MASARYKOVA UNIVERZITA

Přírodovědecká fakulta

Ústav teoretické fyziky a astrofyziky

DIPLOMOVÁ PRÁCE

Automatizace zpracování a analýzy fotometrických dat tranzitujících exoplanet z IČ přístroje HAWK-I observatoře ESO

Martin Blažek

Vedoucí práce: Dr. Petr Kabáth

Brno 2017

Bibliografický záznam

Autor: Martin Blažek
Přírodovědecká fakulta, Masarykova univerzita
Ústav teoretické fyziky a astrofyziky

Název práce: Automatizace zpracování a analýzy fotometrických dat tranzitujících exoplanet z IČ přístroje HAWK-I observatoře ESO

Studijní program: Fyzika

Studijní obor: Teoretická fyzika a astrofyzika

Vedoucí práce: Dr. Petr Kabáth

Akademický rok: 2016/2017

Počet stran: 73

Klíčová slova: Extrasolární planeta, tranzit exoplanety, zákryt exoplanety, atmosféra exoplanety, IČ přístroj, IČ fotometrie, světelná křivka

Bibliographic entry

Author: Martin Blažek
Faculty of Science, Masaryk University
Department of Theoretical Physics and Astrophysics

Title of thesis: Automation of processing and photometric data analysis for transiting exoplanets observed with ESO NIR instrument HAWK-I

Degree programme: Physics

Field of study: Theoretical Physics and Astrophysics

Supervisor: Dr. Petr Kabáth

Academic year: 2016/2017

Number of pages: 73

Keywords: Extrasolar planet, exoplanetary transit, exoplanetary eclipse, exoplanet atmosphere, NIR instrument, NIR photometry, light curve

Abstrakt: Automatizace zpracování a analýzy fotometrických dat tranzitujících exoplanet z IČ přístroje HAWK-I observatoře ESO.

Diplomová práce má za cíl zabývat se tranzity a zákryty (okultacemi) exoplanety WASP-18b získaných na observatoři ESO Paranal přístrojem HAWK-I (High Acuity Wide-field K-band Imager). Hlavním úkolem je automatizace softwaru pro zpracovávání dat a pochopení systematických chyb, tak, aby byla získána co největší přesnost fotometrických měření. Výstupem softwaru bude světelná křivka systému v infračerveném oboru a následné modelování zákrytů pomocí MCMC (Markov-Chain-Monte-Carlo) kódu. Nejdůležitějším výsledkem práce je tedy vytvoření softwarového balíčku, který bude schopen zpracovat data od základní fotometrické redukce až k modelování tranzitu/zákrytu jakéhokoliv systému pro případná budoucí pozorování. Možnosti a vlastnosti softwaru budou předvedeny na skutečných fotometrických datech zákrytu WASP-18b.

Klíčová slova: extrasolární planeta, tranzit exoplanety, zákryt exoplanety, atmosféra exoplanety, IČ přístroj, IČ fotometrie, světelná křivka.

Abstract: Automation of processing and photometric data analysis for transiting exoplanets observed with ESO NIR instrument HAWK-I.

The aim of the thesis is to deal with transits and eclipses (occultations) of an exoplanet WASP-18b obtained at an observatory ESO Paranal with HAWK-I instrument (High Acuity Wide-field K-band Imager). The main task is to develop automation of software for data processing and gain understanding of systematic errors so that the best precision of photometric measurements can be achieved. A light curve in the infrared band will be an output of the software as well as consequential modelling of eclipses via MCMC (Markov-Chain-Monte-Carlo) code. The most important result of the thesis is then creation of the software package (pipeline) which will enable users to process data from basic photometric reduction up to transit/eclipse modelling of any system for eventual future observations. Options and characteristics of the software will be demonstrated on real photometric data of an eclipse of WASP-18b.

Keywords: extrasolar planet, exoplanetary transit, exoplanetary eclipse, exoplanet atmosphere, NIR instrument, NIR photometry, light curve.

MASARYKOVA UNIVERZITA
Přírodovědecká fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Akademický rok: 2016/2017

Ústav: Ústav teoretické fyziky a astrofyziky

Student: Bc. Martin Blažek

Program: Fyzika

Obor: Teoretická fyzika a astrofyzika

Směr: Astrofyzika

Ředitel *Ústavu teoretické fyziky a astrofyziky* PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje diplomovou práci s názvem:

Název práce: Automatizace zpracování a analýzy fotometrických dat tranzitujících exoplanet z IČ přístroje HAWK-I observatoře ESO

Název práce anglicky: Automation of processing and photometric data analysis for transiting exoplanets observed with ESO NIR instrument HAWK-I

Oficiální zadání:
Diplomová práce má za cíl zabývat se tranzity a zákryty (okultacemi) exoplanety WASP-18b získaných na observatoři ESO Paranal přístrojem HAWK-I (High Acuity Wide-field K-band Imager). Hlavním úkolem je automatizace softwaru pro zpracovávání dat a pochopení systematických chyb, tak, aby byla získána co největší přesnost fotometrických měření. Výstupem softwaru bude světelná křivka systému v infračerveném oboru a následné modelování zákrytů pomocí MCMC (Markov-Chain-Monte-Carlo) kódu. Nejdůležitějším výsledkem práce je tedy vytvoření softwarového balíčku, který bude schopen zpracovat data od základní fotometrické redukce až k modelování tranzitu/zákrytu jakéhokoliv systému pro případná budoucí pozorování. Možnosti a vlastnosti softwaru budou předvedeny na skutečných fotometrických datech zákrytu WASP-18b. Práce bude napsána v anglickém jazyce.
Literatura
Berry, R., Burnell, J., 2005, "The Handbook of Astronomical Image Processing"
Winn, J., "Transits and occultations", 2005, arXiv:1001.2010v3
Seager, S., Demming, D., (2010), "Exoplanet Atmospheres", ARAA, 48, 631
Bean et al., 2011, "The Optical and Near-Infrared Transmission Spectrum of the Super-Earth GJ 1214b: Further Evidence for a Metal-rich Atmosphere", ApJ, 743, 92B
Alonso et al., 2009, "Ground-based Near-infrared Observations of the Secondary Eclipse of CoRoT-2b", AJ, 139, 1481A

Jazyk závěrečné práce: angličtina

Vedoucí práce: Bc. Petr Kabáth, Dr. rer. nat.

