debuggingu programu, protože hodnota sféričnosti je jednoduše dohledatelná pro celou řadu těles, které jsem používal pro testování (např. Platónská tělesa a jiná).

3.11.4 Poloha těžiště

Máme-li homogenní čtyřstěn určený polohovými vektory jeho vrcholů $\boldsymbol{q}$, $\boldsymbol{r}$, $\boldsymbol{s}$, $\boldsymbol{t}$, pak jeho těžiště leží v bodě určeném vektorem

$$\boldsymbol{R_i} = \frac{1}{4}(\boldsymbol{q} + \boldsymbol{r} + \boldsymbol{s} + \boldsymbol{t})\,. \tag{3.46}$$

Našim vektorem $\boldsymbol{t}$ je však počátek souřadnic, tedy

$$\boldsymbol{R_i} = \frac{1}{4}(\boldsymbol{q} + \boldsymbol{r} + \boldsymbol{s})\,. \tag{3.47}$$

Z definice těžiště jako váženého průměru polohových vektorů jednotlivých dílčích těžišť:

$$\boldsymbol{R} = \frac{1}{M}\sum_{i=1}^{F} M_i \boldsymbol{R_i} = \frac{1}{V}\sum_{i=1}^{F} V_i \boldsymbol{R_i}\,. \tag{3.48}$$

Vzhledem k této poloze budeme počítat moment setrvačnosti.

3.11.5 Tenzor momentu setrvačnosti

Budeme zacházet s momentem setrvačnosti v tenzorové podobě, pro který platí

$$\boldsymbol{L} = I\boldsymbol{\omega}, \tag{3.49}$$

tedy

$$\begin{pmatrix} L_x \\ L_y \\ L_z \end{pmatrix} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.50}$$

Zde jsem při programování postupoval čistě podle rovnic v článku [3] a uvádím zde pouze použitý výsledek, protože jeho odvození není cílem této práce.

Zavádí zde tenzor P_{jk} pro který vůči tenzoru I_{jk} platí

$$I_{xx} = P_{yy} + P_{zz} \tag{3.51}$$

$$I_{yy} = P_{xx} + P_{zz} \tag{3.52}$$

$$I_{zz} = P_{yy} + P_{xx} \tag{3.53}$$

a

$$I_{yz} = -P_{yz} \tag{3.54}$$

$$I_{xz} = -P_{xz} \tag{3.55}$$

$$I_{xy} = -P_{xy}\,. \tag{3.56}$$

Má užitečnou vlastnost, že jej lze spočítat pro každý čtyřstěn zvlášť a výsledný P_{ij} získat jako součet těchto dílčích tenzorů.

3.11.6 Převedení tenzoru do barycentrické soustavy

Tenzor I_{ij} je v tuto chvíli vztažen k počátku soustavy souřadnic, kterou prochází osy x, y, z. Vztáhneme jej k vypočítané poloze těžiště pomocí Steinerovy věty.

$$I^b_{ij} = I_{ij} - MT\,, \tag{3.57}$$

kde T je matice

$$T = \begin{pmatrix} Y^2 + Z^2 & -XY & -XZ \\ -XY & X^2 + Z^2 & -YZ \\ -XZ & -YZ & X^2 + Y^2\,, \end{pmatrix} \tag{3.58}$$

kde X,Y,Z jsou souřadnice těžiště, které jsme předchozích krocích určili (označovali jsme za $\boldsymbol{R}$).

Toto jsem implementoval pomocí Kroneckerovy Delty, v této podobě:

$$I^b_{ij} = I_{ij} + M(|\boldsymbol{R^2}|\delta_{ij} - R_i R_j)\,, \tag{3.59}$$

kde

$$\delta_{ij} = \begin{cases} 0 & \text{pokud } i \neq j, \\ 1 & \text{pokud } i = j. \end{cases} \tag{3.60}$$

3.11.7 Učení délek os a momentů setrvačnosti kolem nich

Zde bylo postupováno čistě podle [3], implementace v Pythonu je k nalezení v příloze.

3.11.8 Srovnání s nekonvexním modelem a adaptivní optikou

metoda určení	a/b
nekonvexní	1,221
konvexní	1,181
adaptivní optika	1,214

Tabulka 3.2: Srovnání poměru délek hlavních os dynamicky ekvivalentního elipsoidu v případě inverzních modelů (konvexního a nekonvexního) a výsledků z Keckova dalekohledu, využívajícího adaptivní optiku.

model	I_a	I_b	I_c
konvexní	1	0,985	0,821
nekonvexní	1	0,902	0,705

Tabulka 3.3: Srovnání momentů setrvačnosti dynamicky ekvivalentního elipsoidu kolem hlavních os.

3.12 Hrubý odhad velikosti

Existují přibližné vztahy mezi absolutní magnitudou a průměrem asteroidu.

$$D \approx \frac{U}{\sqrt{p}} 10^{-0,2H} (\mathrm{km})\,, \tag{3.61}$$

kde H je absolutní magnituda, kterou lze vyjádřit jako

$$H = \bar{m} - 5 \log \frac{\Delta_S \Delta_E}{1\mathrm{au}^2} + 2{,}5 \log q(\alpha)\,, \tag{3.62}$$

kde d_S, d_E označují vzdálenosti od Slunce a Země, q je při pozorování kolem opozice přibližně rovno $2/3$ a $\bar{m}$ určíme z pozorování jako

$$\bar{m} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} m \; dt \approx \frac{1}{t_2 - t_1} \sum_{t_1}^{t_2} m \; dt\,. \tag{3.63}$$

Faktor U v rovnici (3.61) se určí jako

$$U = 2 \text{ au} \cdot 10^{0,2(H_\odot)} \doteq 1329\,\mathrm{km}\,, \tag{3.64}$$

kde $H_\odot = -26{,}7$ mag je vizuální magnituda Slunce (odpovídá absolutní magnitudě H).

Problém ale je, že tyto magnitudy jsou pouze instrumentální (tedy, vůči nějaké konkrétní hvězdě v zorném poli dalekohledu v konkrétním filtru). Převedení instrumentálních veličin na absolutní hodnoty vizuální magnitudy je možná nejproblematičtějším úkolem běžné pozorovatelské činnosti.

Já jsem použil data z robotizovaných přehlídek (z online databáze SIMBAD), ty jsou ale v současné době problematické. Odchylky mých relativních měření jsou přibližně desetkrát přesnější, než přehlídkových měření.

Při velmi seriózních projektech se používá Landoltových polí, což jsou pečlivě změřená pole na obloze, kde se znají absolutně vizuální magnitudy. Nejlepší pozorovací noci tedy padnou na vzájemné navazování vlastních měření.

