

# Středoškolská odborná činnost

Obor: 2. Fyzika

## Hledání nových proměnných hvězd pomocí družice TESS

## Search for new variable stars using TESS satellite

<b>Autor:</b>	Tomáš Rektořík
<b>Škola:</b>	Slovanské gymnázium Olomouc
<b>Kraj:</b>	Olomoucký kraj
<b>Konzultant:</b>	doc. RNDr. Petr Zasche, Ph.D.

Olomouc 2023

## **Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce 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 Olomouci dne .....

## Poděkování

Rád bych poděkoval svému konzultantovi práce SOČ panu docentovi Zaschemu za věcné poznámky k odborné stránce práce a k jejímu zpracování. Také bych rád poděkoval vedoucím Astronomické expedice za možnost seznámení se s oborem proměnných hvězd. Především děkuji své rodině a přátelům za podporu při psaní práce.

## **Anotace**

Práce se zabývá využitím satelitních dat (TESS & Gaia) k hledání a parametrizaci nových proměnných hvězd. Popisuje zpracování dat a limitace prezentovaných metod. Několik hvězd je zvoleno pro navazující pozorování a analýzu.

## **Klíčová slova**

TESS; Gaia; lightkurve; proměnné hvězdy; zákrytové dvojhvězdy; vícehvězdné systémy; pulzující proměnné hvězdy; rotující proměnné hvězdy; eruptivní proměnné hvězdy; exoplanety; python; fotometrie; hledání period

## **Annotation**

The thesis deals with the use of satellite data (TESS & Gaia) to search for and parameterize new variable stars. It describes the data processing and limitations of the presented methods. Few stars are chosen for follow-up observations and analysis.

## **Keywords**

TESS; Gaia; lightkurve; variable stars; eclipsing binaries; multiple star systems; pulsating variable stars; rotating variable stars; eruptive variable stars; exoplanets; python; photometry; period search



# Obsah

Úvod	8
<b>1 Proměnné hvězdy</b>	<b>9</b>
1.1 Úvod	9
1.2 Teorie	9
1.3 Typy proměnných hvězd	10
1.3.1 Geometrické proměnné hvězdy	11
1.3.1.1 Zákrytové proměnné hvězdy	11
1.3.1.2 Rotující proměnné hvězdy	12
1.3.2 Fyzikální proměnné hvězdy	13
1.3.2.1 Pulzující proměnné hvězdy	13
1.3.2.2 Eruptivní proměnné hvězdy	14
<b>2 Družice TESS</b>	<b>14</b>
2.1 Úvod	14
2.2 Technické parametry	15
2.3 Průběh pozorování	16
2.4 Datové produkty	16
2.4.1 Full Frame Images	17
2.4.2 Target Pixel Files	18
2.4.3 Výběr pozorovaných hvězd	18
2.4.4 Dostupnost datových produktů	19
<b>3 Družice Gaia</b>	<b>20</b>
3.1 Úvod	20
3.2 Datové produkty	21
3.3 Gaia Data Release 3	22
<b>4 Zkoumané hvězdy</b>	<b>23</b>
4.1 TESS Input Catalogue	23
4.2 Zpracování seznamu zkoumaných hvězd	23
4.2.1 Sloučení do jednoho datového souboru	24
4.2.2 Výběr hvězd v rozsahu magnitud	24
4.2.3 Výběr hvězd definované oblasti	25
4.2.4 Odstranění duplikátů	25
4.2.5 Odstranění Gaia proměnných hvězd	26
4.2.6 Odstranění VSX proměnných hvězd	28
4.2.7 Indexace hvězd	28
4.2.8 Astrofyzikální parametry z Gaia DR3	28
4.2.9 Statistika	30

<b>5</b>	<b>Posouzení proměnnosti</b>	<b>34</b>
5.1	Parametry výběru . . . . .	35
5.2	Skript . . . . .	35
5.2.1	Datové produkty pro každý sektor . . . . .	36
5.2.1.1	Světelná křivka . . . . .	36
5.2.1.2	Oříznutá světelná křivka . . . . .	36
5.2.1.3	Periodogram . . . . .	37
5.2.1.4	Fázová křivka . . . . .	38
5.2.2	Datové produkty všech sektorů . . . . .	39
5.2.2.1	Světelná křivka . . . . .	39
5.2.2.2	Periodogram . . . . .	39
5.2.2.3	Fázová křivka . . . . .	40
5.3	Vyskytlé problémy . . . . .	41
5.3.1	Nestažené hvězdy . . . . .	41
5.3.2	Zbytkové systematické chyby . . . . .	41
5.3.3	Nekonzistentní světelné křivky . . . . .	43
5.3.4	Světelné křivky s atypickým rozptylem šumu . . . . .	44
5.3.5	Skoky jasnosti . . . . .	45
5.3.5.1	Mezisektorové skoky jasnosti . . . . .	45
5.3.5.2	Skoky jasnosti uprostřed sektoru . . . . .	46
5.3.5.3	Skoky jasnosti uvnitř sektoru . . . . .	47
5.3.6	Blending . . . . .	48
5.3.7	Odstranění odlehlých bodů . . . . .	48
5.3.8	Objevené proměnné hvězdy . . . . .	49
5.3.8.1	TESS Eclipsing Binaries katalog . . . . .	50
5.3.8.2	Databáze ExoFOP . . . . .	50
5.4	Statistika . . . . .	51
<b>6</b>	<b>Kompletní analýza proměnnosti</b>	<b>52</b>
6.1	Doplnění parametrů do seznamu hvězd . . . . .	52
6.1.1	Zpřesnění magnitud z katalogu TIC v8.2 . . . . .	52
6.1.2	Zpřesnění souřadnic z Gaia DR3 . . . . .	53
6.1.3	Pomocné parametry ke skriptu . . . . .	54
6.2	Skript . . . . .	55
6.2.1	Zpracování dat . . . . .	56
6.2.1.1	Import knihoven, úvodní parametry . . . . .	56
6.2.1.2	Vyhledání okolí hvězdy v Gaia DR3 . . . . .	56
6.2.1.3	Vyhledání TESS dat . . . . .	57
6.2.1.4	Vytvoření adresářů a potřebných seznamů ke zpracování . . . . .	58
6.2.1.5	Úvod zpracování dat v rámci sektoru . . . . .	58
6.2.1.6	Vytvoření aperturní masky . . . . .	60
6.2.1.7	Vytvoření masky pozadí . . . . .	61
6.2.1.8	Odstranění NaN hodnot, vytvoření světelné křivky . . . . .	63
6.2.1.9	Odstranění rozptýleného světla . . . . .	64
6.2.1.10	Odstranění zbytkové systematiky . . . . .	68

6.2.1.11	Odstranění kadencí s jasným pozadím . . . . .	77
6.2.1.12	Odstranění odlehlých hodnot . . . . .	79
6.2.1.13	Uložení světelných křivek . . . . .	79
6.2.1.14	Vytvoření periodogramu . . . . .	79
6.2.1.15	Vytvoření fázové křivky . . . . .	80
6.2.1.16	Úvod zpracování světelných křivek všech sektorů . . . . .	81
6.2.1.17	Oprava mezisektorových skoků jasnosti . . . . .	81
6.2.1.18	Vytvoření periodogramu, výpočet periody a její chyby . . . . .	82
6.2.1.19	Výpočet epochy a její chyby . . . . .	88
6.2.1.20	Vytvoření fázové světelné křivky, výpočet amplitudy . . . . .	89
6.2.2	Vizualizace dat . . . . .	90
6.2.2.1	Model rozptýleného světla . . . . .	90
6.2.2.2	Proložení funkce sigmoid . . . . .	91
6.2.2.3	Světelná křivka sektoru . . . . .	92
6.2.2.4	Detrendovaná světelná křivka sektoru . . . . .	92
6.2.2.5	Fázová světelná křivka sektoru . . . . .	93
6.2.2.6	Fázová detrendovaná světelná křivka sektoru . . . . .	93
6.2.2.7	TargetPixelFile v FFI sektoru . . . . .	94
6.2.2.8	TargetPixelFile v TPF sektoru . . . . .	94
6.2.2.9	Světelná křivka všech sektorů . . . . .	95
6.2.2.10	Detrendovaná světelná křivka všech sektorů . . . . .	95
6.2.2.11	Periodogramy všech sektorů . . . . .	96
6.2.2.12	Detailní periodogramy . . . . .	97
6.2.2.13	Fázová světelná křivka všech sektorů . . . . .	98
6.2.2.14	Fázová detrendovaná světelná křivka všech sektorů . . . . .	98
6.2.2.15	Kompaktní graf periodogramu a fázové světelné křivky . . . . .	99
6.3	Diskuze . . . . .	100
6.4	Klasifikace . . . . .	102
<b>7</b>	<b>Výsledky</b>	<b>102</b>
7.1	Zákrytové dvojhvězdy . . . . .	103
7.2	Eruptivní hvězdy . . . . .	105
7.3	Pulzující DSCT + GDOR hybridy . . . . .	105
7.4	Rotační hvězdy . . . . .	106
7.5	Zákrytový systém . . . . .	107
<b>8</b>	<b>Závěr</b>	<b>109</b>
<b>9</b>	<b>Literatura</b>	<b>111</b>

# Úvod

Práce se zabývá použitím dat družice TESS, vesmírného dalekohledu umístěného na oběžné dráze Země, provádějící fotometrická měření pro téměř každou hvězdu oblohy. Primárním cílem TESS je objevování exoplanet pomocí tranzitní fotometrie, avšak pořízená data jsou extrémně užitečná ke zkoumání a hledání nových proměnných hvězd. Jelikož jsou pořizovány snímky pro téměř celou oblohu, tudíž pro každý objekt v určitém rozmezí magnitud, potenciální množství dat je naprosto enormní a jejich plný rozsah nebyl zatím zcela prozkoumán. V datech se skrývá množství neobjevených exoplanetárních systémů, zákrytových systémů, pulzujících či eruptivních hvězd i spousta dalších kuriozit.

Tato práce se pokouší o představení možnosti použití dat družice TESS k průzkumu proměnnosti hvězd na vybraném vzorku asi dvou tisíc hvězd, nutné zpracování těchto dat a následné použití dat k parametrizaci proměnnosti hvězd a jejich klasifikaci za pomoci dat vesmírné družice Gaia.

# 1 Proměnné hvězdy

## 1.1 Úvod

Studium proměnných hvězd je nedílnou součástí astrofyziky a je nutné k pochopení základních principů našeho vesmíru. První pozorování proměnné hvězdy proběhlo již před více než 3000 lety ve starověkém Egyptě. Perioda 2,85 dne zákrytové dvojhvězdy Algol byla spojována se „šťastnými“ a „nešťastnými“ dny [23]. Supernova SN 1054 byla pozorována čínskými a japonskými astronomy [42]. Vzniklá planetární mlhovina *Messier 1* a pulzar<sup>1</sup> *PSR B0531+21* jsou středem výzkumu v rentgenové a gamma oblasti elektromagnetického spektra [2]. Americká astronomka Henrietta Leavittová se svým objevem závislosti periody pulzací a jasnosti Cefeid v roce 1908 [27] přispěla k objevu rozpínání vesmíru, teoreticky popsáno Alexanderem Friedmannem v roce 1922 [14] a později pozorováními potvrzeného Knutem Lundmarkem v roce 1924 [30]. Studium excentrických dvojhvězdných systémů může sloužit k testování obecné teorie relativity [19]. Pulzující proměnné hvězdy mohou poskytnout unikátní pohled do vnitřích struktur hvězd [1]. Pomocí spektroskopických měření můžeme studovat atmosféry exoplanet, které byly objeveny pomocí fotometrie [31].

## 1.2 Teorie

*Proměnné hvězdy* jsou hvězdy, jejichž jasnost se v čase mění. Tyto změny mohou být vyvolány nejrůznějšími důvody, rychlost a velikost těchto změn se také může dramaticky lišit. Z teoretického hlediska je každá hvězda hvězdou proměnnou, pokud bude pozorována dostatečně dlouho pomocí dostatečně přesného instrumentu. Avšak obecně jsou za proměnné hvězdy považovány hvězdy, jejichž změny jasnosti mají určitý významný fyzikální důvod.

*Fotometrie* se zabývá měřením množství světla (obecněji elektromagnetického záření) přicházejícího od daného objektu v čase. *Tranzitní fotometrie* hledá zákryty pozorované hvězdy jiným tělesem, např. exoplanetou<sup>2</sup>. Při měření astronomických objektů se graf této závislosti nazývá *světelná křivka*. Jas hvězdy můžeme vyjádřit pomocí *toku* v  $e^- \cdot s^{-1}$  (elektrony za sekundu), který se vztahuje k určitému detektoru. Obecně se používá *zdánlivá hvězdná velikost*, bezrozměrná logaritmická škála, kde vyšší magnituda představuje nižší jas hvězdy. Vždy se vztahuje k určitému pásmu vlnové délky. *Absolutní hvězdná velikost* vyjadřuje hvězdnou velikost hvězdy, pokud by se nacházela ve vzdálenosti 10 parseků<sup>3</sup>. *Astrometrie* se zabývá přesným měřením pozic a pohybů hvězd a ostatních těles na obloze. *Spektroskopie* se zabývá měřením spekter elektromagnetického záření astronomických objektů.

---

<sup>1</sup>Pulzar je rapidně rotující neutronová hvězda s pravidelnými změnami intenzity elektromagnetického záření.

<sup>2</sup>Exoplanety (extrasolární planety) jsou planety obíhající jinou hvězdu než je Slunce.

<sup>3</sup>Parsek je jednotka vzdálenosti, definována jako vzdálenost, ze které 1 astronomická jednotka má úhlový rozměr 1 úhlovou vteřinu. Její přibližná hodnota je  $1 \text{ pc} \approx 3,086 \cdot 10^{16} \text{ m}$ . Astronomická jednotka je jednotka vzdálenosti, dříve vyjadřující střední vzdálenost mezi Zemí a Sluncem, od roku 2012 pevně fixována vzhledem k metru na hodnotu  $1 \text{ au} = 149\,597\,870\,700 \text{ m}$ .

Pokud je ve světelné křivce přítomna pravidelná proměnnost, můžeme zkonstruovat *periodogram*. Periodogram vyjadřuje závislost síly signálu (popř. amplitudy) na frekvenci (popř. periodě). Konstrukce periodogramu probíhá pomocí *Fourierovy transformace*, která bude blíže diskutována ve zpracování dat.

Hvězdy klasifikujeme do spektrálních tříd a tříd svítivosti. Tento systém se nazývá Morganova-Keenanova klasifikace. Spektrální třída hvězd je definována na základě jejího spektra<sup>4</sup> a je zpravidla funkcí teploty. V pořadí od nejvyšší do nejnižší teploty nabývá hodnot - *W, O, B, A, F, G, K, M, L, T, Y*. Dále se dělí do podtříd pomocí číslice 0 – 9, přičemž 0 je nejvyšší teplota a 9 je nejnižší teplota; případně se specifikuje malým písmenem určitá vlastnost (např. *e* – emisní čáry, *p* – pekulární).

Třída svítivosti popisuje zářivý výkon (popř. absolutní hvězdnou velikost) vyzařovaný hvězdou. Zpravidla koreluje s poloměrem hvězdy. Nabývá hodnot *0, I, II, III, IV, V, VI, VII* a dále se dělí do podtříd *a, b, c*, kde *a* je nejzářivější. Trpaslíci se označují předponou *d* (dwarf), podtrpaslíci předponou *sd* (subdwarf) a bílí trpaslíci předponou *WD* (white dwarf).

Závislost zářivého výkonu (popř. absolutní hvězdné velikosti) na spektrální třídě (popř. efektivní teplotě<sup>5</sup>) znázorňujeme v *Hertzsprungově-Russellově diagramu*. Většina hvězd se nachází na *hlavní posloupnosti*, na které vodík spalující hvězdy stráví významnou část svého životního cyklu.

V astronomii používáme sférické souřadnicové systémy pro orientaci na obloze. Nej-používanější je *rovníková (ekvatoriální) soustava souřadnic*. Hlavní kružnicí je průmět zemského rovníku na nebeské sféře. Analogicky jsou definovány severní a jižní nebeské póly. Šířka je označována *rektascenze* a výška je označována *deklinace*. *Meridián* je kružnice na nebeské sféře procházející nebeskými póly a zenitem, tudíž závisí na pozici pozorovatele. Hlavní kružnicí *ekliptikální soustavy souřadnic* je průmět roviny oběhu Země kolem Slunce na nebeské sféře, nazývaný *ekliptika*. Sklon jednotlivých souřadnicových soustav je shodný sklonu zemské osy vzhledem k ekliptice, tudíž  $\sim 23,5^\circ$ . Průsečky nebeského rovníku a ekliptiky se nazývají *jarní a podzimní bod*. Kvůli *precesi* a *nutaci* zemské osy musí být souřadnice fixovány na určitou *epochu*, zpravidla se používá J2000.

## 1.3 Typy proměnných hvězd

Proměnné hvězdy se zpravidla dělí do dvou hlavních skupin — geometrická proměnnost (zákryty, deformace povrchu, povrchové skvrny, gravitační čočkování) a fyzikální proměnnost (pulzace, erupce, vzplanutí novy). Změny jasnosti hvězd mohou probíhat na škálách mikrosekundových (např. pulzary), přes škály v rámci dní až měsíců (pulzující či rotující hvězdy), až po škály miliard let při životním cyklu hvězdy.

Následující dělení na jednotlivé typy je adaptováno z databáze VSX [4]. Zmíněny jsou pouze typy, které jsou relevantní k zaměření této práce. Periody, amplitudy, spektrální třídy a třídy svítivosti jsou pouze orientační. Amplitudy se vztahují k vizuálnímu spektru.

<sup>4</sup>Ve spektru hvězdy lze pozorovat *spektrální čáry*, které vypovídají o chemickém složení.

<sup>5</sup>Efektivní teplota hvězdy je teplota, při které by absolutně černé těleso vyzařovalo stejné množství elektromagnetického záření jako daná hvězda.

## 1.3.1 Geometrické proměnné hvězdy

### 1.3.1.1 Zákrytové proměnné hvězdy

#### Zákrytové dvojhvězdy

Je odhadováno, že třetina hvězd v Mléčné dráze se nachází v binárních systémech [26]. V případě, že rovina oběhu hvězd je přibližně rovnoběžná se spojnicí pozorovatel – systém, dochází ke vzájemným zákrytům jednotlivých složek systému z pohledu pozorovatele. Ty se na světelné křivce projeví jako poklesy jasnosti, přičemž za jeden oběh dochází zpravidla ke dvěma zákrytům s rozdílnou hloubkou – tzv. *primární a sekundární minima*, kde primární minimum je hlubší. Poměr hloubek minim je úměrný poměru povrchových teplot jednotlivých složek.

Dvojhvězdy můžeme dělit na tři skupiny – kontaktní, polodotykové a oddělené systémy. Toto rozdělení je na základě vyplnění Rocheových laloků<sup>6</sup> jednotlivých složek. Perioda dvojhvězdy se může v čase měnit, např. kvůli přelivu hmoty z jedné složky na druhou, magnetickému brždění či přítomnosti dalších složek. Tyto změny znázorňujeme v *O – C diagramu*<sup>7</sup>.

#### $\beta$ Persei (EA)

Jinak nazývané *Algolidy*, jsou zákrytové dvojhvězdy se sférickými či mírně elipsoidními komponenty. Komponenty jsou zpravidla oddělené. Z jejich světelné křivky je možné určit časy počátku a konce zákrytů. Světelná křivka mezi zákryty je víceméně konstantní, někdy je možné pozorovat mírné změny jasnosti kvůli elipsoidnímu tvaru komponent, povrchovým skvrnám či reflexním efektům<sup>8</sup>. Sekundární minima nemusí být přítomna. Periody jsou v rozsahu od zlomku dne až několik stovek let. Amplitudy jsou v rozsahu několik milimagnitud až několik magnitud.

#### $\beta$ Lyrae (EB)

Zákrytové dvojhvězdy s elipsoidními komponenty, často polodotykové. Z jejich světelné křivky je nemožné určit přesné počátky a konce zákrytů kvůli neustále se měnící jasnosti. Sekundární minimum je vždy přítomno a jeho hloubka je zpravidla výrazně menší než hloubka primárního minima. Periody jsou v rozsahu 0,5 dne až několik dní. Komponenty zpravidla náležejí spektrálním třídám A a B.

#### *W* UMa (EW)

Zákrytové dvojhvězdy s elipsoidními komponenty téměř v kontaktu. Z jejich světelné křivky je nemožné určit přesné počátky a konce zákrytů. Sekundární minimum je vždy přítomno a je stejně nebo velmi podobně hluboké jako primární minimum. Periody jsou v rozsahu 0,2 dne až 1 den. Komponenty zpravidla náležejí spektrálním třídám F a G.

---

<sup>6</sup>Rocheův lalok je prostorové ohraničení ekvipotenciální plochy systému dvou těles, má tvar dvou dotýkajících se kapek. Při přeplnění Rocheova laloku jedné z hvězd dochází k přelivu hmoty na druhou složku [3].

<sup>7</sup>Observed – Calculated diagram

<sup>8</sup>Světlo vyzářené jednou složkou zahřívá část povrchu druhé složky. Během rotace dvojhvězdy pak zahřátá část povrchu způsobuje změny jasnosti [40].

## Vícehvězdné systémy

Řada hvězd se může nacházet v blízkých, gravitačně vázaných systémech tří nebo více hvězd [53]. Morfologie takových systémů může být různá, mezi časté patří například 2+1 či 2+2. V systému trojhvězdy 2+1 může být prostřední pár hvězd zákrytový a třetí složka obíhat na vzdálené orbitě. Takový systém je možné objevit pomocí změn v O – C diagramu prostřední zákrytové dvojhvězdy. Mezi zajímavé případy patří například šestihvězda TYC 7037-89-1 složená ze tří zákrytových dvojhvězd [38] či devítihvězda QZ Carinae [32].

## Zákryty exoplanetami

Stejně jako u zákrytových dvojhvězd, při přechodu exoplanety přes disk mateřské hvězdy můžeme pozorovat malý pokles jasnosti. Detekovat milimagnitudové změny jasnosti při pozemských pozorováních je velmi náročné, nicméně v poslední dekádě díky vesmírným misím jako Kepler [25] a TESS [41] bylo pomocí fotometrických měření objeveno tisíce nových exoplanet.

### 1.3.1.2 Rotující proměnné hvězdy

#### *Elipsoidní dvojhvězdy (ELL)*

V případě, že rovina oběhu dvojhvězdy není přibližně rovnoběžná se spojnicí pozorovatel – systém, nedochází z pohledu pozorovatele k zákrytům. Nicméně díky elipsoidním tvarům složek, reflexnímu efektu či povrchovým skvrnám můžeme pozorovat změny jasnosti, které odpovídají oběžné době dvojhvězdy. Často se jedná o těsné dvojhvězdy a amplituda je zpravidla jednotky až desítky milimagnitud.

#### *Heartbeat dvojhvězdy (HB)*

Velmi excentrické ( $e > 0,2$ ) dvojhvězdy bez zákrytů. Složky se při průletu periastronem<sup>9</sup> deformují kvůli slapovým jevům a navzájem ozáří, díky čemuž nastane změna jasnosti. Jak z názvu vyplývá, světelné křivky připomínají kardiogram.

#### *Proměnné hvězdy s povrchovými skvrnami (ROT)*

Povrch hvězdy není homogenní, ale nacházejí se na něm hvězdné skvrny. Tyto skvrny mají rozdílnou teplotu oproti okolní fotosféře<sup>10</sup>, tudíž vyzařují jiné množství světla. Hvězdné skvrny jsou vyvolané lokálními magnetickými poli [46]. Při rotaci hvězdy skvrny způsobují změny jasnosti, pomocí kterých lze určit periodu rotace hvězdy [6]. Amplituda je zpravidla méně jak 10 milimagnitud.

---

<sup>9</sup>Nejbližší bod oběžné dráhy hvězdy.

<sup>10</sup>Vnější část atmosféry hvězdy neprostupná pro viditelné fotony.



## 1.3.2 Fyzikální proměnné hvězdy

### 1.3.2.1 Pulzující proměnné hvězdy

Hvězdy jsou v *hydrostatické rovnováze*, při níž je tlak plynu a záření neustále vyrovnáván gravitační silou. Hvězdy kolem tohoto stavu nepřetržitě oscilují. Pulzace se dělí na radiální a neradiální. Při radiálních pulzacích se celý povrch hvězdy cyklicky rozpíná a smršťuje, při neradiálních pulzacích se deformuje tvar hvězdy. Změny jasnosti jsou pak způsobeny změnami povrchové efektivní teploty a rozměrem hvězdy [33].

Aby hvězda byla udržována kolem rovnovážného stavu, proti oscilacím působí síly, které je vyrovnávají. Ty se dělí na *tlakové* (či *akustické*) *mody* (*p-mody*) a *gravitační mody* (*g-mody*). Většina tradičních pulzujících hvězd se nachází v určité části H-R diagramu, nazývané *pás nestability*. Věda zabývající se oscilacemi ve hvězdách se nazývá *asteroseismologie*.

#### $\delta$ Scuti (DSCT)

Hvězdy spektrálních tříd A a F, tříd svítivosti III - V, s periodou pulzací v rozmezí 0,01 až 0,2 dne. Amplituda je v rozsahu 0,003 – 0,9 mag. Tvary světelných křivek se mezi zástupci mohou lišit. Zpravidla dochází k modulaci amplitudy a jsou pozorovány radiální i neradiální pulzace. DSCT pulzují převážně v p-modu.

#### $\gamma$ Doradus (GDOR)

Hvězdy spektrálních tříd A a F, tříd svítivosti IV a V, s periodou pulzací v rozmezí 0,2 až 4 dny. Zpravidla dochází k modulaci amplitudy, pulzace jsou neradiální v g-modu. Amplituda je zpravidla méně než 0,1 mag.

#### $\beta$ Cephei (BCEP)

Hvězdy spektrálních tříd O a B s periodou pulzací v rozmezí 0,1 až 0,6 dne a amplitudou 0,01 až 0,3 mag. Pulzace jsou zpravidla radiální, nicméně neradiální pulzace jsou také pozorovány. Pulzují převážně v p-modu.

#### *Pomalou pulzující B hvězdy* (SPB)

Hvězdy spektrální třídy B hlavní posloupnosti s periodou pulzací v rozmezí 0,4 až 5 dní a amplitudou menší než 0,1 mag. Pulzace jsou zpravidla radiální v g-modu.

#### *Hybridy*

Vesmírné dalekohledy s nevídanou přesností fotometrie změnily představy o pulzujících hvězdách. Většina pulzujících hvězd v pásu nestability pulzuje v p-modu i g-modu, tudíž neexistují „čisté“ DSCT či GDOR hvězdy [18], ale pouze jejich hybridy s dominantním p-modem nebo g-modem. Dokonce, v části H-R diagramu mezi DSCT/GDOR a SPB/BCEP hvězdami se nachází pulzující hvězdy s periodami méně než 0,1 dne, nazývané *Maia* [7].

### 1.3.2.2 Eruptivní proměnné hvězdy

#### *UV Ceti (UV)*

Hvězdy spektrálních tříd KVe a MVe vykazující erupce i zpravidla rotační proměnnost. Amplituda erupcí může být v rozsahu od nedetekovatelných až po několik magnitud. Amplituda je výrazně vyšší v ultrafialové oblasti spektra. Maximum je dosaženo během několika sekund až desítek sekund po začátku erupce, po které se hvězda vrací na původní jasnost během několika minut až desítek minut. Erupce jsou stejně jako na Slunci způsobeny přebytkem energie při *magnetické rekonexi*.

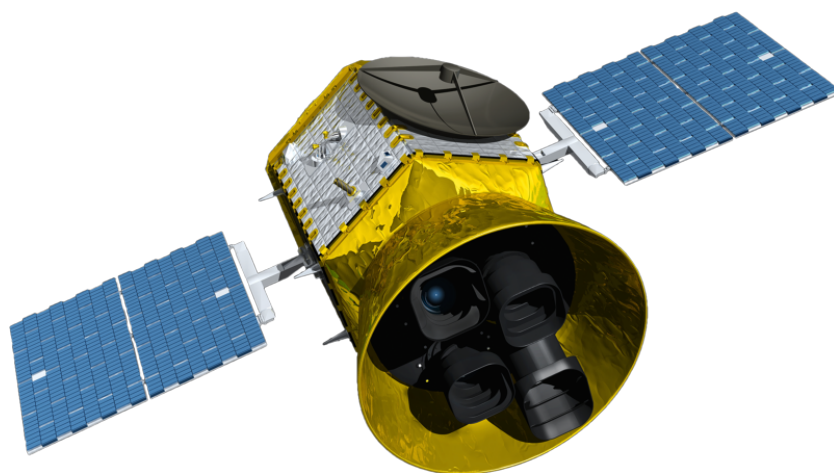
Hvězdy rotují nerovnoměrně, mají tzv. *diferenciální rotaci*, při níž póly rotují pomaleji než rovník. Diferenciální rotace způsobuje změny v magnetickém poli a vytváří lokální magnetická pole, která se projevují jako hvězdné skvrny. Magnetická pole se mohou přepojovat na nižší energie při magnetické rekonexi. Nadbytečná energie pak způsobí erupci.

## 2 Družice TESS

*Následující text čerpá z Ricker et al. (2015) [41] a [50].*

### 2.1 Úvod

TESS (**T**ransiting **E**xoplanet **S**urvey **S**atellite) je vesmírný dalekohled vynesný na orbitu Země 18. dubna 2018. Byl umístěn na eliptickou orbitu s periodou  $\sim 13.7$  dne. Cílem mise je detekce exoplanet kolem jasných blízkých hvězd pomocí tranzitní fotometrie. Jasně hvězdy jsou ideální pro navazující spektroskopická pozorování, která mohou odhalit další parametry exoplanet a hvězdných systémů. Během prvních dvou let primární mise TESS napozorovala fotometrii pro téměř celou oblohu a každým dalším rokem je střídavě znovu pozorována jedna hemisféra.



Obrázek 1: Umělecká představa družice TESS [41] (upraveno).

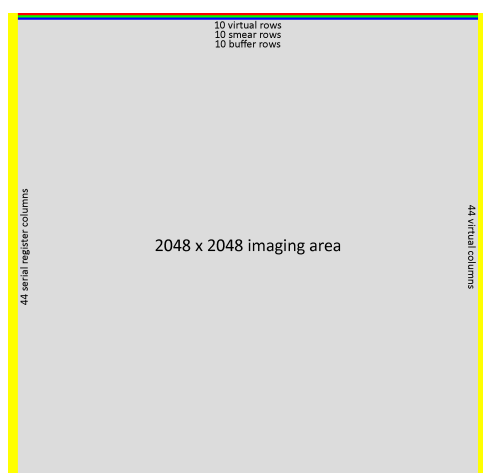
## 2.2 Technické parametry

TESS obíhá na  $108.000 \text{ km} \times 376.000 \text{ km}$  inklinované orbitě v 2:1 resonanci s Měsícem. Na této stabilní orbitě se vyhýbá radiačním pásům Země a stínům Země a Měsíce, díky čemuž se nachází ve teplotně stabilním prostředí s nízkou radiací.

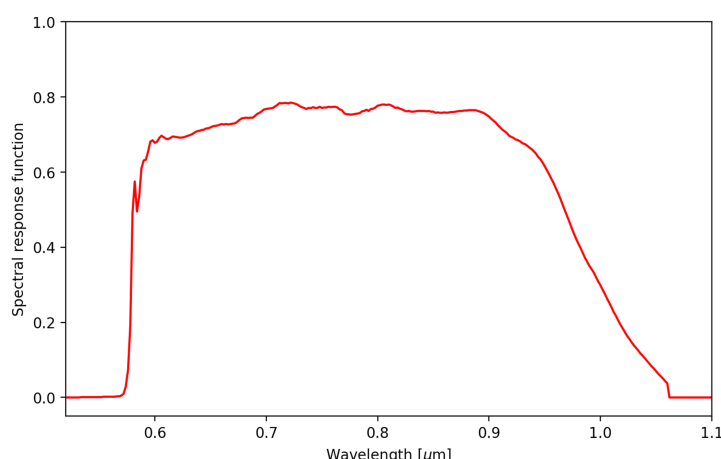
Každou orbitu při průletu perigeem<sup>11</sup> se satelit přeorientuje, aby namířil anténu na Zemi pro stažení nasbíraných dat. Od sektoru<sup>12</sup> 56 stahování dat probíhá dvakrát za jeden oběh, a to v perigeu a v apogeum<sup>13</sup>. Při stahování dat se přeruší pozorování, tudíž na světelné křivce vzniká mezera, pro sektor 56 a dál vznikají mezery tři; s tímto související efekty budou diskutovány při zpracování dat.

TESS používá pro pozorování čtyři širokoúhlé kamery, každá s vlastní soustavou čoček a čtyřmi CCD<sup>14</sup> snímači v soustavě  $2 \times 2$ . Každá kamera má zorné pole  $24^\circ \times 24^\circ$ . Jednotlivé kamery jsou namířené takovým způsobem, že na obloze vytvářejí šestnáctipanelovou mozaiku o celkovém zorném poli  $24^\circ \times 96^\circ$ . Každé CCD má rozlišení  $2048 \times 2048$  pixelů, jeden pixel má tedy na obloze rozměr  $21 \times 21$  úhlových vteřin. Toto nízké úhlové rozlišení má řadu důsledků, které budou diskutovány při zpracování dat.

Primárním cílem TESS je pozorovat hvězdy spektrálních tříd F – M. Tyto hvězdy vyzařují nejvíce na přelomu červené a infračervené oblasti spektra, podle čehož byl zvolen propustný filtr. Nejvyšší citlivost je v oblasti od 600 nm do 1000 nm.



Obrázek 2: Schéma TESS CCD snímače (vlastní tvorba).



Obrázek 3: Křivka spektrální odezvy snímáče TESS. Křivka je výsledkem propustnosti filtru a kvantové účinnosti<sup>15</sup> CCD na dané vlnové délce (vlastní tvorba). Data dostupná online z: <<https://heasarc.gsfc.nasa.gov/docs/tess/documentation.html>>

<sup>11</sup>Nejbližší bod na orbitě Země.

<sup>12</sup>Pojem sektor bude diskutován v další sekci.

<sup>13</sup>Nejvzdálenější bod na orbitě Země.

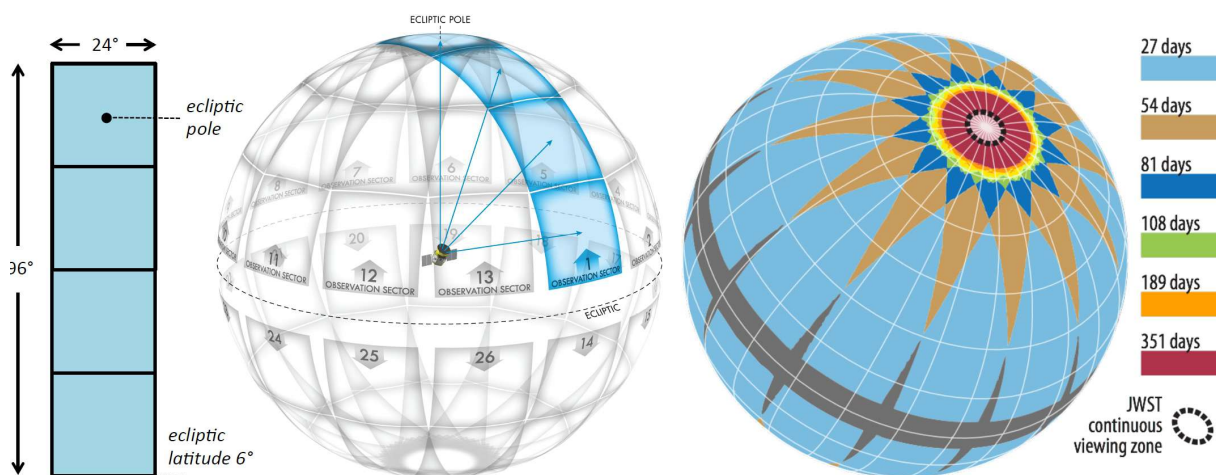
<sup>14</sup>Charge-Coupled Device – typ elektronického snímáče.

<sup>15</sup>Kvantová účinnost vyjadřuje poměr mezi počtem dopadajících fotonů a počtem excitovaných elek-

## 2.3 Průběh pozorování

Pozorování TESS je rozděleno na *sektory*. Jeden sektor je pozorován dva oběhy kolem Země, tedy  $\sim 27$  dní. V rámci jednoho sektoru se pozoruje jedna část oblohy, právě s rozměrem  $24^\circ \times 96^\circ$ . Kamera 4 je centrována na ekliptický pól a celá sada kamer je orientována podle ekliptické šířky, tudíž spodní okraj kamery 1 je  $6^\circ$  od ekliptiky. Po dokončení pozorování sektoru se TESS otočí o  $\sim 27,7^\circ$ , přičemž kamera 4 je stále centrována na ekliptický pól. Tímto způsobem se provede fotometrie pro jednu hemisféru během 13 sektorů, tedy přibližně jednoho roku.

Díky centrování kamery 1 na ekliptický pól vzniká kolem něj oblast, která je téměř nepřetržitě sledována po dobu jednoho roku, čímž se stává ideálním místem pro hledání exoplanet a jejich navazující pozorování vesmírným dalekohledem Jamese Webba. Při pozorování severní hemisféry se avšak musí pozice jednotlivých sektorů upravit tak, aby se vyvarovalo nadměrnému množství rozptýleného světla od Země a Měsíce; s tímto související efekty budou diskutovány při zpracování dat. Během čtvrtého roku pozorování severní hemisféry se také zmapovala dosud nepozorovaná část kolem ekliptiky, a to v sektorech 42 – 46 [47]. V době psaní této práce TESS pozoruje sektor 62.



Obrázek 4: Schéma pozorovaných sektorů primární mise [41].

## 2.4 Datové produkty

Čtyři CCD snímače jsou neustále exponovány a vznikající signál je každé 2 sekundy integrován. Kvůli zvýšenému množství energetického kosmického záření ve vesmíru, které při srážce se snímačem vytváří velké množství falešného signálu, musí být 20 % nejjasnějších integrací jednoho pixelu odstraněno. Ve výsledku je tímto celkový expoziční čas zkrácen na 80 %, nicméně simulace ukazují, že směrodatná odchylka dat se zvýší pouze  $1,03\times$ . 2sekundové integrace se používají na pointaci, která zpřesňuje fixování satelitu vzhledem k obloze a tím zamezuje rozptylu hvězd.

---

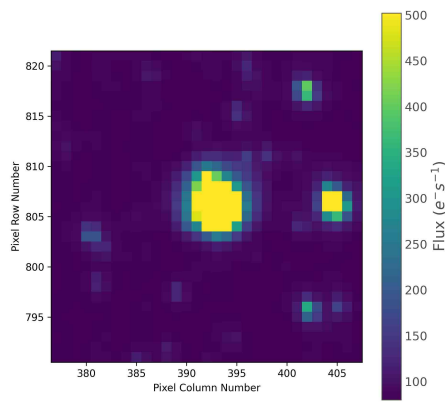
tronů přispívajících na tvorbu digitálního signálu. Jinými slovy, vyjadřuje pravděpodobnost, že foton bude snímačem registrován.

### 2.4.1 Full Frame Images

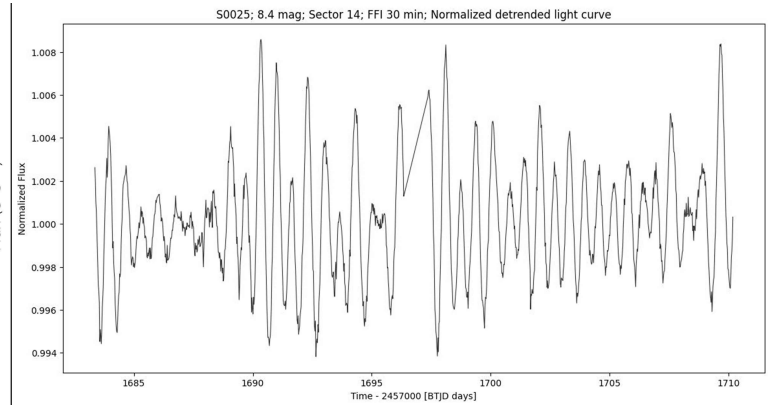


Obrázek 5: Full frame image sektoru 12. Napravo snímku lze vidět Velké Magellanovo mračno, na levé straně se táhne Mléčná dráha. Je zřetelné rozmístění jednotlivých CCD snímačů, které tvoří fotografii celého sektoru. V pravém horním rohu lze vidět odlesky způsobené rozptýleným světlem [35] (upraveno).

*Full frame images* (FFI) jsou jedním ze dvou datových produktů mise TESS. Jedná se o expozice v rozsahu celého snímače. Generování FFI probíhalo v sektorech 1 – 27 s kadencí 30 minut, tudíž efektivní expoziční čas na jeden snímek je 24 minut. Světelná křivka s touto kadencí má  $\sim 1200$  datových bodů. V sektorech 28 – 55 byla kadence snížena na 10 minut, s efektivní expoziční dobou 8 minut. Světelná křivka s touto kadencí má  $\sim 3700$  datových bodů. Od sektoru 56 je kadence 200 sekund, s efektivní expoziční dobou 160 sekund. Stahování dat v těchto sektorech probíhá čtyřikrát za sektor, vytvářející tři mezery ve světelné křivce, která má při této kadenci  $\sim 11.000$  datových bodů.



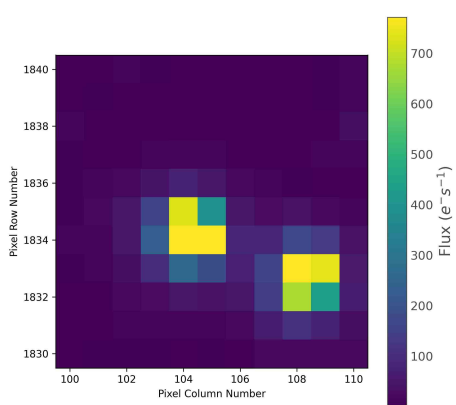
Obrázek 6: Výstřižek  $31 \times 31$  pixelů z FFI sektoru 14 kolem pulzující  $\gamma$  Doradus hvězdy TIC 232524725 pomocí TESScut (vlastní tvorba).



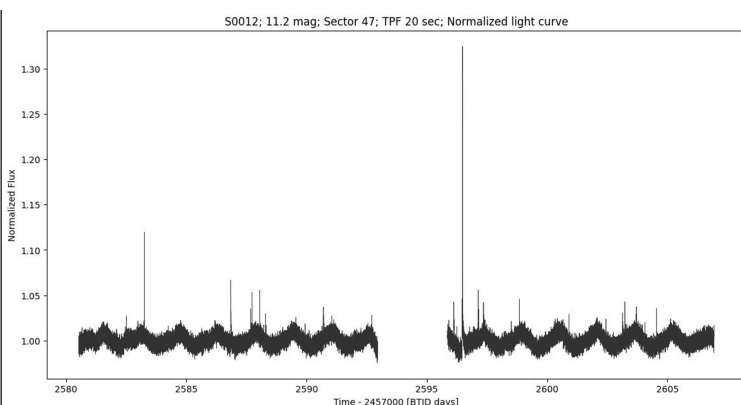
Obrázek 7: FFI světelná křivka pulzující  $\gamma$  Doradus hvězdy TIC 232524725 v sektoru 14 s kadencí 30 minut. Uprostřed světelné křivky lze vidět „złomení“, způsobené chybějícími daty při přeorientování TESS během komunikace se Zemí (vlastní tvorba).

## 2.4.2 Target Pixel Files

*Target pixel files* (TPF) jsou druhým ze dvou datových produktů mise TESS. Jedná se o malé výřezky kolem vybraných hvězd. Většina TPF má rozměr  $11 \times 11$  pixelů, přičemž jasnější hvězdy mají větší výřez. TPF se dělí na dvě kadence – „short“ a „fast“. „Short“ TPF jsou sbírány s kadencí 2 minuty, tudíž efektivní expoziční čas na jeden snímek je 96 sekund. Světelná křivka s touto kadencí má  $\sim 19.000$  datových bodů. „Fast“ TPF jsou dostupné od sektoru 28 a jsou sbírány s kadencí 20 sekund, tudíž efektivní expoziční čas na jeden snímek je 16 sekund. „Fast“ TPF je datový produkt mise TESS s vůbec nejvyšší kadencí, světelná křivka pozorované hvězdy jediného sektoru má potom  $\sim 110.000$  datových bodů.



Obrázek 8: Target Pixel File sektoru 47 eruptivní *UV Ceti* hvězdy TIC 156196475 v levé části. TPF zahrnuje i hvězdu TIC 156196470 napravo (vlastní tvorba).



Obrázek 9: „Fast“ TPF světelná křivka eruptivní *UV Ceti* hvězdy TIC 156196475 v sektoru 47 s kadencí 20 sekund. Opět lze vidět mezeru v polovině sektoru způsobenou přeorientací TESS kvůli komunikaci se Zemí (vlastní tvorba).

## 2.4.3 Výběr pozorovaných hvězd

*Následující text čerpá z Fausnaugh (2021) [13].*

Algoritmický výběr pozorovaných hvězd v rámci TPF má na starost *TESS Payload Operation Center* (POC). POC vytvoří z 6 různých *Candidate Target List* (CTL) seznam hvězd, který je v daném sektoru pozorován. První CTL se skládá z 1920 hvězd používaných jako inženýrské cíle, sloužící pro pointaci a analýzu výkonu CCD. Druhé CTL se skládá ze všech jasných hvězd magnitudy 6 a méně. Třetí CTL je vytvářeno *Target Selection Working Group* (TSWG), obsahující 13.400 hvězd ideálních pro objev exoplanetárních tranzitů. Čtvrté CTL je vytvářeno *TESS Asteroseismic Science Consortium* (TASC), obsahující 751 hvězd vhodných pro asteroseismologii. Páté CTL je vytvářeno *Guest Investigator office*, obsahující 1501 hvězd zvolených výběrovým řízením. Podobně je šesté CTL tvořeno 1501 hvězdami zvolených výběrovým řízením v programu *Director's Discretionary Time*.



Množství pozorovaných hvězd je z technických důvodů omezeno na 20.000 na sektor a 2000 na jedno CCD. V prvních třech sektorech bylo vybráno  $\sim 16.000$  hvězd na pozorování s 2minutovou kadencí. V sektorech 4 – 27 bylo množství rozšířeno na  $\sim 20.000$  hvězd na sektor. Od sektoru 28 se k 20.000 hvězdám pozorované v „short“ kadenci přidalo 1000 hvězd pozorované ve „fast“ kadenci. Od sektoru 56 se počet hvězd „short“ kadence snížil na 12.000, ale počet hvězd ve „fast“ kadenci narostl na 2000.

Kvůli velkému zájmu o pozorování hvězd v TPF kadenci dochází k prioritizaci. Ta probíhá na základě hvězdných parametrů z *TESS Input Catalogue* (TIC) [45]. Katalog TIC bude blíže diskutován při výběru dat.

Je důležité zmínit, že seznam pozorovaných hvězd se sektor od sektoru mění a pouze některé hvězdy jsou pozorované v TPF kadenci ve všech dostupných sektorech.

#### 2.4.4 Dostupnost datových produktů

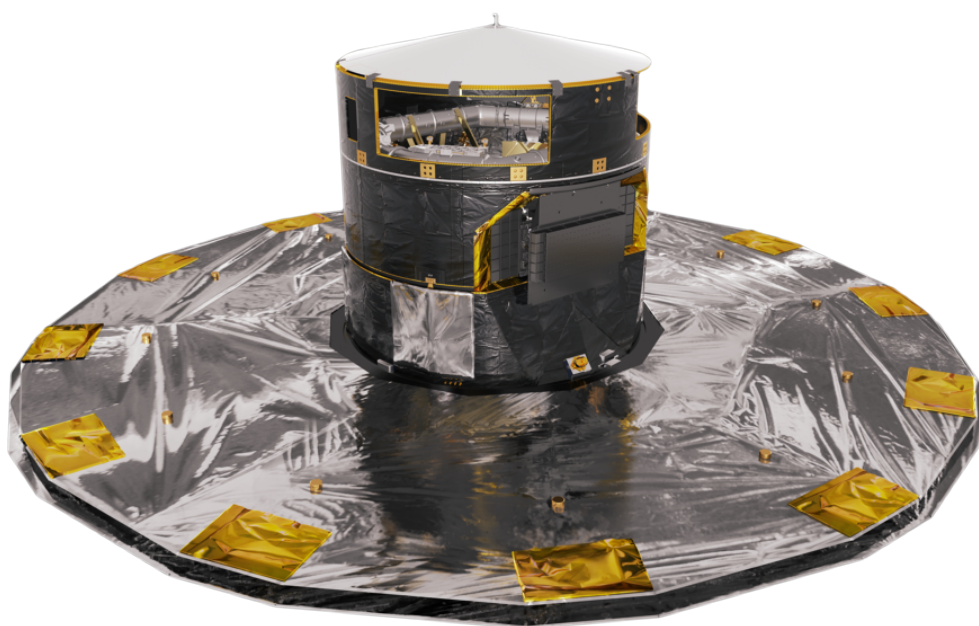
Všechny datové produkty mise TESS jsou dostupné online na *Mikulski Archive for Space Telescopes* (MAST). Kalibrace FFI provádí *Science Processing & Operations Center* (SPOC) [22]. SPOC také zpracovává světelné křivky pro všechny hvězdy pozorované v rámci TPF, navíc vyhledává ve světelných křivkách tranzity exoplanet. Služba **TESScut** [8] provozovaná MAST umožňuje tvoření libovolně velkých výřezků podobným TPF kolem jakékoliv hvězdy z kalibrovaných FFI dat. V této práci jsou použity světelné křivky generované SPOC a služba **TESScut**.