Datum zadání práce: 31. 12. 2014

V Brně dne: 11. 10. 2016

Souhlasím se zadáním (podpis, datum):

.....	.....	.....
Bc. Martin Blažek	Bc. Petr Kabáth, Dr. rer. nat.	prof. Rikard von Unge, Ph.D.
student	**vedoucí práce**	**ředitel Ústavu teoretické fyziky a astrofyziky**

Acknowledgement

I would like to thank to my supervisor Dr. Petr Kabáth (Astronomical Institute, Academy of Sciences, Ondřejov), consultant Mgr. Tereza Klocová, Ph.D. (Astronomical Institute, Academy of Sciences, Ondřejov), and to Mgr. Tereza Jeřábková (Charles University, Prague) and Dipl.-Phys. Felix Rucker (Technical University of Munich) for their help with Python.

I declare that this thesis was written by my own with the use of cited resources.

Brno January 5th, 2017 .....

Martin Blažek

There's a dark but there's a light,
there could be who do not fight.
it is long time and it is far,
we don't still know which is Their star.

But who wants to see, who wants to know,
he upheaves his eyes and searches slow.
On this journey there's not the only cliff,
yet it's worth for those who say "What if".

THE AUTHOR

Contents

1. Introduction **11**
- 1.1. History . . . 12
- 1.2. WASP-18 . . . 18
- 1.3. Exoplanet atmospheres . . . 19
  - 1.3.1. The atmosphere of WASP-18b . . . 19

2. Methods to detect and characterise exoplanets **21**
- 2.1. Geometry of eclipses . . . 22
  - 2.1.1. Transmission spectroscopy . . . 23
  - 2.1.2. Emission photometry . . . 24

3. Instruments **25**
- 3.1. ESO . . . 25
- 3.2. Near-Infrared Instruments . . . 25
- 3.3. High Acuity Wide-field K-band Imager—HAWK-I . . . 26
- 3.4. Fast Photometry Mode . . . 28

4. The data set **30**
- 4.1. The header . . . 31
- 4.2. Airmass during observations . . . 32
- 4.3. Seeing during observations . . . 32
- 4.4. Positions of stars on a chip . . . 34
- 4.5. Reduction frames . . . 36
  - 4.5.1. Dark frames . . . 36
  - 4.5.2. Flat-field frames . . . 37

5. Data reduction **39**
- 5.1. Developed pipeline . . . 39
  - 5.1.1. The script `pipeline.py` . . . 39
- 5.2. Dark frames . . . 40
  - 5.2.1. Dark current measurement . . . 40
  - 5.2.2. The script `darks.py` . . . 41
  - 5.2.3. The script `darksStat.py` . . . 42
- 5.3. Flat-field frames . . . 43
  - 5.3.1. Scripts `flats.py` and `flatsD.py` . . . 44
  - 5.3.2. The script `flatsStat.py` . . . 44
- 5.4. Images before and after reduction compared . . . 45
  - 5.4.1. Scripts `reductFlats.py` and `reductFlatsDarks.py` . . . 47
- 5.5. Bad pixel mask . . . 49

6. Data processing **51**
- 6.1. Aperture photometry . . . 51
  - 6.1.1. The script `photometry.py` . . . 52
- 6.2. Apertures . . . 53
  - 6.2.1. The script `aperts.py` . . . 54
- 6.3. Light curve . . . 54
  - 6.3.1. The script `lightCurve.py` . . . 57
- 6.4. Standard deviation . . . 58
  - 6.4.1. The script `std.py` . . . 60

6.5. Fitting 61
6.6. Discussion of the atmosphere of WASP-18b 65

7. Conclusion **67**

Bibliography **68**

Internet sources **70**

Appendix—Pipeline description **71**

1. Introduction

There are still a lot of people who believe that we, mankind, are alone in the Universe and that the Earth is the only planet in the Universe where a life exists. Yes, we have not so far got any proof that the truth is different. However, there are people, especially scientists, who are searching for those proofs because they simply do not think that a life is connected strictly with the Earth. Another reason for doing that can be the attitude that they do not want to be limited just to the Earth. It is only a tiny dot in the Sun's nearest surroundings, let alone about its importance in the Galaxy and the whole vast Universe [figure 1].

Recently there has been a rapid progress in an exoplanetary field of astronomy and astrophysics. While relatively recently anyone hardly thought about planets beyond the solar system, in the last few decades we discovered about two thousand extrasolar planets and studied their physical characteristics. Moreover, it is even possible to detect their atmospheres and explore their composition and physical processes inside them.

With this progress new questions have emerged, or in other words, the questions of sci-fi have become fundamental ones of this field of astronomy. We can also look at this issue from opposite side. Exoplanetary research started because scientists and people generally had begun to ask those particular questions like following:

- *Is there a life in the Universe?*
- *Is our solar system a unique one in the Galaxy (Universe)?*
- *How does an evolution of planetary systems take place?*
- *What are environments of planetary systems like?*
- *What about a statistical distribution of planetary systems in the Galaxy (Universe)?*

Any answer to these and related questions is neither simple nor direct but definitely it is worth dealing with that. And searching for answers to these questions is the right motivation for me to write my master thesis devoted to this topic.

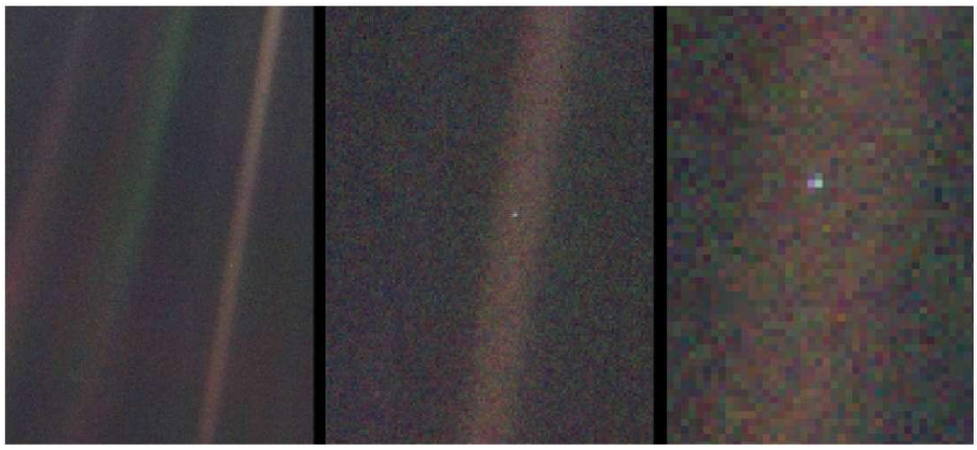


Figure 1: Original (from left) and zoomed images of the Earth taken by Voyager 1 from the distance about 6 billion kilometres. Taken from (E0).