Po dosazení do výše uvedených vztahů, vychází průměr

$$D \approx 300\,\mathrm{km}\,,$$

což dobře odpovídá výsledkům jiných publikací.

Kapitola 4

Závěr

4.1 Výsledky specifické pro asteroid (511) Davida

Z vlastních a nových fotometrických pozorování byla zpřesněna perioda z

$$P = 5{,}129364 \pm 0{,}000001 \text{ h} \tag{4.1}$$

na

$$P = 5{,}1293644 \pm 0{,}0000005 \text{ h}\,, \tag{4.2}$$

což je ovšem dáno tím, že jsme planetku pozorovali naposled a měli jsme tudíž nejdelší pozorovací oblouk.[1]

Dále byla určeny a ověřeny ekliptikální souřadnice rotačního pólu,

$$\lambda = (299{,}5 \pm 3{,}5)^\circ$$
$$\beta = (24 \pm 2{,}5)^\circ$$

a k těmto číslům byly zřejmě poprvé přiřazeny směrodatné odchylky.

4.2 Obecné výsledky pro inverzní metodu

V práci byl vyvinut a popsán způsob charakterizace sady pozorování a její vhodnosti pro použití při inverzním výpočtu tvaru. Tato veličina je v práci označována jako Ψ.

Bylo zjištěno, že tato veličina je zásadní k jednoznačnému určení polohy pólů. Minimální hodnota byla určena přibližně jako

$$\Psi = 0{,}5 \pm 0{,}1\,. \tag{4.3}$$

Je nutné dbát zvýšené opatrnosti, je-li Ψ pozorovací řady v tomto rozsahu.

Také bylo zjištěno, že nad určitou mezí určenou přibližně

$$0{,}8 < \Psi < 0{,}9 \tag{4.4}$$

již pokrytí různých geometrií není zásadním faktorem ovlivňujícím kvalitu modelu. Tímto zásadním faktorem se stává celkový pozorovací čas, který odpovídá počtu pozorovaných bodů a v zásadě také počtu pořízených světelných křivek n. Obecně tedy můžeme posuzovat kvalitu pozorování součinem $n\Psi$, kdy nad

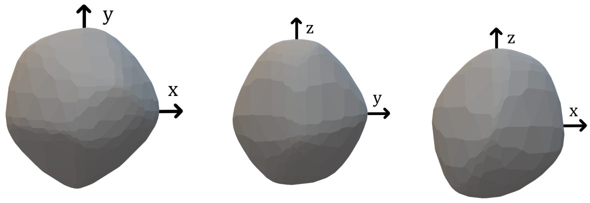
$$n\Psi > 40 \tag{4.5}$$

můžeme tvrdit, že sada pozorování je poměrně reprezentativní a je možné z ní určovat tvar.

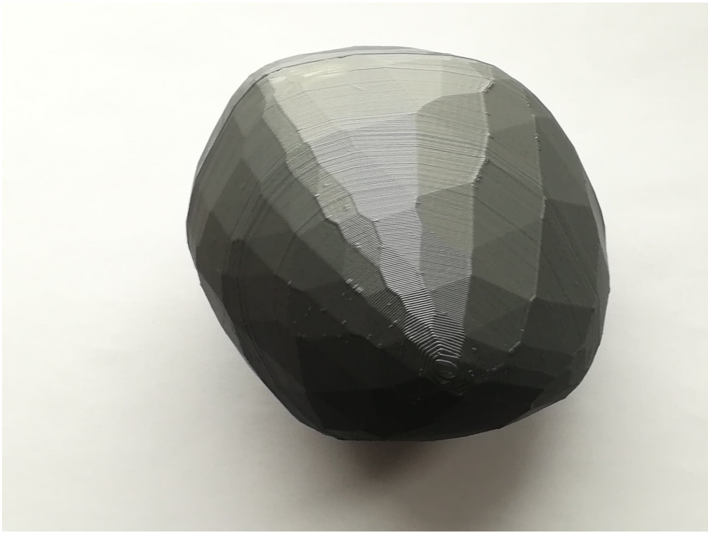
Rozložení geometrií pozorování není ale vždy dostačující podmínkou pro jednoznačné určení polohy pólů. Například planetky (23) Thalia a (24) Themis vykazují nejednoznačnost polohy pólu, i když mají hodnoty $\Psi \approx 1{,}0$.

Dalším přínosem této práce je metoda vyhodnocení heatmap pomocí histogramu. Z histogramů je velmi patrné, zda jsou polohy pólu lokalizovány spolehlivě. Když je mnoho pixelů kolem nejlepšího řešení, lze vidět, že nejlepší řešení není jednoznačně nejlepší.

[1] Může se to zdát jako nesmyslně přesné, ale není. Pro mnoho jiných těles je známa perioda i řádově přesněji.



Obrázek 4.1: Nejlepší konvexní model z dostupných světelných křivek pro planetku (511) Davida.



Obrázek 4.2: Model byl vytištěn na 3D tiskárně.

4.3 Zobrazení modelu

Přestože cílem této práce není přímo něco zjistit o asteroidu (511) Davida, ale spíše poukázat na hranice možnosti využití inverzní metody, je jistě zajímavé se na vypočítaný model podívat.

Seznam použitých programů

- Astrophotography tool
- Stellarium
- Dimension 4
- Muniwin
- periodsearch
- convexinv
- minkowski
- standardtri
- Astrometrica
- Tangra
- a vlastní programy v Pythonu, v příloze.

Seznam použitých knihoven v Pythonu

- math
- sys, os, glob
- numpy, pandas
- matplotlib, Seaborn
- rebound, pyephem
- pickle