## 3 Družice Gaia

*Následující text čerpá z Gaia Collaboration et al. (2016) [15] a (2022) [16].*

### 3.1 Úvod

Družice Gaia je vesmírný dalekohled vynesný do L2 bodu<sup>16</sup> Země 19. prosince 2013. Provádí přesná astrometrická, fotometrická a spektroskopická měření pro celou oblohu. Společně s misí TESS představuje velmi výkonný dataset pro celoooblohové analýzy hvězdné proměnnosti. Mise Gaia poskytla unikátní pohled do historie a dynamiky naší Galaxie. Detailní technický popis mise Gaia nebude v této práci zmíněn, jelikož není relevantní k jejímu zaměření.



Obrázek 10: Umělecká představa družice Gaia [12] (upraveno).

---

<sup>16</sup>Langrangeův bod 2 je stabilní bod v soustavě tří těles, v tomto případě Slunce, Země a Měsíce. Nachází se ve vzdálenosti  $\sim 1.4$  milionu kilometrů od Země na straně opačné Slunci.

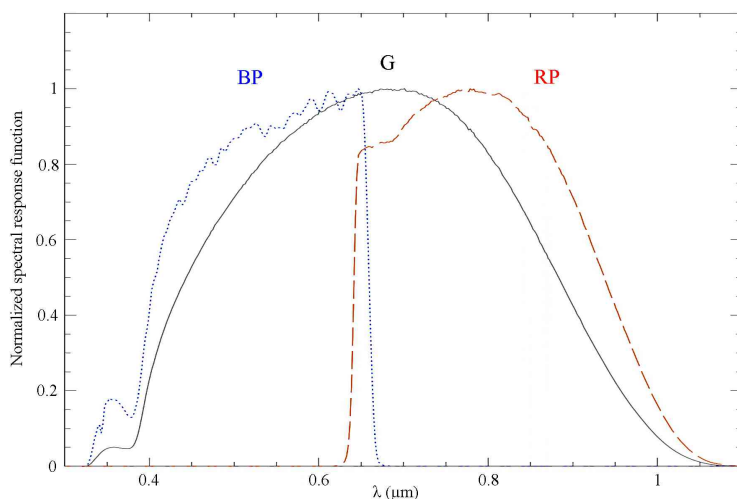


## 3.2 Datové produkty

Data mise Gaia jsou zveřejňovány vědecké komunitě v jednotlivých balíčcích, tzv. *Data Release (DR)*. Gaia pozoruje nejenom hvězdy, ale i víceméně všechny objekty na obloze, včetně galaxií, hvězd mimo naší Galaxii (např. Galaxie v Andromedě, Magellanova mračna) a objektů Sluneční soustavy (např. planety).

Zpracování veškerých dat mise Gaia má na starosti *Data Processing and Analysis Consortium (DPAC)*. Kromě základních dat jako jsou pozice, světelné křivky a spektra DPAC publikuje celou řadu astrofyzikálních parametrů pro vybrané objekty, včetně typů proměnnosti.

Gaia pro fotometrii používá tři filtry: hlavní filtr  $G$ , který se svojí propustností podobá propustnosti filtru TESS. Filtry  $G_{BP}$  a  $G_{RP}$  dělí filtr  $G$  na poloviny, což umožňuje určovat  $G_{BP}-G_{RP}$  index<sup>17</sup>.



Obrázek 11: Křivky spektrální odezvy snímače Gaia ve filtrech  $G$ ,  $G_{BP}$ ,  $G_{RP}$ . Křivky jsou normalizované, tudíž nepředstavují absolutní odezvu, která se v nejvyšších hodnotách pohybuje kolem  $\sim 75\%$  [24] (upraveno).

Data generovaná misí Gaia jsou dostupná v online databázi *Vizier* [36], nebo v rámci *Table Access Protocol (TAP+)*, přístupného pomocí knihovny *astroquery* [17] implementované v jazyce *Python*. V této práci používám knihovnu *astroquery* pro práci s daty Gaia a databázi *Vizier* pro obecné vyhledávání dostupných astronomických dat, včetně Gaia.

<sup>17</sup>Barevný index může přinášet hrubý odhad efektivní teploty, a to i pro velmi málo jasné hvězdy, což může být užitečné v případech, kdy nejsou dostupná spektra.

### 3.3 Gaia Data Release 3

Gaia DR3 je nejnovějším dostupným data release, zveřejněný 13. června 2022. Jedná se o největší astrometrický katalog, s pozicemi pro téměř 2 miliardy objektů a přesnou astrometrii včetně vlastního pohybu<sup>18</sup> hvězd a paralax<sup>19</sup>. Magnitudy nejméně jasných objektů sahají až k 21. magnitudě v G. Pro většinu objektů jsou dostupné i magnitudy ve filtrech  $G_{BP}$  a  $G_{RP}$ , tudíž je možné určit barevný index.

V rámci zpracování dat Gaia určuje základní astrofyzikální parametry pro téměř půl miliardu hvězd, a to např. absolutní magnitudu, efektivní teplotu, spektrální třídu,  $\log(g)$ <sup>20</sup>, poloměr, hmotnost a metalicitu<sup>21</sup>.

Přes 10 milionů hvězd bylo také zařazeno do typu proměnnosti. Bylo objeveno přes 2 miliony zákrytových dvojhvězd, přes 3 miliony rotujících či pulzujících proměnných hvězd a přes 50 tisíc spektroskopických dvojhvězd.

Je důležité zmínit vzhledem k povaze pozorování a použitým metodám, že řada klasifikací může být chybných. Světelné křivky hvězd pozorovaných Gaia mají zpravidla desítky až stovky datových bodů, což je výrazně méně než TESS. Nicméně kombinací odhadů astrofyzikálních parametrů Gaia a světelných křivek TESS můžeme dosáhnout mnohem vyšší pravděpodobnosti správné klasifikace. Bohužel, v některých případech samotná fotometrie kombinovaná s astrofyzikálními parametry není dostatečná a je potřeba dodatečná spektroskopie. Touto problematikou se bude práce zabývat při zpracování dat.

Datový produkt	N (mil.)
Astrometrie	1 812
Úplná astrometrie	1 468
G magnituda	1 806
$G_{RP}$ magnituda	1 555
$G_{BP}$ magnituda	1 542
Astrofyzikální parametry	471
$G_{BP}$ / $G_{RP}$ spektra	219
Světelná křivka	11,8
Proměnné hvězdy	10,5
Nesamotné hvězdy	0,727

Tabulka 1: Počet objektů dostupných v rámci jednotlivých datových produktů [16].

Typ proměnnosti	N (tis.)
Zákrytové dvojhvězdy	2 184
Dlouhoperiodická proměnnost	1 721
Krátkoperiodická proměnnost	472
Rotující proměnnost	474
Pulzátory vrchní HP <sup>22</sup>	54,5
Exoplanety	0,214

Tabulka 2: Typy proměnnosti charakterizované v rámci Gaia DR3 [16].

<sup>18</sup>Hvězdy se v čase na obloze pomalu pohybují při jejich oběhu kolem galaktického jádra.

<sup>19</sup>Paralaxa je úhel, který hvězda během oběhu Země kolem Slunce z pohledu pozemského pozorovatele na obloze opíše. Vzdálenost lze tedy vypočítat jako  $d = \frac{1}{p}$ , kde p je v parsecích.

<sup>20</sup>Dekadický logaritmus gravitačního zrychlení na povrchu.

<sup>21</sup>Metalicita je hojnost prvků těžších než vodík a helium ve hvězdě. Metalicita vypovídá např. o stáří hvězdy.

## 4 Zkoumané hvězdy

První krok této práce je výběr vzorku hvězd, které budou zkoumány. Vzorek musí splňovat několik požadavků. Počet hvězd ve vzorku musí být zvládnutelný v rámci této práce, tedy v řádech tisíců. Hvězdy musí mít dobré pokrytí TESS pozorování, ideálně několik sektorů, tudíž musí být v blízkosti ekliptického pólu. Zároveň by bylo vhodné, aby hvězdy byly pozorované TESS v TPF kadenci pro vysoké časové rozlišení. Vizuální magnituda hvězd musí být v takovém rozsahu, aby nebyly příliš jasné, tudíž s větší pravděpodobností již objevené a příliš rozlehlé na CCD detektoru, a zároveň nebyly příliš málo jasné, jelikož jejich světelné křivky by měly více šumu a byly by špatně rozlišitelné kvůli kontaminaci okolních hvězd. Čistě z našeho zájmu byl výběr hvězd omezen pouze na určitou část oblohy, která je dobře pozorovatelná během většiny roku ze severní hemisféry, především během jarních měsíců, z důvodu navazujících pozorování.

### 4.1 TESS Input Catalogue

*Následující text čerpá z Stassun et al. (2019) [45].*

TESS Input Catalogue (TIC) je katalog vytvořený pro výběr hvězd na pozorování TESS, výpočet kontaminace okolních hvězd pro danou hvězdu a soubor astrofyzikálních parametrů určených především v Gaia DR2. Pomocí TIC se sestavuje také katalog *Candidate Target List (CTL)*, který obsahuje hvězdy vybrané pro pozorování v TPF kadenci. Katalog TIC přiřazuje každé hvězdě unikátní identifikátor ve formátu „*TIC \_identifikátor\_*“. Pro každou hvězdu je také vypočítána její magnituda podle spektrální odezvy detektorů TESS, v této práci označována jako *TESS mag* nebo *T mag*.

### 4.2 Zpracování seznamu zkoumaných hvězd

Segment	Rektascenze (°)	Deklinace (°)
Segment 0	270 – 30	90 – 85
Segment 1	240 – 270	90 – 83
Segment 2	225 – 240	90 – 80
Segment 3	210 – 225	90 – 75
Segment 4	210 – 217,5	70 – 75
Segment 5	195 – 210	90 – 70
Segment 6	202,5 – 210	65 – 70
Segment 7	180 – 195	90 – 75
Segment 8	165 – 180	90 – 80
Segment 9	150 – 165	90 – 85

Pro vzorek jsou výsledně uvažovány následující omezení. Hvězdy musí mít TESS magnitudu v rozsahu 7 – 12. Jelikož se práce zabývá hledáním nových proměnných hvězd, zkoumané hvězdy nesmí být zahrnuté ve VSX [52], ani v proměnných hvězdách charakterizovaných v Gaia DR3 [16]. Oblast výskytu hvězd na obloze je vymezená segmenty v ekvatoriálních souřadnicích v tabulce 3. Výsledná oblast má obsah 1054 čtverečných stupňů.

Tabulka 3: Vybraná oblast zkoumaných hvězd.

Pro sestavení seznamu hvězd jsou použity *Target List* seznamy hvězd pozorované TESS v TPF kadenci dostupné online<sup>23</sup>. Je zde dostupný seznam hvězd pro každý pozorovaný sektor. Seznamy jsou rozděleny na hvězdy pozorované ve „fast“ a „short“ TPF kadenci. Seznamy jsou ve formátu CSV, obsahující následující informace: TIC identifikátor, číslo kamery, číslo CCD, T mag, rektascenze, deklinace.

K vyšetření potřebných seznamů ke stažení je využit modul `tess-point` [10] dostupný online<sup>24</sup>. Jsou použity souřadnice okrajových bodů vymezených v tabulce 3 a stáhnuty *target list* seznamy pro následující sektory: 14, 15, 16, 18, 19, 20, 21, 22, 25, 26, 40, 41, 47, 48, 49, 51, 52, 53, 58, 59, 60.

### 4.2.1 Sloučení do jednoho datového souboru

V této práci používáme prostředí JupyterLab<sup>25</sup> pro psaní kódů. Pro práci s CSV soubory je v této práci použita knihovna `pandas` [48].

Nejdříve jsou *target list* seznamy pro všechny sektory spojeny do jediného pomocí následujícího kódu. Seznamy jsou vytvořeny dva, a to jeden s hvězdami s dostupnou „fast“ kadencí a jeden s hvězdami s dostupnou „short“ kadencí.

```
import pandas as pd

sectors = [14, 15, 16, 18, 19, 20, 21, 22, 25, 26, 40, 41, 47, 48, 49, 51, 52, 53, 58, 59, 60]
name = 'all_targets_S0'
suffix = '_v1.csv'

df_list = [pd.read_csv(f'path{name}{s}{suffix}') for s in sectors]
df_master = pd.concat(df_list, ignore_index = True)

df_master.to_csv('path')
```

Výsledný CSV soubor s hvězdami ve „fast“ kadenci obsahuje 16.106 hvězd, soubor s hvězdami v „short“ kadenci obsahuje 399.000 hvězd.

### 4.2.2 Výběr hvězd v rozsahu magnitud

Pomocí kódu níže je proveden výběr hvězd v rozsahu 7 – 12 TESS magnitudy. Při tomto kroku jsou také odstraněny sloupce „camera“ a „ccd“. Omezením magnitudy do rozsahu 7 – 12 bylo odstraněno 40 % hvězd ve „fast“ kadenci a 28 % hvězd v „short“ kadenci.

```
import pandas as pd

df = pd.read_csv('path')

df.drop(df.columns[['camera', 'ccd']], axis = 1, inplace = True)
df = df[(df['mag'] > 7) & (df['mag'] < 12)]

df.to_csv('path')
```

---

<sup>23</sup>[https://tess.mit.edu/public/target\\_lists/target\\_lists.html](https://tess.mit.edu/public/target_lists/target_lists.html)

<sup>24</sup><https://github.com/tessgi/tess-point>

<sup>25</sup><https://jupyter.org/>

### 4.2.3 Výběr hvězd definované oblasti

Aktuální seznamy obsahují hvězdy po celé obloze a musí být omezeny na oblast definovanou v tabulce 3 pomocí kódu níže.

```
import pandas as pd

df = pd.read_csv('path')

df0 = df[(df['ra'] > 270) & (df['ra'] < 30) & (df['dec'] > 85)]
df1 = df[(df['ra'] > 240) & (df['ra'] < 270) & (df['dec'] > 83)]
df2 = df[(df['ra'] > 225) & (df['ra'] < 240) & (df['dec'] > 80)]
df3 = df[(df['ra'] > 217.5) & (df['ra'] < 225) & (df['dec'] > 75)]
df4 = df[(df['ra'] > 210) & (df['ra'] < 217.5) & (df['dec'] > 70)]
df5 = df[(df['ra'] > 202.5) & (df['ra'] < 210) & (df['dec'] > 65)]
df6 = df[(df['ra'] > 195) & (df['ra'] < 202.5) & (df['dec'] > 70)]
df7 = df[(df['ra'] > 180) & (df['ra'] < 195) & (df['dec'] > 75)]
df8 = df[(df['ra'] > 165) & (df['ra'] < 180) & (df['dec'] > 80)]
df9 = df[(df['ra'] > 150) & (df['ra'] < 165) & (df['dec'] > 85)]

df_master = pd.concat([df0, df1, df2, df3, df4, df5, df6, df7, df8, df9])

df_master.to_csv('path')
```

### 4.2.4 Odstranění duplikátů

Řada hvězd byla zahrnuta v *target list* několika sektorů, tudíž aktuální seznamy obsahují duplikáty. Pro odstranění duplikátů je v knihovně pandas funkce `drop_duplicates()`.

```
import pandas as pd

df = pd.read_csv('path')
df = df.drop_duplicates(subset = 0, keep = 'first', ignore_index = True)

df.to_csv('path')
```

Určité hvězdy jsou také zahrnuté v seznamu „short“ i „fast“ kadence a musí být odstraněny ze seznamu „short“ kadence.

```
import pandas as pd

df_fast = pd.read_csv('path')
df_short = pd.read_csv('path')

for tic in df_fast['tic']:
    df_short = df_short[df_short['tic'] != tic]

df_short.to_csv('path')
```

## 4.2.5 Odstranění Gaia proměnných hvězd

Data družice Gaia [16] jsou v následujícím kódu získána prostřednictvím databáze Vizier [36], přístupné v rámci balíčku `astroquery` [17] v knihovně `astropy` [5].

```
import pandas as pd
from astroquery.vizier import Vizier
import astropy.units as u
import astropy.coordinates as coord
from astropy.coordinates import Angle

Vizier.TIMEOUT = -1 # unlimited server timeout

df = pd.read_csv('path')
```

První problematikou je vyhledání správné hvězdy v katalogu Gaia DR3. Souřadnice uvedené v CSV souborech mají přesnost na 4 desetinná místa ve stupních, tudíž mohou mít odchylku  $\pm 0,25$  úhlové vteřiny od souřadnic uvedených v Gaia DR3. Navíc, zdroj souřadnic v datových souborech je jiný než Gaia DR3, tudíž se přesné souřadnice u některých hvězd mohou lišit, zvláště u hvězd s velkým vlastním pohybem. Pro nejvyšší přesnost správné identifikace hvězd jsou použity následující nástroje.

Hlavní data Gaia DR3 jsou v databázi Vizier dostupná pod katalogovým označením `I/355/gaiadr3`. Pro práci v následujícím kódu potřebujeme číst pouze sloupce `phot_variable_flag` a `phot_g_mean_mag`. Sloupec `phot_variable_flag` nabývá hodnot `NOT_AVAILABLE` a `VARIABLE`. Sloupec `phot_g_mean_mag` obsahuje magnitudu hvězdy ve filtru G. Argument `+r` řadí vyhledané hvězdy podle vzdálenosti od zadaných souřadnic.

```
v = Vizier(columns = ['+r', 'phot_variable_flag', 'phot_g_mean_mag'], row_limit = 100)
```

Okolí hvězdy definované jejími souřadnicemi zapsanými ve sloupcích `ra` a `dec` vyhledáme pomocí `cone search`, zabudované ve funkci `astroquery.vizier.query_object()`. Počáteční poloměr kuželové výseče zvolíme 2 úhlové vteřiny. V případě, že toto okolí neobsahuje žádnou hvězdu, musíme iterativně zvětšovat poloměr této výseče, dokud se v ní hvězda nenachází. Analogicky, v případě, že okolí obsahuje více jak jednu hvězdu, iterativně zmenšujeme poloměr, dokud se v ní nenachází pouze jedna hvězda.

```
for ra, dec, n in zip(df['ra'], df['dec'], range(10000)):
    # Download of photometric variability flag, photometric G band mean magnitude from Gaia DR3 catalogue
    try:
        download = v.query_region(coord.SkyCoord(ra, dec, unit = (u.deg, u.deg)), catalog = "I/355/gaiadr3",
                                   radius = 2 * u.arcsec)

        if len(download) == 1:
            if len(download["I/355/gaiadr3"]) == 1:
                pass
            elif len(download["I/355/gaiadr3"]) > 1:
                for i in range(1, 31):
                    r = 2 - (0.05 * i)
                    download = v.query_region(coord.SkyCoord(ra, dec, unit = (u.deg, u.deg)),
                                                catalog = "I/355/gaiadr3",
                                                radius = r * u.arcsec)

                    if len(download["I/355/gaiadr3"]) == 1:
                        break
                    else:
                        continue
            else:
                pass
        elif len(download) == 0:
```

```

for i in range(1, 27):
    r = 2 + (0.5 * i)
    download = v.query_region(coord.SkyCoord(ra, dec, unit = (u.deg, u.deg)),
                              catalog = "I/355/gaiadr3", radius = r * u.arcsec)

    if len(download) == 1:
        break
    else:
        continue
else:
    continue
if len(download) == 1:
    pass
else:
    continue
download = download["I/355/gaiadr3"]
if len(download) == 1:
    pass
else:
    continue
download = download.to_pandas()
except:
    continue

```

Musíme také ověřit, že vyhledaná hvězda je hvězdou hledanou. To docílíme porovnáním TESS magnitudy hvězdy zapsané v CSV souboru a Gaia G magnitudy vyhledané hvězdy. Spektrální odezva snímačů TESS a Gaia je samozřejmě odlišná, nicméně velmi podobná, tudíž pro většinu hvězd rozdíl magnitud bude malý. Pro několik hvězd s extrémním barevným indexem je manuálně zjištěn rozdíl  $T \text{ mag} - G \text{ mag}$  a je zvolen poměrně velkorysý rozdíl 1,5 magnitudy, který by měl přesně charakterizovat všechny hvězdy ve vzorku a tím je odlišit od okolních hvězd, zpravidla s nižší magnitudou.

Poté již pouze zbývá zapsat *phot\_variable\_flag* do CSV souboru.

```

# Gaia G band mean magnitude cross-check with TESS magnitude
if float(download['Gmag'].loc[download.index[0]]) > 1:
    pass
else:
    continue
mag_dif = abs((float(download['Gmag'].loc[download.index[0]]) - (float(df['mag'].loc[df.index[n]])))
if mag_dif < 1.5:
    pass
else:
    continue

varflag = download['VarFlag'].loc[download.index[0]]
df.iat[n, 4] = varflag
df.to_csv('path', index = False)

```

Hvězdy s hodnotou *VARIABLE* ve sloupci *phot\_variable\_flag* jsou odstraněny pomocí kódu níže.

```

import pandas as pd

df = pd.read_csv('path')
df = df[df['phot_variable_flag'] == 'NOT_AVAILABLE']
df.to_csv('path')

```

Pár procent hvězd neprošlo výše zmíněnými kritérii a jejich hodnoty musely být manuálně extrahovány online z databáze Vizier<sup>26</sup>.

<sup>26</sup><https://vizier.cds.unistra.fr/viz-bin/VizieR-3?-source=I/355/gaiadr3>

## 4.2.6 Odstranění VSX proměnných hvězd

Početnost VSX proměnných hvězd ve zbývajícím vzorku hvězd byla v řádech jednotek, tudíž VSX proměnné hvězdy byly manuálně odstraněny za pomoci databáze Vizier<sup>27</sup>.

## 4.2.7 Indexace hvězd

Pro účely této práce je hvězdám ve zkoumaném vzorku přidělen čtyřmístný identifikátor ve tvaru S0000. Budou v průběhu práce používány místo zdlouhavých TIC identifikátorů. Z praktických důvodů je seznam hvězd rozdělen na CSV soubory obsahující 100 hvězd.

## 4.2.8 Astrofyzikální parametry z Gaia DR3

Při vyšetřování proměnnosti hvězd bez spektroskopických měření můžeme docílit přesnější klasifikace s menší chybovostí za pomoci znalosti některých astrofyzikálních parametrů. Určité typy proměnnosti se vyskytují pouze v určitých částech HR diagramu. Kód níže získává základní parametry z Gaia DR3 za pomoci `astroquery.gaia`, která čerpá data z TAP+.

Pro každou hvězdu jsou extrahovány sloupce *phot\_rp\_mean\_mag*, *phot\_g\_mean\_mag* a *phot\_bp\_mean\_mag*, obsahující magnitudu ve filtrech G, G<sub>BP</sub> a G<sub>RP</sub>. Dále sloupec *teff\_gsp\_phot* obsahující efektivní teplotu, *distance\_gspphot* obsahující vzdálenost, *logg\_gspphot* obsahující logaritmus gravitačního zrychlení na povrchu, *mh\_gspphot* obsahující hojnost železa, *ag\_gspphot* obsahující mezihvězdnou extinkci<sup>28</sup> ve filtru G, *ebpminrp\_gspphot* obsahující zčervenání (reddening)<sup>29</sup>.

Z uvedených parametrů je nadále vypočítána opravená magnituda ve filtru G za pomoci extinkce:

$$G_{ext} = G - Extinction$$

Barevný index hvězdy, nadále opravený o zčervenání:

$$BR_{index} = G_{BP} - G_{RP}$$

$$BR_{index,ext} = G_{BP} - G_{RP} - Reddening$$

Absolutní magnituda ve filtru G:

$$M_G = G_{ext} + 5 - 5 \log_{10}(d)$$

Zářivý výkon v násobcích zářivého výkonu Slunce<sup>30</sup>:

$$\frac{L}{L_{\odot}} = 10^{\frac{4,63 - (G_{ext} + 5 - 5 \log_{10}(d))}{2,5}}$$

<sup>27</sup><https://vizier.cds.unistra.fr/viz-bin/VizieR-3?-source=B/vsx/vsx>

<sup>28</sup>Extinkce (též ztlumení, zeslabení) je oslabování jasů hvězd, jejichž záření k pozorovateli prolétá mezihvězdným prachem. Jas hvězd se poté na obloze jeví nižší než je ve skutečnosti.

<sup>29</sup>Vedlejším efektem extinkce je zčervenání, tedy posun spektra hvězdy na delší vlnové délky. Světlo kratších vlnových délek se v mezihvězdném prachu rozptyluje více než světlo kratších vlnových délek.

<sup>30</sup>Hodnota absolutní magnitudy Slunce ve filtru G byla převzata z [11].



Tentokrát je zvolena jiná metoda pro správné vyhledávání hvězd. Je použita kuželová výšeč kolem hledané hvězdy o poloměru 30 úhlových vteřin, která bezpečně zahrne veškeré hvězdy, včetně těch s vysokým vlastním pohybem či nepřesnými souřadnicemi. Místo magnitudy v G je použita magnituda hvězdy ve filtru G<sub>RP</sub>, která se lépe shoduje s TESS magnitudou. Hvězda s nejmenším rozdílem  $T \text{ mag} - G_{RP} \text{ mag}$  je vybrána jako správná.

```
import pandas as pd
import numpy as np
from astroquery.gaia import Gaia
import astropy.units as u
from astropy.coordinates import SkyCoord

df = pd.read_csv('path')

Gaia.MAIN_GAIA_TABLE = "gaiadr3.gaia_source"
Gaia.ROW_LIMIT = 10000

for i, identifier, ra, dec, mag in zip(range(10000), df['id'], df['ra'], df['dec'], df['mag']):
    obj = SkyCoord(f'{ra} {dec}', unit = (u.deg, u.deg))
    gaia = Gaia.cone_search_async(obj, 30 * u.arcsec).get_results()
    gaia = gaia.to_pandas()
    gaia = gaia[['phot_g_mean_mag', 'phot_rp_mean_mag', 'phot_bp_mean_mag', 'teff_gspphot',
                'distance_gspphot', 'logg_gspphot', 'mh_gspphot', 'ag_gspphot',
                'ebpminrp_gspphot']]

    if len(gaia) < 1:
        continue

    gaia['magdif'] = abs(gaia['phot_rp_mean_mag'] - mag)
    gaia.sort_values('magdif', ascending = True, inplace = True)

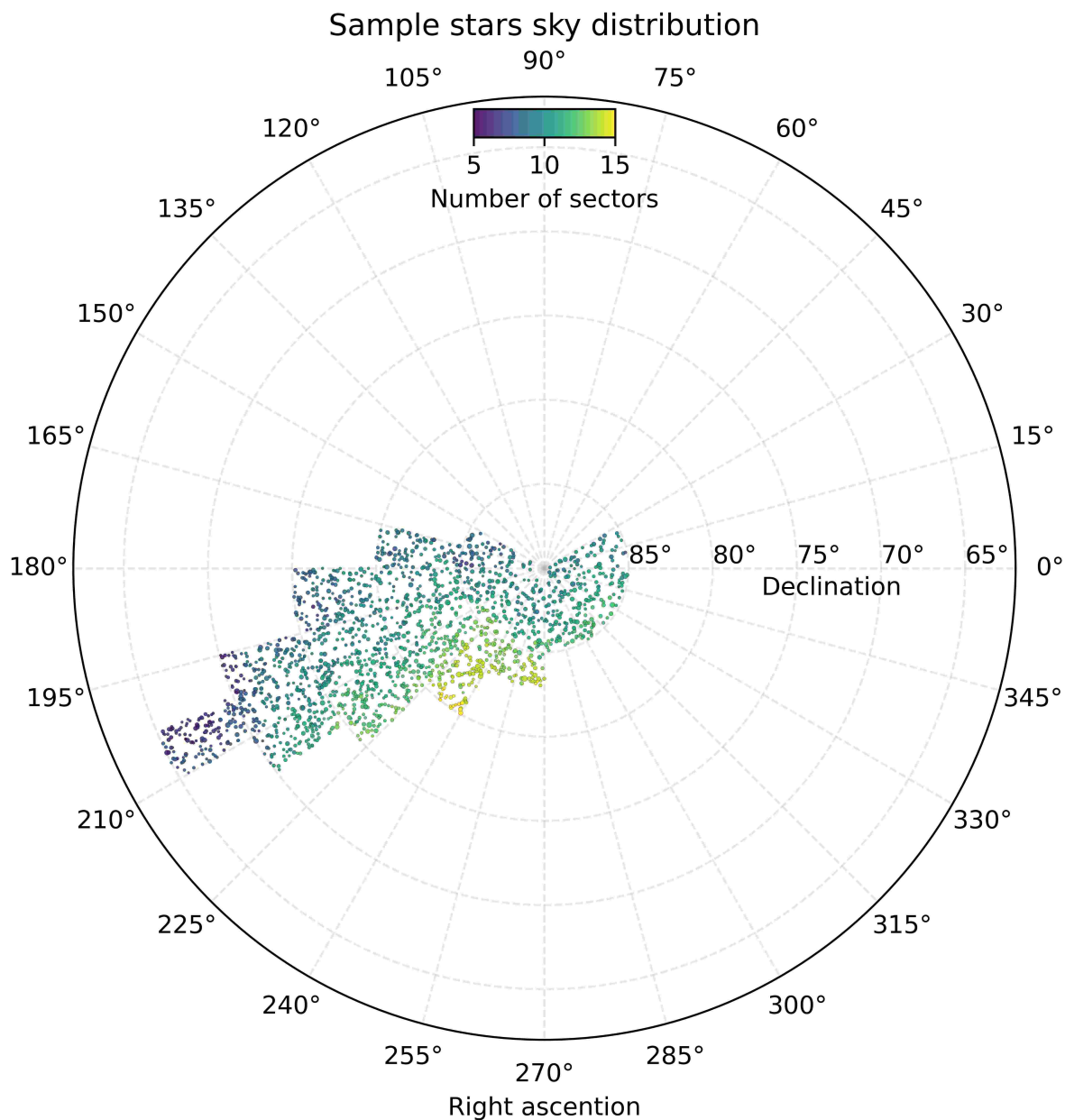
    gmag_c = gaia.loc[0, 'phot_g_mean_mag'] - gaia.loc[0, 'ag_gspphot']
    bprp = gaia.loc[0, 'phot_bp_mean_mag'] - gaia.loc[0, 'phot_rp_mean_mag']
    bprp_c = (gaia.loc[0, 'phot_bp_mean_mag'] - gaia.loc[0, 'phot_rp_mean_mag']) -
            - gaia.loc[0, 'ebpminrp_gspphot']
    absmag = (gaia.loc[0, 'phot_g_mean_mag'] - gaia.loc[0, 'ag_gspphot']) + 5 -
            - 5 * np.log10(gaia.loc[0, 'distance_gspphot'])
    lum = 10 ** ((4.63 - ((gaia.loc[0, 'phot_g_mean_mag'] - gaia.loc[0, 'ag_gspphot']) + 5 -
            - 5 * np.log10(gaia.loc[0, 'distance_gspphot'])))) / 2.5)

    df.at[i, 'rmag'] = gaia.loc[0, 'phot_rp_mean_mag']
    df.at[i, 'gmag'] = gaia.loc[0, 'phot_g_mean_mag']
    df.at[i, 'gmag_c'] = gmag_c
    df.at[i, 'bmag'] = gaia.loc[0, 'phot_bp_mean_mag']
    df.at[i, 'bprp'] = bprp
    df.at[i, 'bprp_c'] = bprp_c
    df.at[i, 'absmag'] = absmag
    df.at[i, 'teff'] = gaia.loc[0, 'teff_gspphot']
    df.at[i, 'dist'] = gaia.loc[0, 'distance_gspphot']
    df.at[i, 'lum'] = lum
    df.at[i, 'logg'] = gaia.loc[0, 'logg_gspphot']
    df.at[i, 'mh'] = gaia.loc[0, 'mh_gspphot']
    df.at[i, 'ext'] = gaia.loc[0, 'ag_gspphot']
    df.at[i, 'redd'] = gaia.loc[0, 'ebpminrp_gspphot']

df.to_csv('path', index = False)
```

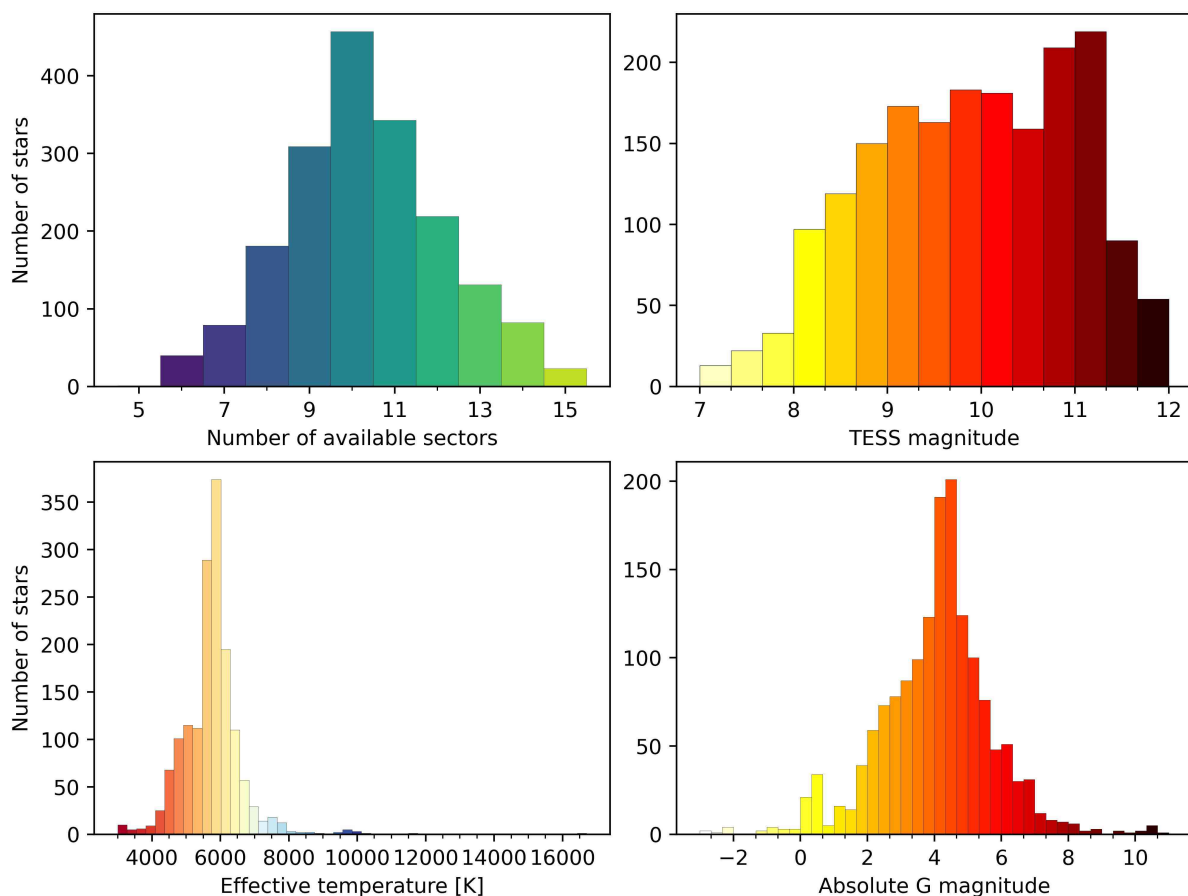
### 4.2.9 Statistika

Původní seznam 415.106 hvězd je očištěn na seznam 1865 neproměnných hvězd pro analýzu. Před odstraněním Gaia a VSX proměnných hvězd seznam obsahoval 2069 hvězd, tudíž 193 hvězd (9,3 %) byly již objevené proměnné hvězdy. 1764 hvězd má dostupnou „short“ TPF kadenci a 101 hvězd má dostupnou „fast“ TPF kadenci.



Obrázek 12: Rozmístění hvězd kolem severního nebeského pólu ve vzorku. Počet dostupných sektorů pro každou hvězdu je barevně znázorněn (vlastní tvorba).

Astrofyzikální parametry z Gaia DR3 nejsou dostupné pro všechny hvězdy. 84,2 % hvězd má dostupnou efektivní teplotu,  $\log(g)$  a hojnost železa, 83 % hvězd má dostupné parametry potřebné k výpočtu absolutní magnitudy.



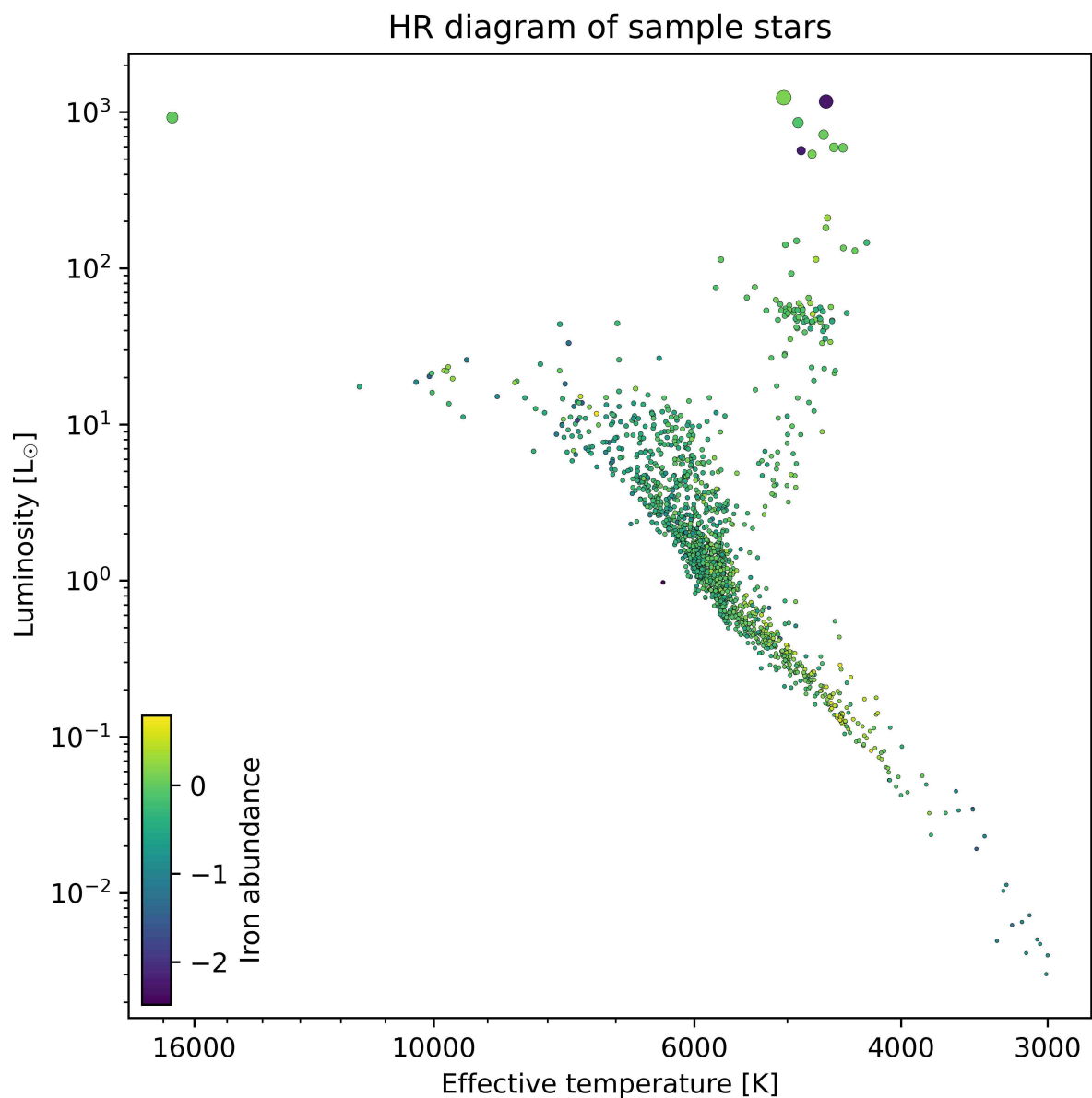
Obrázek 13: Histogramy vybraných parametrů hvězd ve vzorku. Barevná škála je pouze ilustrativní a představuje hodnoty osy x (vlastní tvorba).

Vlevo nahoře: Počet dostupných sektorů pro hvězdy ve vzorku.

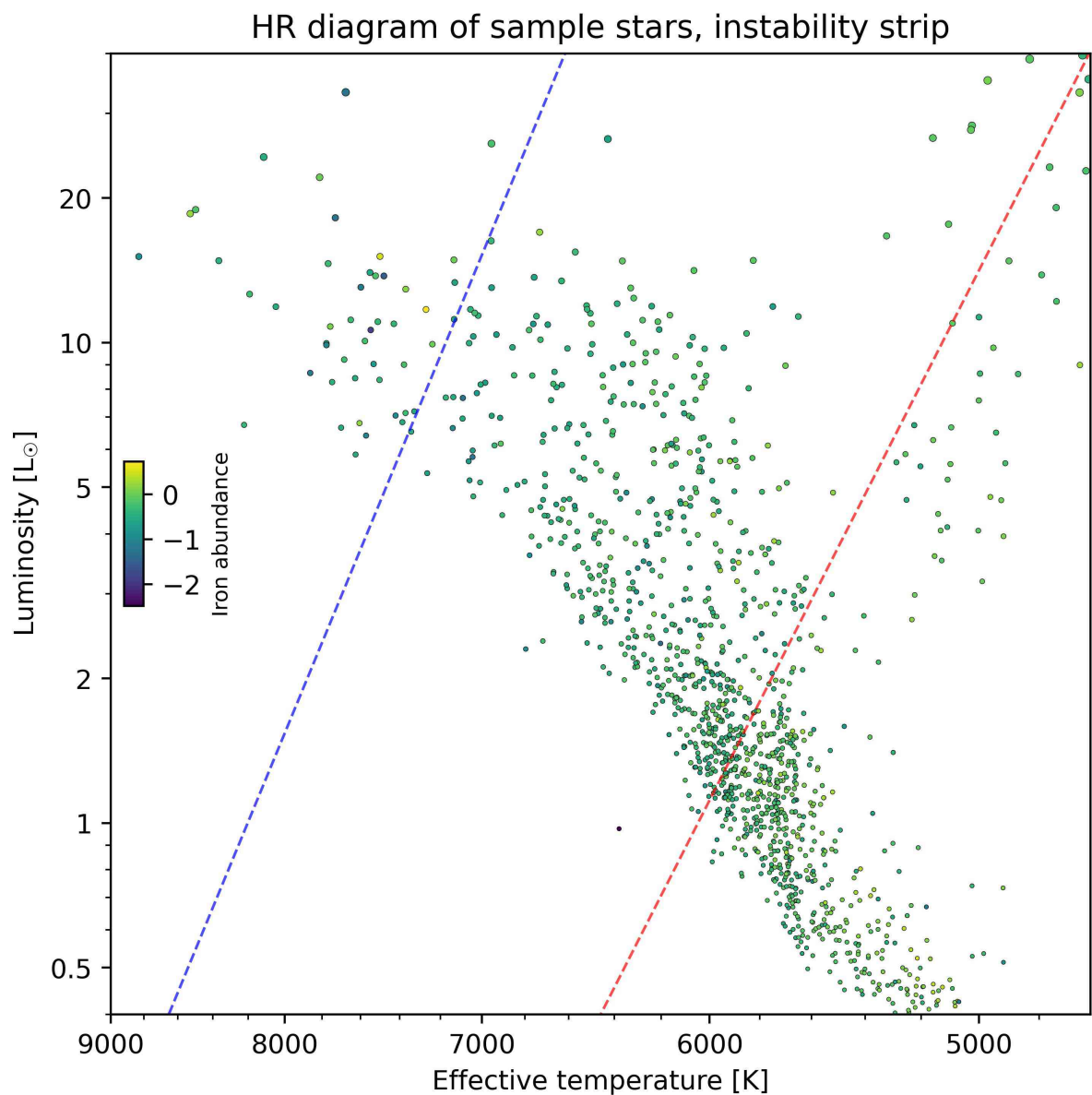
Vpravo nahoře: Zdánlivá TESS magnituda hvězd ve vzorku.

Vlevo dole: Efektivní teplota hvězd ve vzorku. Nejchladnější hvězda ve vzorku má 3000 K. Nejvyšší teplotu má hvězda S0796 s téměř 17.000 K.

Vpravo dole: Absolutní magnituda ve filtru G hvězd ve vzorku. Dvě hvězdy ve vzorku mají více než  $1000\times$  větší zářivý výkon než Slunce (hvězda spektrální třídy G2V). Několik hvězd má méně než setinu zářivého výkonu Slunce.



Obrázek 14: Hertzsprungův-Russellův diagram hvězd ve vzorku. Osy jsou v logaritmickém měřítku, barva hvězd znázorňuje hojnost železa. Závislost je vyjádřena v násobcích zářivého výkonu Slunce na efektivní teplotě. Většina hvězd se nachází na hlavní posloupnosti, nicméně několik desítek hvězd ve vzorku jsou obři či nadobři. Je přítomno také několik červených trpaslíků i hvězd s efektivní teplotou přes 10.000 kelvinů (vlastní tvorba).



Obrázek 15: Bližší pohled na hlavní posloupnost Hertzsprungova-Russellova diagramu hvězd ve vzorku. Dvěmi přímkami je zobrazen pás nestability, ve kterém se nachází většina DSCT a GDOR pulzujících hvězd (vlastní tvorba). Oblast pásu nestability převzata z [34].

## 5 Posouzení proměnnosti

Zpracování světelných křivek a následně periodogramů v této práci probíhá v rámci knihovny `lightkurve` [28], vybudovaná ve spolupráci s `numpy` [20], `scipy` [51], `matplotlib` [21], `astropy` [5], `astroquery` [17] a služby TESScut [8]. Právě knihovna `matplotlib` je použita pro veškerou vizualizaci dat, včetně řady grafů viditelných v této práci.

Díky výběru pouze hvězd s dostupnou TPF kadencí můžeme přímo stahovat a pracovat se světelnými křivkami, které byly zpracovány SPOC [22]. Tento fakt výrazně urychlí proces zpracování dat, jelikož není potřeba stahovat výřezky pro všechny sektory pro všechny hvězdy a provádět kompletní analýzu pro neproměnné hvězdy, ale pouze pro hvězdy vybrané jako potenciálně proměnné.

SPOC světelné křivky pro vybrané hvězdy jsou pouze dostupné v sektorech, ve kterých hvězda byla pozorována v rámci TPF kadence. Proměnnost některých hvězd je tudíž posouzena kupř. pouze na základě světelné křivky v jednom sektoru, i přestože hvězda byla pozorována v rámci kupř. 6 sektorů. Může se proto stát, že hvězda je chybně zařazena mezi neproměnné hvězdy, i přes to, že je v jiných sektorech proměnná. Takový případ může nastat např. u rotujících proměnných hvězd, které měly nízkou aktivitu v zobrazeném sektoru, ale v ostatních (nezobrazených) sektorech se vytvořila povrchová skvrna, která by způsobila proměnnost. Dalším případem mohou být dlouhoperiodické zákrytové systémy či exoplanety. Pro jednoduchost této práce je tento fakt ignorován, jelikož většina hvězd má dostupnout TPF kadenci ve více jak 2 sektorech. Navíc, pro malé procento zbylých hvězd s nízkým pokrytím TPF kadencí je malá pravděpodobnost, že se jedná o dlouhoperiodický zákrytový systém či exoplanetu, nebo o rotující proměnnou hvězdu, která právě v zobrazeném sektoru má nízkou či až nedetekovatelnou proměnnost.

Použité světelné křivky pro posouzení proměnnosti jsou tzv. *Pre-search Data Conditioning Simple Aperture Photometry (PDCSAP)*. Tyto světelné křivky jsou opravené o veškeré systematické chyby pomocí *Co-trending Basis Vectors (CBVs)*. Systematické chyby zahrnují rozptýlené světlo od Měsíce a Slunce, nepřesnosti v pointaci či ostření kamery. SPOC pro každou hvězdu určí ideální aperturní masku a označí kadence, které mají nízkou kvalitu, např. kvůli zásahu kosmickým zářením. Tyto kadence lze filtrovat pomocí parametru `quality_bitmask`.

Aperturní maska označuje pixely, které jsou použity pro vytvoření světelné křivky. Každá hvězda má svojí *Pixel Response Function (PRF)*, která charakterizuje jakým způsobem se hvězda na detektoru promítá kvůli difrakci a dalším optickým jevům. Vybraná aperturní maska musí maximalizovat poměr signálu a šumu (*SNR*), nicméně nezahrnuje veškeré světlo přicházející z dané hvězdy a navíc zahrnuje i část nechtěného světla okolních hvězd. SPOC pomocí PRF vypočítá poměr zahrnutého světelného toku a celkového světelného toku dané hvězdy a také poměr světelného toku dané hvězdy a světelného toku okolních hvězd. Tímto způsobem může světelnou křivku škálovat takovým způsobem, aby měla správnou amplitudu. Amplituda světelné křivky je normalizována, tedy vydělena mediánem.