This thesis is divided into seven chapters. The introductory chapter is devoted to history of exoplanets. Chapter two describes detection methods. It also includes exoplanetary atmospheres, especially what we can find out about them currently. Near-infrared

instrument HAWK-I of ESO and its fast observation mode is presented in chapter three. The processed data set description, observation conditions and basic information about reduction frames are involved in chapter four. The fifth chapter describes data reduction and shows some statistics while the sixth chapter describes the very data processing including the obtained results. The last, seventh chapter, analyses and discusses results and whole the process of my research and concludes the whole thesis. The appendix presents the decription of a developed pipeline.

1.1. History

If we look closer into the history of discovery of exoplanets, we can start with Epicurus (more than 2000 years ago) and his statement: *"There are infinite worlds both like and unlike this world of ours."* He believed that there were living creatures and plants in all worlds. In the 13th century another philosopher Albertus Magnus asked a question to himself if our world is the only one or if there are more worlds. Giordano Bruno was very optimistic about this problem as he asserted following (1584): *"There are countless Suns and countless Earths all rotating around their Suns in exactly the same way as the seven planets of our system ... The countless worlds in the Universe are no worse and no less inhabited than our Earth."* In Newton's age Christopher Huygens even believed (in his book "Cosmotheoros"—Huygens (1698)) that people on similar planets to the Earth had astronomy and other arts (Seager, 2010).

Searching for exoplanets began with an expansion of astrometry between the first and the second half of nineteenth century. The target was a binary system 70 Ophiuchi (this was the first published claim of a planet beyond our solar system—*Jacob, 1855*). In the 1960s a planet around Barnard's star was announced (van de Kamp, 1963). Eventually, none of these two "planets" was confirmed as real planets. The first two real exoplanets, announced in 1992, were orbiting a pulsar—PSR 1257+12 (Wolszczan and Frail, 1992). However, since they are supposed to be dead worlds due to a deadly radiation from their host star, the exoplanet research is focused on Sun-like stars.

At last, first announcement about an exoplanet around solar type star came in 1995. The planet discovered by radial velocity method has got a mass of a 0.5 M_J (Jupiter mass) and of an orbital period of 4.2 days. Its host star is a Sun-like star—51 Peg (Mayor and Queloz, 1995). That is the reason why the word "valid" is used—Sun-like stars can host planets with a life as we know it from the Earth in contrast to probably "dead worlds" orbiting pulsars. Discovery of this planet may be the most important discovery connected with exoplanets. The reason is that this planet has changed our view on planet formation and evolution theories. Giant planets have never been expected to be so close to their parent star as this one is (seven times closer to the Sun than Mercury). These planets are called hot Jupiters because of their size (similar to Jupiter planet of our solar system) and high temperature caused by the short distance to their star.

Since the first discovery of an exoplanet the number of detection techniques has increased. There are five (main) different techniques: Direct imaging, Microlensing, Radial velocity, Timing and Transit technique [figure 2]. Regarding physical characteristics of planets there is a broad diversity of them—vast range of masses, radii, and orbital parameters. This was a very surprising finding. Even though the detection techniques are still improving, we are still very far away from getting an image of an Earthlike planet just like we know images of our Earth from close outer space. At the present time, all the images of exoplanets are spatially unresolved, i.e., only point sources—dots.

But what is actually a planet? According to the definition of the International Astronomical Union (IAU) established by astronomers in 2006, a planet in the Solar System is:

"A celestial body that
a) is in orbit around the Sun,
b) has sufficient mass for its self-gravity to overcome rigid body forces so that it assumes a hydrostatic equilibrium (nearly round) shape, and
c) has cleared the neighbourhood around its orbit" (E1).

Another (working) definition has been established by Working Group on Extrasolar Planets of the IAU. The definition reads as follows (Boss et al., 2007):

"Objects with true masses below the limiting mass for thermonuclear fusion of deuterium (currently calculated to be 13 M_J for objects of solar metallicity) that orbit stars or stellar remnants are 'planets' (no matter how they formed). The minimum mass/size required for an extrasolar object to be considered a planet should be the same as that used in our solar system."

Although there are no official definitions for an exoplanet, some commonly accepted definitions exist. They include a few categories of exoplanets such as Giant planet, Terrestrial planet, Habitable planet, Earth-like planet and so on. Till today there are 3545 confirmed exoplanets in 2659 planetary systems and 598 multiple planet systems[1] (E2).

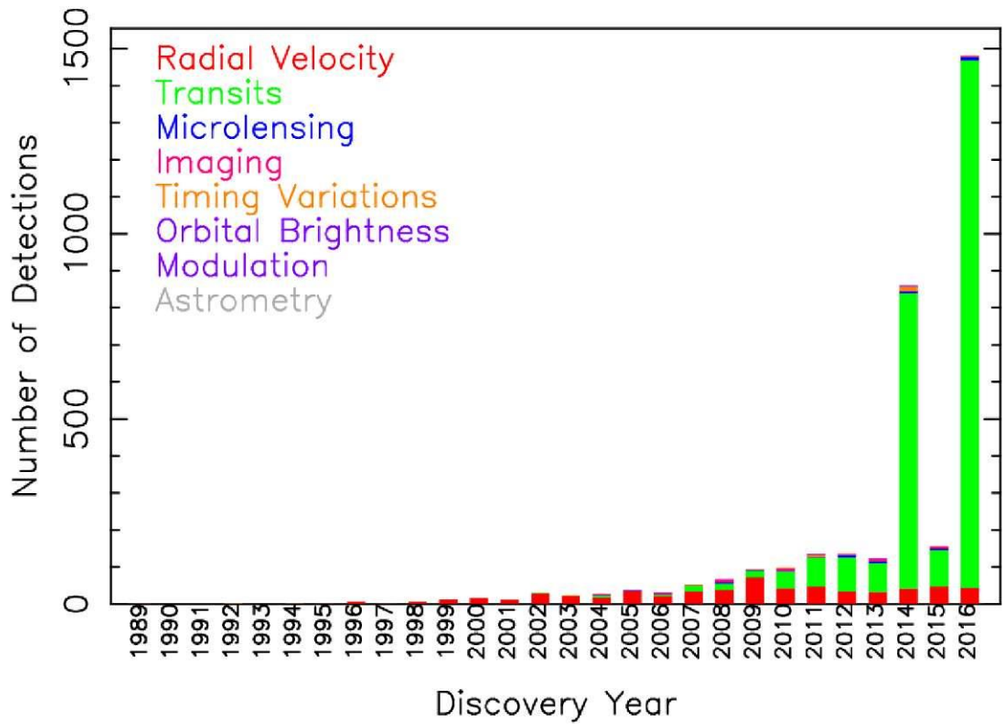


Figure 2: Number of confirmed exoplanets per year according to detection methods (17 November 2016). Taken from (E14).