4.4 Použitá archivní pozorování

N	Datum	publikace
1	1952-01-26.25	Groeneveld & Kuiper (1954)
2	1953-04-08.29	Groeneveld & Kuiper (1954)
3	1958-01-26.35	Gehrels & Owings (1962)
4	1962-12-05.64	Chang & Chang (1963)
5	1968-12-29.33	Vesely & Taylor (1985)
6	1968-12-30.39	Vesely & Taylor (1985)
7	1970-03-21.37	Vesely & Taylor (1985)
8	1972-08-07.49	Vesely & Taylor (1985)
9	1981-04-15.20	Weidenschilling et al. (1987)
10	1981-06-16.18	Weidenschilling et al. (1987)
11	1982-01-09.29	Weidenschilling et al. (1987)
12	1982-01-13.29	Weidenschilling et al. (1987)
13	1982-02-17.22	Weidenschilling et al. (1987)
14	1982-02-18.37	Weidenschilling et al. (1987)
15	1982-02-19.26	Weidenschilling et al. (1987)
16	1982-02-20.28	Weidenschilling et al. (1987)
17	1982-05-20.24	Weidenschilling et al. (1987)
18	1982-07-14.27	Weidenschilling et al. (1987)
19	1982-08-07.18	Weidenschilling et al. (1987)
20	1983-05-21.18	Weidenschilling et al. (1987)
21	1983-05-23.22	Weidenschilling et al. (1987)
22	1983-07-02.32	Weidenschilling et al. (1987)
23	1983-09-14.36	Weidenschilling et al. (1987)
24	1983-09-19.35	Weidenschilling et al. (1987)
25	1983-10-12.30	Weidenschilling et al. (1987)
26	1983-10-16.31	Weidenschilling et al. (1987)
27	1984-06-08.22	Weidenschilling et al. (1987)
28	1984-06-10.30	Weidenschilling et al. (1987)
29	1984-09-23.23	di Martino et al. (1987)
30	1984-09-24.24	di Martino et al. (1987)
31	1984-09-27.21	di Martino et al. (1987)
32	1984-10-18.97	Lagerkvist et al. (1995)
33	1984-10-20.98	Lagerkvist et al. (1995)
34	1984-10-26.92	Lagerkvist et al. (1995)
35	1985-10-22.24	Weidenschilling et al. (1987)
36	1985-10-23.22	Weidenschilling et al. (1987)
37	1985-10-25.25	Weidenschilling et al. (1987)
38	1986-01-20.19	Weidenschilling et al. (1987)
39	1987-04-24.88	Shevchenko et al. (1992)
40	2005-06-28.93	
41	2005-06-29.94	
42	2010-05-28.96	
43	2010-06-03.02	
44	2010-06-03.94	
45	2010-06-04.98	
46	2010-06-08.27	
47	2010-06-12.22	
48	2010-06-13.25	
49	2010-06-21.95	
50	2010-06-24.95	
51	2015-04-10.03	
52	2015-04-11.02	
53	2015-04-11.95	
54	2015-04-18.04	
55	2015-05-11.06	
56	2015-05-11.99	
57	2015-05-17.94	
58	2015-05-17.92	

Tabulka 4.1

Literatura

[1] BROŽ, Miroslav a Marek WOLF. Astronomická měření. Vyd. 3., přeprac. a dopl. Praha: MatfyzPress, 2017. ISBN 978-0-387-36786-6.

[2] DRUMMOND, J.D. a E.K. HEGE. Speckle interferometry of asteroids. Icarus [online]. 1986, 67(2), 251-263 [cit. 2020-02-27]. DOI: 10.1016/0019-1035(86)90107-7. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/0019103586901077

[3] DOBROVOLSKIS, Anthony R. Inertia of Any Polyhedron. Icarus [online]. 1996, 124(2), 698-704 [cit. 2020-02-27]. DOI: 10.1006/icar.1996.0243. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0019103596902432

[4] ĎURECH, J., M. KAASALAINEN, B. D. WARNER, et al. Asteroid models from combined sparse and dense photometric data. Astronomy & Astrophysics [online]. 2009, 493(1), 291-297 [cit. 2020-02-29]. DOI: 10.1051/0004-6361:200810393. ISSN 0004-6361. Dostupné z: http://www.aanda.org/10.1051/0004-6361:200810393

[5] ĎURECH, J., V. SIDORIN a M. KAASALAINEN. DAMIT: a database of asteroid models. Astronomy and Astrophysics [online]. 2010, 513 [cit. 2020-03-07]. DOI: 10.1051/0004-6361/200912693. ISSN 0004-6361. Dostupné z: http://www.aanda.org/10.1051/0004-6361/200912693

[6] HANUŠ, J., J. ĎURECH, M. BROŽ, et al. A study of asteroid pole-latitude distribution based on an extended set of shape models derived by the lightcurve inversion method. Astronomy & Astrophysics [online]. 2011, 530 [cit. 2020-03-10]. DOI: 10.1051/0004-6361/201116738. ISSN 0004-6361. Dostupné z: http://www.aanda.org/10.1051/0004-6361/201116738

[7] GROENEVELD, Ingrid a Gerard P. KUIPER. Photometric Studies of Asteroids. I. The Astrophysical Journal [online]. 1954, 120, 1-18 [cit. 2020-02-27]. DOI: 10.1086/145904. ISSN 0004-637X. Dostupné z: http://adsabs.harvard.edu/doi/10.1086/145904

[8] KAASALAINEN, Mikko a TORPPA, Johanna. Optimization Methods for Asteroid Lightcurve Inversion I. Shape Determination. Icarus [online]. 2001, 153(1), 24-36 [cit. 2020-03-07]. DOI: 10.1006/icar.2001.6673. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0019103501966734

[9] KAASALAINEN, M. Optimization Methods for Asteroid Lightcurve Inversion II. The Complete Inverse Problem. Icarus [online]. 2001, 153(1), 37-51 [cit. 2020-03-07]. DOI: 10.1006/icar.2001.6674. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0019103501966746

[10] MARCHIS, F, M KAASALAINEN, E HOM, J BERTHIER, J ENRIQUEZ, D HESTROFFER, D LEMIGNANT a I DEPATER. Shape, size and multiplicity of main-belt asteroidsI. Keck Adaptive Optics survey. Icarus [online]. 2006, 185(1), 39-63 [cit. 2020-02-29]. DOI: 10.1016/j.icarus.2006.06.001. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0019103506001990

[11] TORPPA, Johanna, Mikko KAASALAINEN, Tadeusz MICHAŁOWSKI, Tomasz KWIATKOWSKI, Agnieszka KRYSZCZYŃSKA, Peter DENCHEV a Richard KOWALSKI. Shapes and rotational properties of thirty asteroids from photometric data. Icarus [online]. 2003, 164(2),

346-383 [cit. 2020-03-04]. DOI: 10.1016/S0019-1035(03)00146-5. ISSN 00191035. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0019103503001465

[12] VOKROUHLICKÝ, David, Petr PRAVEC, Josef ĎURECH, et al. Detailed Analysis of the Asteroid Pair (6070) Rheinland and (54827) 2001 NQ8. The Astronomical Journal [online]. 2017, 153(6) [cit. 2020-03-10]. DOI: 10.3847/1538-3881/aa72ea. ISSN 1538-3881. Dostupné z: http://stacks.iop.org/1538-3881/153/i=6/a=270?key=crossref.adc41cca09849ff3079385f2d9aa262d