I přes všechny zmíněné korekce mohou PDCSAP světelné křivky obsahovat zbytkový signál systematických chyb, nemusí mít přesně správné amplitudy a mohou odstraňovat dlouhodobé změny jasnosti hvězd. Tyto zmíněné faktory je velmi důležité mít na paměti během analýzy proměnnosti hvězd ve vzorku.

## 5.1 Parametry výběru

Jako proměnné jsou označeny všechny hvězdy, u kterých v rámci jednoho sektoru nastala změna větší jak 0,005 normalizovaného světelného toku. Dále také všechny hvězdy, které mají v periodogramu jasně definovaný peak o hodnotě větší jak 0,0005 normalizovaného světelného toku, nebo všechny hvězdy, které mají v periodogramu jasně definovanou skupinu peaků o součtu jejich hodnot větší jak 0,0005 normalizovaného světelného toku.

V případě, že hvězda nesplňuje tyto parametry, ale je nějakým způsobem klasifikovatelná, je také zahrnuta mezi proměnné hvězdy (např. přechod exoplanety může způsobit pouze velmi malé změny jasnosti s nedefinovaným peakem v periodogramu). Je důležité zmínit, že změna jasnosti na světelné křivce nezahrnuje samotný rozptyl bodů a hodnota peaku v periodogramu se počítá nad úroveň okolního šumu.

## 5.2 Skript

Pomocí kódu níže je pro každou hvězdu vygenerováno několik grafů. Pro každý sektor je generována normalizovaná světelná křivka, krátký úsek normalizované světelné křivky a fázová světelná křivka. Pro data všech sektorů je generována normalizovaná světelná křivka, fázová světelná křivka a periodogram. Generování grafů probíhá pomocí knihovny `matplotlib`, která nebude detailně popisována, jelikož nesouvisí se zaměřením práce. Grafy jednotlivých hvězd jsou rozděleny do adresářů pro následující manuální kontrolu. Dokumentace knihovny `lightkurve` je dostupná na <https://docs.lightkurve.org/reference/index.html>.

Funkce `search_lightcurve()` vyhledá dostupné světelné křivky v databázi MAST pro hvězdu s identifikátorem TIC. Parametrem `author` zvolíme světelné křivky zpracované SPOC, parametrem `exptime` zvolíme křivky s „short“ TPF kadencí, popř. „fast“ kadencí pro dostupné hvězdy.

Funkce `download_all()` stáhne všechny světelné křivky splňující zmíněné parametry. Parametru `quality_bitmask` přidělíme hodnotu `'default'`, která ignoruje kadence s závažnými chybami.

```
import pandas as pd
import lightkurve as lk
import matplotlib.pyplot as plt
import astropy.units as u

df = pd.read_csv('path')
for id, tic, mag in zip(df['id'], df['tic'], df['mag']):
    tic = f'TIC {str(tic)}'
    lc_collection = lk.search_lightcurve(tic, author = 'SPOC', exptime = 'short')
    lc_collection = lc_collection.download_all(quality_bitmask = 'default')
    for s in lc_collection.sector:
        lc = lk.search_lightcurve(tic, author = 'SPOC', sector = s, exptime = 'short').download()
```

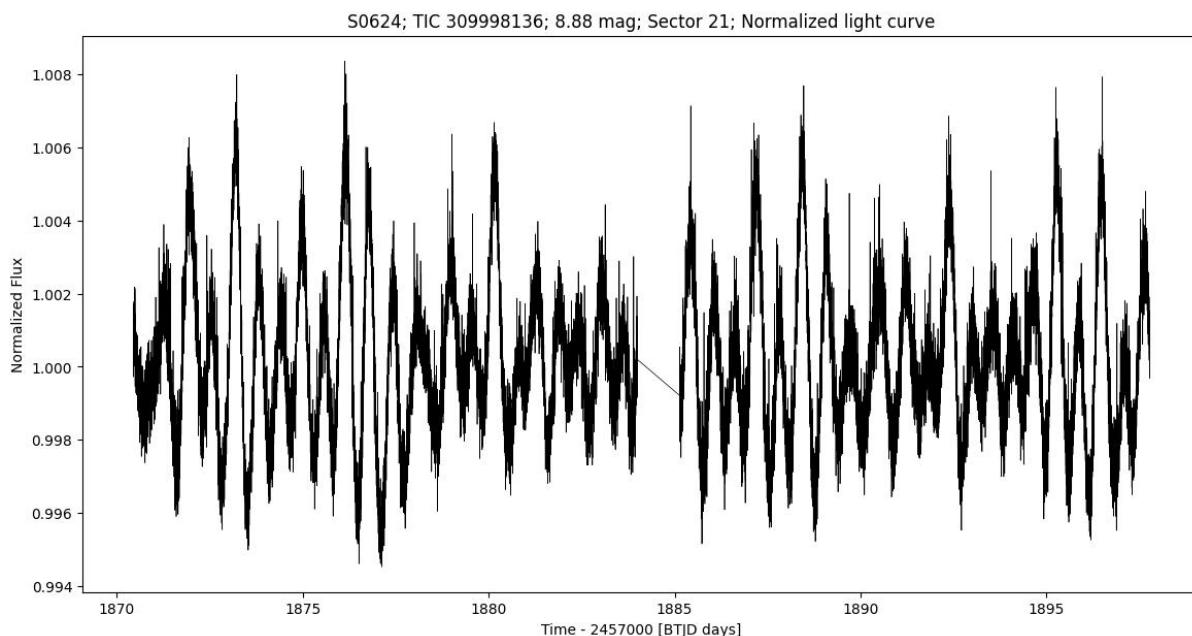
## 5.2.1 Datové produkty pro každý sektor

### 5.2.1.1 Světelná křivka

Stažená světelná křivka zpravidla obsahuje množství odlehlých hodnot, které by při vizualizaci špatně škálovaly osy grafu, proto musí být odstraněny pomocí funkce `remove_outliers()`, která velmi triviálně odstraní všechny hodnoty větší nebo menší než násobek směrodatné odchylky od mediánu. Je použita výchozí hodnota  $5\sigma$ .

Funkce `normalize()` vydělí světelnou křivku jejím mediánem. Absolutní hodnoty světelného toku jsou mezi sektory zpravidla nekonzistentní, proto je důležité používat normalizované hodnoty.

```
lc_norm = lc.remove_outliers().normalize()
```



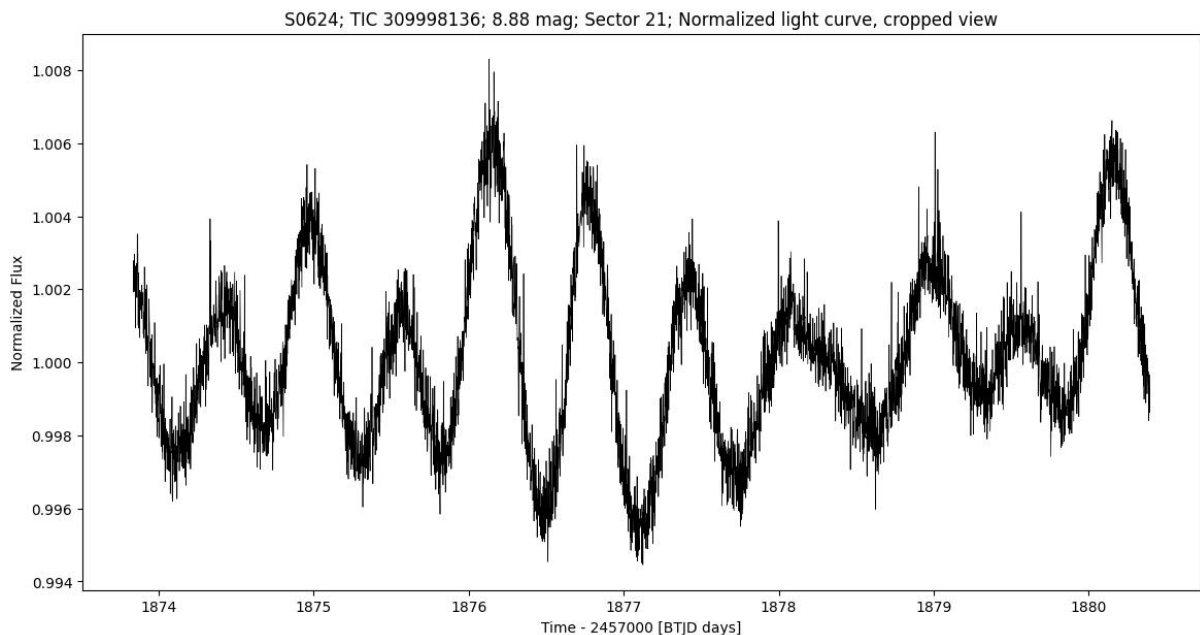
Obrázek 16: Příklad světelné křivky hvězdy S0624 v sektoru 21 (vlastní tvorba).

### 5.2.1.2 Oříznutá světelná křivka

Krátký úsek světelné křivky získáme omezením normalizované světelné křivky do intervalu  $\langle \frac{1}{8}; \frac{3}{8} \rangle$ . Zobrazením krátkého úseku světelné křivky získáme lepší pohled na krátkodobé změny jasnosti. Zvolený interval omezuje světelnou křivku kolem poloviny půlsektoru, ve které je nejméně zbytkového rozptýleného světla (tedy nejvyšší kvalita dat).

```
lc_z = lc_norm[int(len(lc_norm) / 8) : int((3 * len(lc_norm)) / 8)]
```





Obrázek 17: Příklad krátkého úseku světelné křivky hvězdy S0624 v sektoru 21 (vlastní tvorba).

### 5.2.1.3 Periodogram

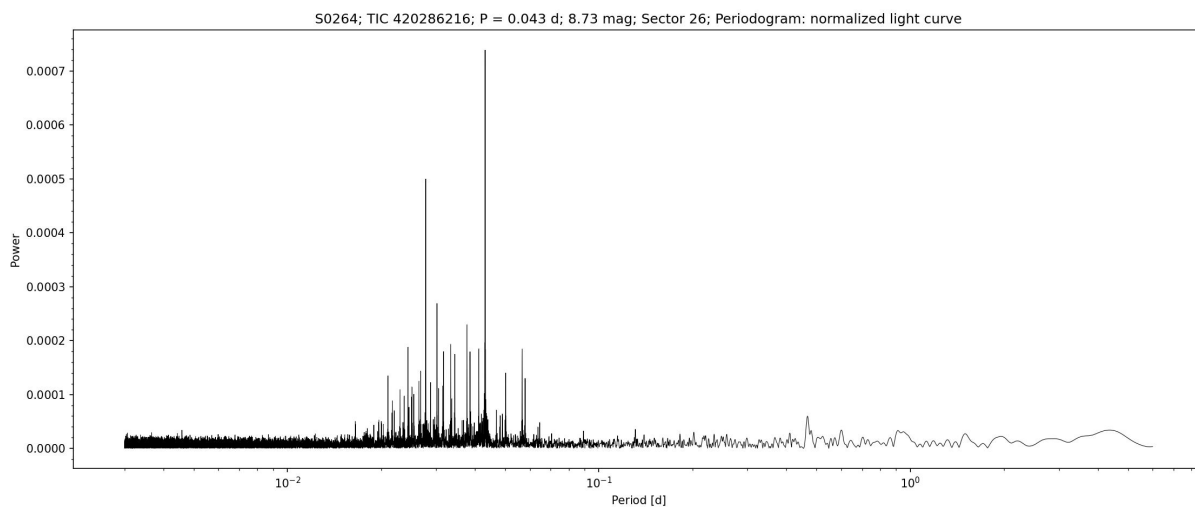
Konstrukce periodogramu probíhá pomocí funkce `to_periodogram()`. Je použit výchozí Lomb-Scargle periodogram (Lomb [29]; Scargle [43]). Rozsah analyzovaných period je zdola omezen *Nyquist frekvencí*<sup>31</sup>, shora omezen periodou 6 dní. Perioda 6 dní je vybrána jako maximální, jelikož během 27 dní pozorování jednoho sektoru má smysl uvažovat periodický signál, který je charakterizován alespoň dvěma fázemi.

Síla signálu je vypočítána pro všechny frekvence od nyquist frekvence až po horní limit frekvence, oddělené vzdáleností  $\frac{1}{\text{délka pozorování}}$ . Takový periodogram není příliš detailní a nelze z něj přímo vyčíst amplitudu. Parametr `oversample_factor` charakterizuje násobek frekvence, se kterou jsou vypočítány hodnoty síly signálu, čímž se zvyšuje rozlišení periodogramu a lze z něj vyčíst hodnoty amplitudy. Hodnota `oversample_factor` je nastavena na 10.

```
pg_norm = lc_norm.to_periodogram(maximum_period = 6 * u.day, oversample_factor = 10)
```

Periodogram pro jediný sektor se generuje pouze v případě, že daná hvězda má dostupnou TPF kadenci pouze v jediném sektoru.

<sup>31</sup>Nyquist frekvence je nejvyšší možná frekvence periodického signálu, která lze analyzovat z diskretních pozorování s konstantní frekvencí vzorkování. Platí  $f_{nyquist} = \frac{1}{2}f_{sampling}$  [49].

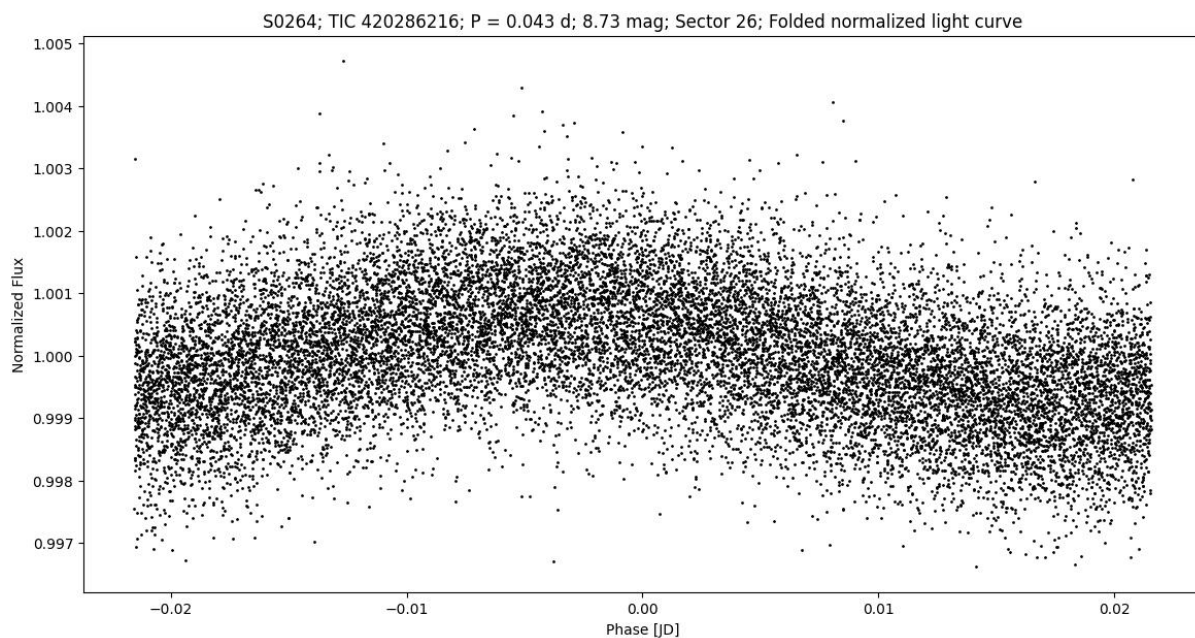


Obrázek 18: Příklad periodogramu hvězdy S0264 v sektoru 26 (vlastní tvorba).

#### 5.2.1.4 Fázová křivka

Fázovou křivku získáme složením světelné křivky podle periody s nejvyšší silou signálu pomocí funkce `fold()`.

```
phase_norm = lc_norm.fold(period = pg_norm.period_at_max_power)
```



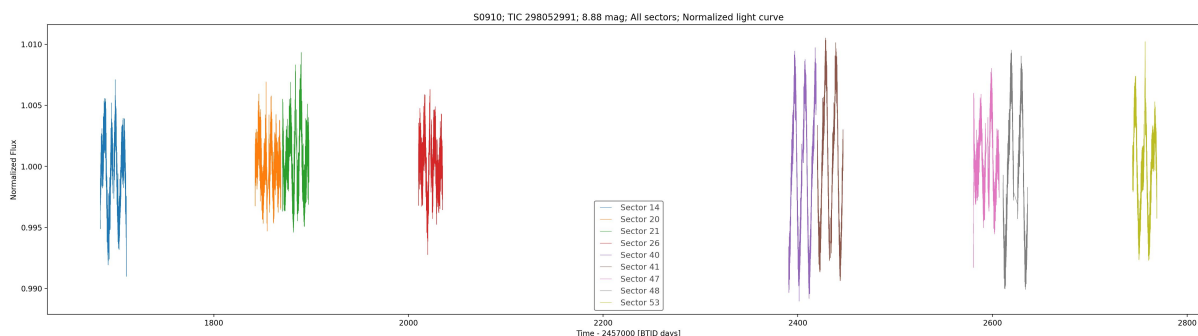
Obrázek 19: Příklad fázové křivky hvězdy S0264 v sektoru 26 (vlastní tvorba).

## 5.2.2 Datové produkty všech sektorů

### 5.2.2.1 Světelná křivka

Světelnou křivku všech sektorů vytvoříme iterativně, světelnou křivku každého sektoru samostatně normalizujeme a odstraníme odlehlé body pomocí funkcí `remove_outliers()` a `normalize()`.

```
if len(search) > 1:
    for lc, s in zip(lc_collection, available_sectors):
        lc = lc.remove_outliers().normalize().plot()
```



Obrázek 20: Příklad světelné křivky hvězdy S0910 v celkově 9 sektorech (vlastní tvorba).

### 5.2.2.2 Periodogram

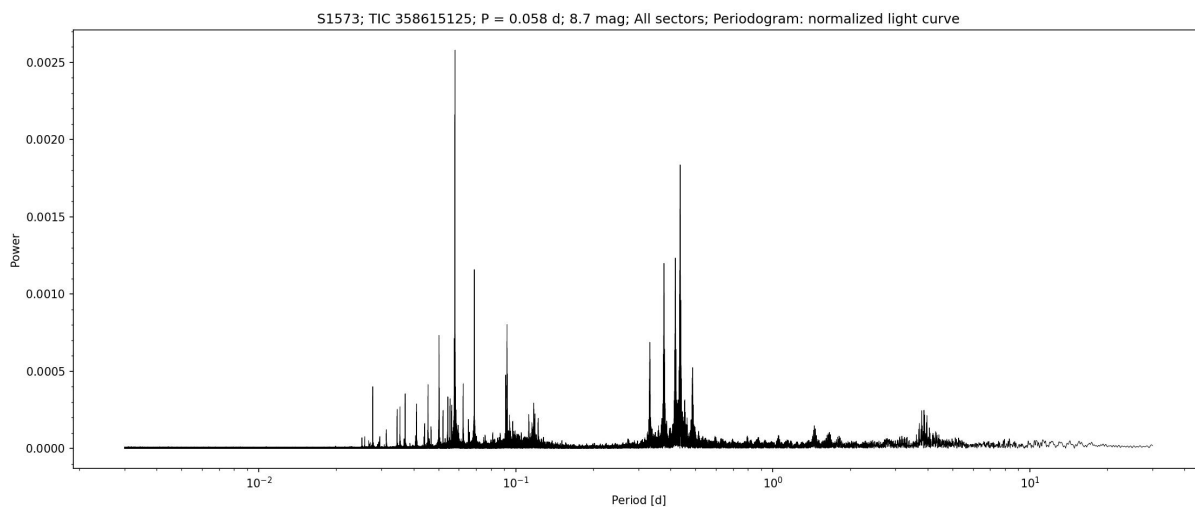
K vytvoření periodogramu ze světelné křivky všech sektorů musíme nejprve jednotlivé světelné křivky sektorů spojit do jednoho `lk.LightCurve` datatypu pomocí funkce `stitch()`. Parametrem `corrector_func` a `lambda` funkcí aplikujeme funkce `remove_outliers()` a `normalize()` na světelnou křivku každého sektoru.

```
lc_all_norm = lc_collection.stitch(corrector_func = lambda lc: lc.remove_outliers().normalize())
```

Tentokrát horní limit je zvýšen na 30 dní, jelikož světelná křivka několika sektorů má lepší časové pokrytí.

```
pg_all_norm = lc_all_norm.to_periodogram(maximum_period = 30 * u.day, oversample_factor = 10)
```

S rostoucím časovým pokrytím a kadencí také roste SNR periodogramu, tedy periodogram světelné křivky všech sektorů vypovídá lépe o proměnnosti dané hvězdy.

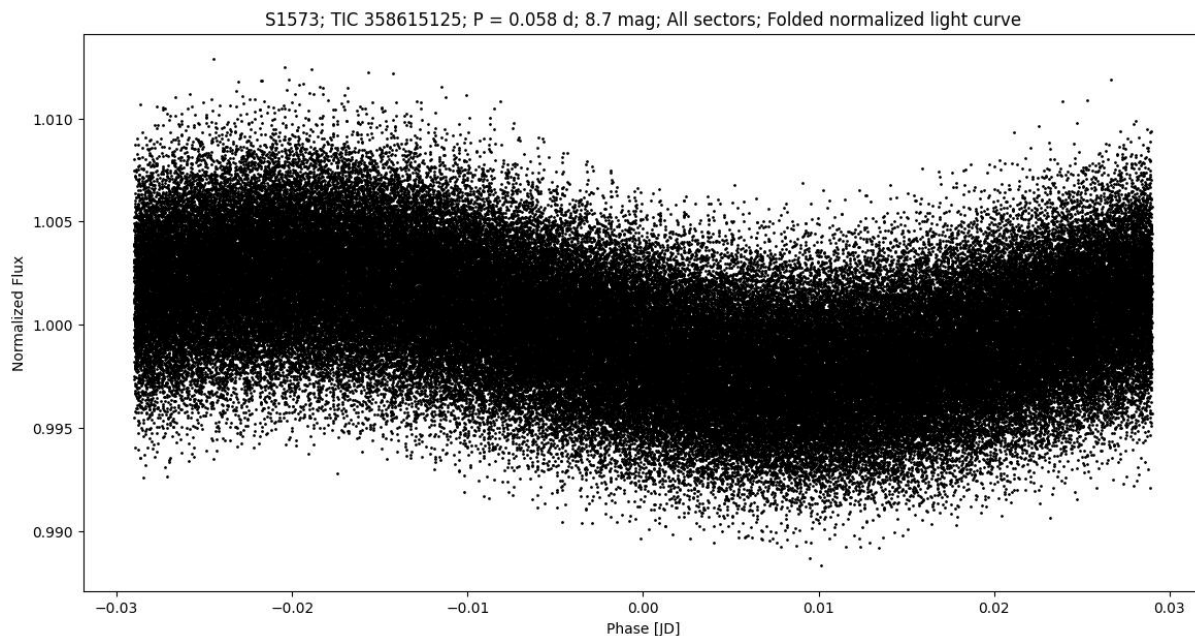


Obrázek 21: Příklad periodogramu hvězdy S1573 ze světelné křivky celkově 8 sektorů (vlastní tvorba).

### 5.2.2.3 Fázová křivka

Fázovou křivku získáme složením světelné křivky všech sektorů podle periody s nejvyšší silou signálu pomocí funkce `fold()`.

```
phase_all_norm = lc_all_norm.fold(period = pg_all_norm.period_at_max_power)
```



Obrázek 22: Příklad fázové křivky hvězdy S1573 ze světelné křivky celkově 8 sektorů (vlastní tvorba).

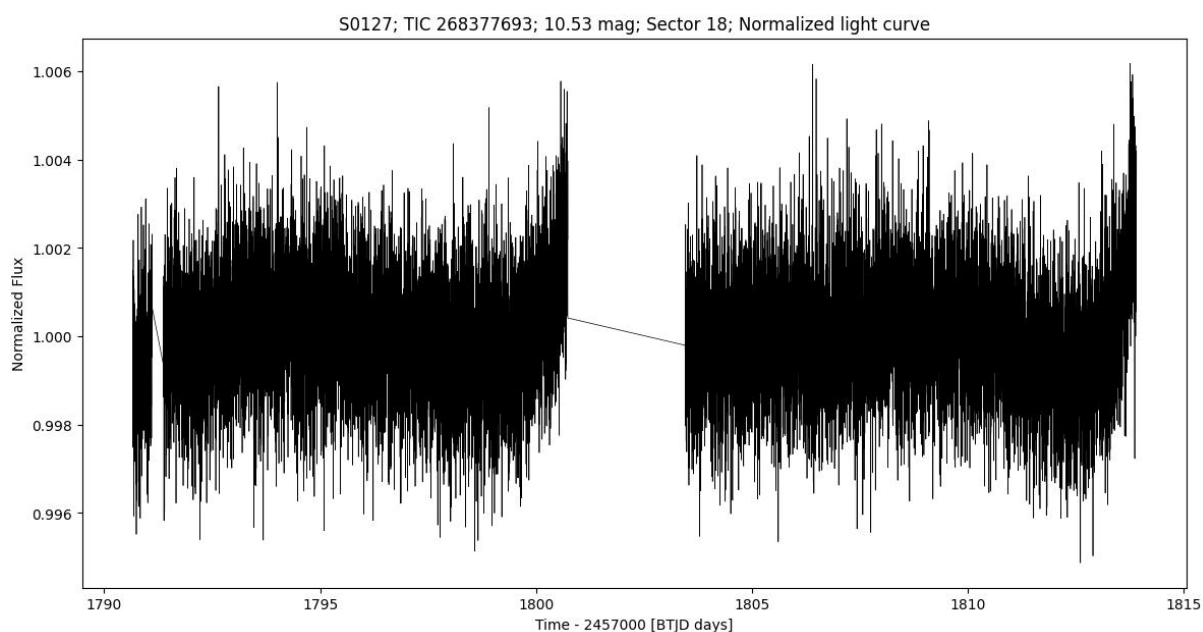
## 5.3 Vyskytlé problémy

### 5.3.1 Nestažené hvězdy

16 hvězd z celkových 1865, tedy méně jak 1 procento, nemá dostupnou světelnou křivku v TPF kadenci. Tyto hvězdy byly pravděpodobně zapsány do *Target List* seznamu, ale z určitého důvodu nemohly být pozorovány. Příčina není blíže zkoumána, tudíž hvězdy jsou odstraněny z posouzení proměnnosti. Jedná se o hvězdy S0049, S0051, S0053, S0100, S0269, S0302, S0401, S0889, S0912, S0988, S1253, S1263, S1270, S1483, S1486, S1626.

### 5.3.2 Zbytkové systematické chyby

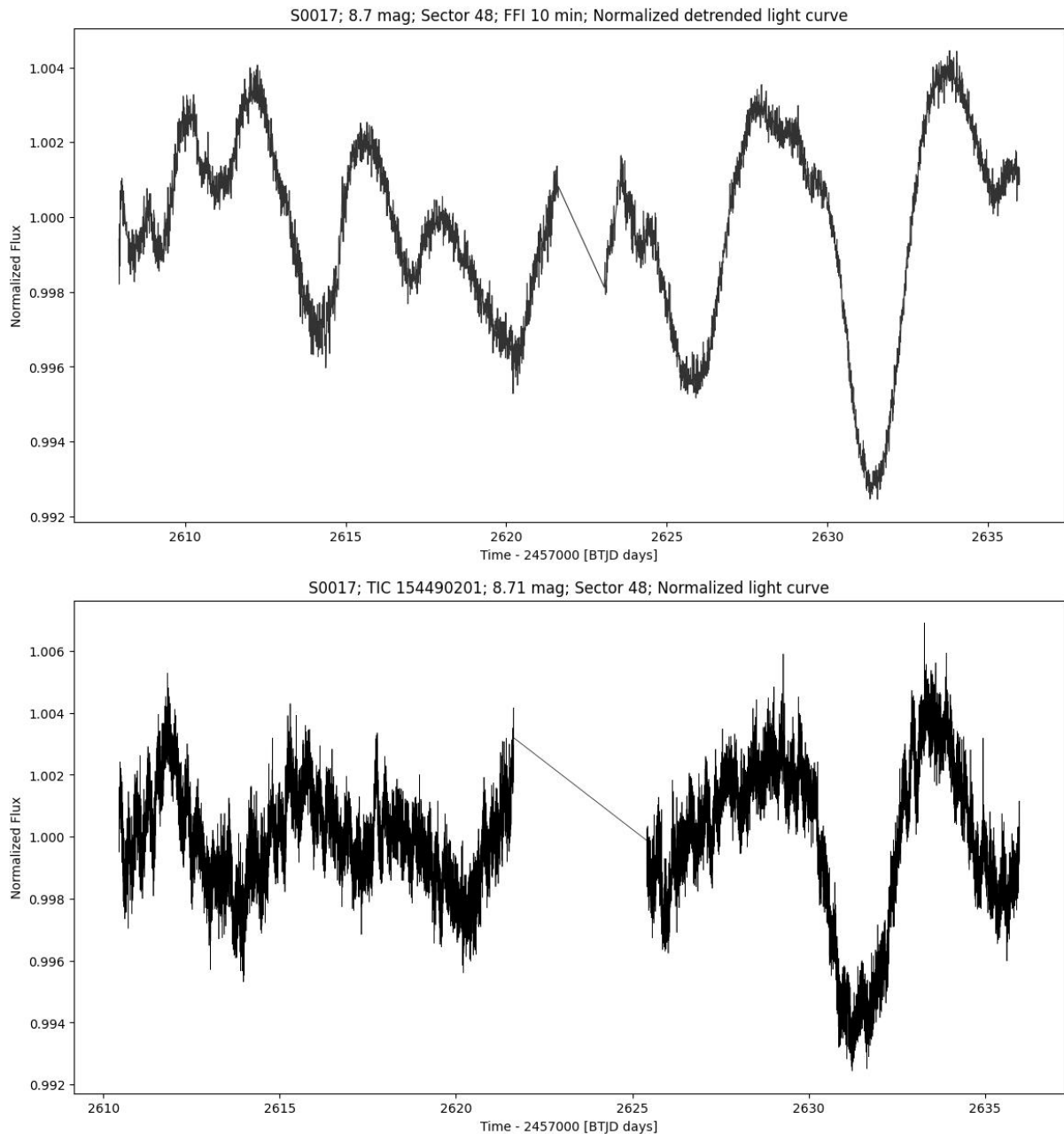
Některé světelné křivky mají nedokonale odstraněný světelný tok způsobený rozptýleným světlem Země a Měsíce. Největší množství rozptýleného světla je zpravidla na začátku, kolem středu a na konci sektoru. Nedostatečná korekce je přítomna u jednotek procent pozorovaných sektorů, zpravidla je ale zbytkový světelný tok malý (v řádech tisícín celkového toku) a postihuje v rámci sektoru 1 – 4 dny pozorování z celkových  $\sim 27$  dní. Zbytkové rozptýlené světlo má na posouzení proměnnosti zanedbatelný vliv, tudíž není potřeba provádět žádnou korekci.



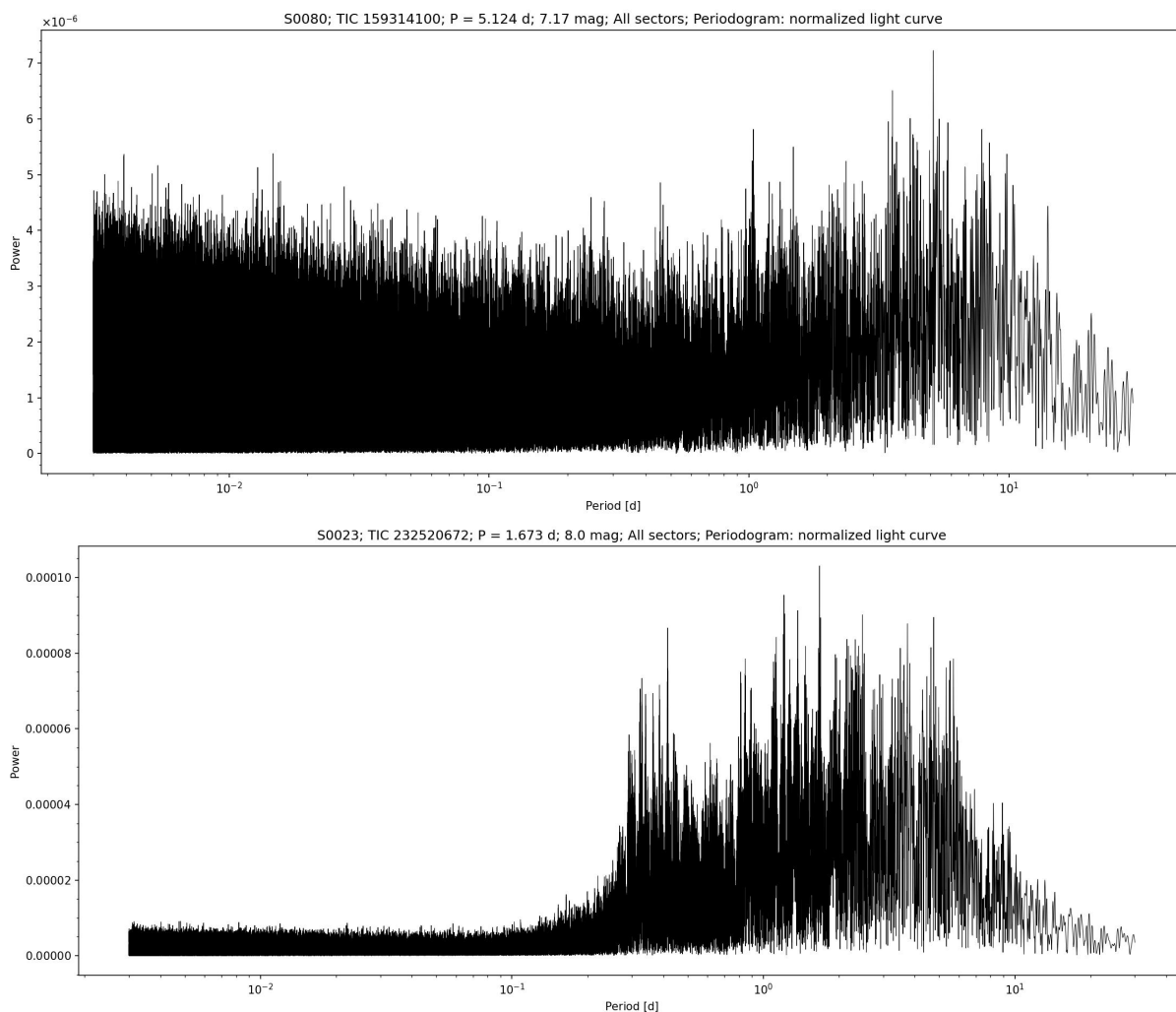
Obrázek 23: Příklad zbytkového rozptýleného světla ve světelné křivce hvězdy S0127 v sektoru 18. V tomto případě lze vidět nadbytečný světelný tok na konci každého půlsektoru, s celkovou dobou poškozených pozorování  $\sim 4$  dny a maximální amplitudou 2 tisíciny celkového světelného toku (vlastní tvorba).

Významným problémem PDCSAP světelných křivek je avšak sporadický šum, který nejspíše vzniká při odstraňování systematických chyb pomocí CBV a analýzy hlavních komponent. Tento problém je viditelný na poměrně velkém množství světelných křivek a má amplitudu od několika desetitisícín až tisícín celkového světelného toku. Zpravidla

neovlivňuje amplitudu změn samotné hvězdy a tudíž nemá významný vliv na posouzení proměnnosti, přesto práce s PDCSAP světelnými křivkami bude v dalších krocích co nejvíce omezena.



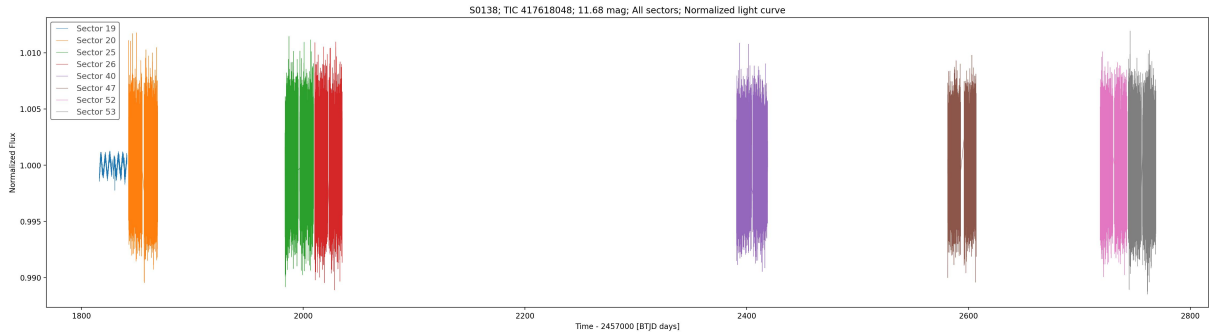
Obrázek 24: Příklad světelné křivky vytvořené manuálně z FFI dat (nahore) a PDCSAP světelné křivky stáhnuté z databáze MAST (dole) hvězdy S0017 v sektoru 48. Amplituda největšího poklesu kolem 2632d je podobná v obou světelných křivkách, nicméně světelná křivka SPOC výrazně potlačuje změny světelného toku způsobené hvězdou a přidává šum s periodou  $\sim 0,5$  dne. Zpracování světelných křivek z FFI dat bude diskutováno ve zpracování dat (vlastní tvorba).



Obrázek 25: Příklad periodogramů dvou neproměnných hvězd, S0080 (nahore) a S0023 (dole). Hvězda S0023 má poškozenou světelnou křivku šumem ze zpracování SPOC (vlastní tvorba).

### 5.3.3 Nekonzistentní světelné křivky

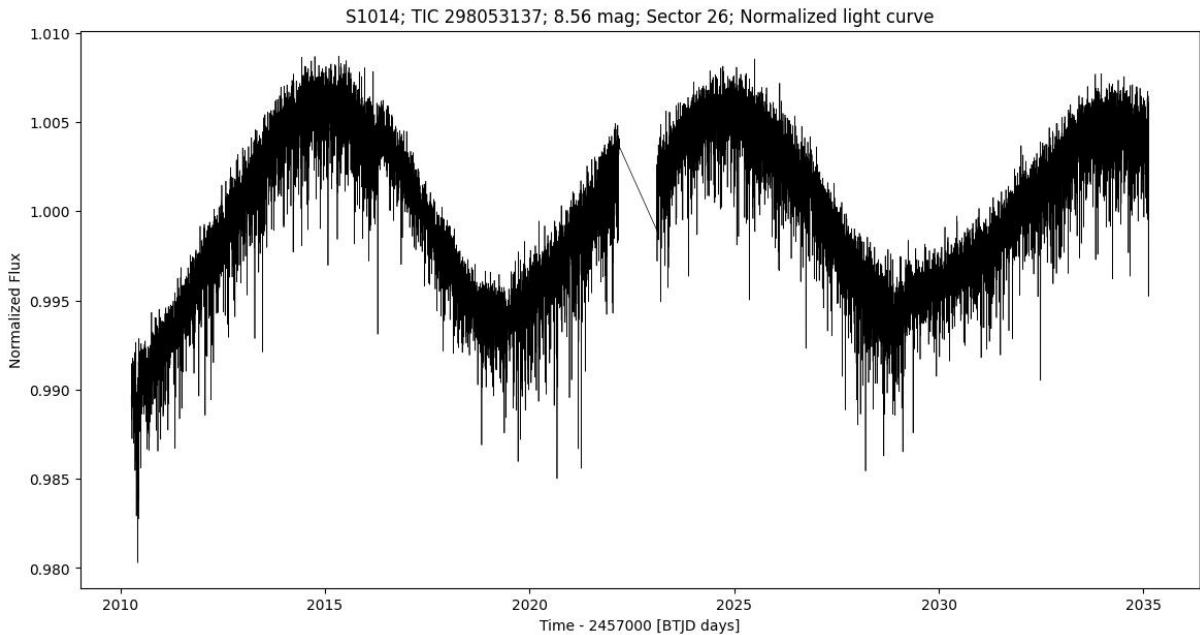
Několik hvězd má evidentně zaměněné, nebo nějakým jiným způsobem navzájem nevhodící se světelné křivky. Zpravidla je hvězda v jednom sektoru proměnná a v následujícím sektoru jakákoliv proměnnost vymizí. Rozptýl měření v sektoru s proměnností je mnohem menší a nekonzistentní s pozorováními okolních sektorů. Jelikož tento problém postihl pouze několik hvězd, není mu věnována větší pozornost. Je důležité zmínit, že se do této kategorie rozhodně nezařazují hvězdy, které mají významnou modulaci amplitudy (např. kvůli mizejícím povrchovým skvrnám), ale pouze hvězdy, které mají viditelně nekonzistentní průběh proměnnosti v rámci několika sektorů.



Obrázek 26: Příklad nekonzistentní světelné křivky hvězdy S0138. Světelná křivka sektoru 19 se viditelně nehodí mezi světelné křivky ostatních sektorů. Rozptyl dat světelné křivky sektoru 19 je výrazně menší než ostatních sektorů, což je situace, která by za normálních podmínek neměla nastat (vlastní tvorba).

### 5.3.4 Světelné křivky s atypickým rozptylem šumu

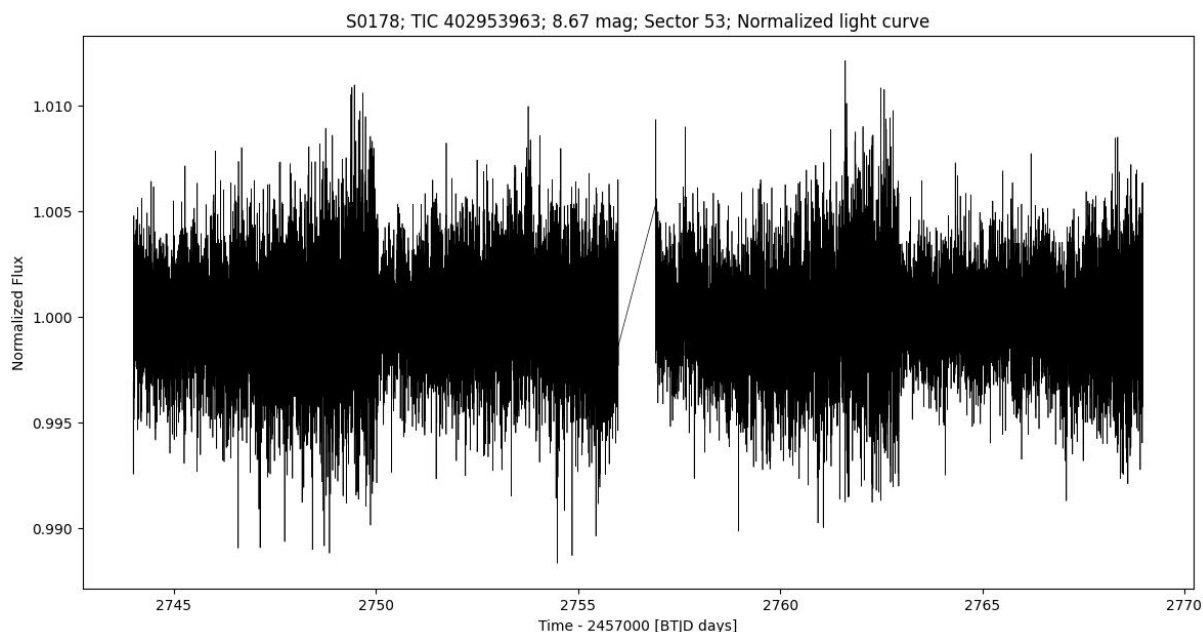
Při stahování světelných křivek byl parametr `quality_bitmask` nastaven na `'default'`, tudíž světelné křivky zahrnují i méně kvalitní kadence a tím i více odlehlých bodů. Některé světelné křivky jsou avšak výrazně poškozené a mohli by se popsat jakýmsi „bleed“ efektem. Jedná se opět pouze o několik hvězd a pokud to kvalita pozorování v ostatních sektorech dovoluje, jsou zařazené mezi proměnné. Je možné, že při nastavení parametru `quality_bitmask` na `'hard'` by byly nekvalitní kadence vyřazeny. Nastavení `'default'` je ponecháno, jelikož je při prvotní analýze potřeba mít maximální počet viditelných kadencí, aby nebylo parametrem odstraněn např. jediný zákryt exoplanetou, jediná erupce či jediný zákryt dlouhoperiodického zákrytového systému, tudíž by zamezil odhalení proměnnosti.



Obrázek 27: Příklad „bleed“ světelné křivky hvězdy S1014 v sektoru 26. I přes nekvalitní světelnou křivku v sektoru 26 je hvězda zařazená mezi proměnné (vlastní tvorba).



Nekvalitní světelné křivky v některých případech připomínali jakési „rázy“, tedy rozptyl světelné křivky se pravidelně zvyšoval a poté se navracel na původní hodnotu. Opět, pokud kvalita pozorování v ostatních sektorech tomu dovoluje, hvězda je zařazena mezi proměnné.



Obrázek 28: Příklad „rázové“ světelné křivky hvězdy S0178 v sektoru 53 (vlastní tvorba).

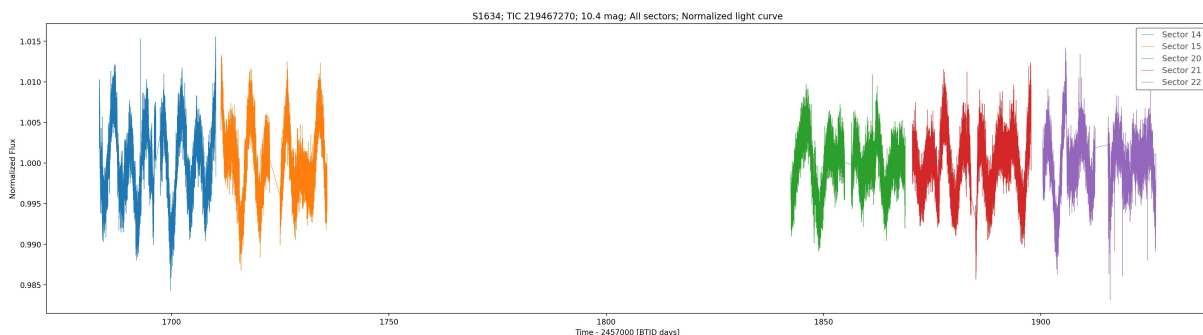
### 5.3.5 Skoky jasnosti

Nejspíše jeden z nejrozšířenějších problémů postihující významnou část hvězd ve vzorku jsou náhlé „skoky“ světelného toku zapříčiněné systematickými chybami.

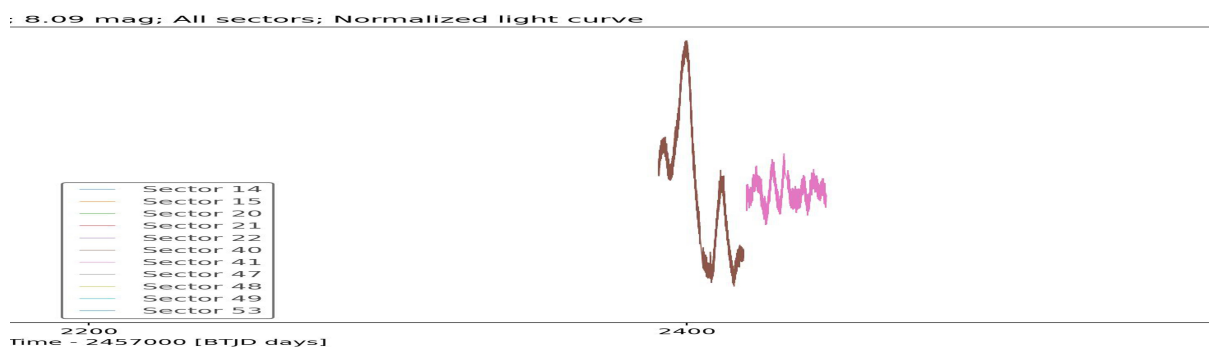
#### 5.3.5.1 Mezisektorové skoky jasnosti

Světelná křivka každého sektoru je normalizována samostatně. Pro hvězdy s periodami změn méně jak  $\sim 5$  dní tento fakt nepředstavuje problém, jelikož světelná křivka jednoho sektoru obsahuje několik fází, tudíž při normalizaci každého sektoru samostatně světelné křivky na sebe dobře navazují. Nicméně, hvězdy s delšími periodami, zpravidla nad 10 dní, mají v rámci jediného sektoru např. pouze 1,5 fáze, tudíž návaznost na světelnou křivku dalšího sektoru závisí na části fáze obsažené ve světelné křivce daného sektoru. Jednoduše řečeno, pokud v jednom sektoru bylo zachyceno maximum hvězdy a v navazujícím sektoru bylo zachyceno minimum hvězdy, při samostatné normalizaci jednotlivých sektorů na sebe světelná křivka nenavazuje a mezi sektory je jakýsi „skok“ světelného toku.