[1] 11 December 2016

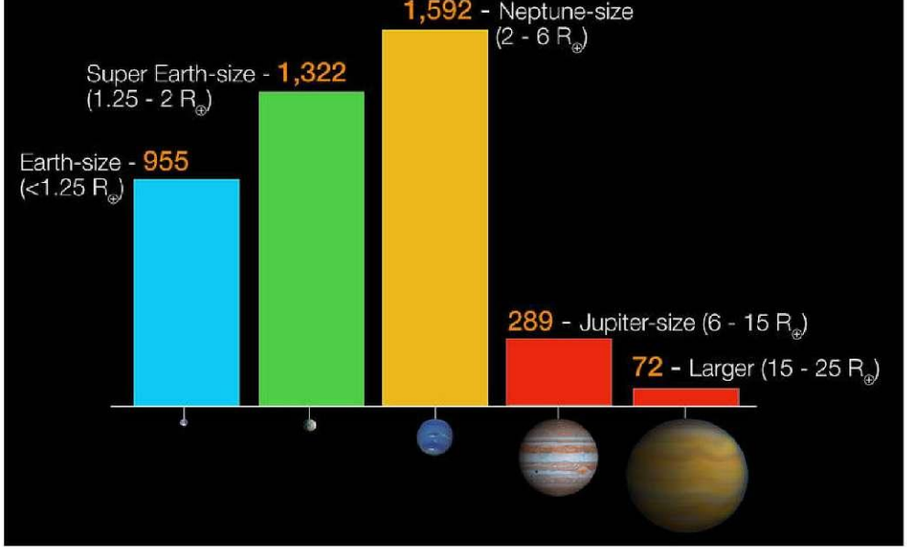


Figure 3: Kepler planet candidates divided into groups according to their sizes (23 July 2015). Taken from (E8).

As previously mentioned, the first exoplanet has been discovered by the radial velocity method [figure 4 on the left]. This method is based on the fact that a star orbits the star-planet common centre of mass. The motion of a star is described by a few components. One of them is the radial velocity of the star's motion which means the motion towards an observer's line-of-sight. The advantage of this technique is a possibility to detect planets with relatively low masses. However, it has got some disadvantages. The main one is a limitation connected with a determination of an exoplanet's mass. What we can get is a product of $\boldsymbol{M_{\mathrm{p}} \sin i}$ ($\boldsymbol{M_{\mathrm{p}}}$ is the planet mass and $\boldsymbol{i}$ is the inclination of the orbit). Since we do not know the inclination angle, we can determine only the minimum mass of the exoplanet. The number of exoplanets discovered by radial velocity technique is almost 500 (E3).

The unknown inclination angle can be determined by a transit method (also called primary eclipse). A transit occurs when, seen from the Earth, a planet passes in front of a star. During this event the planet blocks a certain amount of light of the star and at that moment we can detect a brightness drop which is a typical sign of a planetary transit. This drop of light then can be seen in a light curve (LC)—dependency of a radiation flux on time. To detect a transit, the star-exoplanet plane of orbit has to be aligned with the plane of an observer's sight. The bigger radius of a star and smaller semimajor axis of a planet orbiting the star, the higher the probability of a transit. Small semimajor axis means short period of the orbit of a planet and it implies a fact that by a transit method we detect mainly short-period exoplanets. From transits we can obtain information which can not be obtained by radial velocity technique. The first exoplanet discovered by the transit method was confirmed in 2000 (Charbonneau et al., 2000). The planet is named HD 209458b [figure 4 on the right]. Till today about 1300 exoplanets have been discovered by the transit method (E2).

Since the number of methods had been increasing and new exoplanets had still been discovered, various space missions and ground based projects were launched. The most important are the following:

- **CoRoT (COnvection ROtation and planetary Transits)**—French space mission aimed at searching for exoplanets of a terrestrial size and at detecting astroseismology. It was working between 2007 and 2013 and found about 25 exoplanets (E4).
- **Gaia**—Space observatory of the European Space Agency (ESA) which was launched at the end of 2013 and which is planned to work until 2019. By using astrometry and transit method it should discover exoplanets of higher masses, however, due to a technical problem on Gaia it has not discovered any exoplanet yet (E5).
- **HAT (Hungarian-made Automated Telescope Network)**—network of six telescopes. The goal of the project was to find variable stars and to detect exoplanets using the transit method. This project operated from 1999 to 2001 and discovered about 60 exoplanets (E6). Since 2009 HAT-South (HATS) had joined this project and discovered almost 20 exoplanets till today (E7).
- **Kepler**—NASA space observatory launched in 2009. It is aimed at searching for Earth-like planets nearby habitable zone. Kepler's photometer continually monitors thousands of main sequence stars and for sequential processing on the Earth the transit method is used. Today there are 2330 confirmed exoplanets and 4706 candidates[2] [figure 3] (E8).
- **K2 ("Second Light")**—Due to certain failures of Kepler's spacecraft K2 compensated for Kepler mission. For a planet searching K2 uses a method of transit and gravitational microlensing. It is planned to work until 2018 and till today it has discovered over 170 exoplanets and over 450 are planet candidates (E9).
- **OGLE (Optical Gravitational Lensing Experiment)**—Polish project, mainly aimed at discovering dark matter, uses microlensing technique. The project itself had begun in 1992 but only starting the third phase in 2001 discoveries of exoplanets began. It discovered more than 30 exoplanets till today (E10).
- **WASP (Wide Angle Search for Planets)**—An international project, which began in 1999, uses the method of transit photometry. It found its first exoplanet in 2006 and till today the number of discovered exoplanets by WASP is more than 130 exoplanets (E11).