Příloha: vlastní programy

Pro tuto práci byla napsána řada programů. Začlenění těchto programů do samotné práce by ji ale narušovalo, a proto byly umístěny zde. Stěžejní programy jsou komentované jak ve zdrojovém kódu, tak v samotné práci. Pro časovou náročnost nebyly komentovány všechny programy, ale jsou zde uvedeny, aby byla usnadněna replikovatelnost výsledků.

```python
import math
import numpy as np
import sys

objFile = sys.argv[1]
outFile = sys.argv[2]
parsingOption = len(sys.argv)

f = open(objFile, "r")
lines = f.readlines()

#Data Parsing

vertices = []
for line in lines:
	if line.strip().split()[0] == "v":
		elem = np.array([float(line.strip().split()[1]), float(line.strip().split()[2]), float(line.strip().split()[3])])
		vertices.append(elem)

faces = []
for lineF in lines:
	if lineF.strip().split()[0] == "f":
		elemF =[int(lineF.strip().split()[1])-1, int(lineF.strip().split()[2])-1, int(lineF.strip().split()[3])-1]
		faces.append(elemF)

#Adding tetrahedral volume elements, volume calculated by the vector formula V =1/6((a cross b) dot c)
#a, b vectors connect the points in the facet with vertex R, vector c points from R to 0,0,0


centroids = []
volumes = []

TotalVolume = 0
TotalArea = 0
for face in faces:
	Q, R, S = face[0], face[1], face[2]
	a = np.subtract(vertices[Q], vertices[R])
	b = np.subtract(vertices[S], vertices[R])
	c = np.subtract([0,0,0], vertices[R])
	acrossb = np.cross(a, b)
	vol = abs(1/6*(np.dot(acrossb, c)))
	surf = abs(0.5*np.linalg.norm(acrossb))
	elementCentroid = np.sum([vertices[Q], vertices[R], vertices[S]], axis=0)
	centroids.append(elementCentroid)
	volumes.append(vol)
	TotalVolume += vol
	TotalArea += surf

print("total volume / mass")
print(TotalVolume)

print("total area")
print(TotalArea)

sphericity = (math.pi**(1/3)*(6*TotalVolume)**(2/3))/(TotalArea)
print("sphericity ")
print(sphericity)

#Global Centroid calculation -- centroid elements weighted by volume elements, summed.

centroidsNP = np.array(centroids)
volumesNP = np.array(volumes)

```

```python
weightedCentroidElements = []

for i in range(len(centroidsNP)):
        wCentroidElement = (1/TotalVolume)*volumesNP[i]*centroidsNP[i]
        weightedCentroidElements.append(wCentroidElement)

weightedCentroidElementsNP = np.array(weightedCentroidElements)
Centroid = np.sum(weightedCentroidElementsNP, axis=0)

#Now, the calculation of the moment of inertia product around the x, y, z axes

Ps = []
for l in range(len(faces)):
        IndD, IndE, IndF = faces[l][0], faces[l][1], faces[l][2]
        D, E, F = vertices[IndD], vertices[IndE], vertices[IndF]

        #the product of inertia
        DeltaP = [[0,0,0], [0,0,0], [0,0,0]]

        for j in range(3):
                for k in range(3):
                        DeltaP[j][k] = (volumes[l]/20)*(2*D[j]*D[k] + 2*E[j]*E[k] + 2*F[j]*F[k] + D[j]*E[k] + D[k]*E[j] + D[k]*F[j] + D[j]*F[k] + E[j]*F[k] + E[k]*F[j])
        Ps.append(DeltaP)

PsNP = np.array(Ps)

P = np.sum(PsNP, axis=0)

#the calculation of the proper Inertia tensor
I = [[0,0,0], [0,0,0], [0,0,0]]

I[0][0] = P[1][1] + P[2][2]
I[1][1] = P[2][2] + P[0][0]
I[2][2] = P[1][1] + P[0][0]

I[1][2] = -P[1][2]
I[0][2] = -P[0][2]
I[0][1] = -P[0][1]

I[2][1] = -P[1][2]
I[2][0] = -P[0][2]
I[1][0] = -P[0][1]


#Application of the parallel axis theorem (Steiner's theorem) to transform I -> Iprime in the barycentric coordinate frame

X, Y, Z = Centroid[0], Centroid[1], Centroid[2]
R = math.sqrt(X**2 + Y**2 + Z**2)

#Kronecker delta function
def KronDel(a, b):
        if a == b:
                return 1
        else:
                return 0

TransformationMatrix = [[0,0,0], [0,0,0], [0,0,0]]

#this is the parallel axis theorem in tensor form, using Kronecker's delta
for jA in range(3):
        for kA in range(3):
                TransformationMatrix[jA][kA] = R**2*KronDel(jA, kA) - Centroid[jA]*Centroid[kA]


TransformationMatrixNP = np.array(TransformationMatrix)
```

```
Iprime = I - TotalVolume*TransformationMatrixNP


#Defining helpful constants T, CapitalPi, U, Theta

T = Iprime[0][0] + Iprime[1][1] + Iprime[2][2]
CapitalPi = Iprime[0][0]*Iprime[1][1] + Iprime[0][0]*Iprime[2][2] + Iprime[1][1]*Iprime[2][2] - Iprime[0][1]**2 - Iprime[0][2]**2 - Iprime[1][2]**2

U = math.sqrt(T**2 - 3*CapitalPi)/3


IprimeDet = np.linalg.det(Iprime)

Theta = np.arccos((-2*T**3 + 9*T*CapitalPi - 27*IprimeDet)/(54*U**3))

CoefA = T/3 - 2*U*np.cos(Theta/3)
CoefB = T/3 - 2*U*np.cos(Theta/3 - 2/3*math.pi)
CoefC = T/3 - 2*U*np.cos(Theta/3 + 2/3*math.pi)

#solving for the principal axes!

principal_a = math.sqrt((5*(CoefB + CoefC - CoefA))/(2*TotalVolume))
principal_b = math.sqrt((5*(CoefA + CoefC - CoefB))/(2*TotalVolume))
principal_c = math.sqrt((5*(CoefA + CoefB - CoefC))/(2*TotalVolume))

print("pricipal axis a " + str(principal_a))
print("pricipal axis b " + str(principal_b))
print("pricipal axis c " + str(principal_c))


print("moment of inertia tensor in barycentric frame")
print(Iprime)

"""
if parsingOption == 3:
        #output parsing
        outputFile = open(outFile, "w+")
        outputFile.write("Volume \t" + str(TotalVolume) + "\n")
        outputFile.write("Surface area \t" + str(TotalArea) + "\n" )
        outputFile.write("Sphericity \t" + str(sphericity) + "\n")

        outputFile.write("Principal axis a \t" + str(principal_a) + "\n")
        outputFile.write("Principal axis b \t" + str(principal_b) + "\n")
        outputFile.write("Principal axis c \t" + str(principal_c) + "\n")

        outputFile.write("moment of inertia around the principal axis a \t" + str(CoefA) + "\n")
        outputFile.write("moment of inertia around the principal axis b \t" + str(CoefB) + "\n")
        outputFile.write("moment of inertia around the principal axis c \t" + str(CoefC) + "\n")


        pa_rat = principal_a/principal_c
        pb_rat = principal_b/principal_c
        pc_rat = principal_c/principal_c

        CoefA_rat = CoefA/CoefC
        CoefB_rat = CoefB/CoefC
        CoefC_rat = CoefC/CoefC

        outputFile.write("Principal axis ratio a : b : c \t" + str(pa_rat) + " : " +  str(pb_rat) + " : "  + str(pc_rat) + "\n")
        outputFile.write("Moment of inerta around principal axis ratio a : b : c \t" + str(CoefA_rat) + " : " +  str(CoefB_rat) + " : " + str(CoefC_rat) + "\n")