Tento problém neovlivnil posouzení proměnnosti hvězd, nicméně by bylo vhodné jej vyřešit při úplné analýze proměnnosti.



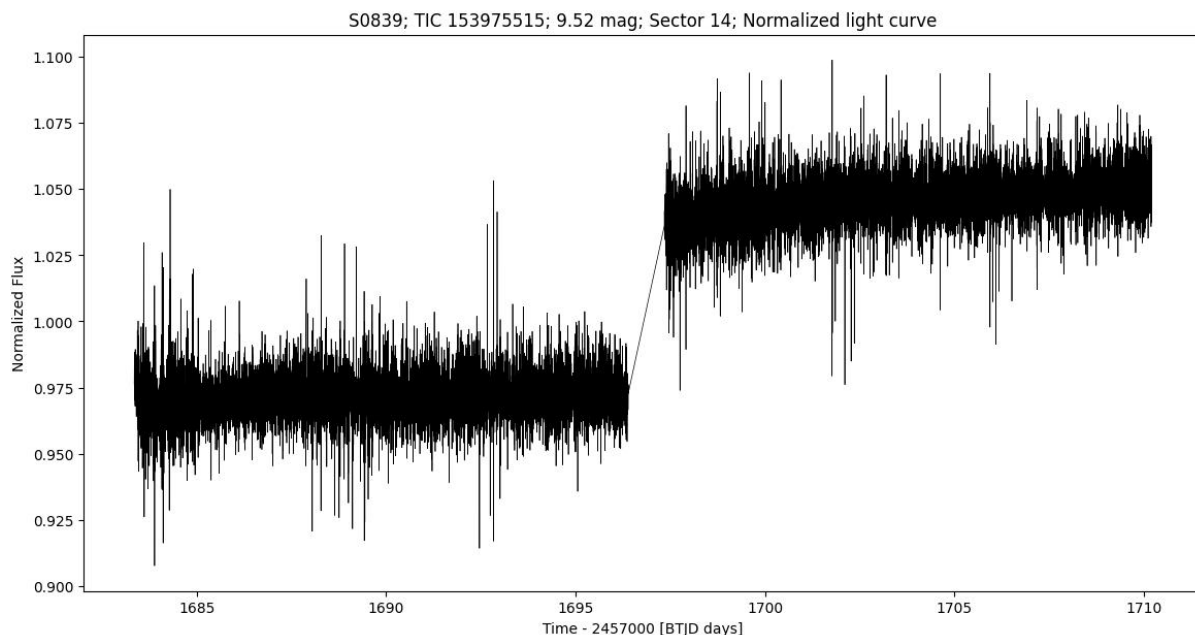
Obrázek 29: Příklad světelné křivky hvězdy S1634 s periodou proměnnosti  $\sim 4$  dny. Perioda je dostatečně krátká na to, aby fáze byla „zprůměrovaná“ a tím nenastávaly mezisketorové „skoky“ světelného toku (vlastní tvorba).



Obrázek 30: Příklad světelné křivky hvězdy S0069 s periodou proměnnosti  $\sim 10$  dní. Lze vidět špatná návaznost světelných křivek sektorů 40 a 41. Graf upraven pro lepší viditelnost (vlastní tvorba).

### 5.3.5.2 Skoky jasnosti uprostřed sektoru

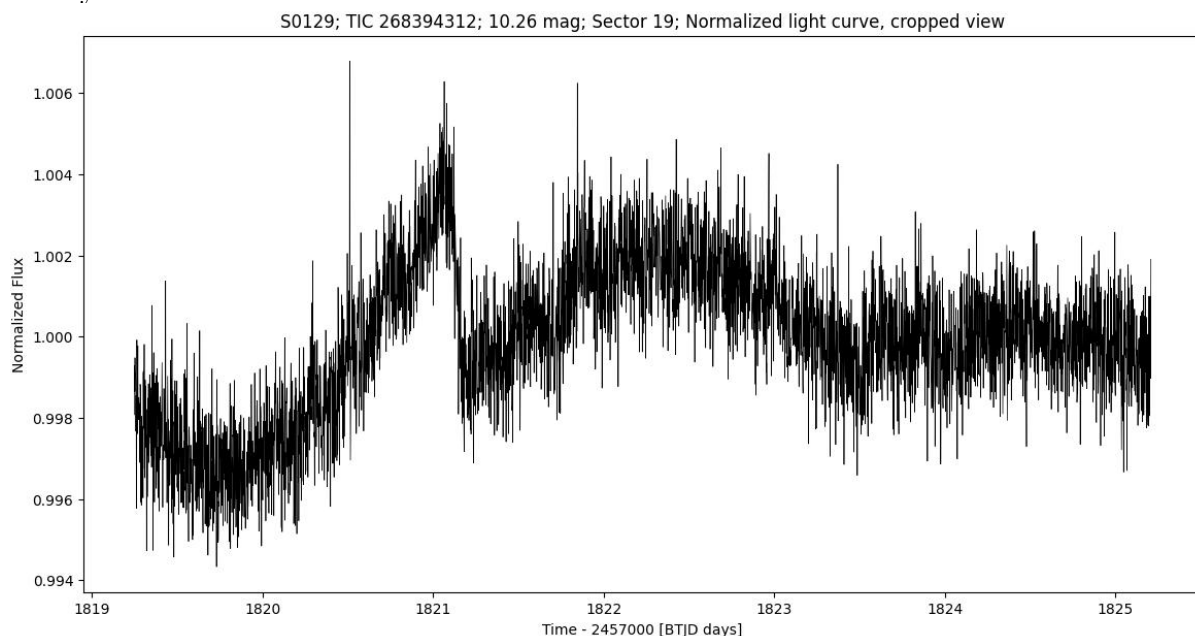
Při přeorientování TESS v polovině každého sektoru se musí satelit opět namířit na přesně stejné místo. Zpravidla není namíření dostatečně přesné, tudíž se hvězdy v polovině sektoru prudce posunou. Aperturní maska avšak zůstává stejná a množství světelného toku se tím pádem změní. Při normalizaci celého sektoru poté vzniká „skok“ jasnosti. Tento problém je zpravidla dobře vyřešen pro většinu hvězd pomocí CBVs v rámci zpracování světelných křivek SPOC, nicméně u některých hvězd přetrvává. „Skok“ jasnosti je zpravidla v řádech tisícín celkového světelného toku, tudíž nemá výrazný vliv na posouzení proměnnosti.



Obrázek 31: Extrémní příklad „skoku“ světelného toku uprostřed sektoru 14 hvězdy S0839 (vlastní tvorba).

### 5.3.5.3 Skoky jasnosti uvnitř sektoru

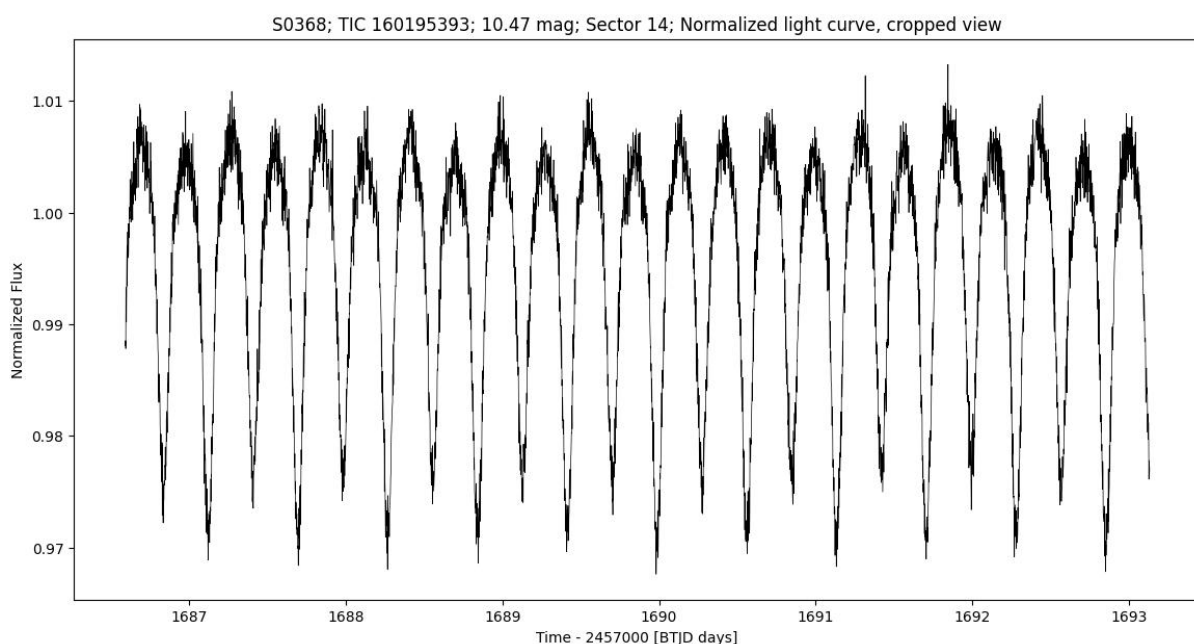
Jedná se o nejrozšířenější druh „skoku“ jasnosti s nejrozsáhlejším dopadem na posouzení proměnnosti. Nastává zpravidla uprostřed každého půlsektoru, tedy v  $\frac{1}{4}$  a  $\frac{3}{4}$  světelné křivky sektoru. Zpravidla je v řádech tisícín celkového světelného toku. V některých případech může připomínat zákryty exoplanetou. Je nejspíše zapříčiněn nedostatečnou korekcí systematických chyb SPOC světelných křivek. V případě, že „skok“ jasnosti snížil amplitudu proměnnosti hvězdy, byla amplituda „skoku“ přičtena k amplitudě proměnnosti hvězdy.



Obrázek 32: „Skoky“ jasnosti v prvním půlsektoru hvězdy S0129 s amplitudou  $\sim 4$  tisíciny a  $\sim 0,5$  tisícín celkového světelného toku hvězdy v sektoru 19. První „skok“ jasnosti lze vidět kolem 1821d, druhý kolem 1823,5d (vlastní tvorba).

### 5.3.6 Blending

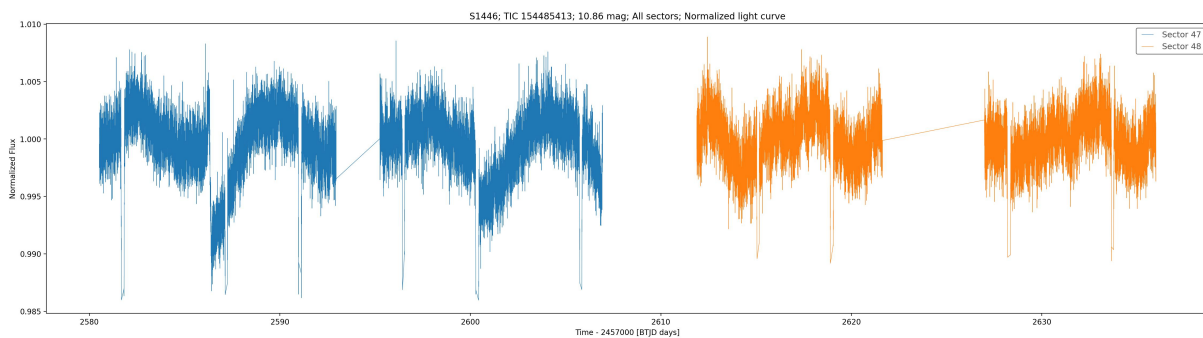
Jedním z limitujících faktorů analýzy proměnnosti hvězd z TESS je tzv. „blending“. Velikost TESS pixelů na obloze je  $\sim 21$  úhlových vteřin, navíc PRF hvězdy má několik pixelů v průměru. S klesající magnitudou roste velikost PRF a tím i rozsah světelného toku hvězdy, např. hvězda magnitudy 10 svým světelným tokem sahá více jak 5 pixelů daleko, tedy  $\sim 1,5$  úhlové minuty. Detailní analýza proměnnosti v okolí hvězd zařazených mezi proměnné bude v rámci zpracování dat. U hvězd podezřelých z kontaminace je okolí dané hvězdy manuálně prozkoumáno v katalogu VSX, který ve většině případech odhalil blízkou proměnnou hvězdu. Kupř. světelná křivka neproměnné hvězdy kontaminovaná blízkou zákrytovou dvojhvězdou s oddělenými složkami může připomínat zákryty exoplanetou. Kontaminované hvězdy jsou vyřazeny z posouzení proměnnosti.



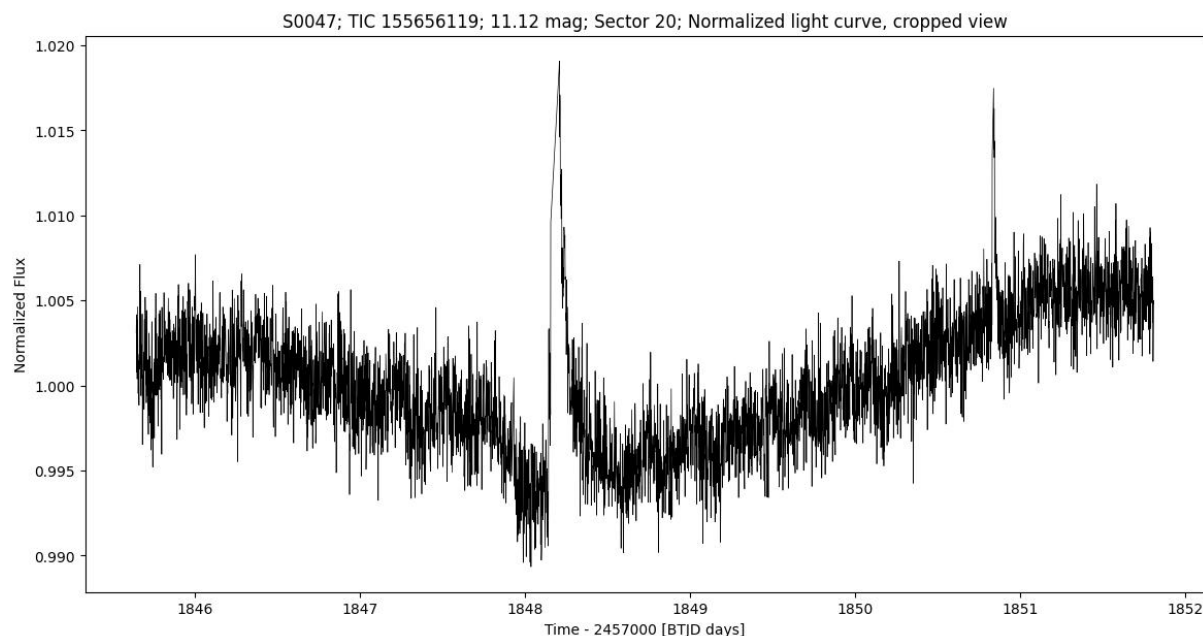
Obrázek 33: Úsek světelné křivky hvězdy S0368 v sektoru 14. Hvězda je kontaminovaná 41 úhlových vteřin ( $\sim 2$  TESS pixely) vzdálenou zákrytovou dvojhvězdou 12. magnitudy s názvem ASASSN-V J171505.23+851208.0 (vlastní tvorba).

### 5.3.7 Odstranění odlehlých bodů

Vedlejším efektem funkce `remove_outliers` je odstranění jakýchkoliv změn jasnosti nad  $5\sigma$ , včetně reálných změn jasnosti hvězdy. Zejména náchylné jsou hvězdy s rychlými změnami jasnosti v rámci krátkého časového úseku. To se týká zejména zákrytových dvojhvězd s oddělenými složkami a eruptivních hvězd. Tyto hvězdy mají „useknuté“ světelné křivky a celé zákryty, popř. erupce nejsou viditelné, nicméně z okolních zachovalých částí je patrné, že se jedná o zákryt, popř. erupci, tudíž posouzení proměnnosti není tímto problémem významně ovlivněno.



Obrázek 34: Příklad „useknuté“ světelné křivky zákrytové dvojhvězdy S1446. Krátké „mezery“ ve světelné křivce jsou okamžiky minima, lze vidět několik „neuříznutých“ kadencí na začátku a konci minima naznačující průběh zákrytu (vlastní tvorba).



Obrázek 35: Příklad „uřízlých“ erupcí viditelných na úseku světelné křivky eruptivní hvězdy S0047 v sektoru 20. Maximum erupce je odstraněno, nicméně pozvolný pokles na původní jasnost je jednoznačným indikátorem erupce (vlastní tvorba).

### 5.3.8 Objevené proměnné hvězdy

Několik hvězd je vyřazeno z navazující analýzy proměnnosti, jelikož se nachází v databázi ExoFOP<sup>32</sup> obsahující exoplanetární kandidáty, nebo v online katalogu *TESS Eclipsing Binaries*<sup>33</sup>, vytvořeného v rámci práce Prša et. al (2022) [39], obsahující 4584 zákrytových systémů pozorovaných družicí TESS.

<sup>32</sup><https://exofop.ipac.caltech.edu/tess/>

<sup>33</sup><http://tessebs.villanova.edu/>

#### **5.3.8.1 TESS Eclipsing Binaries katalog**

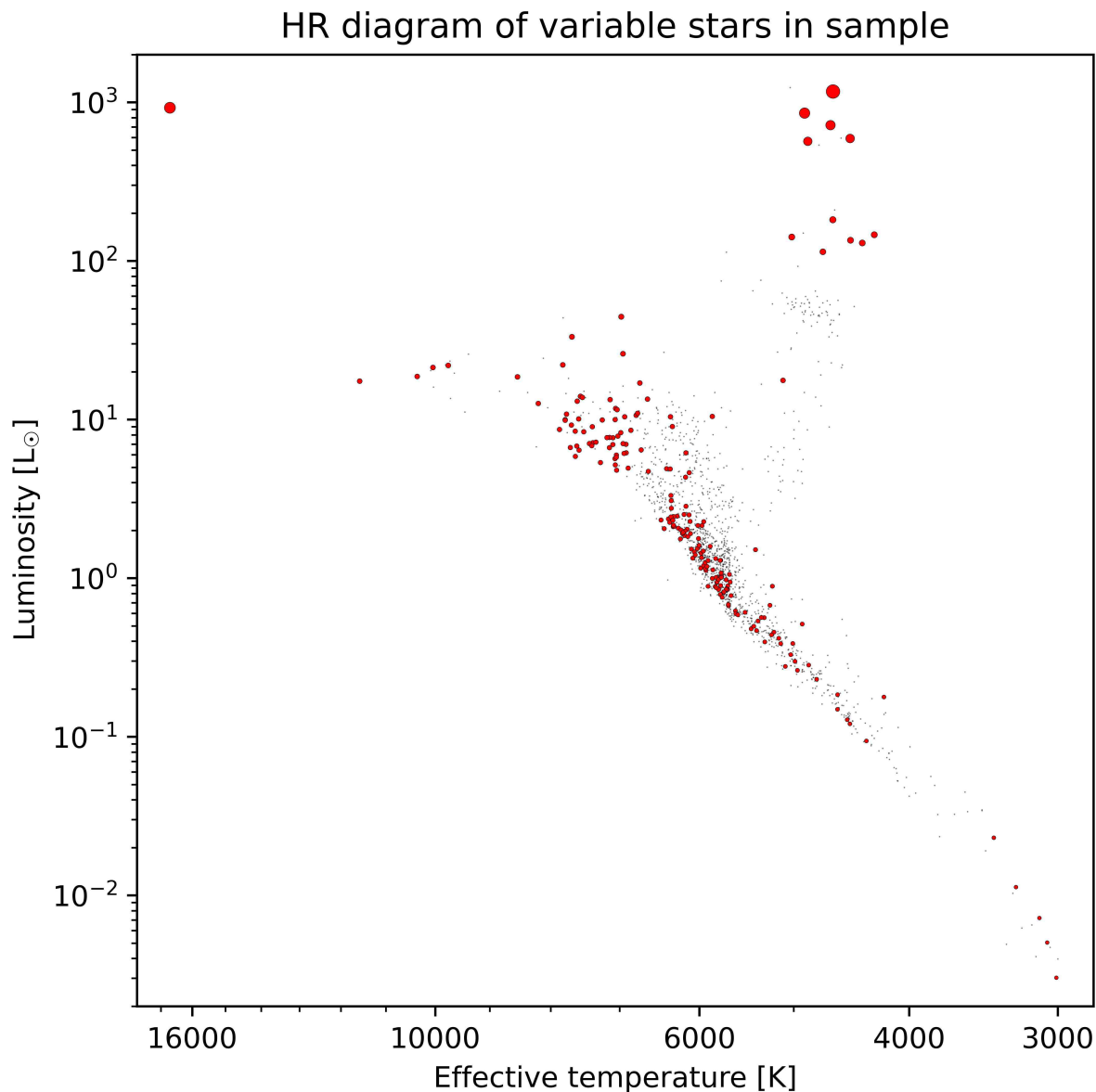
V rámci posouzení proměnnosti je nalezeno celkově pět zákrytových dvojhvězd, z toho jsou tři dostupné v TESS EBs katalogu: S0133, S0357, S1556. Hvězda S0133 je ponechána k navazující analýze z důvodu objevených změn v O – C diagramu, hvězdy S0357 a S1556 jsou odstraněny.

#### **5.3.8.2 Databáze ExoFOP**

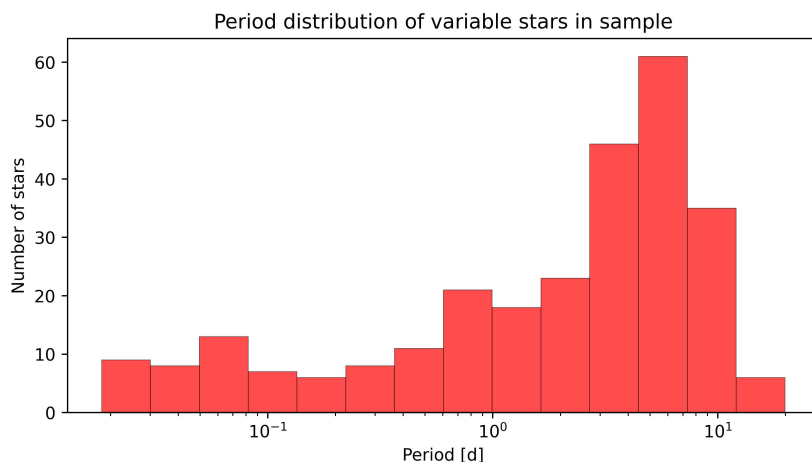
Primárním cílem mise TESS je pozorování exoplanet, tudíž řada hvězd ve vzorku obsahuje exoplanetární tranzity. Tyto hvězdy jsou odstraněny z navazující analýzy. Hvězda S1140 obsahuje zákryty podobající se exoplanetě a není přítomna v databázi ExoFOP. S1140 bude blíže diskutována ve zpracování dat.

## 5.4 Statistika

Z celkových 1865 hvězd ve vzorku je objevena proměnnost u 270 hvězd, tedy  $\sim 14,6$  %.



Obrázek 36: Hertzsprungův-Russellův diagram objevených proměnných hvězd. Osy jsou v logaritmickém měřítku. Závislost je vyjádřena v násobcích zářivého výkonu Slunce na efektivní teplotě. Proměnné hvězdy jsou označeny červeně, neproměnné hvězdy jsou označeny černě. Je zobrazeno 202 proměnných hvězd s dostupnými Gaia astrofyzikálními parametry z celkových 270 proměnných hvězd (vlastní tvorba).



Obrázek 37: Histogram dominantní periody přítomné v periodogramu objevených proměnných hvězd ve vzorku. Osa x je v logaritmickém měřítku (vlastní tvorba).

## 6 Kompletní analýza proměnnosti

Cílem kompletní analýzy je zjištění parametrů proměnnosti dané hvězdy, zahrnující periodu a její chybu, průměrnou a maximální amplitudu, epochu maxima (popř. minima u zákrytových systémů) a její chybu, fázi sekundárního minima v případě zákrytové dvojhvězdy (tzv.  $m - M$ ) a klasifikaci typu proměnnosti. To je dosaženo kombinací automatické a manuální analýzy, jelikož některé parametry není možné spolehlivě určit automaticky a vyžadují „lidský přístup.“

Automatická analýza je zahrnuta v rámci jednoho skriptu, který provádí veškeré zpracování stažených *TargetPixelFiles* do jednotlivých datových produktů pro danou hvězdu. Klasifikace je provedena na základě všech známých astrofyzikálních parametrů a parametrů proměnnosti.

### 6.1 Doplnění parametrů do seznamu hvězd

#### 6.1.1 Zpřesnění magnitud z katalogu TIC v8.2

Magnitudy hvězd převzaté z původního *Target List* seznamu mohou být zastaralé a nepřesné, tudíž je nutno je aktualizovat s nejnovějším *TESS Input Catalog*. Pomocí skriptu níže používající knihovnu `astroquery`, službu `Vizier` a katalog TIC v8.2 je zpřesněna TESS magnituda pro každou zkoumanou hvězdu na řády tisícín.

```
import pandas as pd
import numpy as np
from astroquery.vizier import Vizier
import astropy.units as u
from astropy.coordinates import SkyCoord
from astropy.coordinates import Angle

Vizier.TIMEOUT = 30000 # 8 hour server timeout
df = pd.read_csv('path')
```



Ve funkci `astroquery.vizier()` je parametr `columns`, ve kterém zvolíme požadované informace o hvězdě, tedy pouze `'Tmag'`; `'+_r'` uspořádá výsledky hledání podle vzdálenosti od zadaných souřadnic. Jelikož již známe Tmag hvězdy na přesnost setin, můžeme uvažovat hvězdy s odchylkou magnitud více jak 0,01 jako nesprávné. Opět je zvolen poloměr kuželové výšece pro hledání 30 úhlových vteřin.

```
v = Vizier(columns = ['+_r', 'Tmag'], row_limit = 100)

for i, identifier, ra, dec, mag in zip(range(1000), df['tic'], df['ra'], df['dec'], df['mag']):
    search = v.query_object(f'TIC {str(identifier)}', catalog = "IV/39/tic82", radius = 30 * u.arcsec)
    search = search["IV/39/tic82"]
    search.to_pandas()
    if len(search) < 1:
        print(f'Star {identifier} search failed')
        continue
    magdif = abs(mag - search['Tmag'][0])
    if magdif > 0.01:
        print(f'Star {identifier} search failed')
        continue
    df.at[i, 'mag'] = search['Tmag'][0]
df.to_csv('path', index = False)
```

## 6.1.2 Zpřesnění souřadnic z Gaia DR3

Podobně jako magnitudy, souřadnice v původním *Target List* seznamu jsou nepřesné, zvláště u hvězd s vysokým vlastním pohybem. Gaia DR3 poskytuje vysoce přesné pozice všech hvězd a jejich vlastní pohyb. Pomocí skriptu níže používající knihovnu `astroquery`, službu TAP+ a Gaia DR3 jsou vypočítány souřadnice k epochám J2000 a J2020. Je opět použit poloměr 30 úhlových vteřin pro kuželovou výšece hledání, správná hvězda je opět určena jako ta s nejmenším rozdílem  $T\ mag - G_{RP}\ mag$ .

```
import pandas as pd
from astroquery.gaia import Gaia
import astropy.units as u
from astropy.coordinates import SkyCoord
from astropy.coordinates import Angle

Gaia.MAIN_GAIA_TABLE = "gaiadr3.gaia_source"
Gaia.ROW_LIMIT = 10000

df = pd.read_csv('path')

for i, identifier, ra, dec, mag in zip(range(1000), df['id'], df['ra'], df['dec'], df['mag']):
    try:
        obj = SkyCoord(f'{ra} {dec}', unit = (u.deg, u.deg))
        gaia = Gaia.cone_search_async(obj, 30 * u.arcsec).get_results()
        gaia = gaia.to_pandas()
```

Jsou vybrány pouze relevantní sloupce, tedy magnituda ve filtru  $G_{RP}$  (`'phot_rp_mean_mag'`), rektascenze k J2016 (`'ra'`), deklinace k J2016 (`'dec'`), vlastní pohyb ve směru rektascenze, v jednotkách  $\text{miliarcsec} \cdot \text{rok}^{-1}$  (`'pmra'`) a vlastní pohyb ve směru deklinace, v jednotkách  $\text{miliarcsec} \cdot \text{rok}^{-1}$  (`'pmdec'`). Správné souřadnice k epochám J2000 a J2020 vypočítáme následovně:

$$RA_{2020} = RA_{2016} + \frac{4 \cdot PM_{RA}}{3,6 \cdot 10^5} \quad RA_{2000} = RA_{2016} - \frac{16 \cdot PM_{RA}}{3,6 \cdot 10^5}$$

$$DEC_{2020} = DEC_{2016} + \frac{4 \cdot PM_{DEC}}{3,6 \cdot 10^5} \quad DEC_{2000} = DEC_{2016} - \frac{16 \cdot PM_{DEC}}{3,6 \cdot 10^5}$$

Do sloupců původního CSV souboru `ra` a `dec` jsou zapsány souřadnice k epoše J2020, do nových sloupců `ra2k` a `dec2k` jsou zapsány souřadnice k epoše J2000.

```
gaia = gaia[['phot_rp_mean_mag', 'ra', 'dec', 'pmra', 'pmdec']]

if len(gaia) < 1:
    print(f'Star {identifier} search failed')
    continue

gaia['magdif'] = abs(gaia['phot_rp_mean_mag'] - mag)
gaia.sort_values('magdif', ascending = True, inplace = True)

ra = gaia.loc[0, 'ra'] + (4 * gaia.loc[0, 'pmra']) / 3600000
ra2k = gaia.loc[0, 'ra'] - (16 * gaia.loc[0, 'pmra']) / 3600000
dec = gaia.loc[0, 'dec'] + (4 * gaia.loc[0, 'pmdec']) / 3600000
dec2k = gaia.loc[0, 'dec'] - (16 * gaia.loc[0, 'pmdec']) / 3600000

df.at[i, 'ra'] = ra
df.at[i, 'dec'] = dec
df.at[i, 'ra2k'] = ra2k
df.at[i, 'dec2k'] = dec2k

except:
    print(f'Star {identifier} search failed')

df.to_csv('path', index = False)
```

### 6.1.3 Pomocné parametry ke skriptu

Každá hvězda ve vzorku potřebuje jiné podmínky pro zpracování, kupř. zpracování zákrytové dvojhvězdy se výrazně liší od eruptivní hvězdy. Z tohoto důvodu je pro správnou funkčnost skriptu pro každou hvězdu určeno několik parametrů, které definují jakým způsobem má být hvězda zpracována. Parametry jsou určeny manuálně a jsou zapsány do hlavního CSV souboru, podle kterého se skript řídí. Implementace jednotlivých parametrů ve skriptu bude popsána v kapitole 6.4.

Do sloupce `period` je zapsána nejvýznamější přítomná perioda, zpravidla to je nejvyšší peak v periodogramu dané hvězdy.

Do sloupce `cadence` je zapsána požadovaná kadence pro danou hvězdu, tedy `'ffi'`, `'short'` či `'fast'`. Pokud je parametr nastaven na `'ffi'`, budou pro danou hvězdu staženy výstřižky z FFI dat, popř. pokud bude parametr nastaven na `'short'`, resp. `'fast'`, budou upřednostněny „short“ TPF, resp. „fast“ TPF data. Tento parametr souvisí s rozsahem periodicity dané hvězdy a jejím typem a nyquist limitem dané kadence. FFI data mají kadenci 30 minut (později i 10 minut a 200 sekund), tudíž periodičita pod nyquist limit 1 hodiny bude nepozorovatelná. Navíc, není vhodné se příliš blížit samotnému nyquist limitu, jelikož vzorkování světelné křivky by bylo v takovém případě nedostatečné. Z těchto důvodů je hvězdám s veškerou periodicitou nad 0,2 dne nastaven parametr na `'ffi'`, hvězdám s periodicitou pod 0,2 dne na `'short'`, která má nyquist limit 4 minuty, tedy i s konzervativním faktorem  $5\times$  je limit 20 minut, což zahrnuje všechny rychle pulzující hvězdy. Hodnota `'short'` je také přidělena všem zákrytovým systémům. Hodnota `'fast'` je vyhrazena pro eruptivní hvězdy, jelikož erupce jsou často velice krátké události, tudíž je maximální časové rozlišení „fast“ TPF dat potřeba.

Do sloupce `outlier_low`, resp. `outlier_high` je zapsáno kolik procent odlehlých bodů bude ze světelné křivky odstraněno, a to hodnot příliš nízkých, resp. příliš vysokých. Zákrytové systémy mají tedy parametr `outlier_low` nastavený na 0, aby se zabránilo

„useknutí“ zákrytů, popř. eruptivní hvězdy mají parametr `outlier_high` nastavený na 0, aby se zabránilo „useknutí“ erupcí. Ve všech ostatních případech jsou oba parametry nastaveny na hodnotu 0,001.

Do sloupce `bright` nabývajících booleovských hodnot je zapsáno, zda kadence světelné křivky s příliš jasným pozadím budou odstraněny. Tento parametr je pro většinu hvězd nastaven na `True`, jelikož ztráta  $\sim 20$  % dat nemá významný vliv na analýzu hvězdy, navíc, právě příliš „jasné“ odstraněné kadence jsou často nekvalitní a obsahují zbytkové systematické chyby. Nicméně, zákrytové systémy a eruptivní hvězdy mají tento parametr nastaven na hodnotu `False`, jelikož odstraněním těchto „jasných“ kadencí by mohl být odstraněn zákryt nutný pro konstrukci O – C diagramu či velmi jasná erupce.

Do sloupce `double_fold` nabývajících booleovských hodnot je zapsáno, zda má být fázová světelná křivka složena podle dvojnásobné periody. Tento parametr je nutný ke zpracování zákrytových dvojhvězd, které mají nejvyšší peak v jejich periodogramu na periodě poloviční, jelikož se primární a sekundární minima složí „na sebe.“

Do sloupce `flat_period` je zapsána nejvyšší významná perioda dané hvězdy, za předpokladu, že tuto periodicitu chceme zachovat. Například, pulzující hvězda může mít hlavní periodu pulzací 0,03 dne, nicméně také pulzuje s periodou 1,5 dne. Pokud tuto periodu chceme zachovat, musíme parametr `flat_period` nastavit na danou periodu.

Do sloupce `min_per`, resp. `max_per` jsou zapsány hranice periodicity dané hvězdy, tedy kupř. u pulzujícího DSCT+GDOR hybridu může být rozsah period 0,01 dne – 4 dny.

## 6.2 Skript

Cílem skriptu je kompletně zpracovat veškerá TESS data dané hvězdy, vyhodnotit základní parametry proměnnosti dané hvězdy, vše vizualizovat a veškerá data uložit pro případnou navazující analýzu. Skript se skládá z pěti hlavních částí. V první části jsou stažena potřebná TESS data pro danou hvězdu a prohledáno okolí hvězdy v katalogu Gaia DR3. V následujících třech částech jsou zpracována FFI, „short“ TPF a „fast“ TPF data, včetně jejich vizualizace a uložení. V poslední části je provedeno zpracování všech dat jako celku, včetně jejich vizualizace a uložení. Navíc jsou vypočítány parametry proměnnosti dané hvězdy a jsou uloženy. Tento proces se poté opakuje pro každou hvězdu.

Skript pracuje s následujícími knihovnami, bez kterých by tato práce nebyla možná:

- `numpy` [20], pro práci s array-like datovými strukturami.
- `pandas` [48], pro práci s CSV soubory a doplnění knihovny `numpy`.
- `lightkurve` [28], pro práci s TESS datovými produkty.
- `scipy` [51], pro práci s matematickými funkcemi a jejich fitováním.
- `astropy` [5], pro práci s astronomickými jednotkami a souřadnicemi.
- `astroquery` [17], pro vyhledávání Gaia DR3 dat.
- `matplotlib` [21], pro vizualizaci dat.

## 6.2.1 Zpracování dat

V této kapitole jsou popsány všechny zavedené funkce potřebné ke zpracování dat, jejich použití a nedostatky, včetně odhalených problematik, které řídily jejich evoluci během vytváření a testování samotného skriptu. Funkce vytvořené za účelem vizualizace dat nejsou uvedeny, jelikož nejsou relevantní k zaměření této práce. Správné odsazení často není dodrženo za účelem formátování textu.

### 6.2.1.1 Import knihoven, úvodní parametry

```
import pandas as pd
import numpy as np
import lightkurve as lk
import astropy.units as u
from astropy.coordinates import SkyCoord
from astroquery.gaia import Gaia
import math
import scipy.stats
from scipy.optimize import curve_fit
import time
import gc
import os
import sys
import copy
import warnings
```

Dále jsou nastaveny parametry pro vyhledávání v Gaia DR3, nastavení zdrojového CSV souboru, nastavení adresářů pro ukládání vizualizace, zpracovaných dat a parametrů proměnnosti.

```
Gaia.MAIN_GAIA_TABLE = "gaiadr3.gaia_source"
Gaia.ROW_LIMIT = 10000

df = pd.read_csv('path\\source.csv')

img_folder = r"path\\plots\\"
data_folder = r"path\\data\\"
params_folder = r"path\\params\\"
```

Skript je započat for loop, definující názvy proměnných pro zpracování.

```
for identifier, tic, target_ra, target_dec, mag, cadence, target_period, outlier_l, outlier_h, bright,
double, flat_period, min_per, max_per, in_zip(df['id'], df['tic'], df['ra'], df['dec'], df['mag'],
df['cadence'], df['period'], df['outlier_low'], df['outlier_high'], df['bright'], df['double_fold'],
df['flat_period'], df['min_per'], df['max_per']):
```

### 6.2.1.2 Vyhledání okolí hvězdy v Gaia DR3

Okolí 500 úhlových vteřin kolem dané hvězdy je prohledáno pro následující vizualizaci staženého výstřižku. Hvězdy jsou omezeny do 18. magnitudy ve filtru  $G_{RP}$ .

```
obj = SkyCoord(f'{target_ra} {target_dec}', unit = (u.deg, u.deg))
gaia = Gaia.cone_search_async(obj, 500 * u.arcsec).get_results()
gaia = gaia.to_pandas()
gaia = gaia[['ra', 'dec', 'phot_rp_mean_mag']]
gaia = gaia[gaia['phot_rp_mean_mag'] < 18]
```

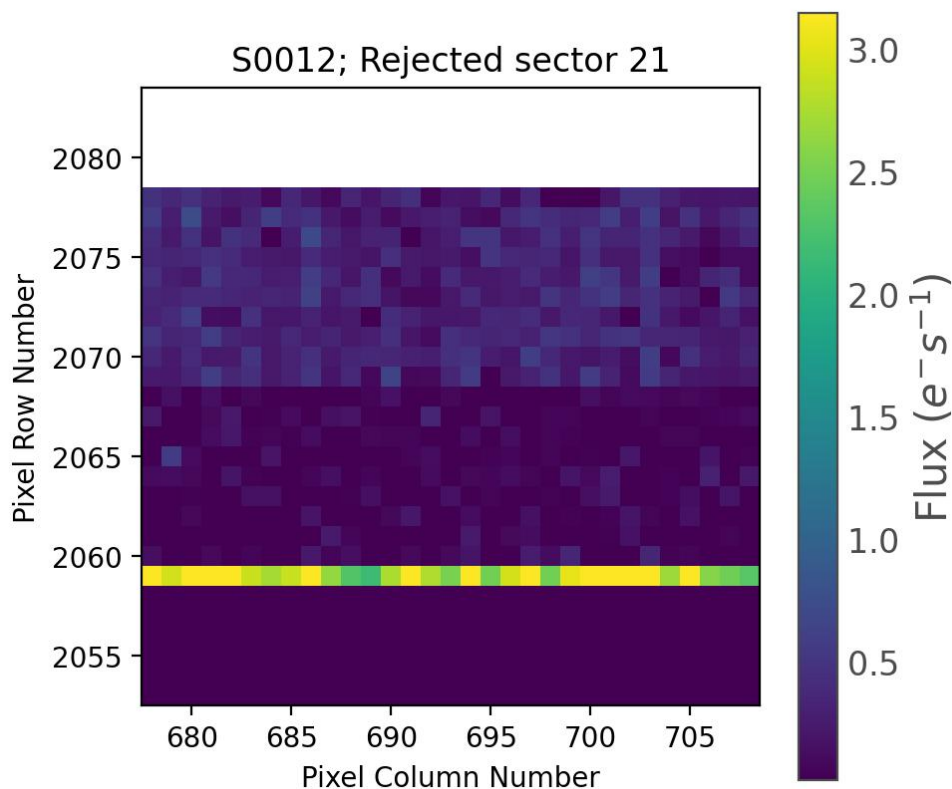
### 6.2.1.3 Vyhledání TESS dat

Nejprve je nutno se ujistit, že daná hvězda byla pozorována.

```
search = lk.search_tesscut(target = obj)
if len(search) == 0:
    print(f'Star {identifier} has no TESS data available')
    gc.collect()
    continue
```

Poté se pro všechny sektory stáhnou elementární  $1 \times 1$  FFI výstřižky, ze kterých je možné zjistit pozici hvězdy na CCD detektoru v daném sektoru. Pokud se hvězda nachází příliš blízko okraji CCD detektoru, musí tento sektor být vyloučen ze zpracování, jelikož by hvězda mohla být „uříznutá“ a data by byla tudíž poškozená. Sektory, v nichž má střed dané hvězdy souřadnice méně jak 16 či více jak 2031 v ose x, popř. méně jak 60 či více jak 2075 v ose y jsou vyřazeny.

```
search = lk.search_tesscut(target = obj).download_all(cutout_size = 1)
rejected_sectors = [int(s.sector) for s in search if (s.row > 2031 or s.row < 16
                                                    or s.column < 60 or s.column > 2075)]
approved_sectors = [int(s.sector) for s in search if not int(s.sector) in rejected_sectors]
```



Obrázek 38: Vyřazený sektor 21 hvězdy S0012. Střed hvězdy je na pixelu [694, 2069], což nevyhovuje nastaveným kritériím (vlastní tvorba).

Následující úsek skriptu vytvoří tři seznamy obsahující čísla sektorů, ve kterých bude daná hvězda zpracována, a to seznam FFI sektorů, „short“ a „fast“ TPF sektorů.

```
ffi_sectors = []
short_sectors = []
fast_sectors = []

if cadence == 'short' or cadence == 'fast':
    search_tpf = lk.search_targetpixelfile(target = obj, author = 'SPOC', exptime = 'short')
    if len(search_tpf) != 0:
        if cadence == 'fast':
            search_tpf_fast = lk.search_targetpixelfile(target = obj, author = 'SPOC', exptime = 'fast')
            if len(search_tpf_fast) != 0:
                fast_sectors = [int(i[-2:]) for i in search_tpf_fast.mission]
                short_sectors = [int(i[-2:]) for i in search_tpf.mission if not int(i[-2:]) in fast_sectors]
                ffi_sectors = [int(i) for i in approved_sectors if not ((int(i) in short_sectors)
                                                                    or (int(i) in fast_sectors))]
            else:
                short_sectors = [int(i[-2:]) for i in search_tpf.mission]
                ffi_sectors = [int(i) for i in approved_sectors if not int(i) in short_sectors]
        else:
            short_sectors = [int(i[-2:]) for i in search_tpf.mission]
            ffi_sectors = [int(i) for i in approved_sectors if not int(i) in short_sectors]
    else:
        ffi_sectors = approved_sectors
else:
    ffi_sectors = approved_sectors
```

#### 6.2.1.4 Vytvoření adresářů a potřebných seznamů ke zpracování

Pro každou hvězdu je vytvořen adresář, který se dále dělí na část s vizualizacemi, zpracovanými daty a ostatní potřebné adresáře.

Je vytvořeno také několik seznamů, do kterých jsou ukládány v průběhu zpracování nejruznější data.

```
df_list = []
df_list_flat = []
df_list_save = []
sectors = []
time_labels = []
```

#### 6.2.1.5 Úvod zpracování dat v rámci sektoru

Jednotlivé funkce jsou popsány chronologicky, jejich implementace se avšak může lišit pro jednotlivé typy sektorů (FFI, „short“ a „fast“ TPF); tyto rozdíly jsou také popsány.

Všechny tři hlavní části skriptu jsou započaty analogicky `for` loop, zpracovávající postupně každý sektor v dané kadenci.

V každém `for` loop jsou nejprve stažena data pro daný sektor. Pokud má daná hvězda nastavený parametr `bright` na `False`, parametr `quality_bitmask` ve funkci `lk.SearchResult.download()` je nastaven na hodnotu `'default'`, v případě, že daná hvězda má hodnotu `True`, parametr `quality_bitmask` je nastaven na hodnotu `'hard'`. Parametr `quality_bitmask` omezuje kvalitu stahovaných kadencí, tudíž u hvězd s parametrem `bright` nastaveným na `False` chceme zachovat veškerá data, byť i méně kvalitní.

Data pro FFI sektory jsou stažena pomocí funkce `lk.search_tesscut()`. Velikost staženého výstřižku je určena parametrem `cutout_size`, v našem případě nastaveného na hodnotu 31.

```

if bright:
    tpf = lk.search_tesscut(target = obj, sector = s).download(cutout_size = 31, quality_bitmask = 'hard')
else:
    tpf = lk.search_tesscut(target = obj, sector = s).download(cutout_size = 31, quality_bitmask = 'default')

```

Data pro TPF sektory jsou stažena pomocí funkce `lk.search_targetpixelfile()`. Velikost výstřižku je již definována, zpravidla se pohybuje kolem hodnot 11 – 13, velikost je vždy lichá a velikost stran výstřižku nemusí být stejná. Parametrem `author` zvolíme *TargetPixelFiles* vytvořené SPOC, parametr `exptime` je v případě „short“ TPF kadence nastaven na hodnotu `'short'`, v případě „fast“ TPF kadence na hodnotu `'fast'`.

```

if bright:
    tpf = lk.search_targetpixelfile(target = obj, author = 'SPOC', sector = s, exptime)
    tpf = tpf.download(quality_bitmask = 'hard')
else:
    tpf = lk.search_targetpixelfile(target = obj, author = 'SPOC', sector = s, exptime)
    tpf = tpf.download(quality_bitmask = 'default')

```

Dále je určena „nejtmavší“ kadence, která je použita pro vytvoření aperturní masky, masky pozadí a vizualizaci. V případě, že bychom použili kadenci, která má velmi kontaminované pozadí rozptýleným světlem, mohla by se maska pozadí špatně vytvořit.

Datotyp `lk.targetpixelfile` představuje originální FITS pixelová data (tedy fotografie) pro danou hvězdu v čase. Více méně si můžeme představit, že se jedná o  $\sim 27$  dní dlouhé video dané hvězdy. Jednotlivé hodnoty můžeme extrahovat pomocí `.time`, `.flux` a `.flux_err`. Tímto získáme `np.ndarray()` s tvarem (*počet kadencí, velikost výstřižku v ose x, velikost výstřižku v ose y*), tedy kupř. (10000, 31, 31). Jednotlivé hodnoty jsou datotyp `astropy.units.quantity.Quantity`, v případě času je jednotka `u.day`, v případě světelného toku a chyby světelného toku je jednotka  $\frac{\text{u.electron}}{\text{u.s}}$ . Pro odstranění jednotky můžeme připsat `.value`, což je často vhodné při práci s `np.ndarray`.

```

d_cad = int(np.argmin(np.nanmedian(tpf.flux.value, axis = (1,2))))

```

V ojedinělých případech může mít z neznámých důvodů jediná kadence nulové hodnoty, tudíž nemohou být správně vytvořeny masky. Příklad takové kadence je na obrázku 39. Tento případ je ošetřen následovně.

```

if np.nanmedian(tpf[d_cad].flux.value) < 0.2 * np.nanmedian(tpf.flux.value):
    d_cad -= 1

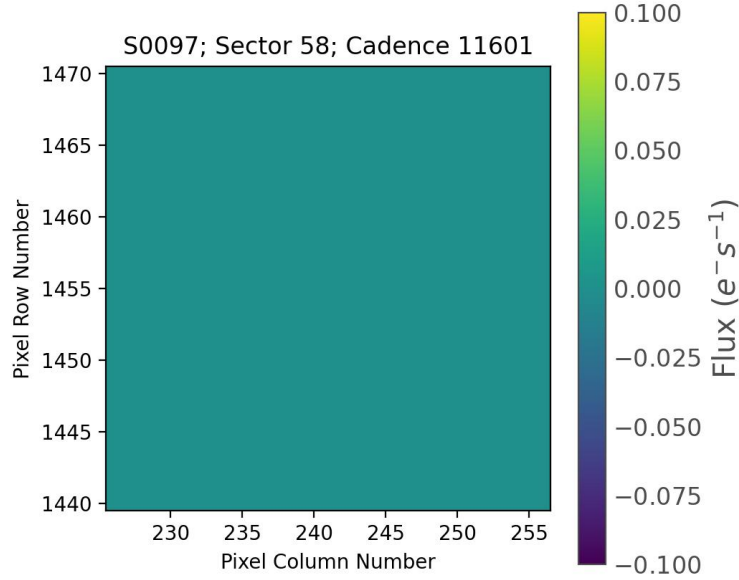
```

Pro práci s různými hodnotami je zavedena pomocná funkce `sr(x, n)`, která zaokrouhlí číslo `X` na `N` platných číslic. Je založená na funkci `np.format_float_positional()`, nicméně doplňuje její nedostatky.

```

def sr(x, n):
    x = np.format_float_positional(x, precision = n, fractional = False, min_digits = n)
    if x[-1] == '.':
        x = x[0:-1]
    if x.find('.') == -1:
        return int(x)
    else:
        return float(x)

```



Obrázek 39: Příklad „nulové“ kadence sektoru 58 hvězdy S0097 (vlastní tvorba).