Currently there are some research projects being planned whose purpose is (among others) exoplanet research. The most important ones are following:

- **E-ELT (European Extremely Large Telescope)**—It will be the largest telescope of the world, working at optical/near-infrared band. It is now under construction on Cerro Armazones mountain in Chile by the European Southern Observatory. Its first light is planned for the year 2024. Apart from other scientific goals E-ELT will search for exoplanets, including the planets whose mass is lower than the mass of the Earth, image Earh-like planets, show Jupiter-like planets details and also study atmospheres of exoplanets (E12).
- **JWST (James Webb Space Telescope)**—Space observatory with the largest infrared telescope, operated by ESA, Canadian Space Agency (CSA), and National Aeronautics and Space Administration (NASA). It is now under construction and planned to be launched in October of 2018. One of main aims of JWST is to study exoplanets by using transit method, atmospheres of exoplanets by spectroscopy method and direct imaging of exoplanets (E13).

[2] 11 December 2016 (E8).

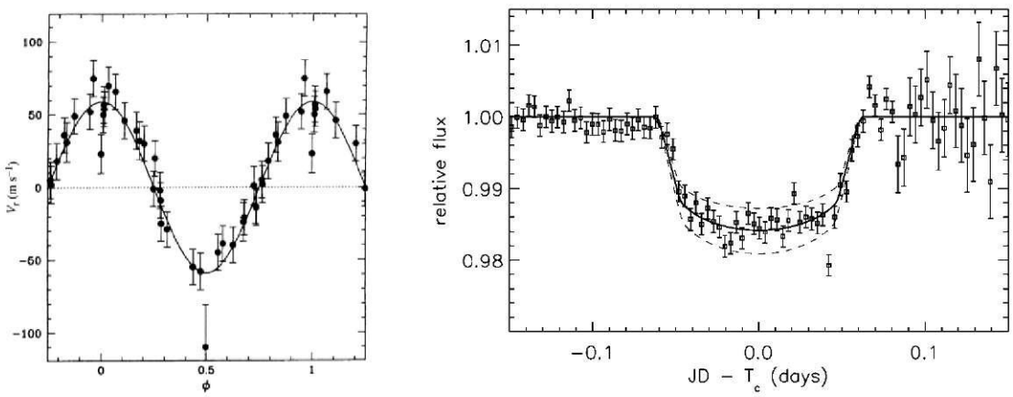


Figure 4: Left: The radial velocity curve of the host star of the first discovered exoplanetary system—51 Pegasi. Taken from Mayor and Queloz (1995). Right: Light curve (original data binned into 5 min averages) of the star HD 209458—the star with the first exoplanet discovered by the method of transit. Taken from Charbonneau et al. (2000).

And what can we say about the properties of all the discovered exoplanets? It is important to say that searching for exoplanets is affected by a selection effect. Although there is a huge variety in masses, distances from their parent stars and other characteristics [figure 5], a lot of discovered planets are giant planets due to selection effect. Among these the most common exoplanets are "hot Jupiters", giant exoplanets orbiting very close to their parent stars, and "super-Earths" (also called "mini-Neptunes"), exoplanets with greater mass than the Earth but lower than Uranus and Neptune. However, latest studies show that the most common are rather smaller planets.

However, as detection methods are still improving and new methods appear, nowadays even small and low-mass exoplanets are being discovered as seen in figure 6. Most of the small exoplanets began to be discovered after Kepler observatory had been launched which was its purpose. Thanks to this mission, some of newly discovered exoplanets were found to be in the habitable zone, which is the distance from a star where the temperature of a planet allows liquid water. This condition supports life as we know it on the Earth. And this is the target of all the searching.

Exoplanets are very diverse—gas planets, rocky planets, hot planets, cold planets etc. There is a huge range of values of physical characteristics. And since new exoplanets are still discovered, we can not say yet that the ranges of characteristics are final. Table 1 shows "extremes" of important physical characteristics of exoplanets to date.

Radius				Mass			
[$R_\oplus$]		[R_J]		[$M_\oplus$]		[M_J]	
min.	max.	min.	max.	min.	max.	min.	max.
$3{\cdot}10^{-1}$	28	$3{\cdot}10^{-2}$	2.5	$7{\cdot}10^{-4}$	$2{\cdot}10^{4}$	$2{\cdot}10^{-6}$	63.3

Semimajor axis [AU]		Orbital period [d]		Eccentricity	
min.	max.	min.	max.	min.	max.
$4.4{\cdot}10^{-3}$	$6.9{\cdot}10^{3}$	$7{\cdot}10^{-2}$	$7.3{\cdot}10^{5}$	0	0.97

Table 1: Diversity of discovered planets—their minimal and maximal values of important physical characteristics (27 March 2016). Taken from (E2).

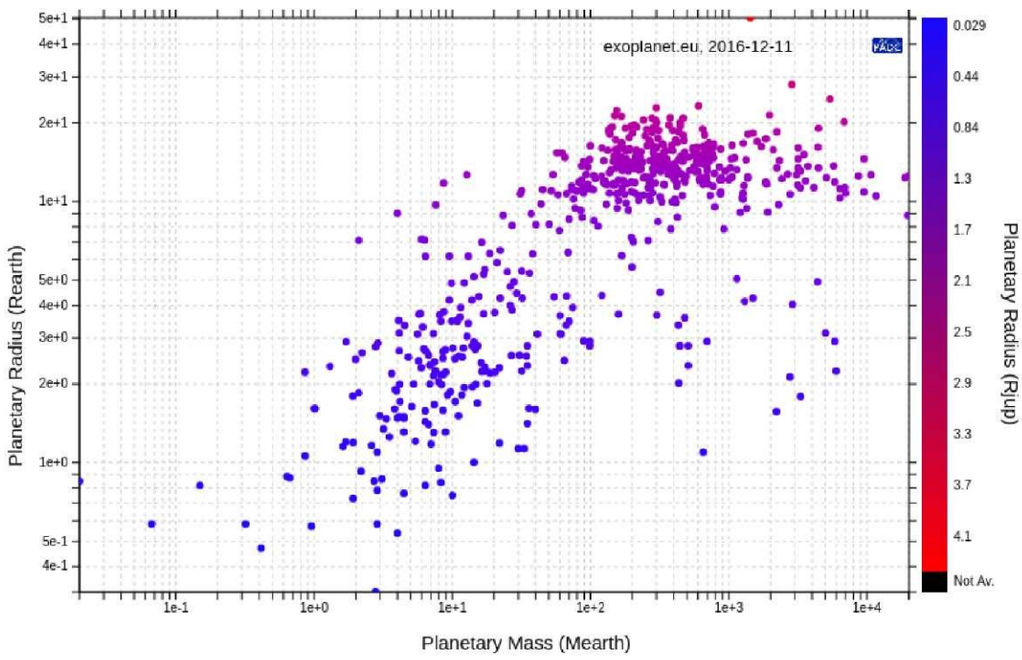


Figure 5: Range of masses and radii of discovered exoplanets (612 of them are represented). Taken from (E2).