```

```
        outputFile.close()

if parsingOption == 4:
        #output parsing NAKED
        outputFile = open(outFile, "w+")
        outputFile.write(str(TotalVolume)+ "\n")
        outputFile.write(str(TotalArea) + "\n")
        outputFile.write(str(sphericity) + "\n")
        outputFile.write(str(principal_a) + " " + str(principal_b) + " " +str(principal_c) +"\n")
        outputFile.write(str(CoefA) + " " + str(CoefB) + " " +  str(CoefC) + "\n")
        pa_rat = principal_a/principal_c
        pb_rat = principal_b/principal_c
        pc_rat = principal_c/principal_c
        CoefA_rat = CoefA/CoefC
        CoefB_rat = CoefB/CoefC
        CoefC_rat = CoefC/CoefC
        outputFile.write(str(pa_rat) + " : " +  str(pb_rat) + " : "  + str(pc_rat) + "\n")
        outputFile.write(str(CoefA_rat) + " : " +  str(CoefB_rat) + " : " + str(CoefC_rat) + "\n")
        outputFile.close()
"""

if sys.argv[3] == "S":
        #output parsing
        outputFile = open(outFile, "a")
        outputFile.write(sys.argv[1] + "\t" + str(sphericity) + "\n")
        outputFile.close()


if sys.argv[3] == "IAB":
        #output parsing
        outputFile = open(outFile, "a")
        CoefA_rat = CoefA/CoefC
        CoefB_rat = CoefB/CoefC
        outputFile.write(sys.argv[1] + "\t" + str(CoefA_rat) + "\t" + str(CoefB_rat) + "\n")
        outputFile.close()


if sys.argv[3] == "AB":
        #output parsing
        outputFile = open(outFile, "a")
        pa_rat = principal_a/principal_c
        pb_rat = principal_b/principal_c
        outputFile.write(sys.argv[1] + "\t" + str(pa_rat) + "\t" + str(pb_rat) + "\n")
        outputFile.close()
```

```python
import matplotlib.pyplot as plt
import os
import glob
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import seaborn as sns
import numpy as np
import pickle
import math

def CalcXi(measured, calculated):
        oFil = open(measured, "r")
        lines = oFil.readlines()

        OLCname = calculated
        OLCFil = open(OLCname, "r")
        OLClines = OLCFil.readlines()

        newlines = []

        for line in lines:
                if len(line.split()) == 8:
                        newlines.append(line)
        N = len(newlines)
        DM = 0
        for i in range(N):
                O = float(newlines[i].split()[1])
                E = float(OLClines[i])
                dM = abs(2.5*math.log((O/E), 10))
                DM += dM**2
        #RMS in magnitudes
        #print("N:  " + str(N) + " Average dM: " + str(((1/N) * DM)**0.5))
        return 1000*math.sqrt((DM)/(N)) ###1000x means MILI magnitude!!!

Betas = []
Lambdas = []
Xis = []

folder ="MajellaHM_OLC" ##############
for fileA in os.listdir(folder):
        Xis.append(CalcXi("RangePsi/Majella.txt", folder + "/" +  fileA)) #############
        name = fileA.strip("OLC").strip(".txt")
        Betas.append(int(name.split("_")[0]))
        Lambdas.append(int(name.split("_")[1]))

print(Betas)
print(Lambdas)
print(Xis)

Nw = len(Betas)
data = []
for rt in range(Nw):
        data.append([Betas[rt], Lambdas[rt], Xis[rt]])

print(data)


#################################### CHANGE PICKLE FILE!
with open("MajellaPickle", 'wb') as fp:
    pickle.dump(data, fp)



"""
ax = plt.axes(projection='3d')
ax.scatter3D(Lambdas, Betas, Xis)
plt.show()
"""
```

```
x = np.array(Lambdas)
y = np.array(Betas)
z = np.array(Xis)

df = pd.DataFrame.from_dict(np.array([x,y,z]).T)
df.columns = ['X_value','Y_value','Z_value']

pivotted= df.pivot('Y_value','X_value','Z_value')
sns.heatmap(pivotted,cmap='Greys_r')

plt.show()
```

```python
import math
import numpy as np
import matplotlib.pyplot as plt
import random
import sys
from statistics import stdev
from mpl_toolkits.mplot3d import Axes3D


filename = sys.argv[1]

#parsing of data to an array
infile = open(filename, "r")
points = infile.readlines()
RAW = []
for point in points:
        stringpoint = point.split()
        Newpoint = [float(stringvalue) for stringvalue in stringpoint]
        if len(Newpoint) == 8:
                RAW.append(Newpoint)
        else:
                print("Parse pass")
                pass



#line corresponds to Newpoints[k] for example -- array of 8 floats
#PAB = phase angle bisector
# returns Numpy array

def PAB(line):
        R_SUN = [line[2], line[3], line[4]]
        R_EARTH = [line[5], line[6], line[7]]
        sumPrep = [R_SUN, R_EARTH]
        #A for Array (normal python array), N for numpy array
        #s, e -- sun, earth
        PABseparateN = np.array(sumPrep)
        PAB = np.sum(PABseparateN, axis=0)
        #normalisation
        PABnormalised = PAB/(np.linalg.norm(PAB))
        return PABnormalised



#returns 3x3 numpy array
def IJK(A, B):
        Iq = A/np.linalg.norm(A)
        Kq = (np.cross(A, B))/(np.linalg.norm(np.cross(A, B)))
        Jq = (np.cross(A, Kq))/(np.linalg.norm(np.cross(A, Kq)))
        baseA = [Iq, Jq, Kq]
        baseN = np.array(baseA)
        return baseN



#transforming the set of N vectors to 3 sets of N numbers, for plotting
def PlotCon(PABnormSet):
        xs, ys, zs = [], [], []
        for PABnorm in PABnormSet:
                x = PABnorm[0]
                y = PABnorm[1]
                z = PABnorm[2]
                xs.append(x)
                ys.append(y)
                zs.append(z)
        PABnormSetOutput = [xs, ys, zs]
        return PABnormSetOutput
```

```
#data is PAB vector list, base is 3x3 numpy array of IJK vectors
#ciruclar residuals represent goodness of projection pairs. if Circres is too high, error
def Plane(dataset, base):
        CircRes = 0
        I = base[0]
        J = base[1]
        K = base[2]
        xs, ys = [], []
        pairs = []
        for vector in dataset:
                x = np.dot(vector, I)
                y = np.dot(vector, J)