Následovně je určena přesná kadence daného sektoru, dále převedená na `string`, který je přidán do seznamu a používán pro popisky grafů. Vypočítán je také nyquist limit pro danou kadenci.

```
exp_time = float(np.nanmedian(np.diff(tpf.time.value)))
if exp_time < 0.003:
    time_label = '200 s'
else:
    time_label = f'{sr(exp_time * 1440, 2)} min'
f_nyq = 1 / (2 * exp_time)
cadences.append(f_nyq)
time_labels.append([s, time_label])
if s < 10:
    sectors.append(f'Sector 0{s}')
else:
    sectors.append(f'Sector {s}')
```

### 6.2.1.6 Vytvoření aperturní masky

Aperturní maska určuje jaké pixely jsou použity k vytvoření světelné křivky. Jedná se o `np.array()` booleovských hodnot se stejným tvarem jako TPF. Je vytvořena pomocí funkce `find_aperture_mask()`, jejímž vstupem je magnituda dané hvězdy, společně s TPF, číslem „nejtmavší“ kadence a identifikátorem hvězdy. Pracuje na základě funkce `lk.targetpixelfile.create_threshold_mask()`.

Ideální počet pixelů zahrnutých v aperturní masce je určen následovně. Jednotlivé hodnoty počtu pixelů v masce jsou pro hvězdu dané magnitudy manuálně určeny, za pomoci funkce `scipy.optimize.curve_fit()` proloženy zvolenou funkcí, v tomto případě polynomem druhého stupně. Analogicky je určena řada ostatních použitých vztahů.

$$N_{\text{pixels}} = 0,928571 \cdot \text{mag}^2 - 23,4143 \cdot \text{mag} + 153,257$$



Výsledkem je aperturní maska obsahující  $\sim 35$  pixelů v případě hvězdy magnitudy 7, maska hvězdy magnitudy 12 poté obsahuje  $\sim 6$  pixelů. Zvolený polynom tvoří aperturní masku spíše menší, čímž se zabrání vysoké kontaminaci okolních hvězd.

Funkce nejprve najde počáteční masku pomocí funkce `lk.create_threshold_mask()`, parametr `threshold` určuje násobek směrodatné odchylky od mediánu, tvořící spodní práh pro potřebný světelný tok ke zvolení daného pixelu do aperturní masky. Je nastavena počáteční hodnota 50. Iterativním zmenšováním, resp. zvětšováním hodnoty `threshold` je poté dosaženo požadované velikosti aperturní masky. Parametrem `reference_pixel` je určen střed masky, v tomto případě tedy vždy střed výstřižku, odpovídající hodnotou je `'center'`.

```
def find_aperture_mask(mag, tpf, d_cad, identifier):
    target_size = 0.928571 * (mag ** 2) - 23.4143 * mag + 153.257
    mask = tpf[d_cad].create_threshold_mask(reference_pixel = 'center', threshold = 50)
    if np.count_nonzero(mask == True) > target_size:
        i = 1
        while np.count_nonzero(mask == True) > target_size:
            mask = tpf[d_cad].create_threshold_mask(reference_pixel = 'center', threshold = 50 + i ** 1.1)
            i += 1
    elif np.count_nonzero(mask == True) < target_size:
        i = 1
        while np.count_nonzero(mask == True) < target_size:
            mask = tpf[d_cad].create_threshold_mask(reference_pixel = 'center', threshold = 50 - i ** 1.1)
            i += 1
    elif np.count_nonzero(mask == True) == target_size:
        pass
    else:
        raise Exception(f'Could not create threshold mask for star {identifier}')
    return mask
```

Vytváření aperturní masky je nutné pouze v FFI sektorech, jelikož „short“ a „fast“ TPF obsahuje aperturní masku vytvořenou SPOC. Ta je získána následovně.

```
mask = tpf.pipeline_mask
```

### 6.2.1.7 Vytvoření masky pozadí

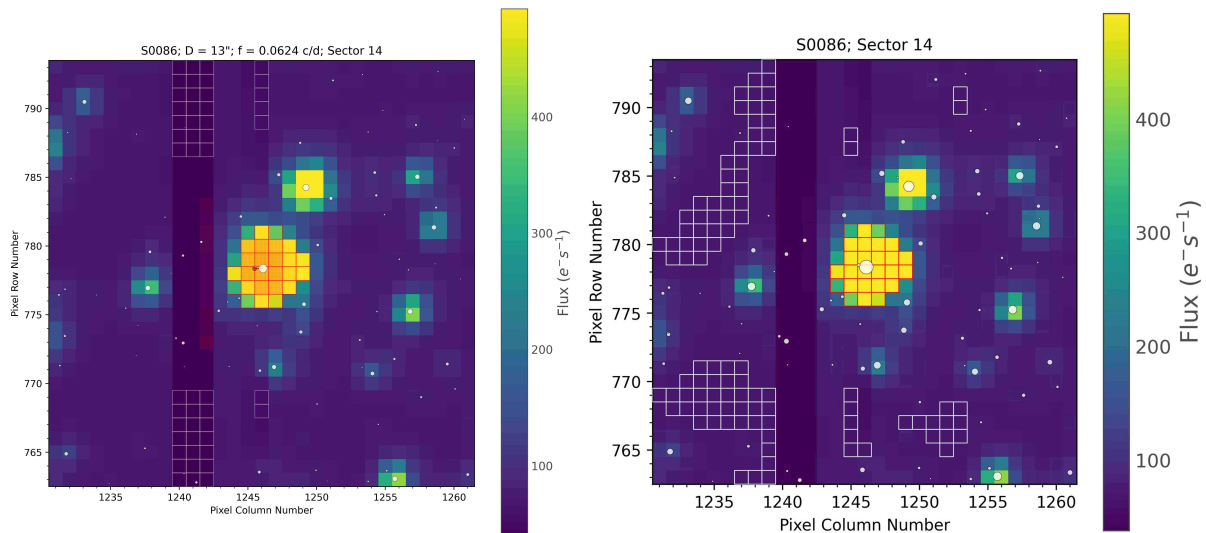
Maska pozadí slouží k charakterizaci přítomných systematických chyb, převážně rozptýleného světla. Je vytvořena pomocí funkce `find_background_mask()`, jejímž vstupem je magnituda dané hvězdy, společně s TPF a číslem „nejtmavší“ kadence. Maska pozadí musí zahrnovat co nejmenší množství světelného toku z okolních hvězd, tudíž jsou víceméně vybrány „nejtmavší“ pixely. Nicméně, maska musí být ošetřena, aby se nevybíraly nechtěné pixely.

Kolem hvězdy je vytvořena ochranná maska, která brání zvolení pixelů v blízkosti pozorované hvězdy. V případě, že by maska pozadí obsahovala světelný tok pozorované hvězdy, mohla by se ze světelné křivky odstranit část proměnnosti. Poloměr ochranné masky je určen následovně.

$$R_{\text{protect\_mask}} = \left\lceil 1,5 \cdot \sqrt{\frac{N_{\text{pixels}}}{\pi}} + 4 \right\rceil$$

V určitých částech CCD se vyskytují dva a tři pixely široké „pruhy“, které mají zpravidla chybné hodnoty a nesmí být zahrnuty do masky pozadí. Zvolení těchto chybných

pixelů je zabráněno volením pixelů jasnější jak  $\times 0,85$  mediánu TPF, všechny pixely nesplňující tuto podmínku jsou zahrnuty v ochranné masce.



Obrázek 40: FFI výstřižky kolem hvězdy S0086 v sektoru 14. Na obou výstřižcích lze vidět dva vertikální pruhy. Pixely označené bílým rámečkem jsou zahrnuty v masce pozadí. V případě nalevo jsou chybně zahrnuty pixely náležící pruhům, napravo je použita ochranná maska zabráňující zvolení těchto pixelů, nicméně je stále zvoleno 8 pixelů náležícím „slabšímu“ z pruhů. V takovém případě je už maska pozadí použitelná (vlastní tvorba).

Po vytvoření ochranné masky zbývá vybrat „nejtmavší“ pixely, které této masce nenáležejí. Je použito 100 nejtmavších pixelů, tedy v případě FFI výstřižku  $31 \times 31 \sim 10\%$  „nejtmavších“ pixelů. Implementace funkce vypadá následovně.

```
def find_background_mask(mag, tpf, d_cad, identifier):
    tpf_d = tpf[d_cad].flux.value[0]
    tpf_m = np.nanmedian(tpf_d)
    bg_arr = np.array([[x, y] for x in range(31) for y in range(31)])
    bg_mask = np.zeros((31, 31), dtype='bool')
    prot_mask = np.ones((31, 31), dtype='bool')
    r = int((((0.928571 * (mag ** 2) - 23.4143 * mag + 153.257) / math.pi) ** 0.5) * 1.5) + 4
    prot_mask[15 - r : 16 + r, 15 - r : 16 + r] = False
    for px in bg_arr:
        if tpf_d[px[0], px[1]] < tpf_m * 0.85:
            prot_mask[px[0], px[1]] = False
    bg_flux = [tpf_d[px[0], px[1]] for px in bg_arr]
    bg_flux_s = sorted(bg_flux)
    bg_px = [bg_flux.index(bg_flux_s[i]) for i in range(len(bg_flux))]
    for px in bg_px:
        if np.count_nonzero(bg_mask) == 100:
            break
        if prot_mask[bg_arr[px, 0], bg_arr[px, 1]]:
            bg_mask[bg_arr[px, 0], bg_arr[px, 1]] = True
    return bg_mask
```

Tato funkce pro nalezení ideální masky pozadí je použita pouze v FFI sektorech. V „short“ a „fast“ TPF sektorech jsou výstřižky zpravidla téměř třetinové, tudíž není dostatek prostoru na vytvoření masky pozadí, proto jsou do masky pozadí zahrnuty pouze

všechny krajní pixely výstřižku. Musíme také nadefinovat velikost výstřižku, která se pro hvězdy různých magnitud liší. Implementace vypadá následovně.

```
size_x, size_y = tpf.shape[1], tpf.shape[2]
bg_mask = np.zeros((size_x, size_y), dtype = 'bool')
bg_mask[0, :], bg_mask[size_x - 1, :], bg_mask[:, 0], bg_mask[:, size_y - 1] = True, True, True, True
```

### 6.2.1.8 Odstranění NaN hodnot, vytvoření světelné křivky

TPF může obsahovat NaN hodnoty, a to ve všech třech dimenzích – čas, světelný tok a chyba světelného toku. Pro správnou funkčnost následujících funkcí musí být odstraněny všechny kadence obsahující alespoň jednu NaN hodnotu na pixelu zahrnutém v aperturní masce nebo masce pozadí.

Nejprve je vytvořena maska kombinující aperturní masku a masku pozadí.

```
com_mask = np.logical_or(mask, bg_mask)
```

Dále jsou nalezeny indexy kadcencí obsahující NaN hodnoty.

```
nan1 = np.argwhere(np.isnan(tpf.time.value))
nan2 = np.argwhere(np.isnan(tpf.flux.value))
nan3 = np.argwhere(np.isnan(tpf.flux_err.value))
```

Odstraníme duplikáty a zahrneme pouze NaN hodnoty nácházející se v kombinované masce.

```
nan1 = np.unique([int(x[0]) for x in nan1 if com_mask[x[1], x[2]]])
nan2 = np.unique([int(x[0]) for x in nan2 if com_mask[x[1], x[2]]])
nan3 = np.unique([int(x[0]) for x in nan3 if com_mask[x[1], x[2]]])
nan = np.unique(np.concatenate([nan1, nan2, nan3], axis = 0).astype('int32'))
```

Struktura TPF se v „short“ a „fast“ TPF sektorech liší, jelikož jsou zpracované SPOC, dimenze světelného toku je již opravena o systematické chyby. Nicméně, jak již bylo zmíněno, korekce SPOC není ideální a v řadě případů odstraňuje proměnnost hvězd. Systematika je uložena ve čtvrté dimenzi TPF s názvem `flux_bkg`. Jednotlivé dimenze jsou nadefinovány a opraveny o NaN kadence. Pro vlastní zpracování systematik jsou dimenze `flux` a `flux_bkg` sečteny.

```
time_cor = np.delete(tpf.time, nan, axis = 0)
flux_cor = np.delete(tpf.flux, nan, axis = 0)
flux_err_cor = np.delete(tpf.flux_err, nan, axis = 0)
flux_bkg_cor = np.delete(tpf.flux_bkg, nan, axis = 0)
flux_total = flux_cor + flux_bkg_cor
```

Světelnou křivku v FFI sektorech vytvoříme pomocí funkce `.to_lightcurve()`, zvolíme námi vytvořenou aperturní masku. Následně ze světelné křivky odstraníme NaN kadence pomocí funkce `.remove_rows()`.

```
lc = tpf.to_lightcurve(aperture_mask = mask)
lc.remove_rows(error_cadences)
```

Světelnou křivku v „short“ a „fast“ TPF sektorech vytvoříme z o NaN kadence opravených `np.array()`, opět z pixelů označených aperturní maskou. Jednotlivé dimenze vložíme do jednoho `lk.LightCurve` datotypu.

```
lc = lk.LightCurve(time = time_cor,
                   flux = np.sum(flux_total[:, mask], axis = 1),
                   flux_err = np.sum(flux_err_cor[:, mask], axis = 1))
```

### 6.2.1.9 Odstranění rozptýleného světla

Funkce pro odstranění rozptýleného světla (popř. ostatních systematik) prošla během vývoje skriptu nejvíce změnami. Zprvu byly použity tradiční metody zabudované do knihovny `lightkurve`, principiálně používané i v oficiálním zpracování SPOC, která vytváří *SAP* a *PDCSAP* světelné křivky. Metody jako `CBVCorrector`, `PLDCorrector` či `RegressionCorrector` zabudované v `lightkurve` použijí masku pozadí k vytvoření vektorů, které mají charakterizovat přítomnou systematiku. Pomocí analýzy hlavních komponent (PCA) jsou vektory zredukovány pouze na několik složek nejlépe charakterizující přítomnou systematiku. Tyto vektory jsou poté ze světelné křivky lineární algebrou zabudovanou v korektoru odečteny takovým způsobem, aby se výsledná světelná křivka co nejvíce blížila k nule. Tato metoda by byla bezchybná pouze za předpokladu, že ve světelné křivce není přítomný žádný signál (tedy proměnnost), což samozřejmě v našem případě je. Proto často dochází k nadměrné korekci. Kvůli povaze hlavního komponentu, čímž je rozptýlené světlo, jehož intenzita se během jednoho sektoru dvakrát opakuje (během jednoho sektoru satelit TESS dvakrát oběhne Zemi), nejvíce trpí nadměrnou korekcí hvězdy s delšími periodami, a to nejvíce kolem  $\sim 27$  dní. Navíc, PCA vektory obsahují poměrně významné množství šumu, který se na světelné křivce také promítne. Všechny tyto efekty lze pozorovat na *PDCSAP* světelných křivkách.

Zde bude demonstrována mnohem robustnější metoda pro odstranění rozptýleného světla, které je největším zdrojem kontaminace TESS světelných křivek. Výsledky této metody budou porovnány s tradiční korekcí pomocí `RegressionCorrector` a oficiálními *PDCSAP* světelnými křivkami vytvořenými SPOC.

Nejdříve je nadefinována pomocná funkce `smooth()`, počítající klouzavý průměr. Jejím vstupem je `np.array()` obsahující závislou proměnnou (nejčastěji tedy světelný tok), šířku okna pro výpočet průměru, `np.array()` obsahující nezávislou proměnnou (nejčastěji tedy čas), a dva argumenty `split` a `niter`. Funkce dokáže vypočítat klouzavý průměr i pro krajní hodnoty `np.array()` díky adaptivnímu snížení velikosti okna. Argument `split` rozdělí vstupní `np.array()` na dva, což je vhodné pro práci s TESS daty obsahující „mezeru“ uprostřed každého sektoru. Argument `niter` provede klouzavý průměr N-krát, s iterativně zvětšujícím se oknem. Využití tohoto argumentu bude uvedeno v následujících kapitolách. Zdrojový kód funkce `smooth()` je dostupný v příloze dokumentu.

Pro odstranění rozptýleného světla je definována funkce `regression()`, jejímž vstupem je světelná křivka, od které chceme odstranit rozptýlené světlo, `flux` dimenze TPF opravená o NaN hodnoty, maska pozadí, aperturní maska, kadence daného sektoru a argument `return_model`. Zdrojový kód funkce `regression()` je dostupný v příloze dokumentu.

Zjednodušeně řečeno funkce počítá průměr světelného toku v pozadí označeného maskou pro každou kadenci. Rozptýlené světlo v pozadí kontaminující světelné křivky má aditivní charakter, tedy světelný tok pozadí musí být odečten, nikoliv vydělen jak je v některých zdrojích uvedeno. Funkce navíc pro každý pixel masky pozadí vypočítá jeho váhu na základě jeho proměnnosti v čase. Pokud by byla maska pozadí kontaminována proměnnou hvězdou, její efekt na světelnou křivku pozorované hvězdy by se významně utlumil. Tato vlastnost funkce také obecně snižuje přivedený šum, jelikož méně časově proměnné pixely mají přidělenou vyšší váhu.

Nejprve funkce vypočítá průměrný světelný tok pozadí pro každou kadenci a medián světelného toku pozadí.

```
def regression(lc_input, tpf, bg_mask, mask, cadence, return_model = True):
    lc = copy.deepcopy(lc_input)
    flux_bg = tpf[:, bg_mask]
    flux_bg_mean = np.nanmean(flux_bg, axis = 1)
    median_bg_flux = np.nanmedian(flux_bg_mean)
```

Dále vypočítá medián světelného toku pro každý pixel masky pozadí v každé kadenci. Medián pro každý pixel je vydělen mediánem všech pixelů v dané kadenci, díky čemuž můžeme normalizovat hodnoty pixelů v každé kadenci na stejnou hladinu.

```
median_flux_px = np.nanmedian(flux_bg, axis = 0)
norm_scale = median_flux_px / median_bg_flux
norm_flux_bg = flux_bg / norm_scale
```

Pomocí těchto normalizovaných hodnot pro každý pixel můžeme vypočítat odchylku každého pixelu od průměru pro každou kadenci.

```
flux_bg_mean_dev = np.stack([norm_flux_bg[:, i] - flux_bg_mean for i in range(len(median_flux_px))],
                             axis = 1)
```

Pro každý pixel následně vypočítáme jeho průměrnou odchylku v čase (tedy směrodatnou odchylku) a vydělíme ji mediánem daného pixelu, čímž ji normalizujeme. Normalizace směrodatné odchylky je provedena, jelikož šum daného pixelu se lineárně zvyšuje s přítomným světelným tokem v daném pixelu, tudíž normalizací získáme veličinu nezávislou na světelném toku daného pixelu.

```
flux_mean_std = np.nanstd(flux_bg_mean_dev, axis = 0)
flux_mean_std_norm = flux_mean_std / median_flux_px
```

Váhy pro každý pixel jsou odvozeny právě z jeho normalizované směrodatné odchylky. Zvolený rozsah vah je  $\langle \frac{1}{3}; 1 \rangle$ , tudíž nejvíce časově proměnné pixely přispívají pro výpočet světelného toku pozadí pouze třetinově. Váha pixelu je vypočtena následovně.

$$W_{px} = \frac{2}{3} \cdot \left( \frac{\max s_i - s_{px}}{\max s_i - \min s_i} + 0,5 \right)$$

Dále implementováno do funkce.

```
flux_weights = (((np.nanmax(flux_mean_std_norm) - flux_mean_std_norm)
                  / (np.nanmax(flux_mean_std_norm) - np.nanmin(flux_mean_std_norm))) + 0.5) / 1.5
```

Finálně je vypočten vážený průměr z pixelů označených maskou pozadí. Výsledný `np.array()` představuje hodnotu světelného toku pozadí v  $e^- \cdot s^{-1}$  v jediném pixelu. Aperturní maska avšak zahrnuje několik pixelů, tudíž k získání celkového světelného toku kontaminujícího světelnou křivku dané hvězdy musíme vypočítaný světelný tok pro jediný pixel vynásobit počtem pixelů zahrnutých v aperturní masce.

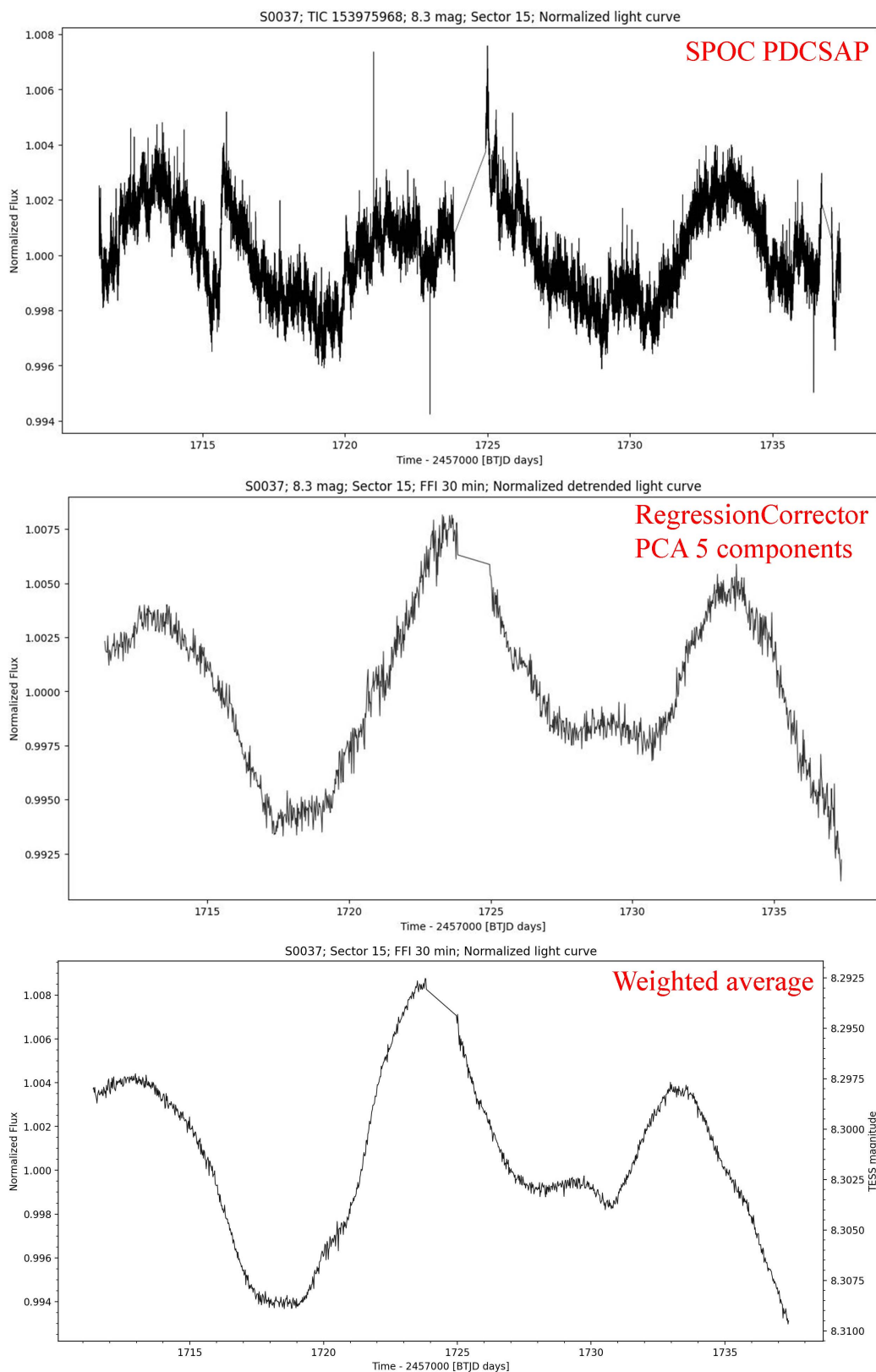
```
weighted_avg_flux = np.average(flux_bg, weights = flux_weights, axis = 1)
scaled_bg_flux = weighted_avg_flux * len(mask[mask == True])
scaled_bg_flux = scaled_bg_flux.value
```

Světelný tok pozadí se periodicky mění v řádech dnů, tudíž v sektorech s vyšší kadencí (200 sekund, 2 minuty a 20 sekund) lze vypočítaný světelný tok dále „vyhladit“, čímž se sníží množství šumu přidaného do světelné křivky. Vyhlazení probíhá pomocí funkce `smooth()`, velikost okna je nastavena pro danou kadenci tak, aby odpovídala 10 minutové kadenci.

```
if cadence == '200 s':
    scaled_bg_flux = smooth(scaled_bg_flux, 3, lc.time.value)
elif cadence == '2 min':
    scaled_bg_flux = smooth(scaled_bg_flux, 5, lc.time.value)
elif cadence == '20 s':
    scaled_bg_flux = smooth(scaled_bg_flux, 30, lc.time.value)
```

Funkce odečte minimální hodnotu světelného toku pozadí z celého `np.array()`, jelikož se předpokládá, že minimální hodnota světelného toku pozadí by měla být v určité době pozorování přibližně nulová, což pravděpodobně není daleko od skutečné hodnoty. Zbývá odečíst světelný tok od světelné křivky a výsledek vložit do jediného `lk.LightCurve` datatypu.

```
scaled_bg_flux -= np.nanmin(scaled_bg_flux)
lc_regress = lk.LightCurve(time = lc.time.value, flux = lc.flux.value - scaled_bg_flux)
if return_model:
    return lc_regress, scaled_bg_flux
else:
    return lc_regress
```



Obrázek 41: Tři stejné světelné křivky, tři různé způsoby odstranění systematicky. Nahoře: PDCSAP světelná křivka vygenerovaná SPOC. Uprostřed: světelná křivka s korekcí pomocí `RegressionCorrector` a analýzy hlavních komponent, v tomto případě bylo použito 5 hlavních komponent. Dole: světelná křivka s korekcí prezentované funkce `regression()` (vlastní tvorba).

Hvězda S0037 v sektoru 15 viditelná na obrázku 41 jasně demonstruje problémy s PD-CSAP světelnými křivkami a světelnými křivkami zpracovanými tradičními metodami pomocí analýzy hlavních komponent. Je důležité zmínit, že PD-CSAP světelná křivka je v „short“ TPF kadenci, tudíž expoziční čas je  $15\times$  kratší než v 30 minutové FFI kadenci, proto má světelná křivka přirozeně větší rozptyl. Nicméně, na PD-CSAP světelné křivce lze vidět velké množství šumu s periodou  $\sim 1$  den, odpovídající periodě změn jasnosti chybně odstraněného rozptýleného světla. V první polovině sektoru lze také vidět dva „skoky“ jasnosti, které byly již diskutovány. Proměnnost samotné hvězdy je výrazně potlačena a neodpovídá skutečnosti. Světelná křivka opravená pomocí analýzy hlavních komponent je výrazně lepší než PD-CSAP světelná křivka, nicméně stále obsahuje výrazně větší množství šumu než světelná křivka opravená pomocí zavedené funkce `regression()` a také mírně potlačuje proměnnost samotné hvězdy. Světelná křivka opravená pomocí prezentované funkce je bezpochyby nejkvalitnější a obsahuje minimální množství zbytkové systematiky, tudíž nejlépe zachovává proměnnost hvězdy.

#### 6.2.1.10 Odstranění zbytkové systematiky

I přes kvalitní odstranění rozptýleného světla světelná křivka obsahuje zbytkové trendy způsobené nepřesností pointace či ostřením. Je zavedena funkce `flatten_lc()`, která se pokouší o odstranění nízkofrekvenčního signálu s ohledem na periodicitu hvězdy. Jejím vstupem je světelná křivka, kterou chceme opravit, parametr `flat_period` označující nejdelší proměnnost, která by měla být zachována a argumenty `return_model`, `split` a `offset`. Zdrojový kód funkce `flatten_lc()` je zahrnutý v příloze dokumentu.

Podobně jako funkce `smooth()` je možnost světelnou křivku rozdělit na dvě během zpracování, tudíž se provede detrendování pro obě poloviny světelné křivky zvlášť. Argument `offset` rozhoduje, zda budou rozdělené poloviny světelné křivky navzájem „vyrovnané“ na stejný medián. Nejprve je na základě periody hvězdy rozhodnuto o typu zpracování. Prezentované nastavení je výsledkem nesčetného testování, nicméně může být upraveno.

```
def flatten_lc(lc_regress, period, return_model = True, split = True, offset = True):
    lc = copy.deepcopy(lc_regress)
    if period > 4:
        polyorder = 0
        offset = False
    elif period > 1 and period <= 4:
        polyorder = 0
    elif period > 0.2 and period <= 1:
        polyorder = 1
    else:
        polyorder = 2
        niter = 2
```

Funkce se dále dělí na dvě části podle nastavení argumentu `split`, přičemž obě části si jsou velmi podobné, pouze v jednom případě probíhá zpracování pro dvě poloviny světelné křivky zvlášť. Zde bude popsána situace pro `split = False`.



V případě, že parametr `polyorder` má hodnotu 1, světelnou křivkou se proloží lineární funkce pomocí funkce `np.polyfit()`.

```
if polyorder == 1:
    idx = np.isfinite(lc_t) & np.isfinite(lc_f)
    fit = np.poly1d(np.polyfit(lc_t[idx], lc_f[idx], polyorder))
    poly_flux = fit(lc_t)
    poly_flux -= np.nanmin(poly_flux)
    lc_f = lc_f - poly_flux
```

V případě, že parametr `polyorder` má hodnotu 2, přítomný trend je vypočítán pomocí dříve zavedené funkce `smooth()`. Šířka okna pro klouzavý průměr je trojnásobek vstupní periody hvězdy. Celočíslný násobek periody hvězdy pro šířku okna by měl v teoretickém případě harmonického průběhu proměnnosti maximálně redukovat proměnnost hvězdy samotné. Klouzavý průměr je pomocí argumentu `niter` proveden dvakrát.

```
elif polyorder >= 2:
    wl = int((period / np.nanmedian(np.diff(lc.time.value))) * 3)
    poly_flux = smooth(lc_f, wl, lc_t, split = False, niter = niter)
    poly_flux -= np.nanmin(poly_flux)
    lc_f -= poly_flux
```

Pokud je světelná křivka rozdělena na poloviny a současně je parametr `offset = True`, je vypočten medián pro obě poloviny světelné křivky a obě křivky jsou posunuty na střední hodnotu jejich mediánů.

```
if offset:
    offset1 = np.nanmedian(lc1_f) - np.nanmean([np.nanmedian(lc1_f), np.nanmedian(lc2_f)])
    offset2 = np.nanmedian(lc2_f) - np.nanmean([np.nanmedian(lc1_f), np.nanmedian(lc2_f)])
    lc1_f -= offset1
    lc2_f -= offset2
    poly_flux1 += offset1
    poly_flux2 += offset2
```

Detrendovaný světelný tok je navrácen do jednoho `lk.LightCurve` datatypu, světelná křivka je také normalizována pomocí funkce `.normalize()`.

```
lc_flat = lk.LightCurve(time = lc.time, flux = lc_f)
lc_flat = lc_flat.normalize()
if return_model:
    return lc_flat, poly_flux
else:
    return lc_flat
```

Druhou zavedenou funkcí pro odstranění zbytkové systematiky je funkce `fix_sector_jump()`, která se pokouší o opravení „skoků“ jasnosti v rámci sektoru. Jejím vstupem je světelná křivka, pro kterou chceme zavést korekci, parametr `flat_period`, primární periody dané hvězdy, parametry `min_per` a `max_per` popisující rozsah periodicity dané hvězdy, kadence daného sektoru, identifikátor hvězdy, číslo sektoru a argument `normalize`. Zdrojový kód funkce `fix_sector_jump()` je zahrnut v příloze dokumentu.

Funkce provádí korekci pouze pro hvězdy s primární periodou delší jak 3 dny, jelikož během testování bylo zjištěno, že implementace takové funkce není možná pro krátkoperiodické hvězdy, protože často docházelo k nadměrné či chybné korekci.

```
def fix_sector_jump(lc_input, period, per, min_int, max_int, exp_time, identifier, s, normalize = True):
    lc = copy.deepcopy(lc_input)
    if per > 3:
```

Funkce se dělí na dvě části. V první části provede korekci „skoku“ jasnosti uprostřed obou půlsektorů, v druhé části provede korekci skoku jasnosti uprostřed sektoru.

Nejprve je světelná křivka rozdělena na poloviny a jsou separátně nadefinovány jednotlivé dimenze. Zde v popisu funkce bude uveden případ pouze pro první polovinu světelné křivky, jelikož proces pro druhou polovinu je identický.

```
time = lc.time.value
flux = lc.flux.value
flux1, flux2 = np.split(flux, [np.nanargmax(np.diff(time)) + 1])
time1, time2 = np.split(time, [np.nanargmax(np.diff(time)) + 1])
lc1 = lk.LightCurve(time = time1, flux = flux1)
lc1_t = lc1.time.value
lc1_f = lc1.flux.value
```

Následně jsou odstraněny všechny odlehlé body nad  $5\sigma$ , díky čemuž se u zákrytových systémů odstraní zákryty, popř. u eruptivních hvězd erupce.

```
lc1p = lc1.remove_outliers(sigma = 5)
```

Iterativním odstraňováním nejsilnější přítomné periody ve světelné křivce je odstraněna veškerá periodicitá hvězdy. Pro danou kadenci se vypočítá prahová hodnota SNR, kterou daný signál v periodogramu musí mít, aby byl ze světelné křivky odstraněn. Pokud v periodogramu již není přítomná perioda nad definovanou prahovou hodnotu SNR, `for loop` je přerušen. Prahová hodnota SNR je funkcí kadence daného sektoru, vypočtena pomocí následujícího polynomu, kde kadence je ve dnech.

$$\text{SNR}_{\min} = -27.527.200 \cdot \text{cadence}^3 + 888.375 \cdot \text{cadence}^2 - 7493,9 \cdot \text{cadence} + 26,4506$$

Prahová hodnota SNR je pro 30 minutovou FFI kadenci  $\sim 7$ , pro 20 sekundovou TPF kadenci  $\sim 25$ . Periodogram je tvořen v intervalu definovaným parametry `min_per` a `max_per`. Model nejsilnější přítomné periody je vytvořen funkcí `.model()`, následně odečten od původní světelné křivky. Křivka je detrendována pomocí dříve zavedené funkce `flatten_lc()`. Nakonec jsou odstraněny odlehlé body nad  $4\sigma$ . Tento proces se iterativně opakuje dokud je přítomný signál nad prahovou hodnotu SNR, avšak maximální počet iterací je omezen na 20.

```
min_snr_c = -27527200 * exp_time ** 3 + 888375 * exp_time ** 2 - 7493.9 * exp_time + 26.4506
for i in range(20):
    pgz = lc1p.to_periodogram(oversample_factor = 2)
    pga = lc1p.to_periodogram(minimum_period = min_int, maximum_period = max_int, oversample_factor = 5)
    snr = (np.nanmax(pga.power.value) / np.nanstd(pgz.power.value))
    if snr < min_snr_c:
        break
    model = pga.model(time = lc1p.time, frequency = pga.frequency_at_max_power)
    lc1p = lk.LightCurve(time = lc1p.time, flux = (lc1p.flux.value - model.flux.value + 1))
    lc1p = flatten_lc(lc1p, period, return_model = False, split = False)
    lc1p = lc1p.remove_outliers(sigma = 4)
```

Po odstranění periodicity je světelná křivka „vyhlazena“ pomocí dříve zavedné funkce `smooth()`, s šířkou okna jako funkcí kadence daného sektoru.

$$L_{\text{window}} = \left\lfloor \frac{25}{36 \cdot \text{cadence}} \right\rfloor$$

```

lc1p_f, lc1p_t = lc1p.flux.value, lc1p.time.value
winlen = int(1 / (1.44 * exp_time))
if winlen % 2 != 1:
    winlen += 1
lc1p_fs = smooth(lc1p_f, winlen, lc1p_t, split = False)

```

Poté se stejnou šířkou okna je vypočten rozsah hodnot závislé proměnné pro každou hodnotu nezávislé proměnné. Velmi zjednodušeně si můžeme tento proces představit jako derivaci diskrétní funkce s nenulovým  $\Delta x$ .

```

hwinlen = int(winlen / 2)
jump_est1 = np.array([[t, np.ptp(lc1p_fs[i - hwinlen : i + hwinlen])] for i, t in enumerate(lc1p_t)
                      if not i < hwinlen or i > len(lc1p_t) - winlen])

```

Výsledný `np.array()` je omezen do intervalu  $\langle \frac{1}{3}; \frac{2}{3} \rangle$ , jelikož se „skok“ jasu nachází zpravidla uprostřed půlsektoru. Je nalezena nejvyšší hodnota, která by měla představovat čas „skoku“ jasnosti, také je nadefinováno číslo kadence.

```

jump_est1 = jump_est1[int(jump_est1.shape[0] / 3) : int((2 * jump_est1.shape[0]) / 3)]
jump_time1 = jump_est1[np.nanargmax(jump_est1[:, 1]), 0]
jump_cad1 = np.argwhere(lc1p_t == jump_time1).ravel()[0]

```

Jelikož už známe okamžik, ve kterém došlo ke „skoku“ jasnosti, stačí pouze zjistit jeho „výšku“. Vráťme se ke zpracované světelné křivce s odstraněnou periodicitou a odlehými body před jejím „vyhlazením“. Křivku kolem zjištěného času „skoku“ omezíme do intervalu následující šířky, která je funkcí kadence daného sektoru ve dnech.

$$L_{\text{fit}} = \left\lfloor \frac{5}{18 \cdot \text{cadence}} \right\rfloor$$

```

fit_wl = int(5 / (18 * exp_time))
if fit_wl % 2 == 1:
    fit_wl += 1
lc1_f_fit = lc1p_f[(jump_cad1 - int(fit_wl / 2)) : (jump_cad1 + int(fit_wl / 2))]
lc1_t_fit = lc1p_t[(jump_cad1 - int(fit_wl / 2)) : (jump_cad1 + int(fit_wl / 2))]

```

Pro zjištění „výšky“ „skoku“ jasnosti je zavedena funkce sigmoid, která je nadefinovaným oknem proložena.

$$f(x) = \frac{a}{1 + e^{\frac{\mu - x}{s}}} + b$$

Implementovaná do skriptu.

```

def sigmoid(x, m, s, a, b):
    return (a / (1 + np.exp((m - x) / s))) + b

```

„Výšku“ „skoku“ v kladném smyslu poté můžeme zjistit z rozdílu limit této funkce.

$$H_{\text{jump}} = \lim_{x \rightarrow \infty} f(x) - \lim_{x \rightarrow -\infty} f(x) = a$$

Parametr  $a$  funkci „škáluje“, parametr  $b$  posouvá funkci ve směru osy  $y$ , parametr  $\mu$  posouvá funkci ve směru osy  $x$  a parametr  $s$  určuje sklon funkce. Znalostí vlastností této funkce můžeme vyhodnotit kvalitu proložení a popř. proložení zamítnout. „Skok“ jasnosti totiž nastává pouze v malém procentu případů, jelikož je redukován předcházejícím správným odstraněním systematik. V praxi potom nejčastěji dochází k zamítnutí proložení kvůli konzervativně zvoleným mezím hodnot parametrů sigmoid funkce.

Pomocí knihovny `scipy` je vytvořeno optimální proložení funkce sigmoid zvoleným oknem funkcí `scipy.optimize.curve_fit()`. Parametrem `p0` definujeme počáteční hodnoty proložení pro vyšší úspěšnost. Počáteční hodnota parametru  $\mu$  je čas „skoku“, parametru  $s$  a  $a$  hodnota 0,01 a parametru  $b$  průměrná hodnota světelného toku. Hodnota  $-a$  je potom světelný tok, který musíme přičíst k druhé polovině půlsektoru, aby byl „skok“ opraven.

```
popt1, pcov1 = curve_fit(sigmoid, lc1_t_fit, lc1_f_fit,
                        p0 = [jump_time1, 0.01, 0.01, np.nanmean(lc1_f_fit)])
jump_size1 = -popt1[2]
```

Pokud proložení nesplňuje alespoň jednu z následujících podmínek, je odmítnuto.

- $|\mu - \overline{time}| > 0,05$
- $|s| > 0,05$
- $|a| \notin (0,0005; 0,1)$
- $|b - \overline{flux}| > 0,1$

Zavedeno do skriptu.

```
if abs(popt1[0] - np.nanmean(lc1_t_fit)) > 0.05
    or abs(popt1[1]) > 0.05
    or abs(jump_size1) > 0.1
    or abs(jump_size1) < 0.0005
    or abs(popt1[3] - np.nanmean(lc1_f_fit)) > 0.1:

    jump_size1 = 0
```

Zbývá se vrátit k originální světelné křivce, kterou o „skoky“ jasnosti opravíme.

```
split_cad1 = np.argwhere(lc1_t == jump_time1).ravel()
split_cad2 = np.argwhere(lc2_t == jump_time2).ravel()
split_cad1 = split_cad1[0]
split_cad2 = split_cad2[0]
flux11, flux12 = np.split(lc1_f, [split_cad1])
flux21, flux22 = np.split(lc2_f, [split_cad2])
flux12 += jump_size1
flux22 += jump_size2
flux_fix = np.concatenate([flux11, flux12, flux21, flux22])
lc_fix = lk.LightCurve(time = time, flux = flux_fix)
```

V druhé části funkce `fix_sector_jump()` se opraví „skok“ uprostřed sektoru. Principiálně se jedná o stejný proces, nicméně jsou zavedeny jiné parametry a omezení.

Znovu zavedeme prahovou hodnotu SNR jako funkce kadence daného sektoru. Byl zvolen polynom, který má mírně vyšší prahovou hodnotu.

$$\text{SNR}_{\min} = -54.902.600 \cdot \text{cadence}^3 + 1.713.400 \cdot \text{cadence}^2 - 13.248,5 \cdot \text{cadence} + 37,7867$$

Opět z křivky odstraňujeme iterativně periodicitu, dokud se nedosáhne prahové hodnoty SNR, avšak maximálně proběhne 20 iterací.

```

lcp = lc_fix.remove_outliers(sigma = 5)
for i in range(20):
    pgz = lcp.to_periodogram(oversample_factor = 2)
    pga = lcp.to_periodogram(minimum_period = min_int, maximum_period = max_int, oversample_factor = 5)
    snr = (np.nanmax(pga.power.value) / np.nanstd(pgz.power.value))
    if snr < min_snr:
        break
    model = pga.model(time = lcp.time, frequency = pga.frequency_at_max_power)
    lcp = lk.LightCurve(time = lcp.time, flux = (lcp.flux.value - model.flux.value + 1) * (u.electron / u.s))
    lcp = flatten_lc(lcp, period, return_model = False, split = False)
    lcp = lcp.remove_outliers(sigma = 4)
lcp = lcp.remove_outliers(sigma = 3)

```

Nadefinuujeme šířku okna pro vyhlazení křivky. Šířka okna pro proložení funkce sigmoid je dvojnásobná. Určit čas „skoku“ je triviální, jelikož se jedná o největší „mezeru“ ve světelné křivce.

$$L_{\text{window}} = \left\lfloor \frac{1}{3 \cdot \text{cadence}} \right\rfloor$$

```

wl = int(1 / (3 * exp_time))
flux1, flux2 = np.split(lcp.flux.value, [np.nanargmax(np.diff(lcp.time.value)) + 1])
flux1s = smooth(flux1, int(wl), 0, split = False)
flux2s = smooth(flux2, int(wl), 0, split = False)
fluxs = np.concatenate([flux1s, flux2s])
lcp_f, lcp_t = fluxs, lcp.time.value
jump_cad = np.nanargmax(np.diff(lcp_t))
lc_f_fit = lcp_f[(jump_cad - int(wl)) : (jump_cad + int(wl))]

```

Analogicky je křivka proložena s úvodními parametry p0. Pro jednoduchost je odstraněna dimenze času.

```

popt, pcov = curve_fit(sigmoid, np.arange(len(lc_f_fit)), lc_f_fit,
                       p0 = [len(lc_f_fit) / 2, 0.01, 0.01, np.nanmean(lc1_f_fit)])
jump_size = -popt[2]

```

Tentokrát jsou zavedeny mírně odlišné omezení pro odmítnutí proložení.

- $\left| \mu - \frac{\text{card}(\text{flux})}{2} \right| > \left| \frac{\text{card}(\text{flux})}{2} \right|$
- $|s| > 0,05$
- $|a| > 0,1$
- $|b - \overline{\text{flux}}| > 0,1$

Zavedeno do skriptu.

```

if abs(popt[0] - (len(lc_f_fit) / 2)) > (len(lc_f_fit) / 6)
or abs(popt[1]) > 0.05
or abs(jump_size) > 0.1
or abs(popt[3] - np.nanmean(lc_f_fit)) > 0.1:

    jump_size = 0

```

Zbývá opravit „skok“ jasnosti, vložit světelnou křivku zpět do jediného `lk.LightCurve` datatypu a normalizovat pomocí funkce `.normalize()`, v případě, že na začátku funkce byl zvolen parametr `normalize = True`.

```

flux1, flux2 = np.split(lc_fix.flux.value, [np.nanargmax(np.diff(lc_fix.time.value)) + 1])
flux2 += jump_size
flux_fix = np.concatenate([flux1, flux2])
lc_fix = lk.LightCurve(time = lc_fix.time, flux = flux_fix)
if normalize:
    lc_fix = lc_fix.normalize()
else:
    return lc_fix

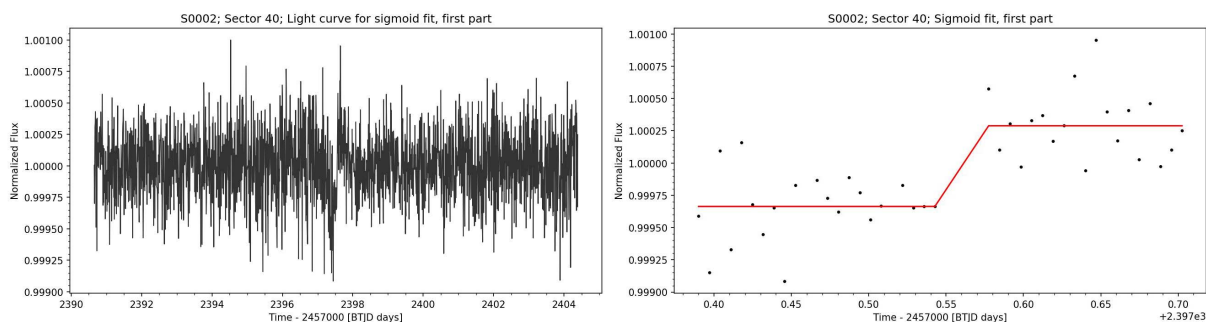
```

Aplikace funkcí `flatten_lc()` a `fix_sector_jump()` ve skriptu vypadá následovně. Originální světelná křivka, na kterou byla aplikována pouze funkce `regression()` je zachována. Na světelnou křivku s odstraněnou systematikou je aplikována funkce `flatten_lc()` dvakrát, poprvé před funkcí `fix_sector_jump()` s argumentem `split = True`, podruhé po funkci `fix_sector_jump()` s argumentem `split = False`. Křivkám je navržena dimenze `flux_err`.