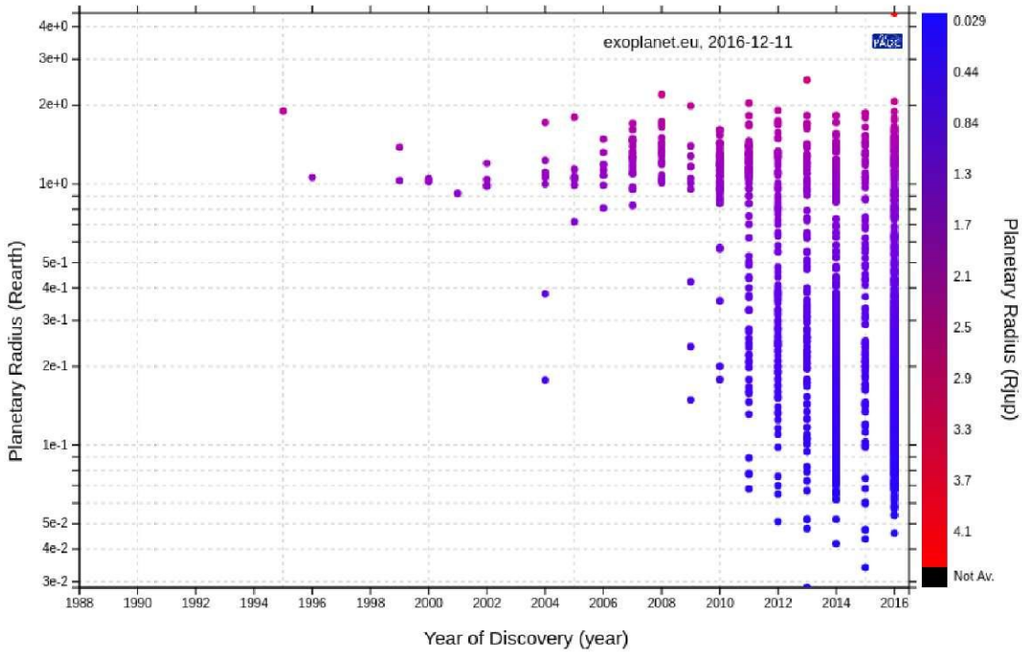


Figure 6: Radii of exoplanets as the range of them was increasing since the first exoplanets were discovered (2735 of them are represented). Taken from (E2).

1.2. WASP-18

The object of our interest is the exoplanet WASP-18b. It was discovered by an observation of a primary transit in 2009 (Hellier, 2009) and belongs to hot-Jupiter type of exoplanets. Its mass is (10.43 ± 0.30) M_J (Nymeyer et al., 2010) and its radius is (1.165 ± 0.055) R_J (Nymeyer et al., 2010). The orbital period is $(0.941 \pm 4 \cdot 10^{-7})$ days (Nymeyer et al., 2010) with semi-major axis (0.02026 ± 0.00068) AU (Hellier, 2009). The inclination is (86.0 ± 2.5) deg (Hellier, 2009) and orbital eccentricity 0.0091 ± 0.0012 (Nymeyer et al., 2010). Due to the short distance from its parent star and relatively high temperature of the star, the surface temperature of the planet is about 3200 K (which is a temperature of an early M-class stars) (Nymeyer et al., 2010).

The host star of the exoplanet, WASP-18 (HD 10069), is a star of spectral type F6 with effective temperature (6400 ± 100) K (Hellier, 2009). The apparent magnitude at filter V is 9.3 mag (Hellier, 2009). Its mass is (1.25 ± 0.13) $M_\odot$ (Hellier, 2009) and its radius is (1.216 ± 0.067) $R_\odot$ (Hellier, 2009). The distance of the star from the Earth is (100 ± 10) pc (Hellier, 2009). This star is found in the constellation of Phoenix and its equatorial coordinates are following: $\mathcal{RA}$: $01^h37^m24.95^s$, $\mathcal{DEC}$: $-45°40'40.80''$.

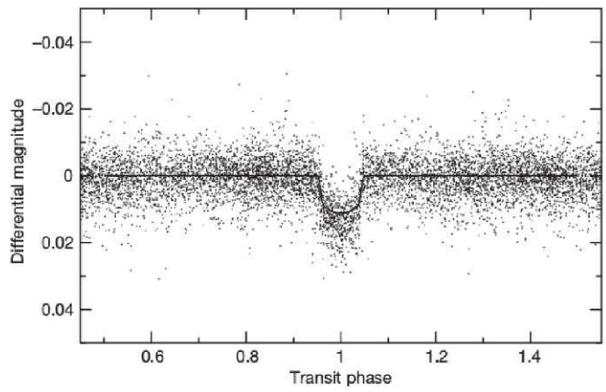


Figure 7: Discovering light curve together with a model curve of WASP-18. Data were obtained during years 2006 and 2007. Taken from Hellier (2009).

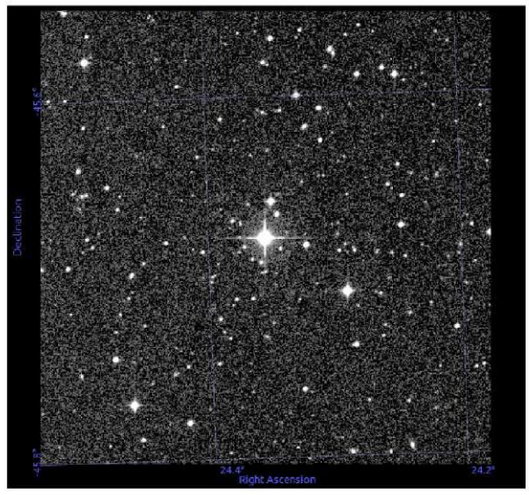


Figure 8: The star WASP-18 (the bright one in the center) with marked equatorial coordinates. Taken from (E15).

1.3. Exoplanet atmospheres

As there is a large diversity in types of exoplanets and their characteristics, the same is valid for their atmospheres as well because a formation and a composition of atmospheres depends on the place of formation, its evolution, and so on. To understand atmospheres of exoplanets there is an important pre-requisite—to understand the atmospheres of solar system planets as the physics is the same everywhere. To determine the composition and temperature of a planet atmosphere, we use a spectrum of the atmosphere.