                CR = abs(1 - (x**2 + y**2))
                CircRes += CR
                xs.append(x)
                ys.append(y)
                pairs.append([x,y])
        Lsorted = []
        #print(CircRes)
        for position in pairs:
                L = math.atan2(position[1],position[0])
                triplet = [position[0], position[1], L]
                Lsorted.append(triplet)
        if CircRes > 4:
                return None
        else:
                Lsorted.sort(key=lambda x:x[2])
                return Lsorted

#for well covered measurements, calculates area of polygon
def centerFullHull(coordinatesIN):
        coors = []
        S = 0
        for Ltrim in coordinatesIN:
                Ltrim[2] = 0
                coors.append(np.array(Ltrim))

        for i in range(len(coors)-1):
                dS = np.linalg.norm(np.cross(coors[i], coors[i+1]))
                S += dS/2
                #print(S, coors[i], coors[i+1])
        S += np.linalg.norm(np.cross(coors[-1], coors[0]))/2
        return S


#for small datasets, where maximum distance is < 180 deg
def centerEmptyHull(coordinatesIN):
        print("empty")
        coors = []
        S = 0
        for Ltrim in coordinatesIN:
                Ltrim[2] = 0
                coors.append(np.array(Ltrim))

        for i in range(len(coors)-1):
                dS = np.linalg.norm(np.cross(coors[i], coors[i+1]))
                S += dS/2
                #print("dS" + str(dS))
                #print(coors[i], coors[i+1])
        return S


#maximum separation in degrees using L values
def Lmax(triplets):
        maxL = 0
        for i in range(len(triplets)-2):
                deltaL = triplets[i+1][2]-triplets[i][2]
                if deltaL > maxL:
```

```
            maxL = deltaL
    deltaL = (math.pi - triplets[len(triplets)-1][2]) + math.pi +triplets[0][2]
    if deltaL > maxL:
        maxL = deltaL
    return maxL



def PSI(CIN):
    if Lmax(CIN) > math.pi:
        newS = centerEmptyHull(CIN)
    elif Lmax(CIN) <= math.pi:   ####### !!!!!!!!!!!
        newS = centerFullHull(CIN)
    return newS/math.pi


PABset = []
for testline in RAW:
    PABset.append(PAB(testline))


print("PSI")




##input == PABset!!!!!!, not Planar output
def PSIaverager(data,k):
    psis = []
    for j in range(k):
        Qs = random.sample(range(0, len(data)),2)
        Q0 = Qs[0]
        Q1 = Qs[1]
        try:
            BASE = IJK(PAB(RAW[Q0]), PAB(RAW[Q1]))
            Qtriplet = Plane(data,BASE)
            psiN = PSI(Qtriplet)
            print(psiN)
            psis.append(psiN)
        except:
            print("pass")
            pass

    print(psis)
    return [sum(psis)/len(psis), stdev(psis)]


LL = PSIaverager(PABset, 50)
print("PSI average: "  + str(LL[0]) + " stdev: " + str(LL[1]))


outF= open("psiOutput.txt", "a")
outF.write(filename + "\t" + str(LL[0]) + "\t" + str(LL[1]) + "\n")
outF.close()




#print(len(PABset))



Qs = random.sample(range(0, len(PABset)),2)
Q0 = Qs[0]
Q1 = Qs[1]
```

```
XXS = []
YYS = []
ZZS = []
BASEA = IJK(PAB(RAW[Q0]), PAB(RAW[Q1]))

#print(PABset)
for xxx in Plane(PABset, BASEA):
        XXS.append(xxx[0])
        YYS.append(xxx[1])
        ZZS.append(xxx[2])

#print(XXS)
###display planar chart based on convertToPlaneLs
an = np.linspace(0, 2*np.pi, 1000000)
plt.plot(np.cos(an), np.sin(an), "r-")
plt.fill(XXS, YYS)
plt.scatter(XXS, YYS, s=50, c="k", marker="x", zorder=2)
plt.axis('equal')
plt.savefig(filename + "-planar0.png")








"""
rset = PlotCon(PABset)

ax = plt.axes(projection='3d')
ax.scatter3D(rset[0], rset[1], rset[2], marker="x", c="k")
plt.savefig(filename + "-3d.png")
"""
```

```python
import sys

filename_in = sys.argv[1]
openFile = open(filename_in, "r")

file_out = sys.argv[2]

vf = openFile.readline().split()

NumOfV = int(vf[0])
NumOfF = int(vf[1])

allLines = openFile.readlines()


with open(file_out, "w") as f:
	for i in range(0,NumOfV):
		f.write("v {}".format(allLines[i]))
	for j in range(NumOfV, NumOfV+NumOfF):
		f.write("f {}".format(allLines[j]))
```

```python
import rebound
import math
import matplotlib
import matplotlib.pyplot as plt
import sys

from datetime import datetime


#speed of light in au/day
c = 173.144633

#UNITS. years, AU, and Solar masses => G=1

#INPUT PARSING

inputFile = sys.argv[1]
outputFile = sys.argv[2]

# CURRENT TIME

year = datetime.now().year
month = datetime.now().month
day = datetime.now().day
hour = datetime.now().hour
minute = datetime.now().minute
second = datetime.now().second

#determining current julian date
JDtoday = 367 * year - 7 * (year + (month + 9)/12)/4 - 3 * ((year + (month -
9)/7)/100 + 1)/4 + 275 * month/9 + day + hour/24 + minute/1440 + second/86400
+ 1721029


#CAUTION! MJD, JD ETC. HAS TO BE MODIFIED MANUALLY
modJDtoday = JDtoday


#particles added to the simulation

sim = rebound.Simulation()

sim.units = ("yr", "AU", "Msun")

sim.add("Sun")
sim.add("Earth")


#other planets can be added optionally

"""
#sim.add("Mercury")
#sim.add("Venus")
sim.add("Mars")
sim.add("Jupiter")
sim.add("Saturn")
sim.add("Uranus")
sim.add("Neptune")
"""


#add the selected asteroid to the N particle simulation
sim.add("Davida")

#iteration count
q = 0

#Dictionary for the paramenters, expected time, mag and mag error. Cartesian
coordinates calculated for the sun and earth in
```

```python
#the asteroids frame of reference -- cardinal points fixed at the N. ecliptical pole, vernal equinox and the "summer solstice point"

data = {'time': [], 'mag': [], 'mag_err': [], "sun_x": [], "sun_y": [], "sun_z": [],"earth_x": [], "earth_y": [], "earth_z": [], "asteroid_centric_JD": [], "earth_asteroid_distance": [], "Intensity": []}
# Load data
with open(inputFile, 'r') as file_in:
        for line in file_in:
                t, m, m_err = line.split()
                data['time'].append(float(t))
                data['mag'].append(float(m))
                data['mag_err'].append(float(m_err))
                dt = modJDtoday - float(t)
                # calculate untill time dt, where dt is the difference in days between now and the time given in inputFile
                sim.integrate(dt/365.25)
                #Necessary to asign correctly!!!
                E = sim.particles[1]
                S = sim.particles[0]
                A = sim.particles[2]
                data['sun_x'].append(S.x - A.x)
                data['sun_y'].append(S.y - A.y)
                data['sun_z'].append(S.z - A.z)
                data['earth_x'].append(E.x - A.x)
                data['earth_y'].append(E.y - A.y)
                data['earth_z'].append(E.z - A.z)
                #correction for the travel time of light (LTE) from the asteroid to Earth.
                Earth_Asteroid_distance = math.sqrt((E.x - A.x)**2 + (E.y - A.y)**2 + (E.z - A.z)**2)
                data["earth_asteroid_distance"].append(Earth_Asteroid_distance)
                asteroid_centric_correction = Earth_Asteroid_distance/c
                data["asteroid_centric_JD"].append(float(t) - asteroid_centric_correction)
                #magnitude to intensity units conversion from pogson equation
                Inten = 0.12*1.8*0.78*10**(-float(m)/2.5)/1.2
                data["Intensity"].append(Inten)
                q += 1


#output file parsing
#keep white space
outfile = open(outputFile, "w+")
for j in range(q):
        ascjd, inten, x1, y1, z1, x2, y2, z2  = data["asteroid_centric_JD"][j], data["Intensity"][j], data['sun_x'][j], data['sun_y'][j], data['sun_z'][j], data['earth_x'][j], data['earth_y'][j], data['earth_z'][j]
        outfile.write('{}  {}   {} {} {}   {} {} {}\n'.format(ascjd, inten, x1, y1, z1, x2, y2, z2))
outfile.close()

plt.scatter(data["time"], data["earth_asteroid_distance"], label="Earth-Asteroid distance")
plt.scatter(data["time"], data["mag"], label="Asteroid magnitude")
plt.scatter(data["time"], data["Intensity"], label="Asteroid intensity")
plt.savefig("ZbynekDavida.png")
plt.show()
```

```python
import matplotlib.pyplot as plt
import os
import glob
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import seaborn as sns
import numpy as np


Betas = []
Lambdas = []
Sphers = []
As = []
Bs = []
Mas = []


for fileA in os.listdir("outs"):
        opnFileA = open("outs/" + fileA, "r")
        objlines = opnFileA.readlines()
        spher = float(objlines[0])
        a = float(objlines[1])
        b = float(objlines[2])
        Ma = float(objlines[4])
        name = fileA.strip("OBJ").strip(".txt.obj.out")
        Betas.append(int(name.split("_")[0]))
        Lambdas.append(int(name.split("_")[1]))
        Sphers.append(spher)
        As.append(a)
        Bs.append(b)
        Mas.append(Ma)

print(Betas)
print(Lambdas)
print(Sphers)

Nw = len(Betas)
data = []
for rt in range(Nw):
        data.append([Betas[rt], Lambdas[rt], Mas[rt]])

print(data)



"""
ax = plt.axes(projection='3d')
ax.scatter3D(Lambdas, Betas, Xis)
plt.show()
"""

plt.ylabel("Beta")
plt.xlabel("Lambda")

x = np.array(Lambdas)
y = np.array(Betas)
z = np.array(Mas)

df = pd.DataFrame.from_dict(np.array([x,y,z]).T)
df.columns = ['X_value','Y_value','Z_value']

pivotted= df.pivot('Y_value','X_value','Z_value')
sns.heatmap(pivotted,cmap='RdBu')

plt.show()
```

```python
import os
import math
import sys
from shutil import copyfile


dirs = ["A", "B", "C", "D", "E", "F", "G", "H"]
updirs = ["a", "b", "c", "d", "e", "f", "g", "h"]

NAME = "RheinlandHM" #################### 1/3
P = str(4.27371) ###################### 2/3
t0 = str(2454601) ###################### 3/3

def makedirs(name, DIRS, UPDIRS):
        os.mkdir(name)
        for i in range(8):
                os.mkdir("{}/{}".format(name, UPDIRS[i]))
                os.mkdir("{}/{}/{}".format(name, UPDIRS[i], DIRS[i]))


makedirs(NAME, dirs, updirs)

intext = sys.argv[1]



def createHM(name, P, t0):
        for Bet in range(-90, 95, 5):
                if Bet <= -67.5:
                        for Lam in range(0, 365, 5):
                                filename = name +"/a/A/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= -45:
                        for Lam in range(0, 365, 5):
                                filename = name +"/b/B/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= -22.5:
                        for Lam in range(0, 365, 5):
                                filename = name +"/c/C/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
```

```
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= 0:
                        for Lam in range(0, 365, 5):
                                filename = name+"/d/D/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= 22.5:
                        for Lam in range(0, 365, 5):
                                filename = name+"/e/E/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= 45:
                        for Lam in range(0, 365, 5):
                                filename = name+"/f/F/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                                CP = open(filename, "w+")
                                CP.write(str(Lam) + " " + "0\n")
                                CP.write(str(Bet) + " " + "0\n")
                                CP.write(P + " " + "0\n")
                                CP.write(t0 + " " + "0\n")
                                CP.write("0" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("6" + " " + "6\n")
                                CP.write("5\n")
                                CP.write("0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("-0.5" + " " + "0\n")