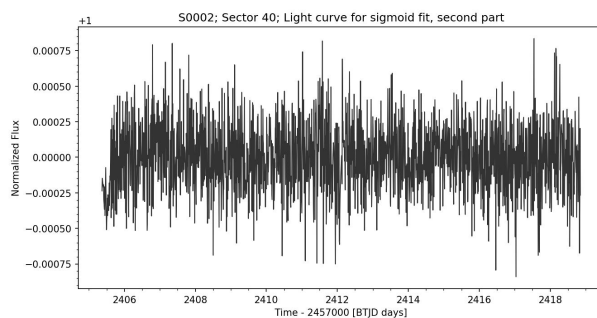
```

lc_flat = flatten_lc(lc_regress, flat_period, return_model = False)
lc_flat = fix_sector_jump(lc_flat, flat_period, target_period, min_per, max_per,
                          exp_time, identifier, s)
lc_flat = flatten_lc(lc_flat, flat_period, return_model = False, split = False, offset = False)
lc_regress = lk.LightCurve(time = lc_regress.time.value,
                           flux = lc_regress.flux.value,
                           flux_err = lc.flux_err.value)
lc_flat = lk.LightCurve(time = lc_flat.time.value,
                        flux = lc_flat.flux.value,
                        flux_err = lc.flux_err.value / np.nanmedian(lc_regress.flux.value))

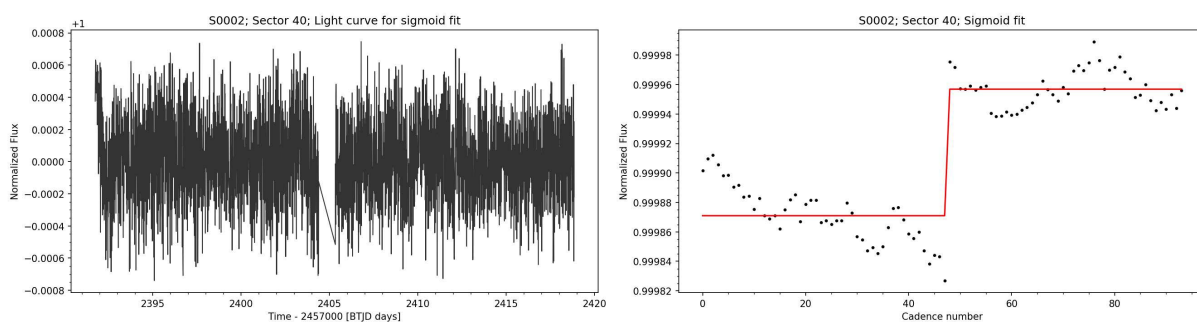
```



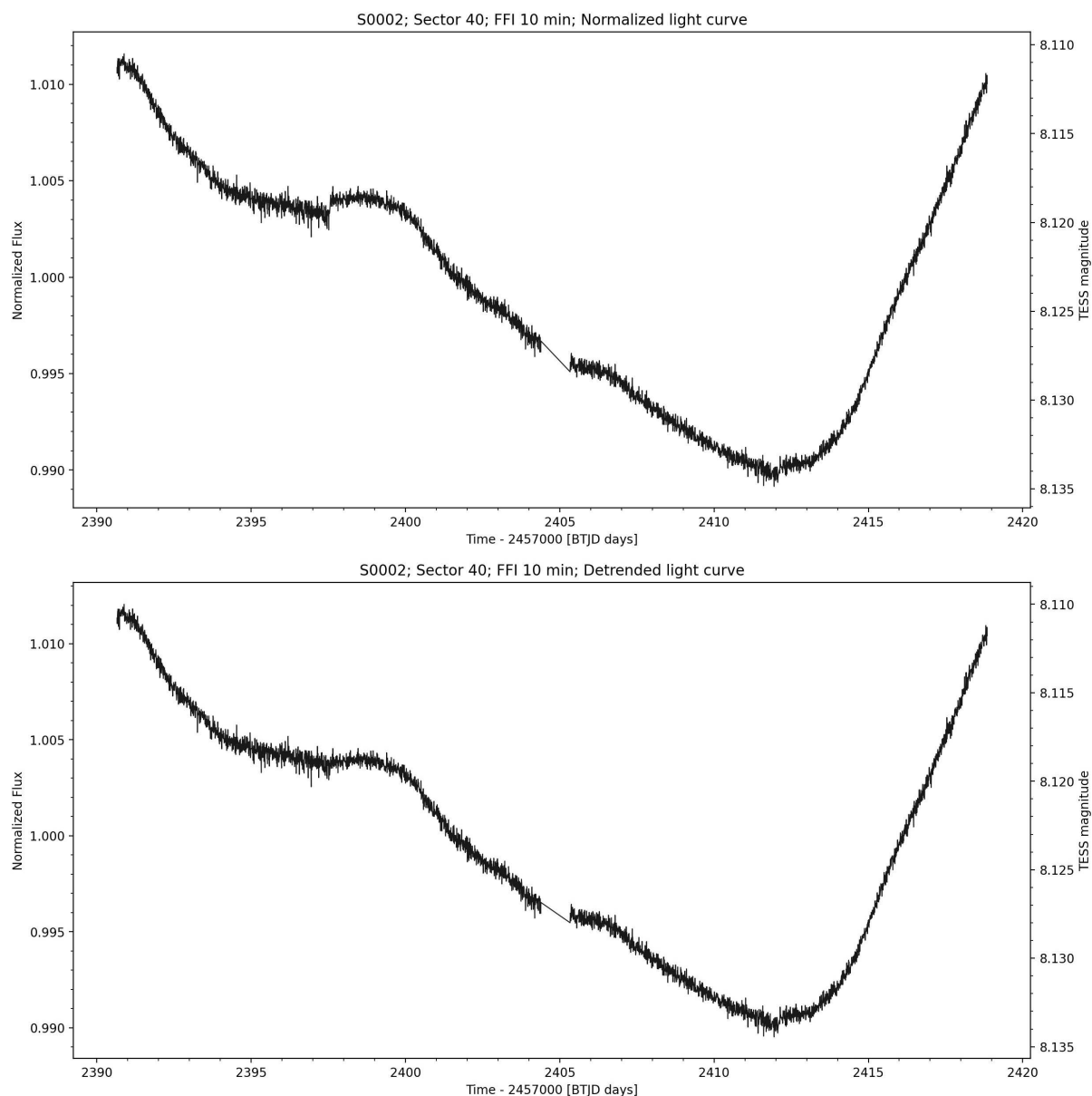
Obrázek 42: Nalevo: zpracovaná světelná křivka první poloviny sektoru 40 hvězdy S0002 s odstraněnou periodicitou a odlehlými body pro proložení. Uprostřed křivky lze vidět „skok“ jasnosti. Napravo: detail na „skok“ jasnosti a jím proloženou sigmoid funkcí (vlastní tvorba).



Obrázek 43: Zpracovaná světelná křivka druhé poloviny sektoru 40 hvězdy S0002 s odstraněnou periodicitou a odlehlými body pro proložení. „Skok“ jasnosti není dostatečně velký, tudíž parametry proložení byly odmítnuty (vlastní tvorba).



Obrázek 44: Nalevo: zpracovaná světelná křivka sektoru 40 hvězdy S0002 s odstraněnou periodicitou a odlehlými body pro proložení. Napravo: detail na „skok“ jasnosti uprostřed sektoru a jím proloženou sigmoid funkcí (vlastní tvorba).



Obrázek 45: Nahoře: světelná křivka hvězdy S0002 v sektoru 40 s odstraněným rozptýleným světlem. Dole: světelná křivka s opravenými „skoky“ jasnosti. Nejvýraznější změna lze pozorovat v první polovině světelné křivky. Druhá polovina světelné křivky nebyla upravena, jelikož parametry proložení nesplňují definované podmínky (vlastní tvorba).



### 6.2.1.11 Odstranění kadencí s jasným pozadím

Popsaná metoda korekce rozptýleného světla je velmi dobrá, avšak není bezchybná. V některých sektorech světelný tok rozptýleného světla je několikanásobně silnější než světelný tok samotné hvězdy a i po korekci může část světelné křivky obsahovat zbytkový trend. V případě, že daná hvězda má nastavený parametr `bright = True`, jsou odstraněny kadence s příliš jasným pozadím. Ztráta několika procent dat pro většinu hvězd není velkým zásahem do jejich analýzy, avšak zákrytové systémy či eruptivní hvězdy mají zachována veškerá dostupná data.

Model pozadí extrahovaný z funkce `regression()` je použit pro výpočet prahové hodnoty světelného toku pozadí. Nejprve je k modelu přičten jeho medián, také je nadefinována proměnná ukládající dimenzi času.

```
model_shift = model + np.nanmedian(model)
model_time = np.array(lc_regress.time.value)
```

Z modelu jsou odstraněny všechny hodnoty větší jak 0,9 kvantil, poté je definována proměnná `median`.

```
model_clip = model_shift[model_shift < np.nanquantile(model_shift, 0.9, axis = 0)]
median = np.nanmedian(model_clip)
```

Dále je definována proměnná `flux_range`, vypočtena jako 0,8 kvantil – 0,2 kvantil.

```
flux_range = np.nanquantile(model_clip, 0.8, axis = 0) - np.nanquantile(model_clip, 0.2, axis = 0)
```

Prahová hodnota světelného toku je definována následovně. Veličina `median(model)` je medián původního modelu pozadí.

$$F_{\max} = 6 \cdot \text{flux\_range} + \frac{\text{median}}{120} - \text{median}(\text{model})$$

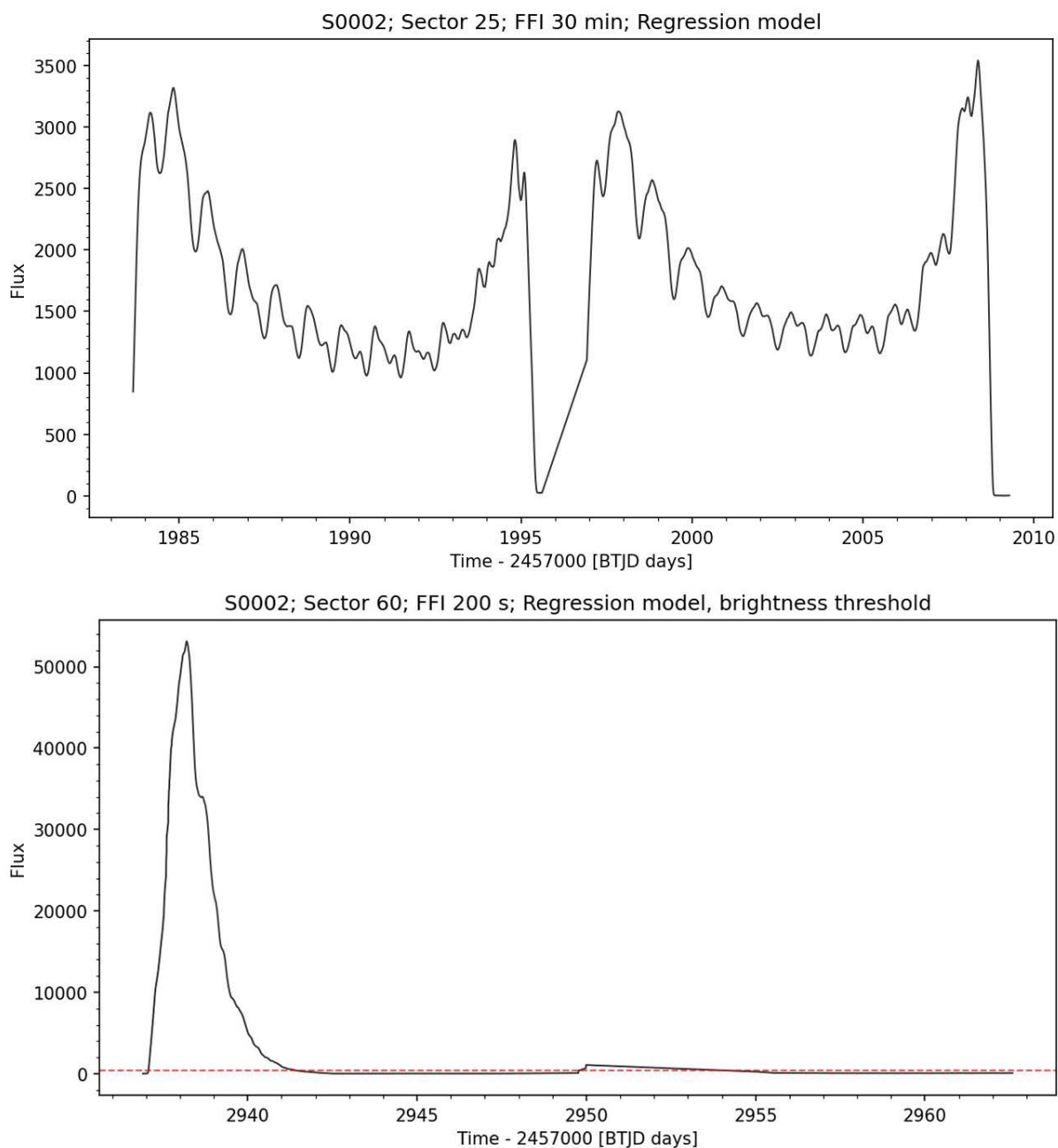
Implementováno do skriptu.

```
max_bg_flux = 6 * flux_range + median / 120 - np.nanmedian(model)
```

Pomocí funkce `.remove_rows()` jsou z obou světelných křivek odstraněny kadence s hodnotou pozadí větší než prahová. Světelná křivka s odstraněným rozptýleným světlem je také normalizována. V případě, že `bright = False`, proběhne pouze normalizace světelné křivky.

```
bright_cadences = np.argwhere(model > max_bg_flux).ravel().tolist()
lc_regress.remove_rows(bright_cadences)
lc_regress = lc_regress.normalize()
lc_flat.remove_rows(bad_cadences)
```

Výpočet prahové hodnoty je výsledkem nesčetného testování. Je schopen rozpoznat mezi dvěma situacemi, které nastávají. V některých sektorech se světelný tok rozptýleného světla výrazně nemění, tudíž nemá smysl odstraňovat žádné kadence. V některých sektorech dochází avšak na začátku či na konci půlsektoru k prudkému zvýšení světelného toku rozptýleného světla a kvalita světelné křivky je v těchto částech oproti celé světelné křivce nízká, tudíž je vhodné tyto části odstranit.



Obrázek 46: Modely rozptýleného světla generované funkcí `regression()` hvězdy S0002 v sektorech 25 (nahore) a 60 (dole). Světelný tok rozptýleného světla se v sektoru 25 mění poměrně málo a zůstává po celou dobu trvání pozorování na přiměřené hladině, všechny kadence mají podobnou kvalitu a tudíž není potřeba žádné odstraňovat. Světelný tok rozptýleného světla v sektoru 60 prudce naroste na začátku pozorování, čímž výrazně snižuje kvalitu zasažených kadcencí. Prahová hodnota světelného toku je vyznačena červenou přerušovanou čarou (vlastní tvorba).

### 6.2.1.12 Odstranění odlehlých hodnot

Pomocí funkce `scipy.stats.norm.ppf()` je vypočtena odpovídající hodnota násobků směrodatné odchylky od mediánu pro nastavené parametry `outlier_l` a `outlier_h` udávající procentuální počet odlehlých hodnot ve světelné křivce.

```
sigma_low = abs(scipy.stats.norm.ppf(outlier_l / 200))
sigma_high = abs(scipy.stats.norm.ppf(outlier_h / 200))
```

Odlehlé hodnoty jsou odstraněny ze světelných křivek pomocí funkce `.remove_outliers()` a jejích parametrů `sigma_lower` a `sigma_upper`.

```
lc_regress = lc_regress.remove_outliers(sigma_lower = sigma_low, sigma_upper = sigma_high)
lc_flat = lc_flat.remove_outliers(sigma_lower = sigma_low, sigma_upper = sigma_high)
```

Tímto krokem končí zpracování světelných křivek.

### 6.2.1.13 Uložení světelných křivek

Světelné křivky jsou nejprve konvertovány do `pd.DataFrame` datatypu pomocí funkce `.to_pandas()`, poté uloženy do CSV souboru pomocí funkce `.to_csv()`. Obě světelné křivky jsou také zvlášť zařazeny do seznamů společně s číslem sektoru.

```
lc_regress_pd = lc_regress.to_pandas()
lc_flat_pd = lc_flat.to_pandas()
lc_regress_pd.to_csv('path', header = None, index = False)
lc_flat_pd.to_csv('path', header = None, index = False)
df_list.append([s, lc_regress_pd])
df_list_save.append([s, lc_regress])
df_list_flat.append([s, lc_flat_pd])
```

### 6.2.1.14 Vytvoření periodogramu

Pro každou světelnou křivku jsou vytvořeny tři periodogramy. První periodogram `pga` slouží k výpočtu mediánu a směrodatné odchylky. Parametr `oversample_factor` je nastaven na hodnotu 1. Bude uvedeno zpracování pouze světelné křivky `lc_regress`, jelikož zpracování pro světelnou křivku `lc_flat` je identické.

```
pga = lc_regress.to_periodogram(oversample_factor = 1)
```

Periodogram `pg` pro určení primární periody je omezen parametry `min_per` a `max_per`, parametr `oversample_factor` je nastaven na hodnotu 4.

```
pg = lc_regress.to_periodogram(minimum_period = min_per, maximum_period = max_per, oversample_factor = 4)
```

Kolem primární periody je vytvořen interval, pro který se vytvoří periodogram s vysokým rozlišením `pg_c` pro přesné určení periody a amplitudy. Krajní meze intervalu jsou definovány jako funkce primární periody.

$$P_{min,max} = P \pm \left( \frac{P}{10} \right)^{\frac{3}{2}}$$

Implementován do skriptu.

```
min_int = pg.period_at_max_power.value - (pg.period_at_max_power.value / 10) ** 1.5
max_int = pg.period_at_max_power.value + (pg.period_at_max_power.value / 10) ** 1.5
```

Parametr `oversample_factor` je také nadefinován jako funkce primární periody.

$$F_{os} = \frac{320,629 \cdot P + 339,536}{P^2 + 15,3714 \cdot P + 59,0423}$$

Zaveden jako funkce ve skriptu.

```
def find_os_factor(p):
    return (339.536 + 320.629 * p) / (p ** 2 + 15.3714 * p + 59.0423)

os_fac = find_os_factor(pg.period_at_max_power.value)
```

Hodnota `oversample_factor` je poté  $\sim 11$  pro hvězdu s periodou 10 dní,  $\sim 6$  pro hvězdu s periodou 0,01 dne. S definovanými parametry můžeme zkonstruovat periodogram `pg_c` pro určení periody a amplitudy.

```
pg_c = lc_regress.to_periodogram(minimum_period = min_int, maximum_period = max_int,
                                oversample_factor = os_fac)
```

Poměr signálu a šumu (SNR) můžeme vypočítat jako podíl rozdílu nejsilnějšího signálu a mediánu a směrodatné odchylky.

$$\text{SNR} = \frac{A_{max} - \mu}{s}$$

Aplikované do skriptu.

```
max_power = pg_c.max_power.value
median = np.nanmedian(pga.power.value)
std = np.nanstd(pga.power.value)
snr = (max_power - median) / std
```

### 6.2.1.15 Vytvoření fázové křivky

Z dříve vytvořeného periodogramu definujeme primární periodu hvězdy pro složení světelné křivky. V případě, že dané hvězda má parametr `double = True`, je použita dvojnásobná perioda.

```
if double:
    period = 2 * pg_c.period_at_max_power
else:
    period = pg_c.period_at_max_power
```

Fázová světelná křivka je vytvořena pomocí funkce `lk.LightCurve.fold()`, přičemž parametr `normalize_phase` je nastaven na hodnotu `True`.

```
lc_phase = lc_regress.fold(period = period, normalize_phase = True)
```

Fázové křivky se tradičně zobrazují na dvě fáze pro lepší viditelnost. Stačí světelný tok zkopírovat a odečíst, resp. přičíst půl fáze. Hodnoty poté navrátíme do jediného `lk.FoldedLightCurve` datatypu.

```
lc_phase = lk.FoldedLightCurve(time = np.array([lc_phase.phase.value - 0.5,
                                                lc_phase.phase.value + 0.5]).ravel(),
                                flux = np.array(2 * [lc_phase.flux.value]).ravel())
```

Tímto končí zpracování ve třech hlavních `for loop`, které zpracovávají data jednotlivých sektorů. Následuje zpracování kompletní světelné křivky všech sektorů dostupných pro danou hvězdu.

### 6.2.1.16 Úvod zpracování světelných křivek všech sektorů

Seznamy `sectors`, `df_list_save`, `time_labels`, `df_list` a `df_list_flat` jsou seřazeny podle pořadí sektorů a je zaveden nový seznam `df_list_flat_fix`. Světelná křivka `lc_regress` je uložena do CSV souboru.

```
sectors.sort()

df_list_save.sort(key = lambda item: item[0])
df_list_save = [item[1] for item in df_list_save]
merged_df_save = pd.concat(df_list_save, axis = 0)
merged_df_save.to_csv('path', header = None, index = False)

time_labels.sort(key = lambda item: item[0])
time_labels = [item[1] for item in time_labels]

df_list.sort(key = lambda item: item[0])
df_list = [item[1] for item in df_list]

df_list_flat.sort(key = lambda item: item[0])
df_list_flat = [item[1] for item in df_list_flat]
df_list_flat_fix = []
```

### 6.2.1.17 Oprava mezisektorových skoků jasnosti

Pro hvězdy s primární periodou nad 4 dny je provedena korekce mezisektorových skoků jasnosti. Funkce se pokouší o lepší návaznost jednotlivých sektorů. Hvězdy s kratšími periodami takovou korekci nepotřebují, jelikož samotná normalizace křivky je dostačující.

Funkce prokládá světelnou křivku každého sektoru funkcí sinus, která dobře aproximuje průběh jasů proměnných hvězd v čase.

$$f(x) = a \cdot \sin[f \cdot (x - b)] + c$$

Zavedená do skriptu.

```
def sine(x, f, a, b, c):
    return a * np.sin(f * (x - b)) + c
```

Parametr  $a$  je amplituda, parametr  $f$  je frekvence, parametr  $b$  je fáze a parametr  $c$  značí střední hodnotu funkce. Funkce je pomocí funkce `scipy.optimize.curve_fit()` proložena světelnou křivkou každého sektoru. Pomocí parametru  $c$  zjistíme „skutečnou“ hodnotu mediánu sektoru, která je poté z křivky odečtena. Je vypočten medián všech křivek před korekcí, kterým je každá křivka po korekci vydělena. Jsou zvoleny úvodní parametry pro proložení, a to  $f = \frac{2\pi}{P}$ ,  $a = 0,01$ ,  $b = 1$  a  $c = 1$ . Pokud je odchylka vypočtené střední hodnoty vyšší jak 0,01 je proložení zamítnuto a použita nulová odchylka.

```
lc_i_t = np.concatenate([lc.time.value for lc in df_list_flat], axis = 0)
lc_i_f = np.concatenate([lc.flux.value for lc in df_list_flat], axis = 0)
lc_i = lk.LightCurve(time = lc_i_t, flux = lc_i_f)
pg_i = lc_i.to_periodogram(minimum_period = 2 * u.day, oversample_factor = 10)
for s, lc in zip(sectors, df_list_flat):
    fit_t = lc.time.value
    fit_f = lc.flux.value
    popt, pcov = curve_fit(sine, fit_t, fit_f, p0 = [(2 * math.pi) / pg_i.period_at_max_power.value,
                                                    0.01, 1, 1])

    jump = 1 - popt[3]
    if jump > 0.01:
        jump = 0
```

```

lc_flux_cor = lc.flux.value + jump
lc_fix = lk.LightCurve(time = lc.time, flux = lc_flux_cor, flux_err = lc.flux_err)
df_list_flat_fix.append(lc_fix)
median = np.nanmedian(np.concatenate([lc.flux.value for lc in df_list_flat_fix], axis = 0))
for i, lc in enumerate(df_list_flat_fix):
    lc = lk.LightCurve(time = lc.time, flux = lc.flux.value / median, flux_err = lc.flux_err)
    df_list_flat_fix[i] = lc

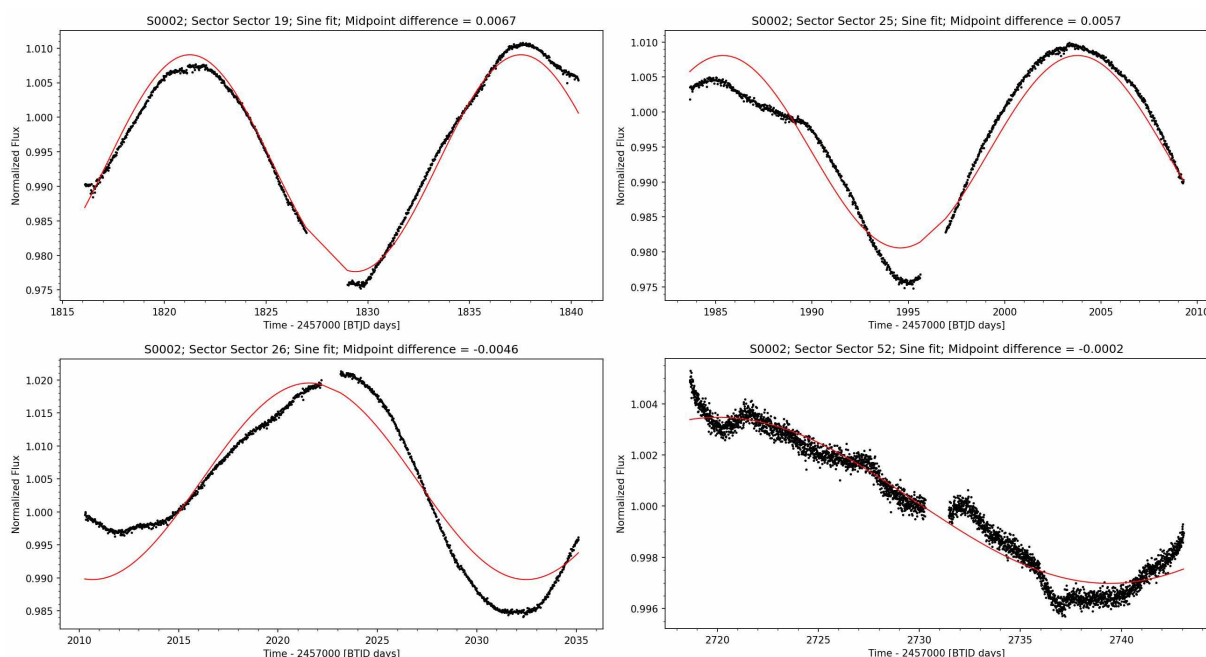
```

Opravená světelná křivka je uložena do CSV souboru.

```

merged_df_save_flat = pd.DataFrame({'time' : np.concatenate([lc.time.value for lc in df_list_flat_fix]),
                                   'flux' : np.concatenate([lc.flux.value for lc in df_list_flat_fix]),
                                   'flux_err' : np.concatenate([lc.flux_err.value for lc in df_list_flat_fix])})
merged_df_save_flat.to_csv('path', header = None, index = False)

```



Obrázek 47: Ukázka čtyř světelných křivek hvězdy S0002 v sektorech 19, 25, 26 a 52. Proložené funkce sinus jsou znázorněny červeně (vlastní tvorba).

### 6.2.1.18 Vytvoření periodogramu, výpočet periody a její chyby

Nejprve všechny světelné křivky musí být vloženy do jediného `lk.LightCurve()` datatypu. Opět bude popsáno zpracování pouze pro jednu z křivek, jelikož zpracování detrendované křivky je od této chvíle identické.

```

lc_all = lk.LightCurve(time = np.concatenate([lc.time.value for lc in df_list]),
                       flux = np.concatenate([lc.flux.value for lc in df_list]),
                       flux_err = np.concatenate([lc.flux_err.value for lc in df_list]))
lc_all_flat = lk.LightCurve(time = np.concatenate([lc.time.value for lc in df_list_flat_fix]),
                           flux = np.concatenate([lc.flux.value for lc in df_list_flat_fix]),
                           flux_err = np.concatenate([lc.flux_err.value for lc in df_list_flat_fix]))

```

Je zavedena funkce `create_pg()`, která efektivně vytvoří periodogram bez přítomného aliasingu, způsobeného různými kadencemi jednotlivých sektorů. Vstupem funkce je seznam světelných křivek a jejich kadencí. Vytvořený periodogram bude sloužit k určení primární periody a pro vizualizaci.

Periodogram pro vizualizaci chceme vytvořit v rozsahu period od 0,01 dne do 40 dní. Sektory s kadencí 200 sekund, 120 sekund a 20 sekund nevytváří žádný aliasing v periodogramu s maximální frekvencí  $100 \text{ den}^{-1}$ , avšak sektory s kadencí 30 minut a 10 minut aliasing vytvářejí, proto jejich světelné křivky musí být použity pouze v rozsahu frekvencí, ve kterých aliasing nevytváří. Jsou vytvořeny tři seznamy obsahující indexy světelných křivek rozdělené podle jejich kadence.

```
def create_pg(lcs, c):
    if '30 min' in c:
        ind1 = [i for i, x in enumerate(c) if x == '30 min']
    else:
        ind1 = []
    if '10 min' in c:
        ind2 = [i for i, x in enumerate(c) if x == '10 min']
    else:
        ind2 = []
    if '200 s' in c or '2 min' in c or '20 s' in c:
        ind3 = [i for i, x in enumerate(c) if x == '200 s' or x == '2 min' or x == '20 s']
    else:
        ind3 = []
```

Jsou vytvořeny tři světelné křivky obsahující všechny sektory, dále všechny sektory kromě 30 minutových FFI a nakonec pouze 200, 120 a 20 sekundové sektory.

```
lc1 = lk.LightCurve(time = np.concatenate([lc.time.value for i, lc in enumerate(lcs)
                                             if i in ind1 or i in ind2 or i in ind3], axis = 0),
                    flux = np.concatenate([lc.flux.value for i, lc in enumerate(lcs)
                                             if i in ind1 or i in ind2 or i in ind3], axis = 0))

if len(ind2) == 0 and len(ind3) == 0:
    lc2 = lk.LightCurve(time = [], flux = [])
else:
    lc2 = lk.LightCurve(time = np.concatenate([lc.time.value for i, lc in enumerate(lcs)
                                                if i in ind2 or i in ind3], axis = 0),
                        flux = np.concatenate([lc.flux.value for i, lc in enumerate(lcs)
                                                if i in ind2 or i in ind3], axis = 0))

if len(ind3) == 0:
    lc3 = lk.LightCurve(time = [], flux = [])
else:
    lc3 = lk.LightCurve(time = np.concatenate([lc.time.value for i, lc in enumerate(lcs)
                                                if i in ind3], axis = 0),
                        flux = np.concatenate([lc.flux.value for i, lc in enumerate(lcs)
                                                if i in ind3], axis = 0))
```

Pro každou světelnou křivku je vytvořen periodogram takovým způsobem, aby byl minimalizován šum a zároveň bylo zabráněno vzniku aliasingu. Pomocí argumentu `frequency` a funkce `np.linspace()` je vytvořen `np.array()` frekvencí, pro které je vytvořen periodogram.

Ze světelné křivky obsahující všechny sektory je vytvořen periodogram v intervalu  $\langle \frac{1}{40}; 22 \rangle \text{ den}^{-1}$ , ze světelné křivky obsahující všechny sektory kromě 30 minutových FFI sektorů je vytvořen periodogram v intervalu  $\langle 22; 65 \rangle \text{ den}^{-1}$  a ze světelné křivky obsahující 200, 120 a 20 sekundovou kadenci je vytvořen periodogram v intervalu  $\langle 65; 100 \rangle \text{ den}^{-1}$ .

Pokud světelná křivka neobsahuje sektory s kadencí 200, 120 nebo 20 sekund, periodogram není vykreslen až po frekvenci  $100 \text{ den}^{-1}$ , ale je omezen uvedenými hodnotami.

Ve funkci `np.linspace()` je vytvořeno takové množství prvků, aby oddělení jednotlivých frekvencí odpovídalo vzorkování  $\frac{1}{3 \cdot 10^4} \text{ den}^{-1}$ .

```

if len(lc1.time.value) > 0:
    pg1 = lc1.to_periodogram(frequency = np.linspace(1 / 40, 22, 65925), ls_method = 'fast')
else:
    pg1 = None
if len(lc2.time.value) > 0:
    pg2 = lc2.to_periodogram(frequency = np.linspace(22, 65, 129000), ls_method = 'fast')
else:
    pg2 = None
if len(lc3.time.value) > 0:
    pg3 = lc3.to_periodogram(frequency = np.linspace(65, 100, 105000), ls_method = 'fast')
else:
    pg3 = None

```

Spojité periodogram pro celý rozsah frekvencí je vytvořen spojením jednotlivých periodogramů do jediného `lk.periodogram.LombScarglePeriodogram` datatypu.

```

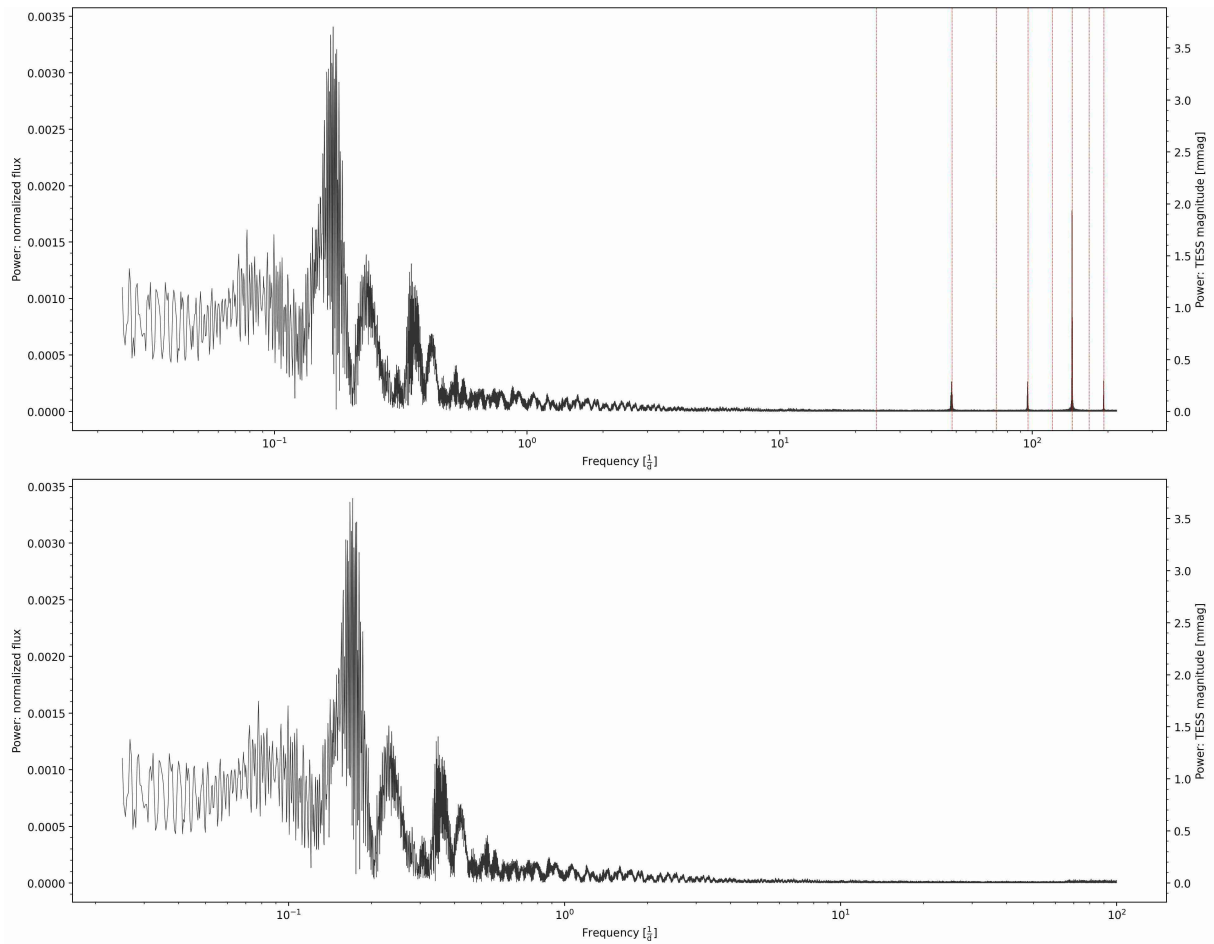
if pg1 != None and pg2 != None and pg3 != None:
    pg = lk.periodogram.LombScarglePeriodogram(frequency = np.concatenate([pg1.frequency.value,
                                                                           pg2.frequency.value,
                                                                           pg3.frequency.value], axis = 0)
                                               * (1 / u.day),
                                               power = np.concatenate([pg1.power.value,
                                                                        pg2.power.value,
                                                                        pg3.power.value], axis = 0)
                                               * u.dimensionless_unscaled)
elif pg1 != None and pg2 != None:
    pg = lk.periodogram.LombScarglePeriodogram(frequency = np.concatenate([pg1.frequency.value,
                                                                           pg2.frequency.value], axis = 0)
                                               * (1 / u.day),
                                               power = np.concatenate([pg1.power.value,
                                                                        pg2.power.value], axis = 0)
                                               * u.dimensionless_unscaled)
else:
    pg = pg1
return pg

```

Funkce v skriptu je jednoduše aplikována.

```
pg = create_pg(df_list, time_labels)
```





Obrázek 48: Periodogramy hvězdy S0017. Horní periodogram byl vytvořen tradičním způsobem ze všech světelných křivek. Červenými přerušovanými vertikálními čarami jsou vyznačeny teoretické frekvence aliasingu. Pro konstrukci dolního periodogramu byla použita funkce `create_pg()`. Kolem  $65 \text{ den}^{-1}$  lze pozorovat změna úrovně šumu způsobená změnou zdrojové světelné křivky pro konstrukci této části periodogramu, obsahující méně dat (vlastní tvorba).

Je zavedena funkce `error()`, jejímž vstupem je právě vytvořený periodogram, světelná křivka všech sektorů a parametry dané hvězdy `min_per` a `max_per` omezující rozsah periodicity.

```
def error(pg, lc, min_per, max_per):
```

Funkce `error()` sestojí velmi detailní periodogram okolo primární periody v definovaných mezích, pomocí kterého je určena perioda, amplituda a chyba periody, popř. frekvence.

Vstupní periodogram je omezen do definovaného intervalu a je určena frekvence primární proměnnosti dané hvězdy.

```
min_freq = 1 / max_per
max_freq = 1 / min_per
ind = np.argwhere((pg.frequency.value > min_freq) & (pg.frequency.value < max_freq)).ravel()
pg_crop_f = pg.frequency.value[ind]
pg_crop_p = pg.power.value[ind]
```

Detailní periodogram je vytvořen v intervalu  $\langle \frac{f_{max}}{2}; 2 \cdot f_{max} \rangle$   $\text{den}^{-1}$ . Počet prvků ve funkci `np.linspace()` je nastaven tak, aby na každou jednotku  $\text{den}^{-1}$  připadalo  $10^5$  vzorků.

```
f_range = max_freq - min_freq
max_f = pg_crop_f[np.nanargmax(pg_crop_p)]
pg_p = lc.to_periodogram(frequency = np.linspace(max_f / 1.9, max_f * 1.9, int(10000 * f_range)),
                        ls_method = 'fast')
```

Je definována perioda, frekvence a síla primárního signálu.

```
period = 1 / pg_p.frequency.value
frequency = pg_p.frequency.value
power = pg_p.power.value
```

Peak primární frekvence v periodogramu může svým tvarem připomínat normální rozdělení. Čím více se vzdalujeme od peaku, tím více se blížíme nulové síle signálu. Chyba dané frekvence by se poté mohla vypočítat jako FWHM (full width at half maximum) daného peaku. Pokud bychom vytvořili fázovou křivku na frekvenci s téměř nulovou silou nacházející se těsně vedle peaku, maxima a minima světelné křivky by se ve fázové křivce viditelně „překryly“. Tato metoda také dobře popisuje hvězdy s vysokou modulací periody, jako např. rotující proměnné hvězdy.

Porovnáním této metody výpočtu chyby s Generalizovaným Lomb-Scargle periodogramem [9], který chybu počítá z chyb měření, zjistíme, že je popisovaná metoda výpočtu chyby spíše konzervativní, avšak lépe modeluje reálné situace nastávající v periodogramech proměnných hvězd.

Místo šířky na poloviční síle dané frekvence je použita hodnota  $\frac{2}{3}$ , která se lépe shoduje s ostatními metodami odhadu chyby, navíc méně dochází k chybné detekci u multiperiodických hvězd.

Pomocí dříve definované funkce `sr()` je chyba zaokrouhlena na jednu platnou číslici.

```
hv = np.ndarray.flatten(np.argwhere(power > ((np.nanmax(pg_p.power.value) * (2 / 3))))
li, ui = np.nanmin(hv), np.nanmax(hv)
hwhm_per = abs((period[li] - period[ui]) / 2)
hwhm_freq = abs((frequency[li] - frequency[ui]) / 2)
if hwhm_per < 1:
    hwhm_per = float(sr(hwhm_per, 1))
else:
    hwhm_per = int(sr(hwhm_per, 1))
if hwhm_freq < 1:
    hwhm_freq = float(sr(hwhm_freq, 1))
else:
    hwhm_freq = int(sr(hwhm_freq, 1))
return hwhm_per, hwhm_freq, pg_p
```

Funkce je jednoduše aplikována ve skriptu.

```
per_err, freq_err, pg_peak = error(pg, lc_all, min_per, max_per)
```

Z detailního periodogramu je definována perioda hvězdy, pokud má daná hvězda parametr `double = True`, je použita dvojnásobná perioda. Je také vypočteno SNR.

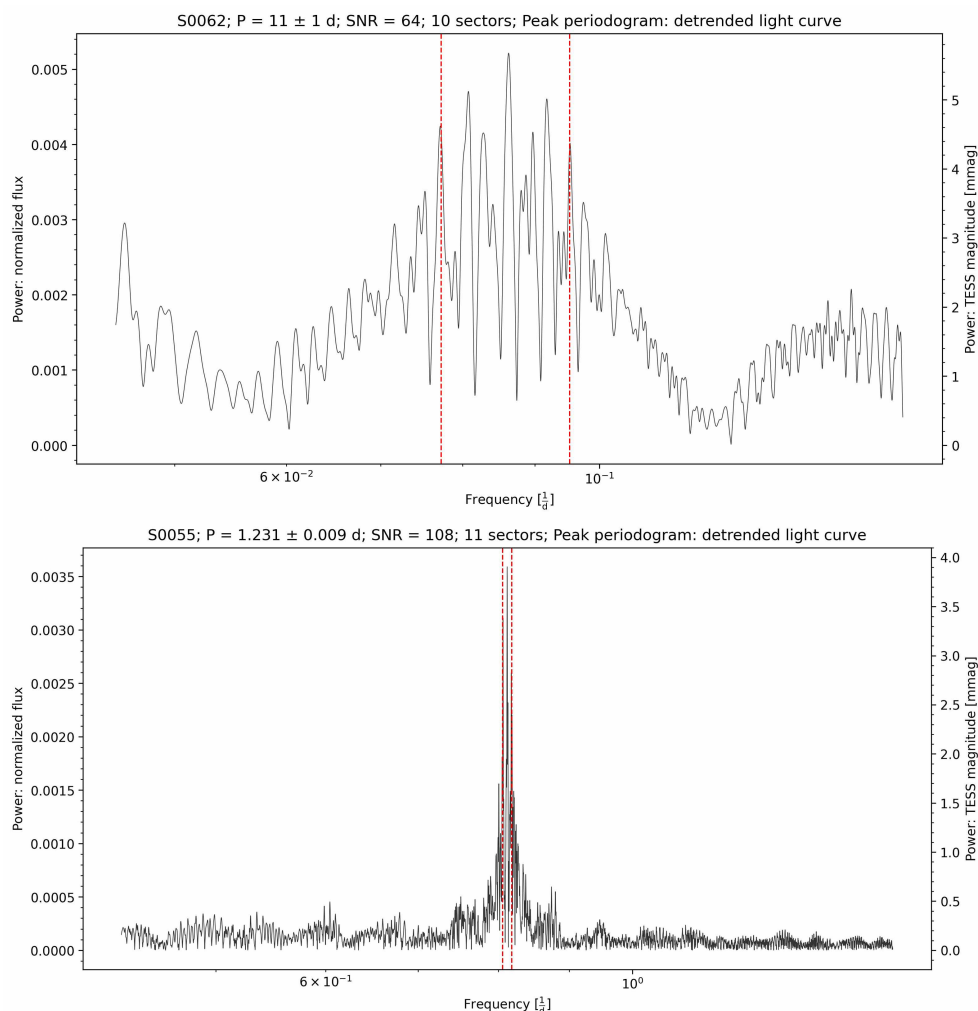
```
if double:
    period = 2 * pg_peak.period_at_max_power
else:
    period = pg_peak.period_at_max_power
frequency = 1 / period
snr = (pg_peak.max_power.value - np.nanmedian(pg.power.value)) / np.nanstd(pg.power.value)
```

Je definována funkce `sr()`, jejímž vstupem je perioda a její chyba. Funkce zaokrouhlí periodu na stejný počet desetinných míst jako má její chyba.

```
def sr(v, e):
    if e < 1:
        v, e = str(v), str(e)
        v = v[: v.find('.') + len(e[e.find('.') + 1:]) + 1]
    else:
        v, e = str(v), str(e)
        v_i = len(v[: v.find('.')])
        v = v[: v.find('.') - len(e) + 1]
        v += (v_i - len(v)) * '0'
    if v.find('.') == -1:
        return int(v)
    else:
        return float(v)
```

Funkce je jednoduše použita ve skriptu.

```
period_sig = sr(period.value, per_err)
frequency_sig = sr(frequency.value, freq_err)
```



Obrázek 49: Detailní periodogramy hvězd S0062 (nahore) a S0055 (dole). Chyba je vy-  
značena červenými přerušovanými vertikálními čarami. Hvězda S0062 evidentně prochází  
mnohem větší modulací periody (vlastní tvorba).

Z periodogramu celého rozsahu frekvencí jsou odstraněny NaN hodnoty a je uložen do CSV souboru.

```
pg_err1 = np.argwhere(np.isnan(pg.frequency.value))
pg_err2 = np.argwhere(np.isnan(pg.power.value))
nan = np.unique(np.concatenate([pg_err1, pg_err2], axis = 0).astype('int32'))
pg_freq = np.delete(pg.frequency.value, nan, axis = 0)
pg_power = np.delete(pg.power.value, nan, axis = 0)
pg_df = pd.DataFrame({'frequency' : pg_freq, 'power' : pg_power})
pg_df.to_csv('path', header = None, index = False)
```

### 6.2.1.19 Výpočet epochy a její chyby

Je zavedena funkce `bin_lc()`, jejímž vstupem je `np.array()`, počet binů a metoda. Funkce provede binning do nadefinovaného počtu binů. Pomocí argumentu `method` je možnost výběru mezi výpočtem průměru nebo mediánu binu.

```
def bin_lc(arr, bins, method):
    binsize = int(len(arr) / bins)
    arr = arr[:-(len(arr) % binsize)]
    arr_s = np.array(np.array_split(arr, int(len(arr) / binsize)))
    if method == 'median':
        return np.nanmedian(arr_s, axis = 1)
    elif method == 'mean':
        return np.nanmean(arr_s, axis = 1)
```

Funkce `bin_lc()` je použita pro výpočet okamžiku maxima. Epocha je definována jako okamžik maxima světelné křivky, v případě zákrytových systémů se definuje okamžik primárního minima. Parametry proměnnosti pro zákrytové systémy jsou určeny manuálně. Binovaná fázová křivka použitá pro výpočet epochy je vytvořena následovně. Je použito 100 binů.

```
bins = 100
lc_all_phase_bin_i = lc_all.fold(period = period, normalize_phase = True)
lc_all_phase_bin_i = lk.FoldedLightCurve(time = bin_lc(np.concatenate([lc_all_phase_bin_i.phase.value - 0.5,
                                                                    lc_all_phase_bin_i.phase.value + 0.5],
                                                                    axis = 0).ravel(),
                                                                    bins, method = 'mean'),
    flux = bin_lc(np.array(2 * [lc_all_phase_bin_i.flux.value]).ravel(),
    bins, method = 'median'))
```

Dále je určena fáze maxima.

```
max_phase = lc_all_phase_bin_i.phase.value[np.nanargmax(lc_all_phase_bin_i.flux.value)]
```

Fáze v `lk.FoldedLightCurve` se vztahuje k počáteční hodnotě času světelné křivky, tudíž epochu maxima vypočítáme následovně.

$$E_{max} = T_0 + \Phi_{max} \cdot P$$

Aplikované ve skriptu.

```
epoch = np.nanmin(lc_all.time.value) + max_phase * period.value
```

Chyba epochy je vypočtena z chyby periody a vzorkovací vzdálenosti binů.

$$E_{error} = \sqrt{P_{error}^2 + \left(\frac{P}{bins}\right)^2}$$

Chyba epochy je pomocí funkce `sr()` zaokrouhlena na jednu platnou číslici.

```
epoch_err = sr(abs(((per_err) ** 2 + ((1 / bins) * period.value) ** 2) ** 0.5), 1)
```

Epocha je poté zaokrouhlena na stejný počet desetinných míst, jako má její chyba pomocí funkce `rs()`.

```
epoch_sig = rs(epoch, epoch_err)
```

### 6.2.1.20 Vytvoření fázové světelné křivky, výpočet amplitudy

Opět je vytvořena fázová světelná křivka, tentokrát se správnou hodnotou epochy, definovatelné pomocí argumentu `epoch_time`. Nápodobně je vytvořena binovaná fázová světelná křivka pro výpočet mediánové amplitudy. Fázové světelné křivky jsou vytvořeny v rozsahu dvou fází pro lepší viditelnost.

```
lc_all_phase_bin = lc_all.fold(period = period, epoch_time = epoch, normalize_phase = True)
lc_all_phase_bin = lk.FoldedLightCurve(time = bin_lc(np.concatenate([lc_all_phase_bin.phase.value - 0.5,
                                                                    lc_all_phase_bin.phase.value + 0.5],
                                                                    axis = 0).ravel(),
                                                                    bins, method = 'mean'),
                                      flux = bin_lc(np.array(2 * [lc_all_phase_bin.flux.value]).ravel(),
                                      bins, method = 'median'))
lc_all_phase = lc_all.fold(period = period, epoch_time = epoch, normalize_phase = True)
lc_all_phase = lk.FoldedLightCurve(time = np.concatenate([lc_all_phase.phase.value - 0.5,
                                                         lc_all_phase.phase.value + 0.5],
                                                         axis = 0).ravel(),
                                   flux = np.array(2 * [lc_all_phase.flux.value]).ravel())
```

Je definována funkce `to_mag()`, konvertující pomocí magnitudy hvězdy normalizovaný světelný tok na magnitudu.

$$m = m_0 - 2,5 \log_{10}(F_{norm})$$

Zavedená do skriptu.

```
def to_mag(mag, flux):
    return mag - 2.5 * np.log10(flux)
```

Jsou definovány minimální a maximální hodnoty mediánové amplitudy z binované fázové světelné křivky, a to v jednotkách normalizovaného světelného toku a TESS magnitudy. Hodnoty jsou zaokrouhleny na desetitisíciny.

```
amp_f_min, amp_f_max = np.nanmin(lc_all_phase_bin.flux.value), np.nanmax(lc_all_phase_bin.flux.value)
amp_m_min, amp_m_max = to_mag(mag, amp_f_min), to_mag(mag, amp_f_max)
amp_f_min, amp_f_max = round(amp_f_min, 4), round(amp_f_max, 4)
amp_m_min, amp_m_max = round(amp_m_min, 4), round(amp_m_max, 4)
```

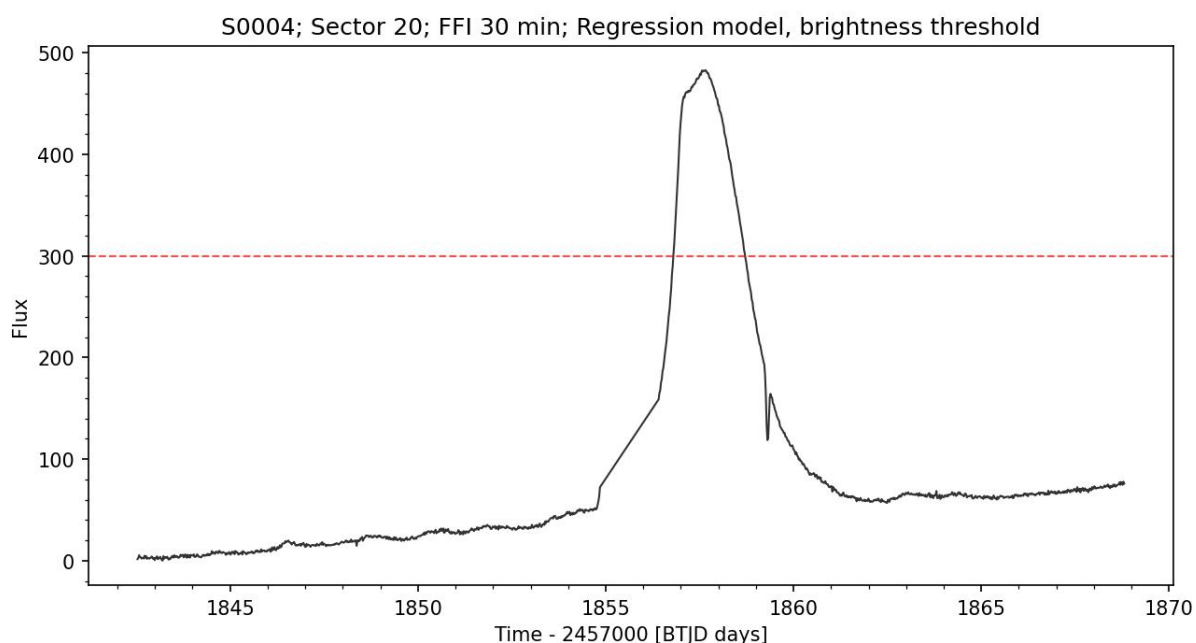
Všechny doposud vypočtené parametry proměnnosti jsou uloženy do jednoho CSV souboru. Tímto končí hlavní `for` loop a proces se opakuje pro další hvězdu.