Generally, it is not easy to detect an atmosphere of an exoplanet directly. The problem is an enormous star-planet contrast. For example—Jupiter-Sun contrast (in visible wavelength) is $1.4 \cdot 10^{-9}$ and for the Earth it is one order higher. Fortunately this problem is less evident in the case of planets orbiting close to their parent star (hot Jupiters and others). Due to relatively high temperature of these planets the star-planet contrast is about 10^{-3} which is more favourable than in the case of the solar system. We are able to detect this ratio of brightness and for this we use the method of transit (see chapter 2. for details) (Seager, 2010).

When we focus on the composition of planet atmospheres, we begin with atmospheres of the solar system planets because these are best described and thus we use them for comparison with detected atmospheres of exoplanets. The solar system includes two categories of planets. Four inner, terrestrial planets (Mercury, Venus, Earth, Mars) are composed of rock and metals and their atmospheres are thin and have changed from primordial state. On the other hand, four outer, giant planets (Jupiter, Saturn, Uranus, Neptune) are very different. Jupiter and Saturn are composed of gas ("gas giants"), mainly of hydrogen and helium. Uranus and Neptune are mostly composed of water, methane and ammonia ices ("ice giants") and their atmospheres contain hydrogen and helium. The atmospheres of giant planets are primitive and due to almost no evolution they are roughly the same as when they were formed. These findings are a good starting point when we observe and study atmospheres of exoplanets.

The first observation of an exoplanet atmosphere was published in 2003 when upper atmosphere around exoplanet HD 209458b was observed by the method of repeated transits through hydrogen absorption (Vidal-Madjar et al., 2003). It was more than ten years ago. Now we are able to detect water in atmosphere and even clouds or colour of sky through Rayleigh scattering (GJ 3470b; Dragomir et al. (2015)). Later at the planet HD 209458b sodium and water were also detected.

The purpose of this thesis is to develop an automated routine to reduct and analyse data of ESO VLT HAWK-I instrument (see chapter 3.3.). The analysis will be focused on exoplanetary atmospheres. The real data prepared for testing the pipeline contain a secondary eclipse of exoplanet WASP-18b. Finally, obtained results will be discussed as well as their implication in a context of atmospheric models.

1.3.1. The atmosphere of WASP-18b

The exoplanet WASP-18b is a massive and very hot planet and therefore it is a suitable candidate to study exoplanet atmospheres. The former studies of the exoplanet atmosphere also dealed with redistribution of thermal energy in the atmosphere and furthermore, Iro and Maxted (2013) studied that for a case of no rotation of the atmosphere and for fast rotating atmosphere. They observed in infrared Spitzer bandpasses. Their central wavelengths, however, differs from the wavelength used for taking images being processed in this thesis (3.6–8.0 μm compared to 2.09 μm).

From the viewpoint of the thesis the most important results of the two mentioned papers are models of spectrum of the WASP-18b atmosphere (dependency of a flux ratio on a wavelength). Figure 9 shows the spectrum and observations in the four Spitzer bandpasses (the black circles with error bars). From that model we can estimate the eclipse depth for images processed in this thesis which is supposed to be roughly about 0.2 %. In the case of Spitzer bandpasses the depth is between 0.30–0.41 % with respect to the bandpasses.

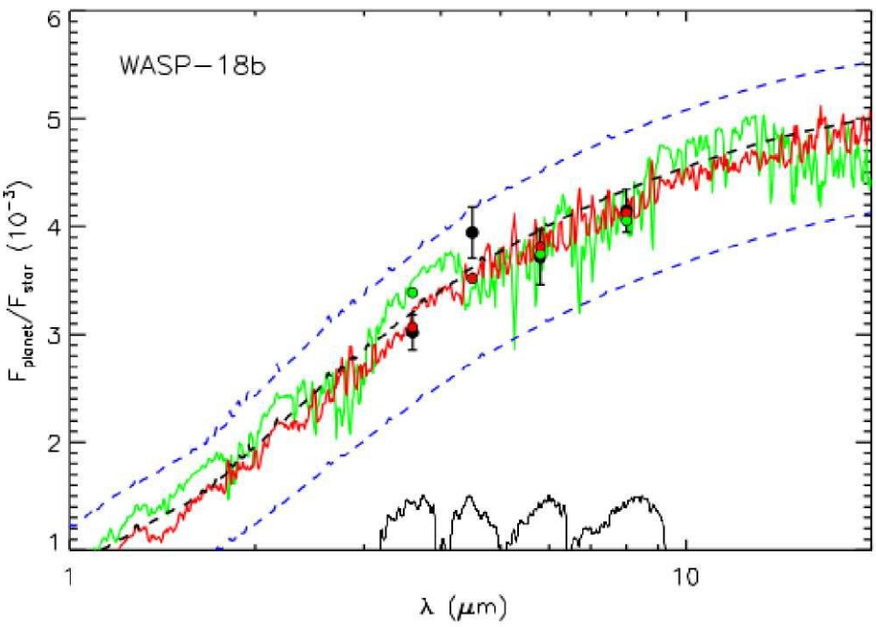


Figure 9: Dayside spectrum of WASP-18b made of observations of *Spitzer's* Infrared Array Camera. The black dashed line is a blackbody spectrum at 3200 K, the blue dashed lines are blackbody spectra which correspond to the minimum and maximum temperatures in the exoplanet atmosphere. There are also two model spectra on the figure—inversion model (the red curve) and non-inversion model spectrum (the green curve). Taken from Nymeyer et al. (2010).

2. Methods to detect and characterise exoplanets

The method of eclipses uses a state of two celestial objects when one obscures another and thus from the view of an observer one object is in front of the second. If sizes of the objects are different then we distinguish between two types of eclipses—transits and occultations.

The situation when smaller object passes in front of the larger one is called *transit* or *primary eclipse*. But if the smaller object is hidden behind the larger one, we call it *occultation* or *secondary eclipse* (or sometimes *secondary transit*).

When we observe a transit, a planet obscures a part of its star, so an amount of light from the star is blocked by the planet and the flux is lower than during no transit. When we observe an occultation, the total flux drops again but due to the fact that the planet is hidden beyond the star and thus it does not contribute to the observed flux [see Figure 10].

Following chapters give same basics of this method for observing exoplanets and describe used techniques which result in obtaining some physical characteristics of the observed exoplanets.