                                CP.write("0.1" + " " + "0\n")
                                CP.write("50")
                elif Bet <= 67.5:
                        for Lam in range(0, 365, 5):
```

```
                filename = name+"/g/G/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                CP = open(filename, "w+")
                CP.write(str(Lam) + " " + "0\n")
                CP.write(str(Bet) + " " + "0\n")
                CP.write(P + " " + "0\n")
                CP.write(t0 + " " + "0\n")
                CP.write("0" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("6" + " " + "6\n")
                CP.write("5\n")
                CP.write("0.5" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("-0.5" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("50")
        elif Bet <= 90:
            for Lam in range(0, 365, 5):
                filename = name+"/h/H/" + str(Bet).zfill(2) + "_" + str(Lam).zfill(3) + ".txt"
                CP = open(filename, "w+")
                CP.write(str(Lam) + " " + "0\n")
                CP.write(str(Bet) + " " + "0\n")
                CP.write(P + " " + "0\n")
                CP.write(t0 + " " + "0\n")
                CP.write("0" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("6" + " " + "6\n")
                CP.write("5\n")
                CP.write("0.5" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("-0.5" + " " + "0\n")
                CP.write("0.1" + " " + "0\n")
                CP.write("50")



def writebat(letter,fold, inLC):
    batfile = open(fold + "/AutoPipe.bat", "w+")
    batfile.write("for %%f in ({}/*) do ( \n".format(letter))
    batfile.write("..\..\convexinvListopad.exe -v -o A%%f -p P%%f {}\%%f OLC%%f <{} \n".format(letter, inLC))
    batfile.write(")")
    batfile.close()

def cpLC(fold, inLC):
    copyfile(inLC, "{}/{}".format(fold, inLC))


createHM(NAME, P, t0)
for i in range(8):
    writebat(dirs[i], "{}/{}".format(NAME,updirs[i]), intext)
    cpLC("{}/{}".format(NAME, updirs[i]), intext)
```

```python
from PIL import Image
import pickle
import sys

picname = sys.argv[1]

with open(picname, 'rb') as fp:
    data = pickle.load(fp)

Lambdas, Betas, Xis = [], [], []


im = Image.new( 'RGB', (73,37)) # create a new black image
mapa = im.load()

VS = []
for qq in data:
        VS.append(qq[2])
print(min(VS), max(VS))


#color range setup
DV = max(VS) - min(VS)
for item in data:
        j = int(item[1]/5)
        i = int((item[0]+90)/5)
        v = int((item[2] - min(VS)) * 255/DV)
        mapa[j, i] = (v,v,v)

im.save(picname + "wrap.png")
```

```
from scipy import fftpack
import sys
import numpy as np
import matplotlib.pyplot as plt
import matplotlib


#input lightcurve from damit or own data
filename = sys.argv[1]
oFil = open(filename, "r")
linesA = oFil.readlines()
lines = []

for line in linesA:
        if len(line.split()) == 8:
                lines.append(line.split())

print(lines)

t, I = [], []
for k in range(len(lines)):
        t0, I0 = float(lines[k][0]), float(lines[k][1])
        t.append(t0)
        I.append(I0)

print(t)

  # Frequency, in cycles per second, or Hertz
f_s = 10  # Sampling rate, or number of measurements per second


fig, ax = plt.subplots()

X = fftpack.fft(I)
freqs = fftpack.fftfreq(len(I)) * f_s


ax.scatter(freqs, fftpack.fftshift(abs(X)))



matplotlib.rc('xtick', labelsize=20)
matplotlib.rc('ytick', labelsize=20)
plt.ylabel("I", size=20)
plt.xlabel("JD (asteocentic)", size=20)
#ax.scatter(t, I, marker="P", c="k")

plt.show()
```

```python
from scipy import fftpack
import sys
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import math
import scipy.signal as signal


#input lightcurve from damit or own data
filename = sys.argv[1]
oFil = open(filename, "r")
linesA = oFil.readlines()
lines = []

for line in linesA:
        if len(line.split()) == 8:
                lines.append(line.split())

print(lines)

t, I = [], []
for k in range(len(lines)):
        t0, I0 = float(lines[k][0]), float(lines[k][1])
        t.append(t0)
        I.append(I0)

print(t)

fs = np.linspace(13, 80, 10**4)

pgram = signal.lombscargle(t, I, fs, normalize=True)

plt.plot(fs, pgram, c="k")
plt.tick_params(axis='both', which='major', labelsize=25)
plt.title('Lomb-Scargle periodogram', fontsize = 35) # title with fontsize 20
plt.xlabel('$\omega = 2 \pi/P $', fontsize = 35) # x-axis label with fontsize 15
plt.ylabel('Power', fontsize = 35) # y-axis label with fontsize 15

plt.show()
ps = []
for fr in fs:
        ps.append(24*2*math.pi/fr)



plt.plot(ps, pgram, c="k")
plt.tick_params(axis='both', which='major', labelsize=25)
plt.title('Lomb-Scargle periodogram', fontsize = 35) # title with fontsize 20
plt.xlabel('P (h)', fontsize = 35) # x-axis label with fontsize 15
plt.ylabel('Power', fontsize = 35) # y-axis label with fontsize 15

plt.show()
```

```python
import os
import numpy as np
import matplotlib.pyplot as plt


lambdas = []
betas = []

for filename in os.listdir("/home/marco/Desktop/pyt/COORPsyche/data"):
        openFile = open("data/" + str(filename), "r")
        lines = openFile.readlines()
        Lambda = float(lines[0].split()[0])
        Beta = float(lines[0].split()[1])
        lambdas.append(Lambda)
        betas.append(Beta)

print(lambdas)
print(betas)
n = range(len(lambdas))




plt.figure(figsize=(20,10))
fig, ax = plt.subplots()
ax.scatter(lambdas, betas, marker="+")

plt.xlim(31, 35.5)
plt.ylim(-7, -12)


for i, txt in enumerate(n):
    ax.annotate(txt, (lambdas[i], betas[i]))

outF = open("psyche-map.txt", "w+")

for i in range(len(lambdas)):
        outF.write(str(n[i]) + " " + str(betas[i]) + " " + str(lambdas[i]) + "\n")
outF.close()

plt.savefig("PsycheMapPoles6.png")
```