## 6.2.2 Vizualizace dat

V této kapitole budou představeny generované vizualizace, avšak nebude uveden detailní popis jejich tvorby, jelikož nesouvisí se zaměřením práce. Vizualizace jsou nedílnou součástí analýzy, jelikož např. určování maximální amplitudy je provedeno manuálně. Řada vizualizací je vytvářena pouze pro sledování správné funkcionality zavedených metod.

### 6.2.2.1 Model rozptýleného světla

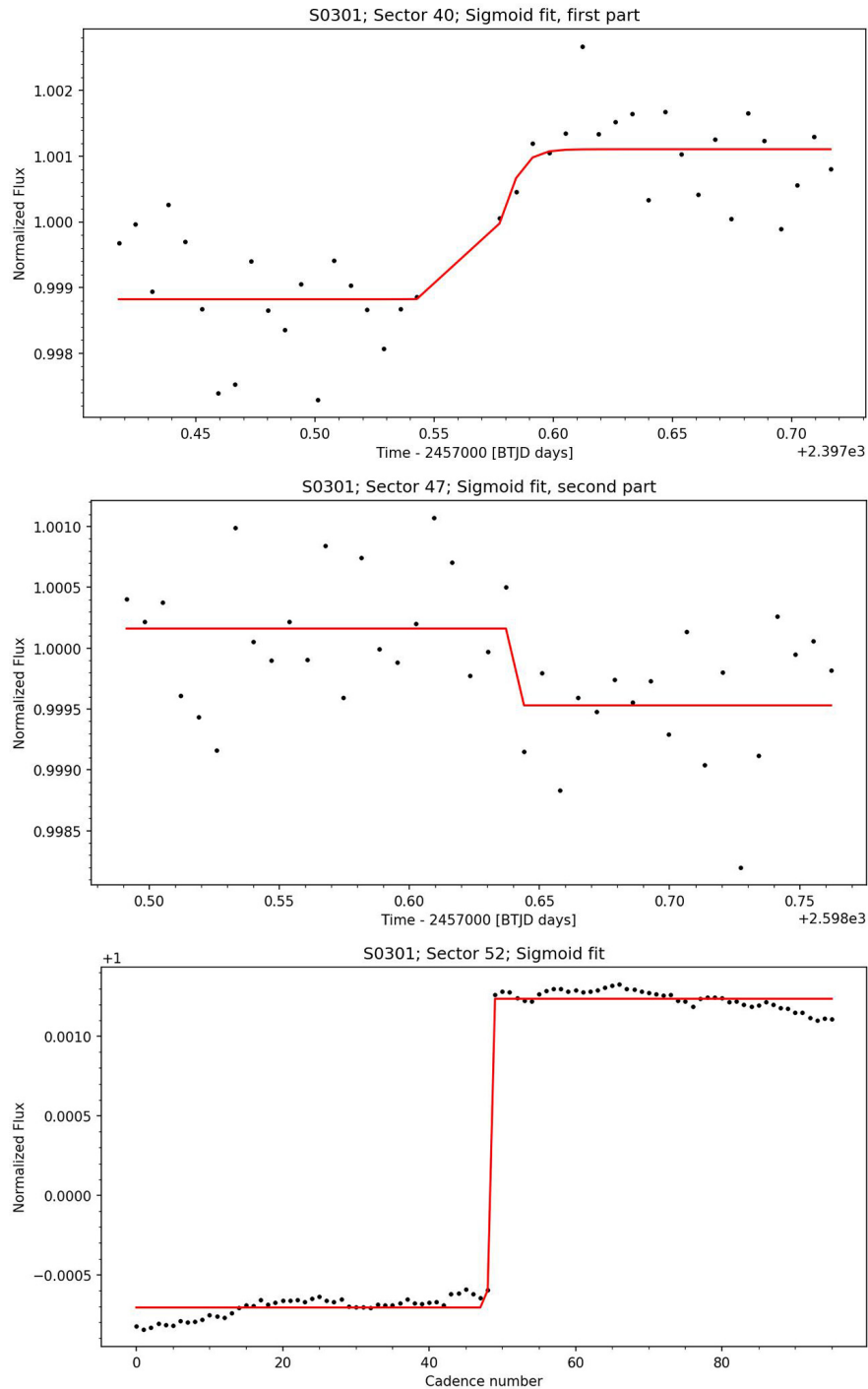
Graf světelného toku rozptýleného světla slouží k validaci prahové hodnoty světelného toku pro odstranění příliš jasných kadencí. V případě, že byly kadence odstraněny, je zobrazen i práh světelného toku.



Obrázek 50: Příklad modelu světelného toku rozptýleného světla odstraněného ze světelné křivky hvězdy S0004 v sektoru 19. V tomto případě došlo k odstranění příliš „jasných“ kadencí, tudíž je zobrazen práh světelného toku (vlastní tvorba).

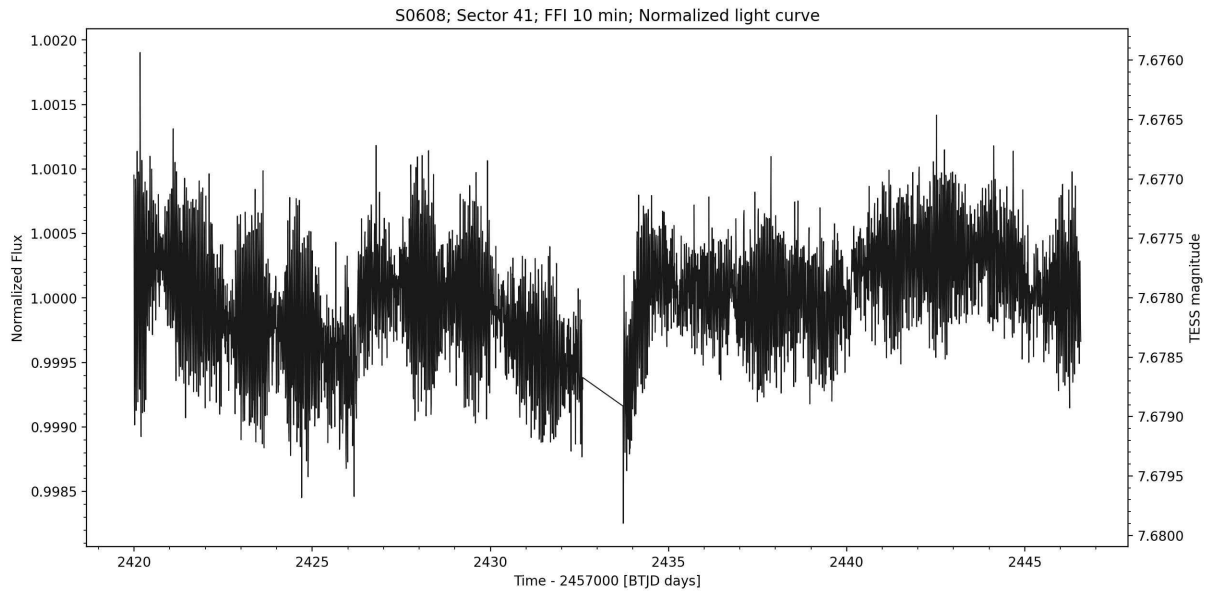
Pro konverzi normalizovaného světelného toku na magnitudu je použita dříve nadefinovaná funkce `to_mag()`. Magnituda je zobrazena na pravou stranu grafu.

### 6.2.2.2 Proložení funkce sigmoid



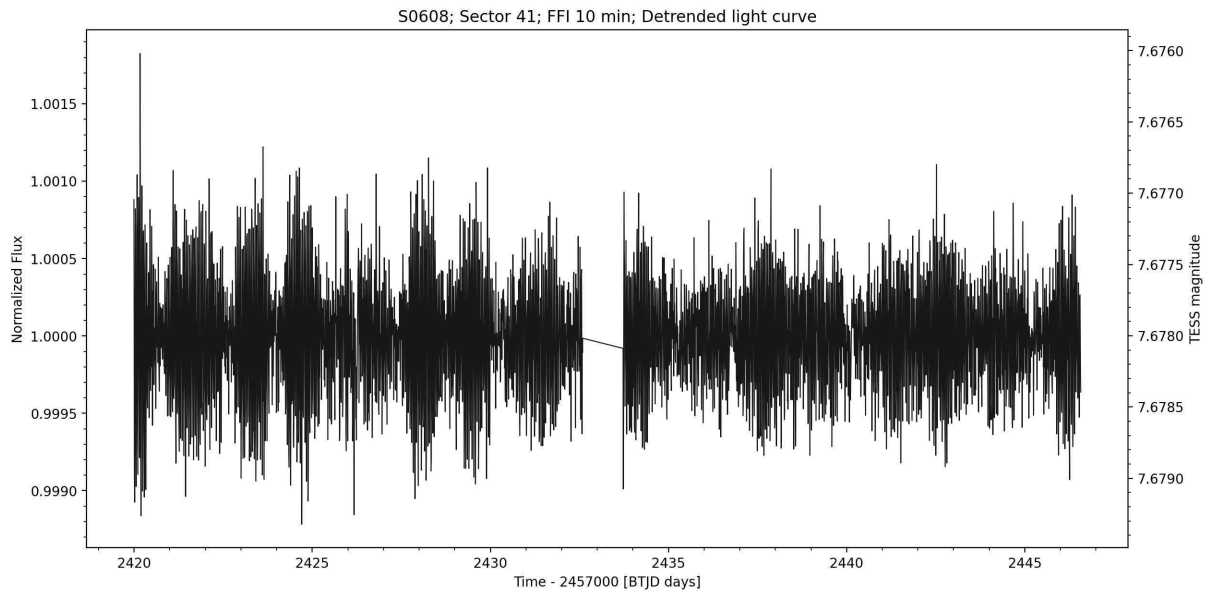
Obrázek 51: Příklad světelných křivek hvězdy S0301 a jimi proložené funkce sigmoid pro odstranění „skoku“ jasnosti. Nahoře: proložená světelná křivka prvního půlsektoru 40. Uprostřed: proložená světelná křivka druhého půlsektoru 47. Dole: proložená světelná křivka sektoru 52 (vlastní tvorba).

### 6.2.2.3 Světelná křivka sektoru



Obrázek 52: Příklad světelné křivky DSCT pulzující hvězdy S0608 v sektoru 41 s odstraněným rozptýleným světlem. Je zřetelné, že se v křivce nachází zbytkové systematické chyby (vlastní tvorba).

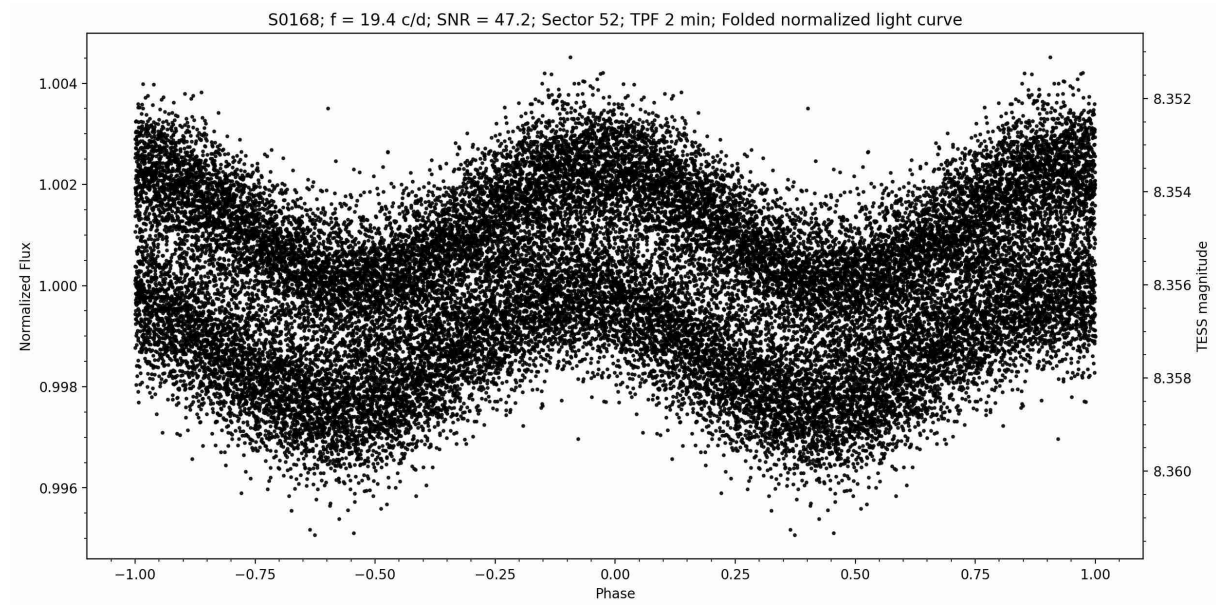
### 6.2.2.4 Detrendovaná světelná křivka sektoru



Obrázek 53: Příklad detrendované světelné křivky DSCT pulzující hvězdy S0608 v sektoru 41 s odstraněnou zbytkovou systematickou. Korekce zbytkové systematiky bývá u krátkoperiodických hvězd velmi efektivní (vlastní tvorba).

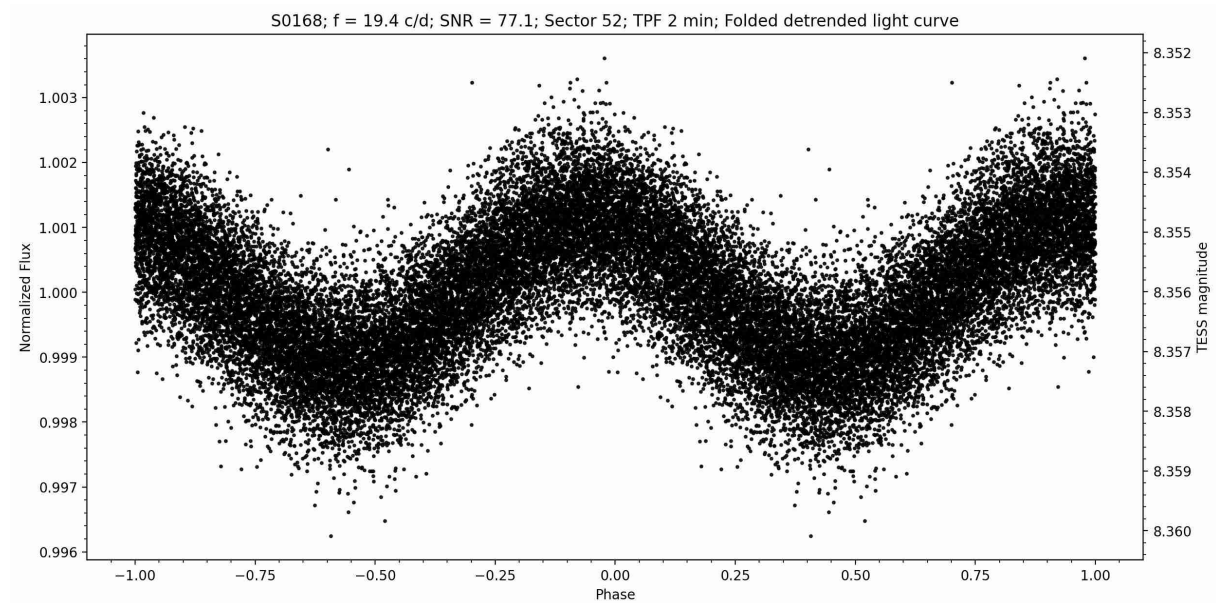


### 6.2.2.5 Fázová světelná křivka sektoru



Obrázek 54: Příklad fázové světelné křivky DSCT pulzující hvězdy S0168 v sektoru 52. Je vidět „zdvojení“ způsobené nedostatečnou korekcí systematik (vlastní tvorba).

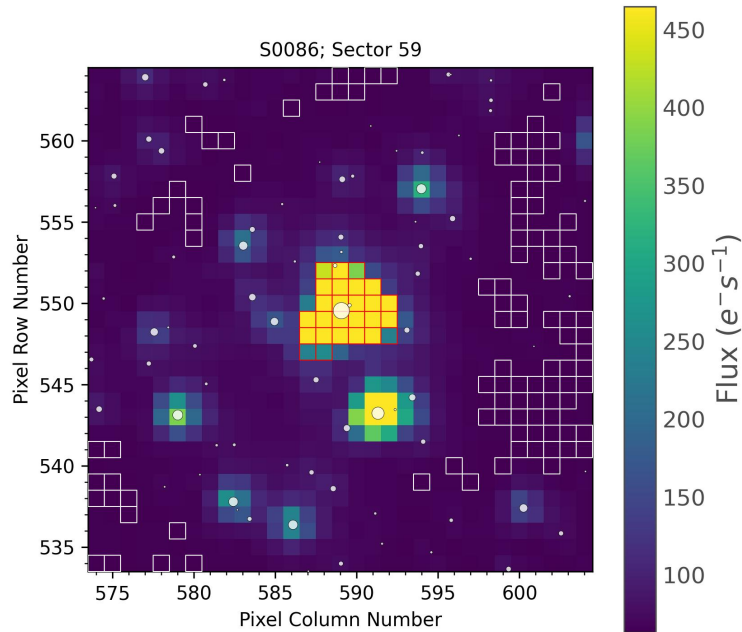
### 6.2.2.6 Fázová detrendovaná světelná křivka sektoru



Obrázek 55: Příklad fázové detrendované světelné křivky DSCT pulzující hvězdy S0168 v sektoru 52. Zbytková systematika je opravena, díky čemuž se výrazně zvýší SNR (vlastní tvorba).

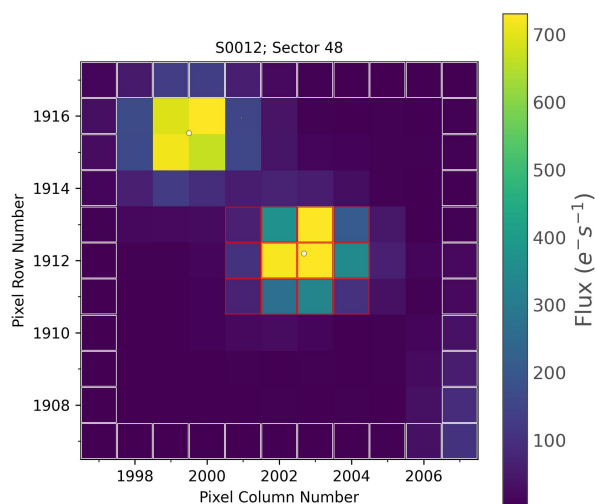
### 6.2.2.7 TargetPixelFile v FFI sektoru

Pro každý sektor je zobrazen *TargetPixelFile* zobrazující zdroj originálních dat. Červenými rámečky jsou označeny pixely náležící aperturní masce, bílými rámečky jsou označeny pixely náležící masce pozadí. Bílými kolečky jsou zobrazeny Gaia DR3 hvězdy do 18. magnitudy.



Obrázek 56: Příklad TargetPixelFile výstřížku 31×31 pixelů v FFI sektoru 59 hvězdy S0086 (vlastní tvorba).

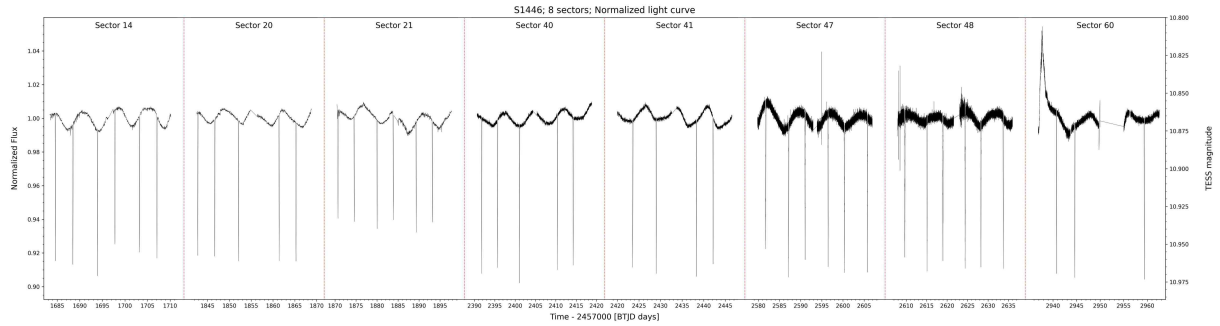
### 6.2.2.8 TargetPixelFile v TPF sektoru



Obrázek 57: Příklad TargetPixelFile 11×11 pixelů v „fast“ TPF sektoru 48 hvězdy S0012 (vlastní tvorba).

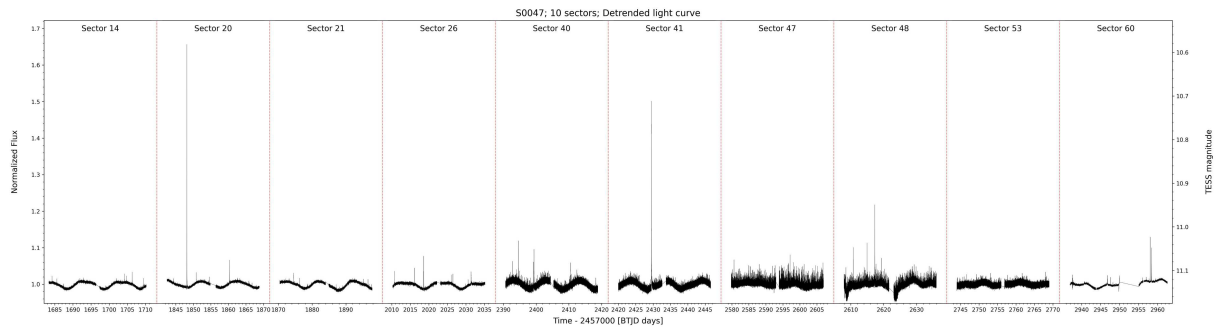
### 6.2.2.9 Světelná křivka všech sektorů

Světelná křivka všech sektorů má odstraněny dlouhé „mezery“ mezi sektory, tudíž osa x není spojitá. Jednotlivé sektory jsou oddělené červenými přerušovanými vertikálními čarami. Každá světelná křivka je popsána jejím číslem sektoru.



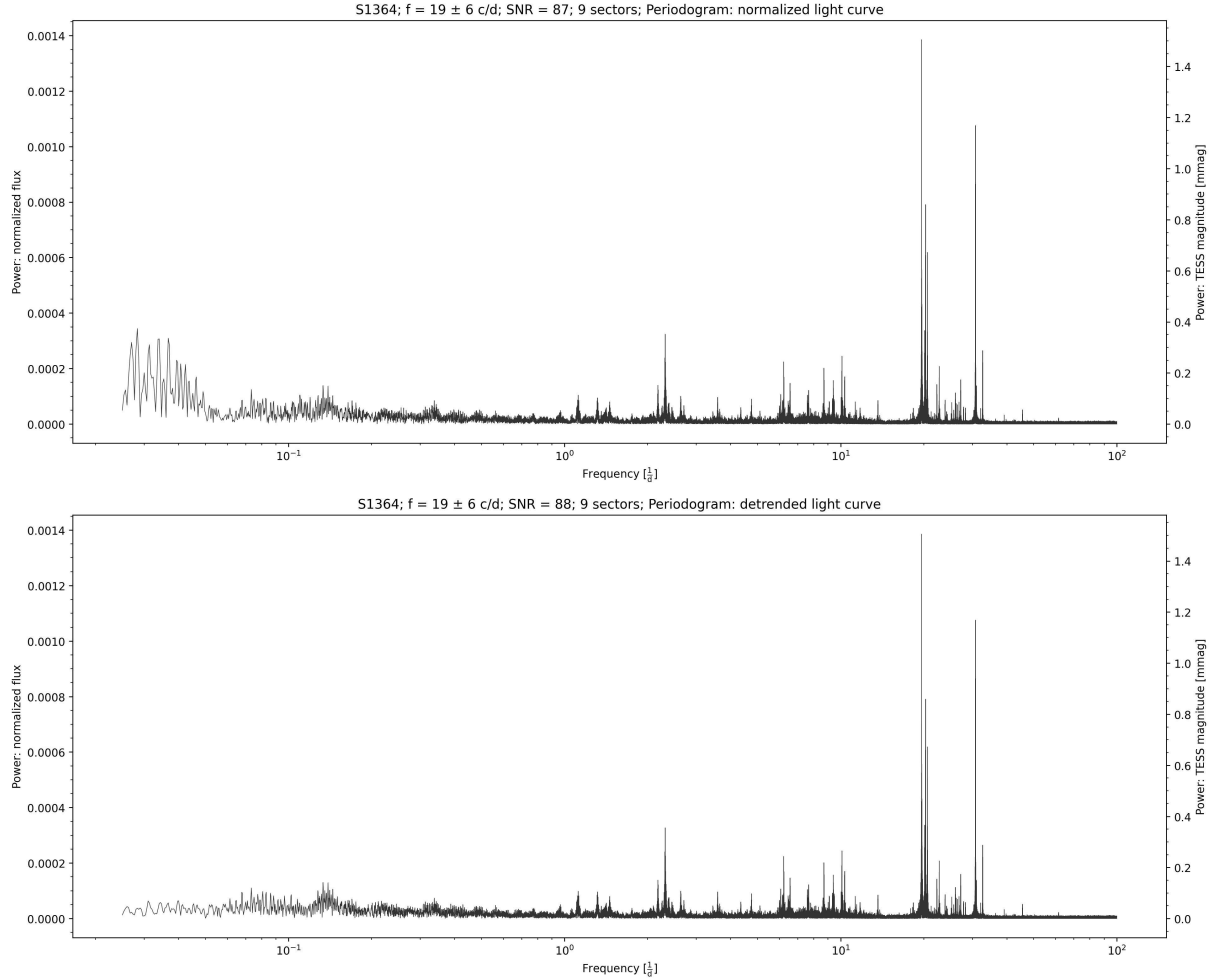
Obrázek 58: Příklad světelné křivky 8 sektorů zákrytové dvojhvězdy S1446 (vlastní tvorba).

### 6.2.2.10 Detrendovaná světelná křivka všech sektorů



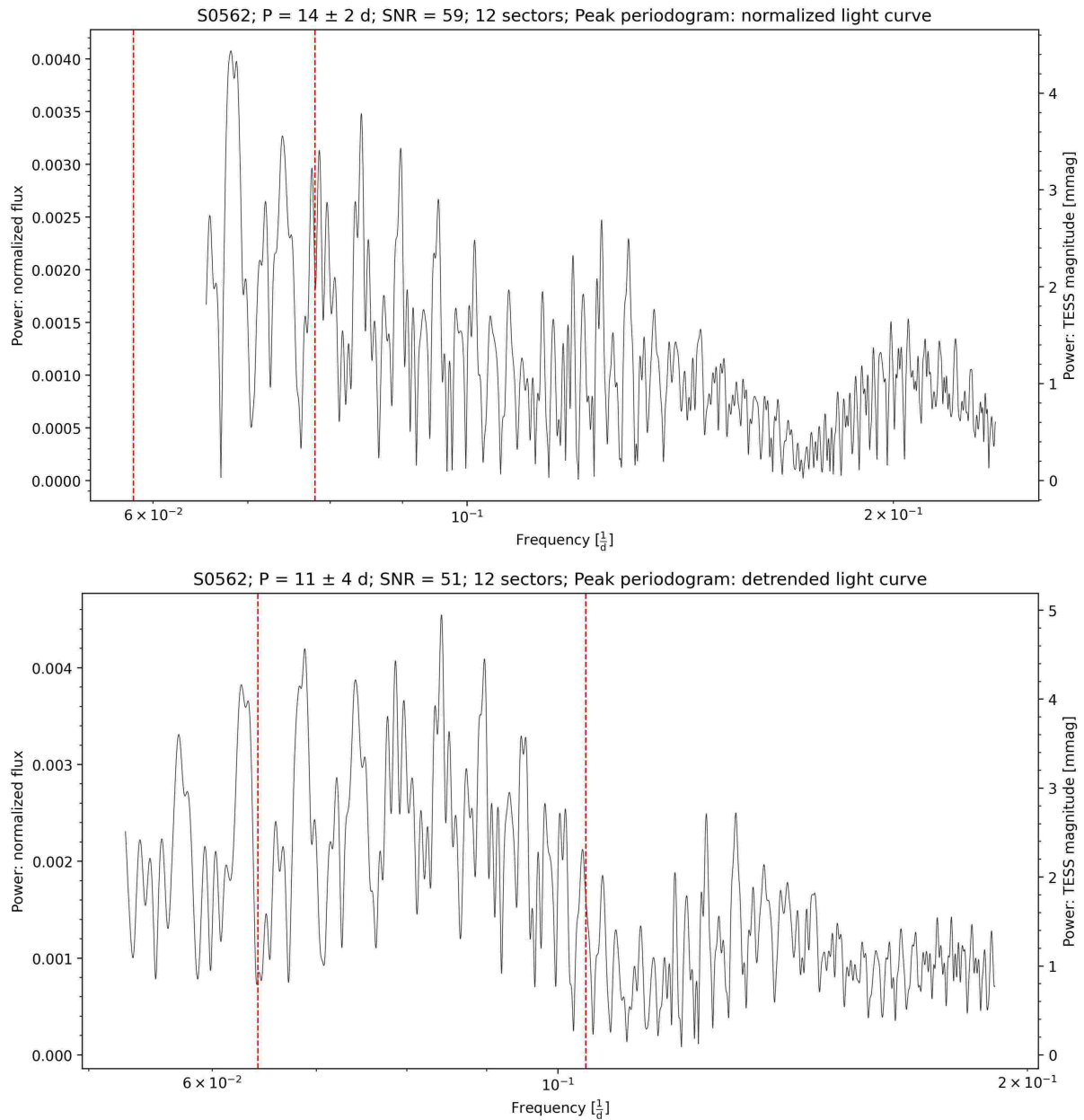
Obrázek 59: Příklad detrendované světelné křivky 10 sektorů eruptivní hvězdy S0047 (vlastní tvorba).

### 6.2.2.11 Periodogramy všech sektorů



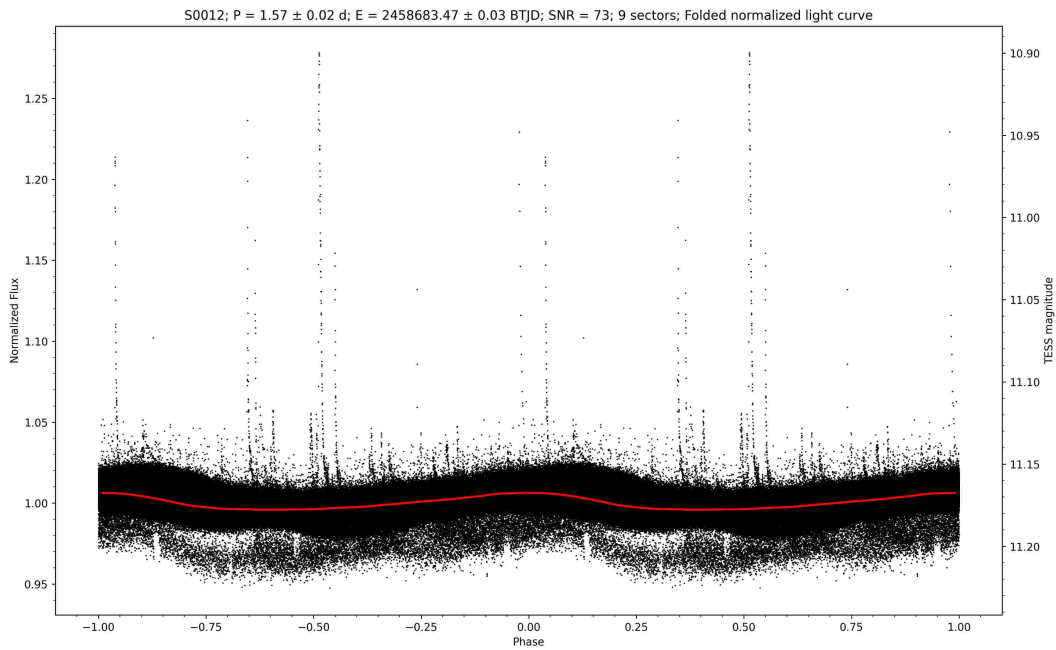
Obrázek 60: Příklad periodogramů DSCT+GDOR hybridu S1364 vytvořených z 9 sektorů pozorování. Nahoře: periodogram vytvořený ze světelné křivky s odstraněným rozptýleným světlem. Dole: periodogram vytvořený z detrendované světelné křivky. Lze vidět jasný pokles síly nízkých frekvencí oproti periodogramu nahoře (vlastní tvorba).

### 6.2.2.12 Detailní periodogramy



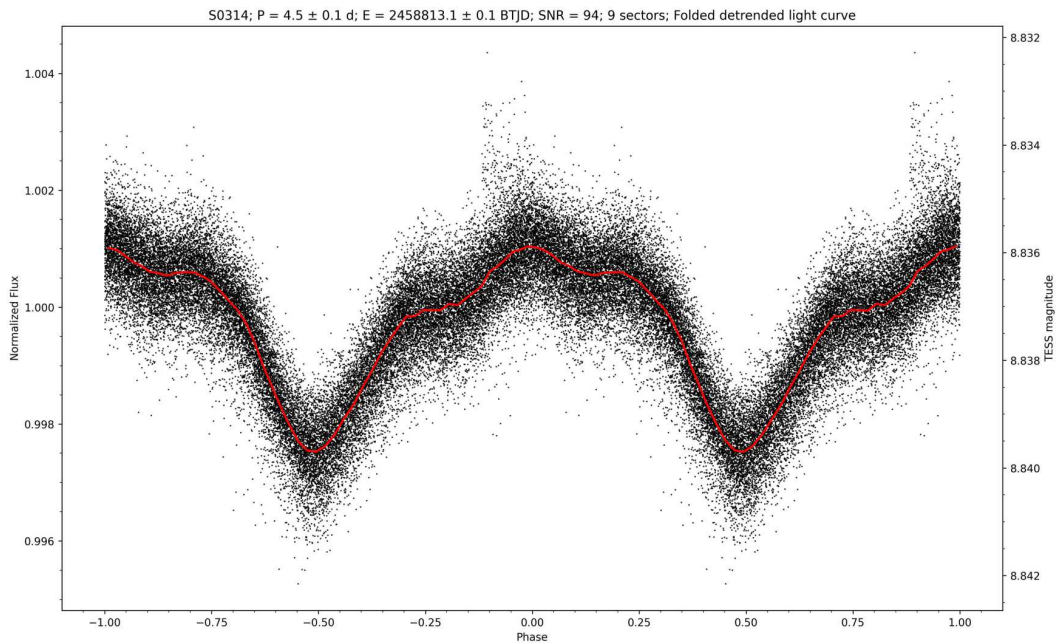
Obrázek 61: Příklad detailních periodogramů hvězdy S0562 vytvořených z 12 sektorů pozorování. Nahoře: periodogram vytvořený ze světelné křivky s odstraněným rozptýleným světlem. Dole: periodogram vytvořený z detrendované světelné křivky. Po odstranění nízkofrekvenčního signálu ze světelné křivky došlo ke zvolení jiné periody (vlastní tvorba).

### 6.2.2.13 Fázová světelná křivka všech sektorů



Obrázek 62: Příklad fázové světelné křivky 9 sektorů eruptivní hvězdy S0012. Červenou křivkou je vyznačena binovaná fázová světelná křivka (vlastní tvorba).

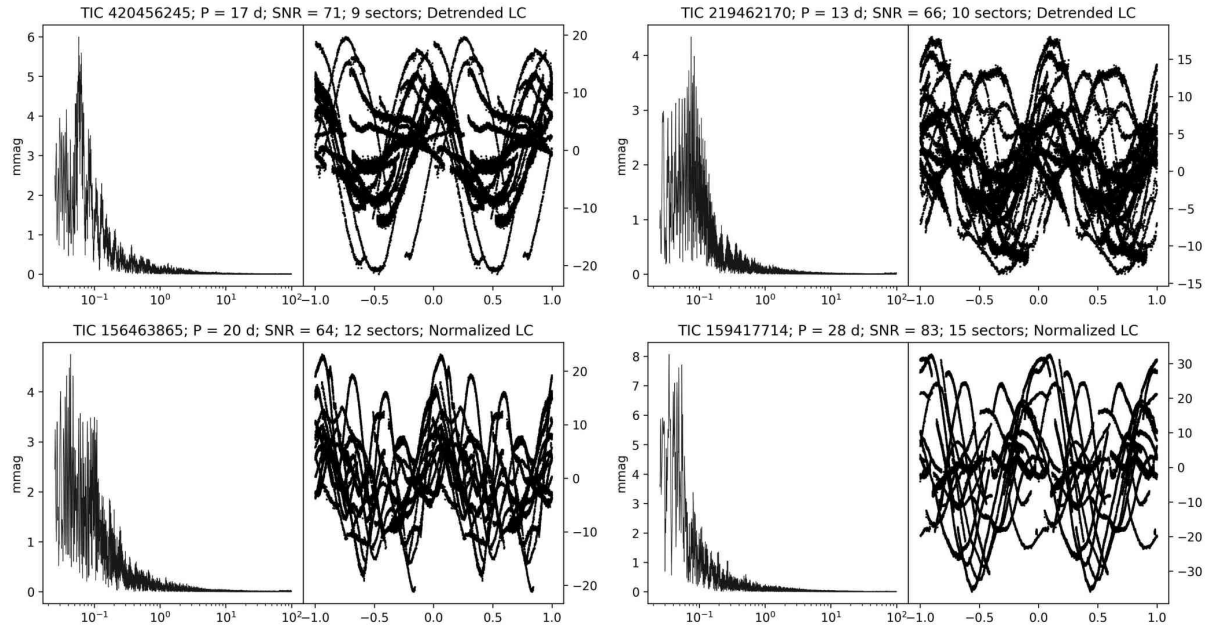
### 6.2.2.14 Fázová detrendovaná světelná křivka všech sektorů



Obrázek 63: Příklad fázové detrendované světelné křivky 9 sektorů rotující proměnné hvězdy S0314. Červenou křivkou je vyznačena binovaná fázová světelná křivka (vlastní tvorba).

### 6.2.2.15 Kompaktní graf periodogramu a fázové světelné křivky

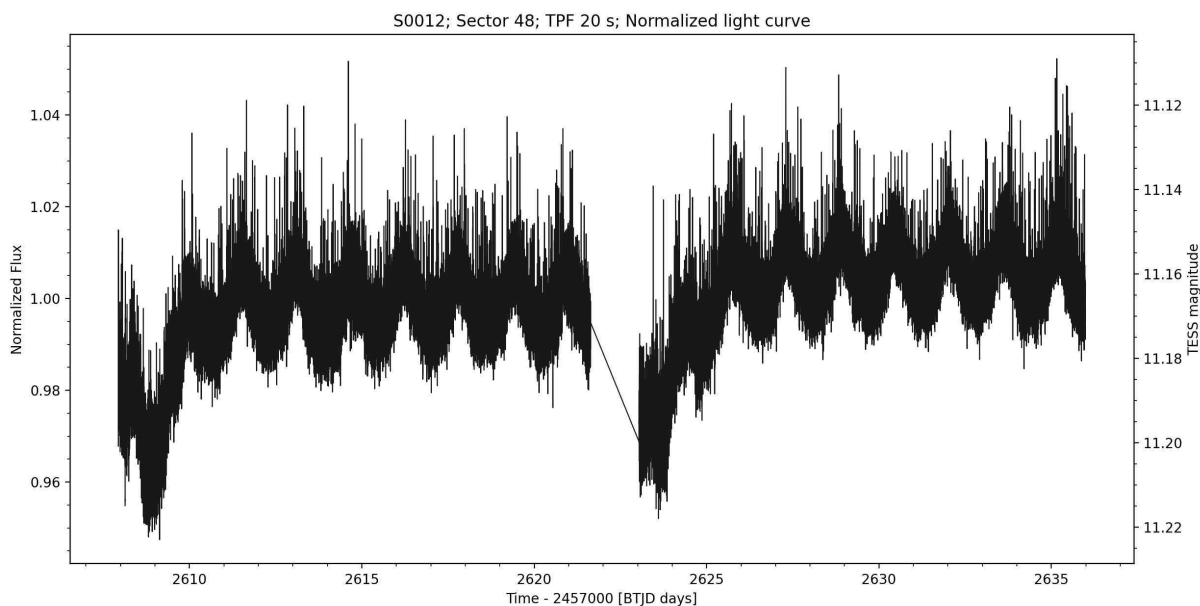
Grafy byly také generovány v kompaktnější formě. V nadpise jsou zahrnuty nejdůležitější informace o hvězdě: TIC identifikátor, perioda, SNR, počet sektorů a typ světelné křivky. Osa y je zobrazena jako amplituda v milimagnitudách.



Obrázek 64: Příklad čtyř různých dlouhoperiodických proměnných hvězd s významnou modulací periody a amplitudy. Pravděpodobně se jedná o rotující proměnné hvězdy s vyvíjecími se povrchovými skvrnami. Vlevo nahoře: hvězda S0002, vpravo nahoře: hvězda S0052, vlevo dole: hvězda S0069, vpravo dole: hvězda S0086 (vlastní tvorba).

## 6.3 Diskuze

Při konečném spuštění skriptu nebyly zaznamenány žádné významné problémy. Stojí za zmínku, že prezentované metody nejsou zdaleka bezchybné. Funkce `regression()` avšak byla schopna úspěšně odstranit světelný tok rozptýleného světla v naprosté většině případů, pouze v malém procentu sektorů lze vidět nedostatečná korekce. Prezentovaná metoda se osvědčila jako evidentně velmi úspěšná.



Obrázek 65: Příklad nedostatečné korekce v sektoru 48 hvězdy S0012 (vlastní tvorba).

Funkce `regression()` by mohla být vylepšena v „short“ a „fast“ TPF sektorech, ve kterých malá velikost TPF nedovoluje tvorbu dostatečně kvalitní masky pozadí. Pomocí interpolace světelného toku extrahovaného z FFI dat by mohl být světelný tok rozptýleného světla efektivněji odstraněn v TPF sektorech.

Funkce `flatten_lc()` a `fix_sector_jump()` se pokouší o korekci zbytkových systematických chyb, a to ve většině případů také úspěšně. Důkazem jsou periodogramy s vyšším SNR detrendovaných světelných křivek. Velkým problémem avšak zůstává korekce samotné proměnnosti hvězdy.

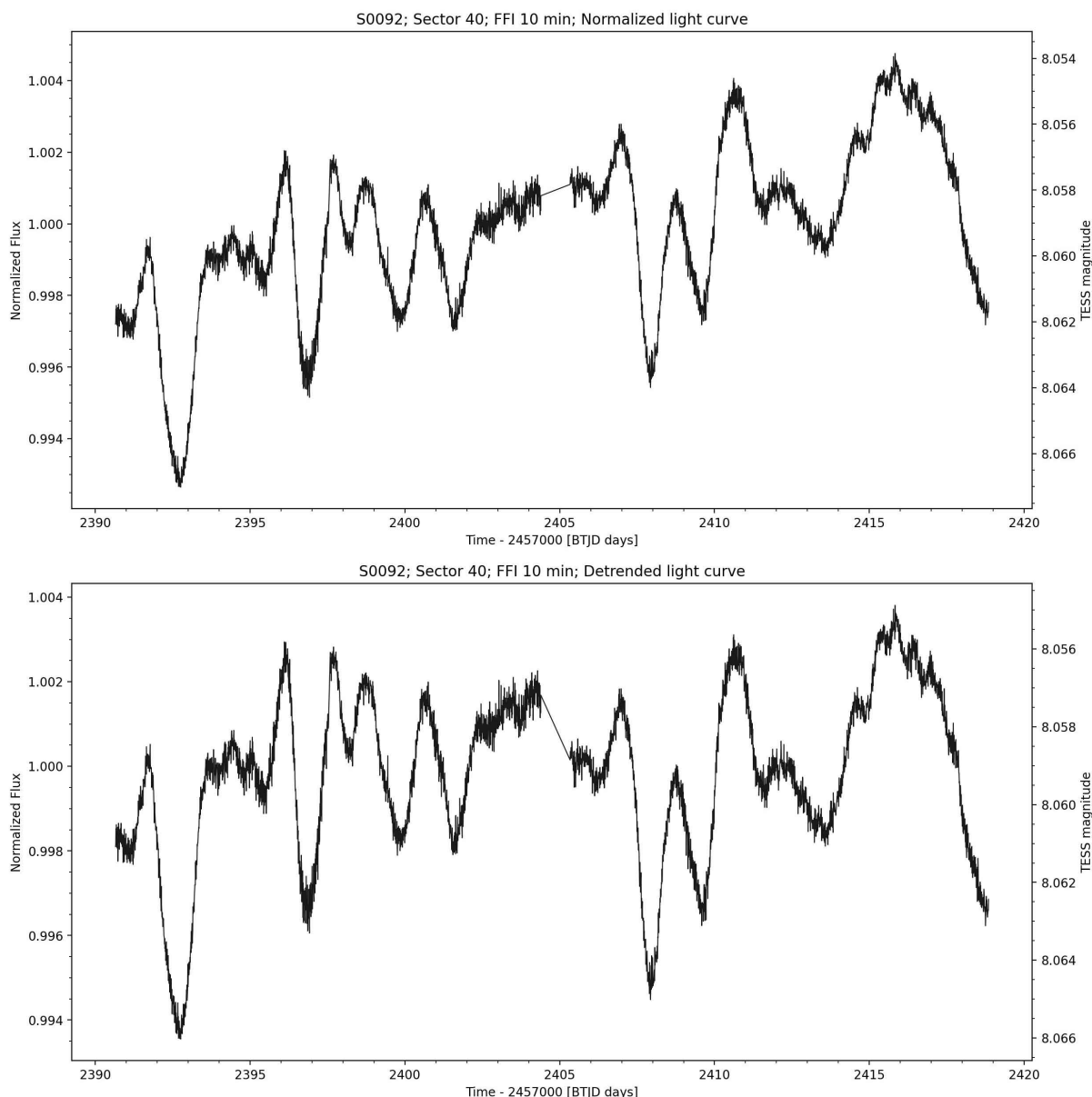
Prezentované funkce se pokouší o korekci fundamentálně hlubšího problému, čímž jsou nepřesnosti v pointaci, ostření kamer a ostatní systematické chyby. Možným řešením je pečlivé modelování těchto změn světelného toku na okolních neproměnných hvězdách, za využití metody hlavních komponent a jejího správného použití.

Největším problémem při zpracování světelné křivky všech sektorů byl odhad chyby periody. Multiperiodické hvězdy (převážně pulzující) často kvůli několika blízkým peakům v periodogramu měly vypočtenou chybu mnohem vyšší než je ve skutečnosti. Pro tyto hvězdy musela být správná chyba periody určena znovu manuálně. Lepší metoda pro výpočet chyby periody by byla vhodná.

Aktuálně probíhá práce na skriptu vypočítávající pozici centroidů proměnnosti kolem pozorovaných hvězd pro ověření jejich proměnnosti. Velké TESS pixely ( $\sim 21$  úhlových



vteřin) často obsahují světelný tok několika hvězd, tudíž není zprvu jasné, zda proměnnost opravdu pochází z pozorované hvězdy. Řada proměnných hvězd objevená v rámci této práce má velmi nízkou amplitudu, což znemožňuje centroidovou analýzu, jelikož samotné pixely nemají dostatečné SNR. Pozemské navazující pozorování těchto hvězd je často také velmi náročné, či dokonce v některých případech nemožné kvůli vyšší magnitudě, nízké amplitudě a dlouhé periodě.



Obrázek 66: Selhaná korekce „skoku“ jasnosti uprostřed sektoru 40 hvězdy S0092. V tomto případě by bylo nejspíše nejlepší řešení proložení celé světelné křivky lineární funkcí pro odstranění dlouhodobého trendu. Nahoře: světelná křivka s odstraněným světelným točím rozptýleného světla. Dole: světelná křivka s aplikovanými funkcemi `flatten_lc()` a `fix_sector_jump()` (vlastní tvorba).

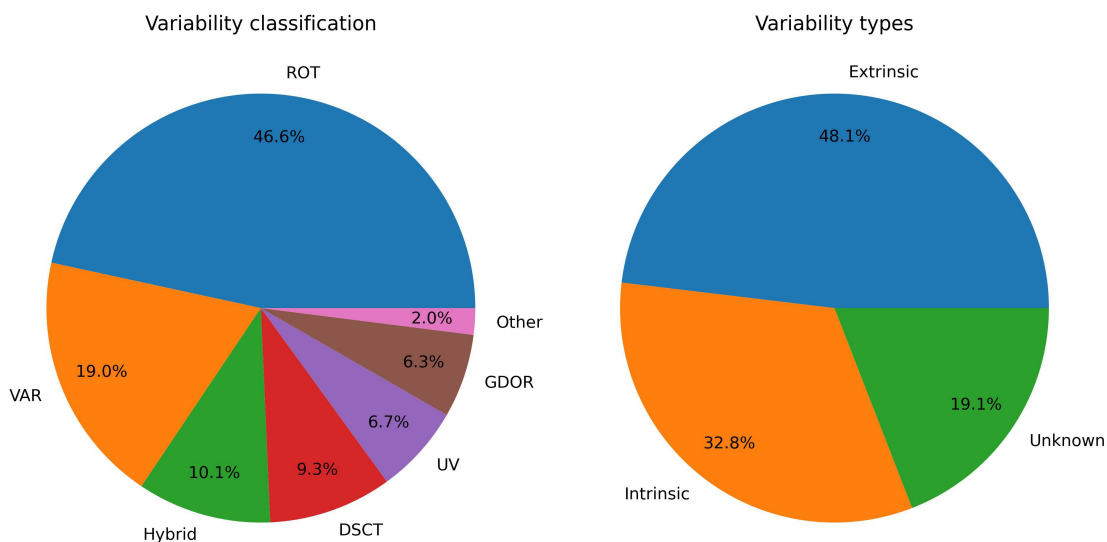
## 6.4 Klasifikace

Hvězdy jsou rozděleny do klasifikačních tříd zavedených ve VSX [4]. Všechny rotující proměnné hvězdy jsou označeny zkratkou ROT. Znaménko + znamená dva současné typy proměnnosti. Hvězdy DSCT a GDOR jsou odděleny hranicí  $5 \text{ den}^{-1}$ . Pokud je hvězda hybrid DSCT a GDOR, je napsán nejdříve typ, který přispívá více do periodicity dané hvězdy. Znaménko / značí nejistotu mezi dvěma typy proměnnosti, znaménko : značí nejistotu v typu proměnnosti. Hvězdy, jejichž typ proměnnosti není možný určit je přidělena zkratka VAR. U hvězd S1195 a S1749 nebyla prokázána dostatečná proměnnost, tudíž byly vyřazeny.

Klasifikace probíhá na základě zhodnocení všech dostupných informací, jako je tvar, amplituda či perioda světelné křivky, popř. multiperiodicita. Další faktory ovlivňující klasifikaci jsou jak široký je peak primární periody v periodogramu či jak se světelná křivka vyvíjí dlouhodobě. Jsou uváženy taky známé hvězdné parametry z Gaia DR3 a pozice hvězdy v HR diagramu. Typy proměnnosti jednotlivých hvězd jsou uvedeny v příloženém CSV souboru, který ukládá informace vyprodukované skriptem.

## 7 Výsledky

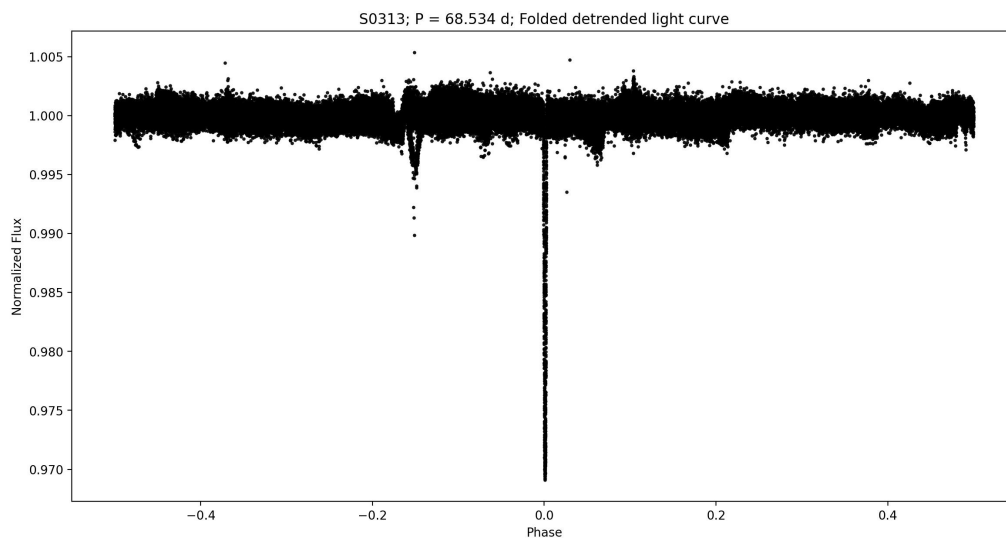
Je provedena kompletní automatická parametrizace 268 nových proměnných hvězd. Hvězdy jsou klasifikovány podle typu proměnnosti. XLSX soubor s automaticky určenými parametry a typem proměnnosti je dostupný v příloze dokumentu. V této kapitole bude ukázáno několik nejzajímavějších objevených proměnných hvězd.



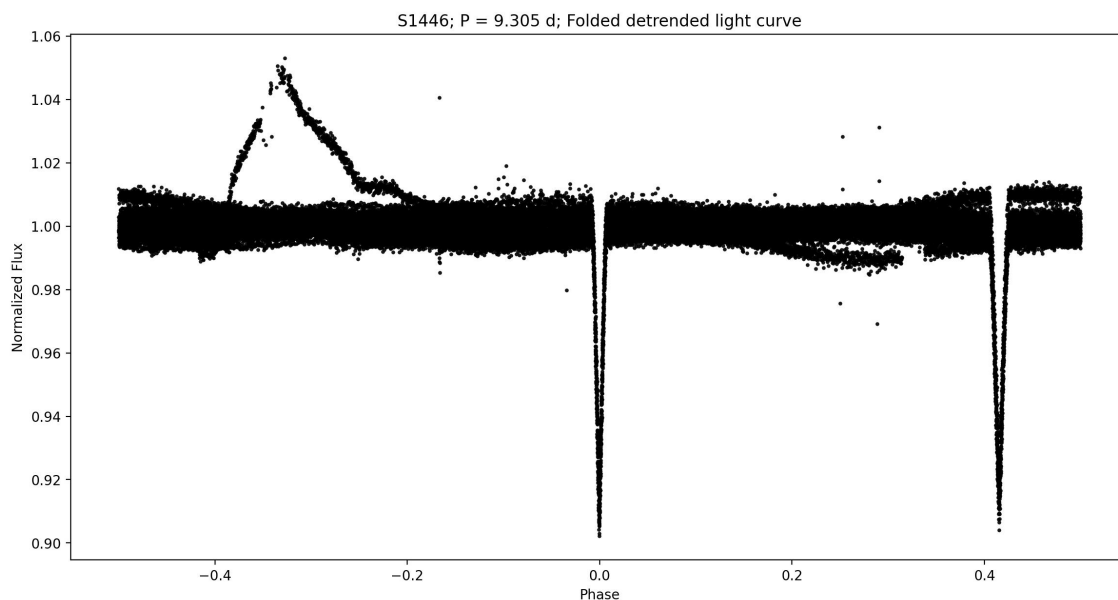
Obrázek 67: Nalevo: Klasifikace 268 objevených proměnných hvězd do jednotlivých typů proměnnosti definovaných v kapitole 6.4. Kategorie *Hybrid* zahrnuje hvězdy DSCT+GDOR i GDOR+DSCT. Napravo: Rozdělení objevených proměnných hvězd na geometrické (*extrinsic*) a fyzické (*intrinsic*) proměnné hvězdy. Mezi geometrické byly zařazeny všechny hvězdy s označením ROT (vlastní tvorba).

## 7.1 Zákrytové dvojhvězdy

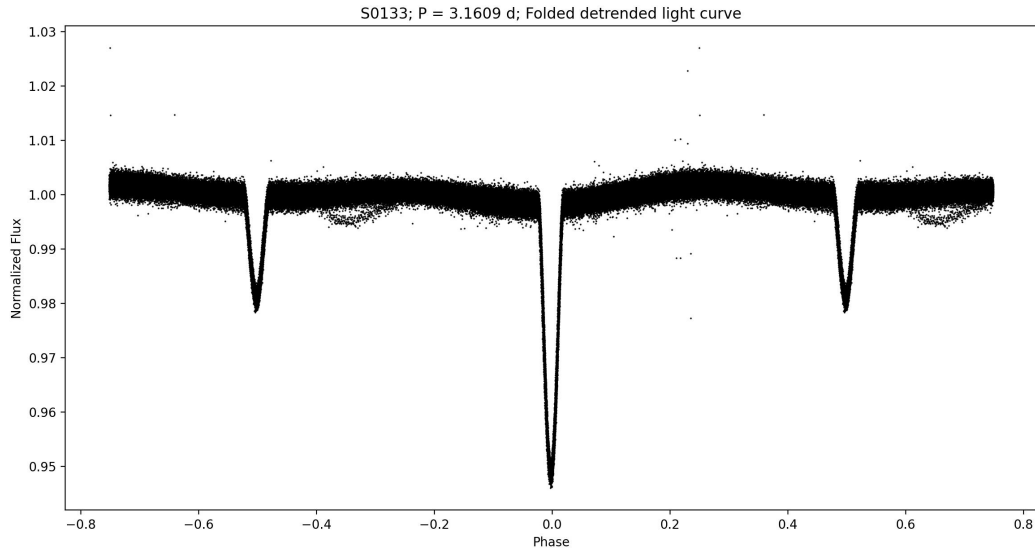
Jsou objeveny dvě zákrytové dvojhvězdy, S0313 a S1446.



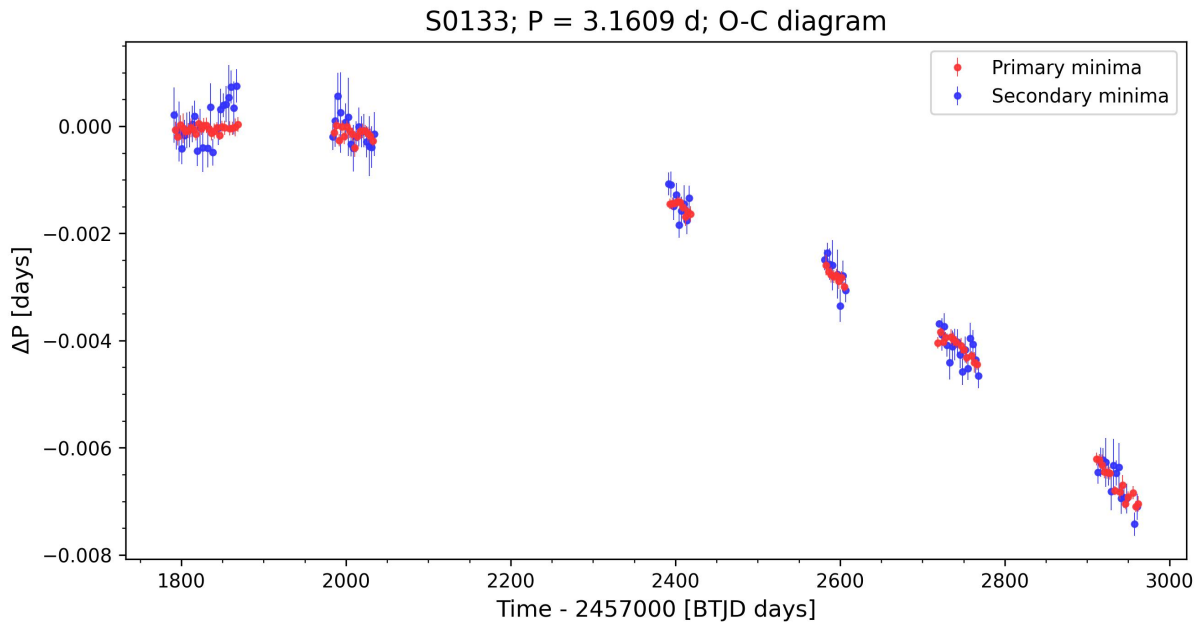
Obrázek 68: Fázová detrendovaná křivka zákrytové dvojhvězdy S0313. Hvězda nemá sekundární zákryty. Perioda je přes 68 dní a jedná se o hvězdu s nejdelší periodou ve vzorku (vlastní tvorba).



Obrázek 69: Fázová detrendovaná křivka zákrytové dvojhvězdy S1446. Hvězda má periodu  $\sim 9$  dní. Sekundární minimum má velmi podobnou hloubku jako primární, avšak je širší. Nastává kolem fáze 0,42 (vlastní tvorba).



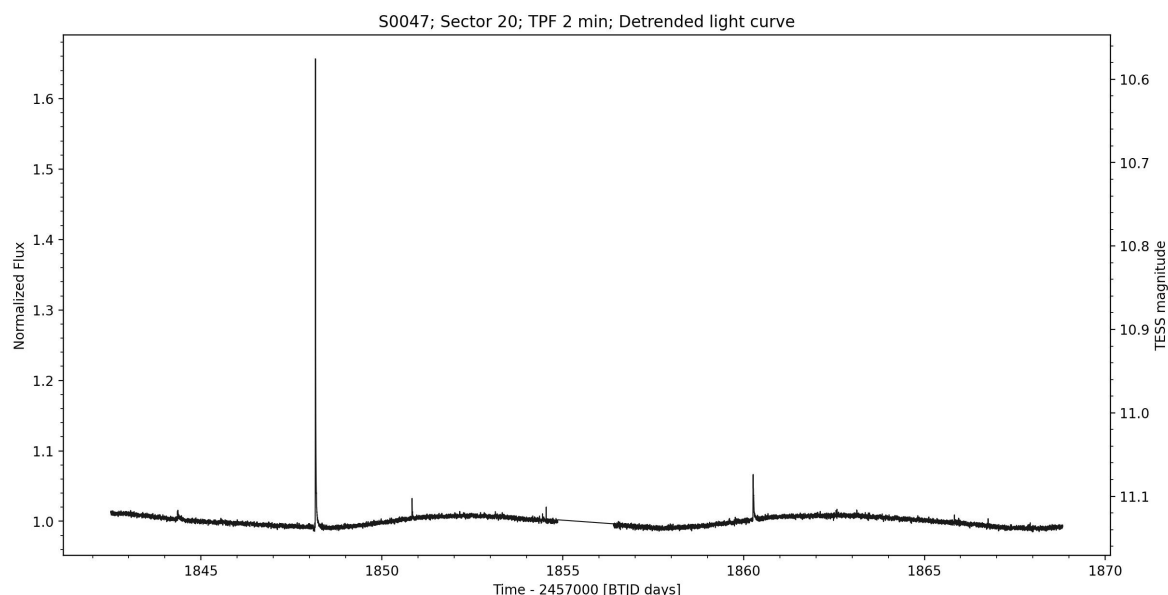
Obrázek 70: Fázová dentrendovaná křivka zákrytové dvojhvězdy S0133 (vlastní tvorba).



Obrázek 71: O – C diagram hvězdy S0133. Hvězda byla objevena v práci [39], avšak nebyla popsána její změna v O – C diagramu. Okamžiky minim byly určeny proložením světelné křivky v softwaru SILICUPS, popsaného v práci [37] (*Appendix A*). Bylo proloženo celkově 170 minim, z toho 84 primárních a 86 sekundárních. Okamžiky sekundárních minim mají větší rozptyl i chybu kvůli jejich menší hloubce. Perioda hvězdy v říjnu 2019 byla 3,1609 dne, během následujících  $\sim 1200$  dní pozorování se její perioda zkrátila o  $\sim 0,2\%$ , nebo  $\sim 10$  minut. Zkracování periody může být způsobeno jiným tělesem obíhajícím v tomto systému, přelivem hmoty mezi složkami, magnetickým bržděním či jiným efektem (vlastní tvorba).

## 7.2 Eruptivní hvězdy

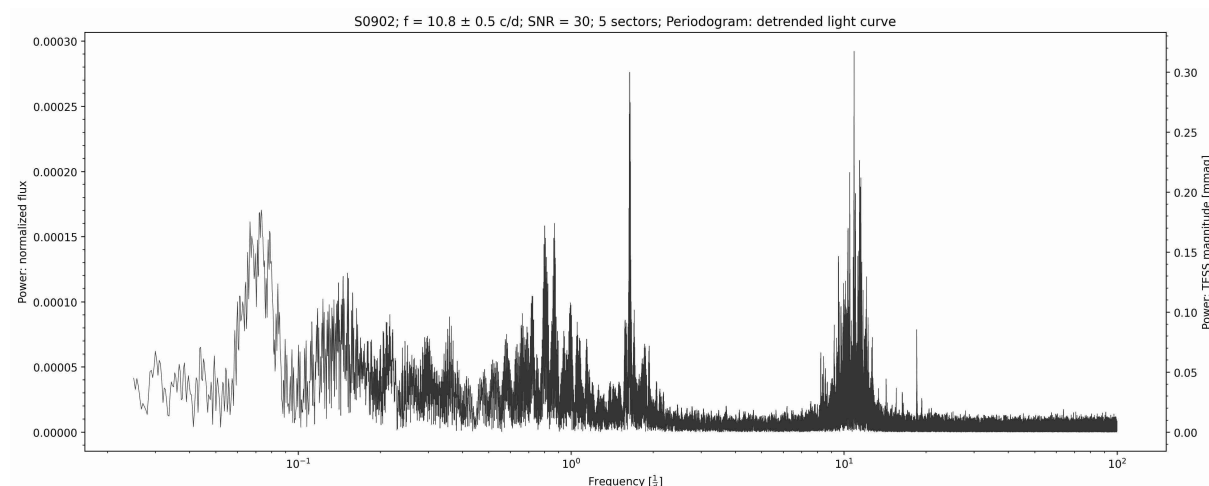
Jsou pozorovány erupce u 16 proměnných hvězd z celkových 268 proměnných hvězd ( $\sim 6\%$ ). Eruptivní hvězdy jsou často doprovázeny rotační proměnností díky jejich aktivnímu povrchu.



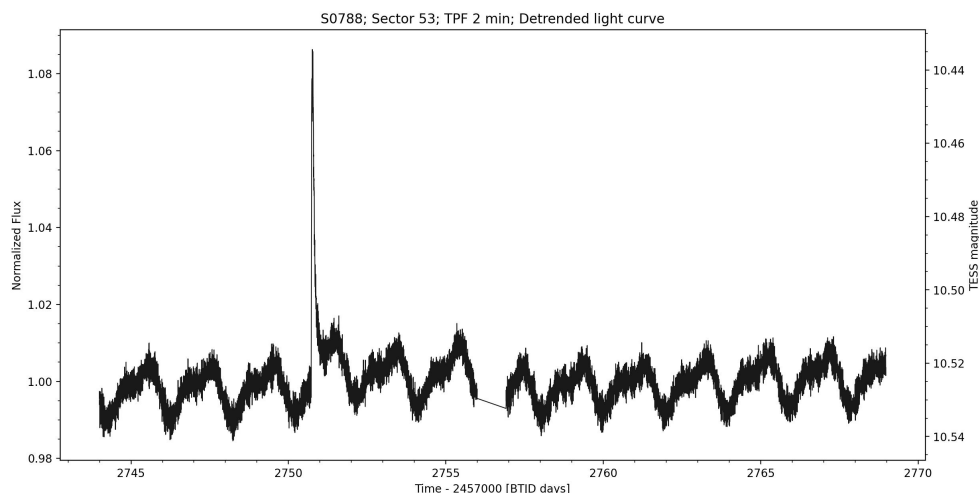
Obrázek 72: Erupce v sektoru 41 eruptivní UV Ceti hvězdy S0047. Světelný tok hvězdy dočasně narostl téměř na  $1,7\times$  normální hodnoty (vlastní tvorba).

## 7.3 Pulzující DSCT + GDOR hybridy

Pulzace jsou pozorovány u 72 z celkových 268 hvězd ( $\sim 27\%$ ). Řada pulzujících hvězd vykazuje pulzace jak nad tak pod hranici 0,2 dne.



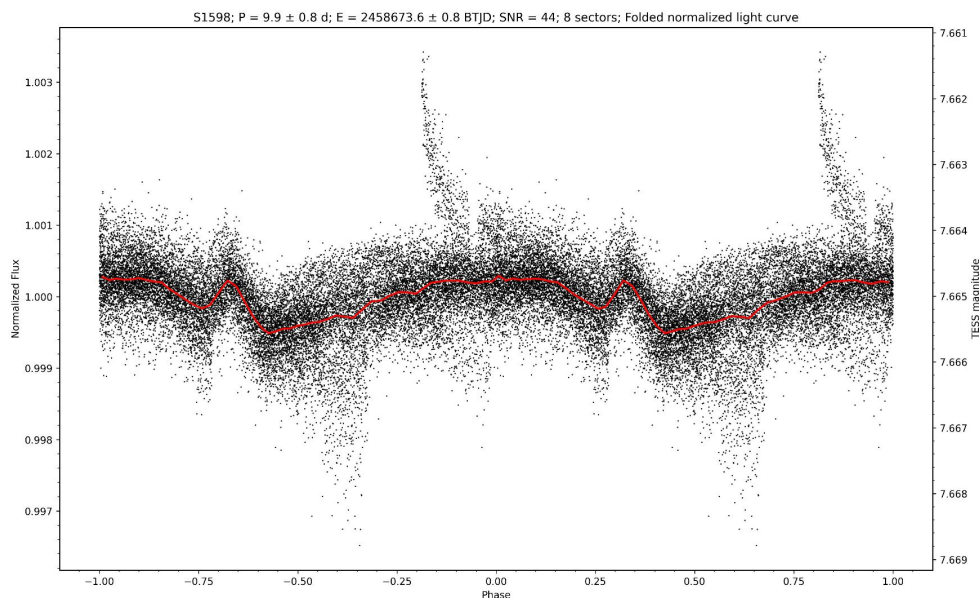
Obrázek 73: Periodogram hybrid DSCT+GDOR hvězdy S0902 (vlastní tvorba).



Obrázek 74: Detrendovaná světelná křivka hvězdy S0788 v sektoru 53. Hvězda S0788 je eruptivní s viditelnou rotační proměnností, navíc se ve světelné křivce vyskytují DSCT pulzace. Je možné, že se jedná o „blend“ (vlastní tvorba).

## 7.4 Rotační hvězdy

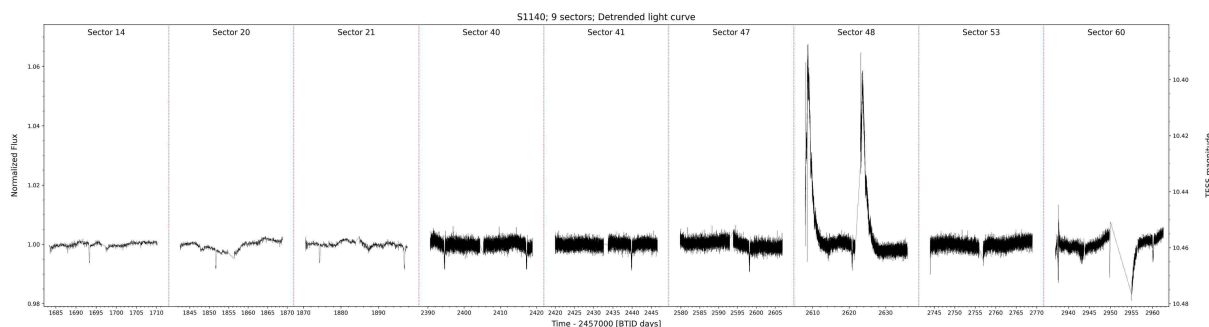
Téměř polovina hvězd ve vzorku projevují rotační proměnnost. Většina rotačně proměnných má délky period v rámci několika dní až desítek dní. Rotační proměnnost je zpravidla také pozorována u eruptivních hvězd.



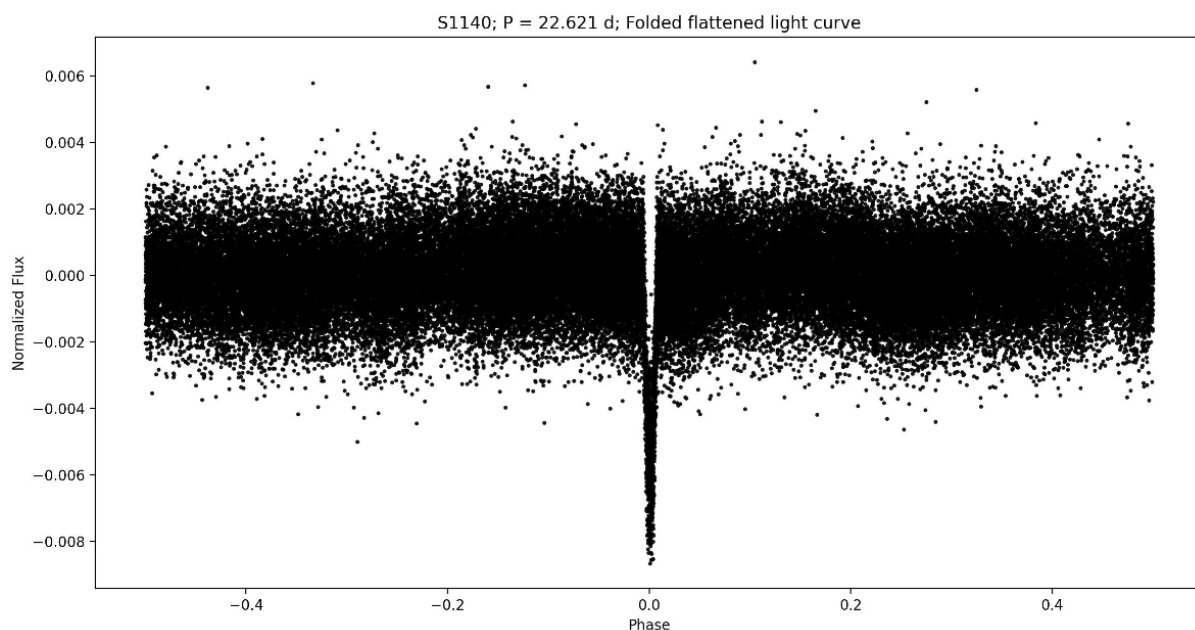
Obrázek 75: Fázová světelná křivka hvězdy S1598 složená z 8 sektorů pozorování. Hvězda S1598 je jedinou heartbeat hvězdou ve vzorku. Heartbeat hvězdy jsou extrémním případem elipsoidních dvojhvězd na excentrické orbitě, při průletu periastronem se obě složky deformují slapovými jevy a navzájem ohřejí reflexní efektem. Perioda oběhu heartbeat hvězdy S1598 je  $\sim 10$  dní (vlastní tvorba).

## 7.5 Zákrytový systém

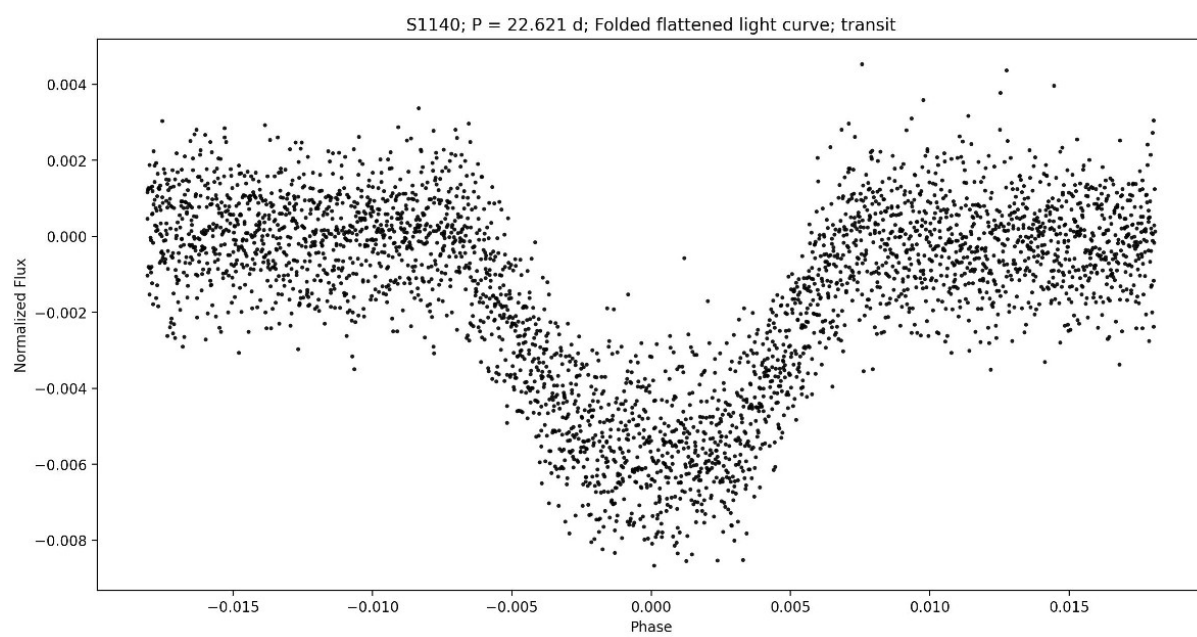
Světelná křivka hvězdy S1140 obsahuje v 9 sektorech pozorování 11 zákrytů s hloubkou  $\sim 7$  milimagnitud a periodou  $\sim 22,6$  dne. Sekundární minima nejsou pozorována. Okolí hvězdy bylo prohledáno a nejedná se o „blend“. Spektroskopické pozorování je třeba pro zjištění hmotnosti obíhajícího objektu, který by teoreticky mohl být větší exoplaneta, nebo hnědý či červený trpaslík.



Obrázek 76: Detrendovaná světelná křivka hvězdy S1140 (vlastní tvorba).



Obrázek 77: Fázová křivka hvězdy S1140. Světelná křivka byla před sfázováním detrendována pomocí funkce `.flatten()` (vlastní tvorba).



Obrázek 78: Fázová křivka hvězdy S1140 s bližším pohledem na tranzit (vlastní tvorba).



## 8 Závěr

Práce započala parametrizaci objevených proměnných hvězd automatickým zpracováním. Před publikací výsledků je nutno dokončit analýzu centroidů proměnnosti pro ověření proměnnosti pozorovaných hvězd. Další ověření proměnnosti může být provedeno za pomoci ostatních dostupných prohlídkových dat, jako je například Gaia [15] či ASAS-SN [44]. Některé parametry proměnnosti musí být dále zpřesněny manuální analýzou a typy proměnnosti ověřeny, za cílem snížení počtu hvězd s nejistou klasifikací VAR. Po dokončení analýzy budou výsledky publikovány do katalogu CzeV<sup>34</sup>. Příloha dokumentu obsahuje parametry hvězd určené pro detrendovanou světelnou křivku, zahrnující SXXX identifikátor hvězdy, typ proměnnosti, periodu, amplitudu, epochu a SNR.

Objevené proměnné hvězdy v této práci jsou velmi rozmanité a ideální pro navazující pozorování. Pulzující proměnné hvězdy nalezené v rámci této práce mají díky širokému pokrytí pozorování TESS vysoké SNR, tudíž jsou vhodné i pro analýzu vnitřní struktury hvězd za pomoci asteroseismologie. Objevené zákrytové dvojhvězdy mohou být dlouhodobě monitorovány pro vykreslení jejich O – C diagramů. Navazující spektroskopická pozorování hvězdy S1140 mohou přinést zajímavé výsledky o tomto zákrytovém systému. Řada eruptivních hvězd objevená v rámci této práce může být použita pro studium hvězdných magnetických polí a modelování hvězdných erupcí.

Práce demonstrovala metody práce s daty satelitu TESS a navrhla nové řešení problémů pozorovaných v těchto datech. Práce ukázala, že více jak desetina hvězd jsou doposud neobjevené, neparametrizované proměnné hvězdy, o jejichž parametrizaci se pomocí manuálních a automatických metod pokusila. V rámci práce je nalezeno 268 nových proměnných hvězd, včetně řady velmi zajímavých systémů ideálních pro navazující analýzu.

---

<sup>34</sup><http://var2.astro.cz/czev.php?lang=cz>

# Acknowledgement

This thesis includes data collected by the TESS mission, which are publicly available from the Mikulski Archive for Space Telescopes (MAST). Funding for the TESS mission is provided by NASA’s Science Mission directorate. We acknowledge the use of public TESS data from pipelines at the TESS Science Office and at the TESS Science Processing Operations Center.

This thesis has made use of data from the European Space Agency (ESA) mission *Gaia*<sup>35</sup>, processed by the *Gaia* Data Processing and Analysis Consortium (DPAC)<sup>36</sup>. Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

Some of the data presented in this thesis were obtained from the Mikulski Archive for Space Telescopes (MAST)<sup>37</sup>. STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5-26555. Support for MAST for non-HST data is provided by the NASA Office of Space Science via grant NNX13AC07G and by other grants and contracts.

This thesis has made use of the VizieR catalogue access tool, CDS, Strasbourg, France (DOI : 10.26093/cds/vizier)<sup>38</sup>. The original description of the VizieR service was published in 2000, A&AS 143, 23 [36].

This thesis has made use of the International Variable Star Index (VSX) database<sup>39</sup>, operated at AAVSO, Cambridge, Massachusetts, USA.

This thesis made use of Lightkurve, a Python package for Kepler and TESS data analysis (Lightkurve Collaboration, 2018).

This thesis made use of Astropy:<sup>40</sup> a community-developed core Python package and an ecosystem of tools and resources for astronomy [5].

This thesis has made use of the Exoplanet Follow-up Observation Program (ExoFOP; DOI: 10.26134/ExoFOP5) website, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

This thesis makes use of Python (Python Software Foundation)<sup>41</sup>.

---

<sup>35</sup><https://www.cosmos.esa.int/web/gaia>

<sup>36</sup><https://www.cosmos.esa.int/web/gaia/dpac/consortium>

<sup>37</sup><https://archive.stsci.edu/>

<sup>38</sup><https://vizier.cds.unistra.fr/viz-bin/VizieR>

<sup>39</sup><https://www.aavso.org/vsx/index.php>

<sup>40</sup><http://www.astropy.org>

<sup>41</sup><https://www.python.org/>

## 9 Literatura

- [1] AERTS, C. Probing the interior physics of stars through asteroseismology. Reviews of Modern Physics. Jan 2021, 93, s. 015001. doi: 10.1103/RevModPhys.93.015001. Dostupné z: <<https://link.aps.org/doi/10.1103/RevModPhys.93.015001>>.
- [2] ALBERT, J. et al. VHE  $\gamma$ -Ray Observation of the Crab Nebula and its Pulsar with the MAGIC Telescope. The Astrophysical Journal. feb 2008, 674, 2, s. 1037. doi: 10.1086/525270. Dostupné z: <<https://dx.doi.org/10.1086/525270>>.
- [3] ALDEBARAN GROUP FOR ASTROPHYSICS. Aldebaran [online]. Dostupné z: <<https://www.aldebaran.cz/glossary/print.php?id=294>>.
- [4] AMERICAN ASSOCIATION OF VARIABLE STAR OBSERVERS. Variable Star Index [online]. Dostupné z: <<https://www.aavso.org/vsx/index.php?view=about.vartypes>>.
- [5] ASTROPY COLLABORATION et al. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. The Astrophysical Journal. August 2022, 935, 2:167. doi: 10.3847/1538-4357/ac7c74.
- [6] BALONA, L. A. Starspots on A stars. Monthly Notices of the Royal Astronomical Society. 02 2017, 467, 2, s. 1830–1837. ISSN 0035-8711. doi: 10.1093/mnras/stx265. Dostupné z: <<https://doi.org/10.1093/mnras/stx265>>.
- [7] BALONA, L. A. – OZUYAR, D. Pulsation among TESS A and B stars and the Maia variables. Monthly Notices of the Royal Astronomical Society. 03 2020, 493, 4, s. 5871–5879. ISSN 0035-8711. doi: 10.1093/mnras/staa670. Dostupné z: <<https://doi.org/10.1093/mnras/staa670>>.
- [8] BRASSEUR, C. E. et al. Astrocut: Tools for creating cutouts of TESS images. Astrophysics Source Code Library, record ascl:1905.007, May 2019.
- [9] BRETTHORST, G. L. Generalizing the Lomb-Scargle periodogram—the nonsinusoidal case. AIP Conference Proceedings. 2001, 568, 1, s. 246–251. doi: 10.1063/1.1381889. Dostupné z: <<https://aip.scitation.org/doi/abs/10.1063/1.1381889>>.
- [10] BURKE, C. J. et al. TESS-Point: High precision TESS pointing tool. Astrophysics Source Code Library, record ascl:2003.001, March 2020.
- [11] CASAGRANDE, L. – VANDENBERG, D. A. On the use of Gaia magnitudes and new tables of bolometric corrections. Monthly Notices of the Royal Astronomical Society. September 2018, 479, 1, s. L102–L107. doi: 10.1093/mnrasl/sly104.
- [12] EUROPEAN SPACE AGENCY. Artist’s impression of Gaia [online]. Dostupné z: <<https://esahubble.org/images/heic1719f/>>.

- [13] FAUSNAUGH, M. et al. The TESS Mission Target Selection Procedure. Publications of the Astronomical Society of the Pacific. September 2021, 133, 1027: 095002. doi: 10.1088/1538-3873/ac1d3f.
- [14] FRIEDMANN, A. Über die Krümmung des Raumes. Zeitschrift für Physik. January 1922, 10, s. 377–386. doi: 10.1007/BF01332580.
- [15] GAIA COLLABORATION et al. The Gaia mission. Astronomy & Astrophysics. November 2016, 595:A1. doi: 10.1051/0004-6361/201629272.
- [16] GAIA COLLABORATION et al. Gaia Data Release 3: Summary of the content and survey properties. arXiv e-prints. July 2022, art. arXiv:2208.00211. doi: 10.48550/arXiv.2208.00211.
- [17] GINSBURG, A. et al. astroquery: An Astronomical Web-querying Package in Python. The Astronomical Journal. March 2019, 157, 3:98. doi: 10.3847/1538-3881/aafc33.
- [18] GRIGAHCÈNE, A. et al. HYBRID  $\gamma$  DORADUS –  $\delta$  SCUTI PULSATORS: NEW INSIGHTS INTO THE PHYSICS OF THE OSCILLATIONS FROM KEPLER OBSERVATIONS. The Astrophysical Journal Letters. mar 2010, 713, 2, s. L192. doi: 10.1088/2041-8205/713/2/L192. Dostupné z: <<https://dx.doi.org/10.1088/2041-8205/713/2/L192>>.
- [19] GUINAN, E. F. – MALONEY, F. P. The apsidal motion of the eccentric eclipsing binary DI Herculis - an apparent discrepancy with general relativity. The Astronomical Journal. August 1985, 90, s. 1519–1528. doi: 10.1086/113865.
- [20] HARRIS, C. R. et al. Array programming with NumPy. Nature. September 2020, 585, 7825, s. 357–362. doi: 10.1038/s41586-020-2649-2. Dostupné z: <<https://doi.org/10.1038/s41586-020-2649-2>>.
- [21] HUNTER, J. D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007, 9, 3, s. 90–95. doi: 10.1109/MCSE.2007.55.
- [22] JENKINS, J. M. et al. The TESS science processing operations center. In CHIOZZI, G. – GUZMAN, J. C. (Ed.) Software and Cyberinfrastructure for Astronomy IV, 9913, s. 99133E. International Society for Optics and Photonics, SPIE, 2016. doi: 10.1117/12.2233418. Dostupné z: <<https://doi.org/10.1117/12.2233418>>.
- [23] JETSU, L. – PORCEDDU, S. Shifting Milestones of Natural Sciences: The Ancient Egyptian Discovery of Algol’s Period Confirmed. PLOS ONE. dec 2015, 10, 12, s. e0144140. doi: 10.1371/journal.pone.0144140. Dostupné z: <<https://doi.org/10.1371/journal.pone.0144140>>.
- [24] JORDI, C. et al. Gaia broad band photometry. Astronomy & Astrophysics. November 2010, 523:A48. doi: 10.1051/0004-6361/201015441.

- [25] KOCH, D. G. et al. KEPLER MISSION DESIGN, REALIZED PHOTOMETRIC PERFORMANCE, AND EARLY SCIENCE. The Astrophysical Journal Letters. mar 2010, 713, 2, s. L79. doi: 10.1088/2041-8205/713/2/L79. Dostupné z: <<https://dx.doi.org/10.1088/2041-8205/713/2/L79>>.
- [26] LADA, C. J. Stellar Multiplicity and the Initial Mass Function: Most Stars Are Single. The Astrophysical Journal. feb 2006, 640, 1, s. L63–L66. doi: 10.1086/503158. Dostupné z: <<https://doi.org/10.1086/503158>>.
- [27] LEAVITT, H. S. – PICKERING, E. C. Periods of 25 Variable Stars in the Small Magellanic Cloud. Harvard College Observatory Circular. March 1912, 173, s. 1–3.
- [28] LIGHTKURVE COLLABORATION et al. Lightkurve: Kepler and TESS time series analysis in Python. Astrophysics Source Code Library, December 2018.
- [29] LOMB, N. R. Least-Squares Frequency Analysis of Unequally Spaced Data. Astrophysics and Space Science. February 1976, 39, 2, s. 447–462. doi: 10.1007/BF00648343.
- [30] LUNDMARK, K. The determination of the curvature of space-time in de Sitter’s world. Monthly Notices of the Royal Astronomical Society. June 1924, 84, s. 747–770. doi: 10.1093/mnras/84.9.747.
- [31] LUSTIG-YAEGER, J. et al. A JWST transmission spectrum of a nearby Earth-sized exoplanet, 2023. Dostupné z: <<https://arxiv.org/abs/2301.04191>>.
- [32] MAYER, P. et al. Towards a consistent model of the hot quadruple system HD 93206 = QZ Carinae - I. Observations and their initial analyses. Astronomy & Astrophysics. 2022, 666, s. A23. doi: 10.1051/0004-6361/202142108. Dostupné z: <<https://doi.org/10.1051/0004-6361/202142108>>.
- [33] MIKULÁŠEK, Z. Proměnné hvězdy [online]. Dostupné z: <<https://astro.physics.muni.cz/download/documents/skripta/>>.
- [34] MURPHY, S. J. et al. Gaia-derived luminosities of Kepler A/F stars and the pulsator fraction across the  $\delta$  Scuti instability strip. Monthly Notices of the Royal Astronomical Society. 03 2019, 485, 2, s. 2380–2400. ISSN 0035-8711. doi: 10.1093/mnras/stz590. Dostupné z: <<https://doi.org/10.1093/mnras/stz590>>.
- [35] NASA’S GODDARD SPACE FLIGHT CENTER. TESS Southern Hemisphere Sector Images [online]. Dostupné z: <<https://svs.gsfc.nasa.gov/13267>>.
- [36] OCHSENBEIN, F. – BAUER, P. – MARCOUT, J. The VizieR database of astronomical catalogues. Astronomy and Astrophysics Supplement. April 2000, 143, s. 23–32. doi: 10.1051/aas:2000169.
- [37] PEJCHA, O. et al. The complex dynamical past and future of double eclipsing binary CzeV343: Misaligned orbits and period resonance. Astronomy & Astrophysics. November 2022, 667:A53. doi: 10.1051/0004-6361/202244335.

- [38] POWELL, B. P. et al. TIC 168789840: A Sextuply Eclipsing Sextuple Star System. The Astronomical Journal. mar 2021, 161, 4, s. 162. doi: 10.3847/1538-3881/abddb5. Dostupné z: <<https://dx.doi.org/10.3847/1538-3881/abddb5>>.
- [39] PRŠA, A. et al. TESS Eclipsing Binary Stars. I. Short-cadence Observations of 4584 Eclipsing Binaries in Sectors 1-26. The Astrophysical Journal Supplement Series. January 2022, 258, 1:16. doi: 10.3847/1538-4365/ac324a.
- [40] PRŠA, A. et al. PHYSICS OF ECLIPSING BINARIES. II. TOWARD THE INCREASED MODEL FIDELITY. The Astrophysical Journal Supplement Series. dec 2016, 227, 2, s. 29. doi: 10.3847/1538-4365/227/2/29. Dostupné z: <<https://dx.doi.org/10.3847/1538-4365/227/2/29>>.
- [41] RICKER, G. R. et al. Transiting Exoplanet Survey Satellite (TESS). Journal of Astronomical Telescopes, Instruments, and Systems. January 2015, 1:014003. doi: 10.1117/1.JATIS.1.1.014003.
- [42] ROSSI, B. B. THE CRAB NEBULA ANCIENT HISTORY AND RECENT DISCOVERIES. 1 1969. Dostupné z: <<https://www.osti.gov/biblio/4127351>>.
- [43] SCARGLE, J. D. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. The Astrophysical Journal. December 1982, 263, s. 835–853. doi: 10.1086/160554.
- [44] SHAPPEE, B. J. et al. The Man behind the Curtain: X-Rays Drive the UV through NIR Variability in the 2013 Active Galactic Nucleus Outburst in NGC 2617. The Astrophysical Journal. June 2014, 788, 1:48. doi: 10.1088/0004-637X/788/1/48.
- [45] STASSUN, K. G. et al. The Revised TESS Input Catalog and Candidate Target List. The Astronomical Journal. October 2019, 158, 4:138. doi: 10.3847/1538-3881/ab3467.
- [46] STRASSMEIER, K. G. Starspots. The Astronomy and Astrophysics Review. September 2009, 17, 3, s. 251–308. doi: 10.1007/s00159-009-0020-6.
- [47] TESS Observations [online]. Dostupné z: <<https://tess.mit.edu/observations>>.
- [48] THE PANDAS DEVELOPMENT TEAM. pandas-dev/pandas: Pandas, February 2023. Dostupné z: <<https://doi.org/10.5281/zenodo.7658911>>.
- [49] VANDERPLAS, J. T. Understanding the Lomb–Scargle Periodogram. The Astrophysical Journal Supplement Series. may 2018, 236, 1, s. 16. doi: 10.3847/1538-4365/aab766. Dostupné z: <<https://dx.doi.org/10.3847/1538-4365/aab766>>.
- [50] VANDERSPEK, R. et al. TESS Instrument Handbook [online]. Dostupné z: <<https://archive.stsci.edu/missions-and-data/tess>>.
- [51] VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020, 17, s. 261–272. doi: 10.1038/s41592-019-0686-2.

- [52] WATSON, C. L. – HENDEN, A. A. – PRICE, A. The International Variable Star Index (VSX). Society for Astronomical Sciences Annual Symposium. May 2006, 25, s. 47.
- [53] ZASCHE, P. – HENZL, Z. – MAŠEK, M. Multiply eclipsing candidates from the TESS satellite. Astronomy & Astrophysics. August 2022, 664:A96. doi: 10.1051/0004-6361/202243723.