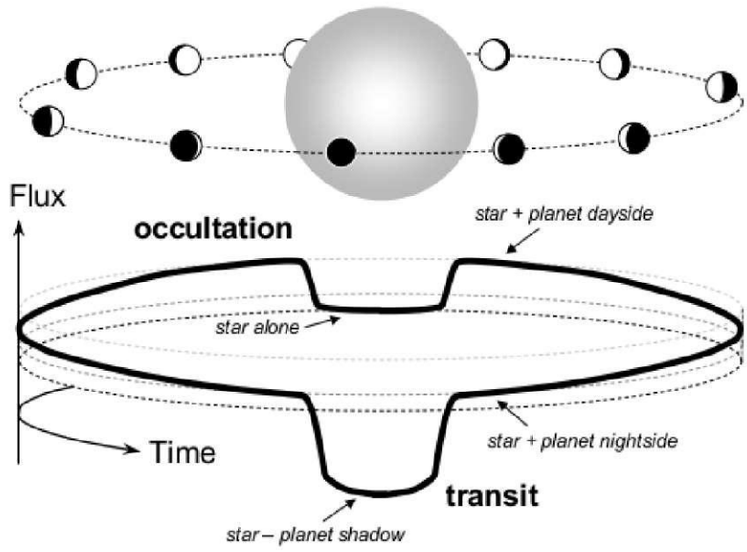


Figure 10: Illustration of transits and occultations. The influence of a planet in total observed flux is also shown. Taken from Winn (2010).

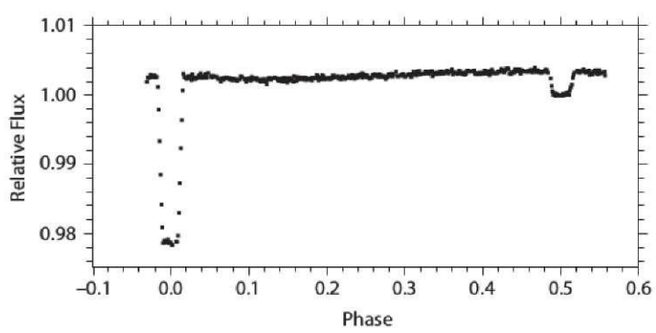


Figure 11: Light curve of HD 189733A and b observed in infrared band. Transit (left dip) and also occultation (right dip) are visible here. Taken from Knutson (2007).

2.1. Geometry of eclipses

The main requirement to observe an eclipse is that orbital plane of an exoplanet has to be oriented towards the Earth. This important fact can be written mathematically—an eclipse appears if

$$R_{\mathrm{p}} + R_{*} \geq a \cos i, \tag{1}$$

where R_{p} is the planet radius, R_{*} is the star radius, a is the semi-major axis and i is the inclination angle, which is an angle between the orbital plane of an exoplanet and a plane perpendicular to the observing plane. From this equation we can derive a minimal inclination angle when an eclipse is still detectable:

$$i_{\mathrm{min}} = \arccos\left(\frac{R_{\mathrm{p}} + R_{*}}{a}\right). \tag{2}$$

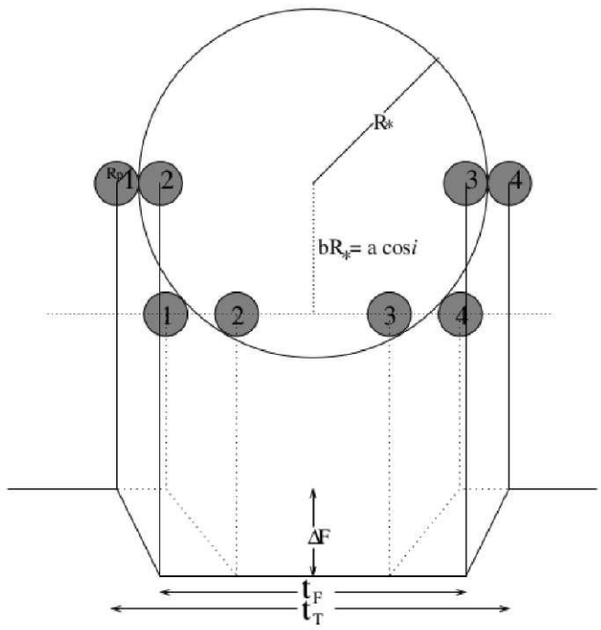


Figure 12: The principle of detecting transits (and occultations) with corresponding light curves and marked fundamental physical characteristics. Taken from Seager and Mallén-Ornelas (2003).

In figure 12 a transit is drawn as it is in progress. There are two different positions of the planet and two light curves matching the positions. The shape of a light curve is dependent on planet radius R_{p} and impact parameter b which corresponds to orbital inclination angle. The depth of a transit ΔF is given by the planet radius while its duration is given by the impact parameter. Duration of the whole transit is marked t_{T}—the time between the first and the last visual star-planet contact. Duration of the central part of a transit (between ingress and egress) is marked t_{F} which is the time when the whole planet disc is superimposed on the disc of its star. Figures 1–4 note star-planet visual contacts as the planet goes from left to right.

The figure 12 does not take into account limb darkening of a star which corresponds to temperatures of different parts of star. If we look at a star, the central part is brighter than limbs of its. This is given by the fact that central parts are hotter and physical

characteristics of stars are also projected to shapes of light curves obtained during transit. The shape of a light curve is then rounded, especially between contacts 1–2 and 3–4, in moments when a planet begins and then ends obscuring its host star.

From measured transit depth we can compute planet radius if the radius of the parent star is known. If F is the total observed flux, then the drop of the flux, which is equal to the transit depth, is

$$\delta_{\mathrm{tra}} \equiv \frac{F_{\text{no transit}} - F_{\text{transit}}}{F_{\text{no transit}}} = \left(\frac{R_{\mathrm{p}}}{R_*}\right)^2. \tag{3}$$

All the radiative flux coming to us from a system is a flux both from a planet and its host star which is time-dependent: $F(t)$. The flux is combined flux from the both bodies, so outside an eclipse we can simply write:

$$F(t) = F_*(t) + F_{\mathrm{p}}(t). \tag{4}$$

Here and further F_* is a flux from a star and F_{p} is a flux from a planet of the star. If an eclipse occurs, the flux drops and we have to subtract from the equation 4 a function describing a corresponding loss of flux according to the eclipse type.

2.1.1. Transmission spectroscopy

The method of transmission spectroscopy uses physical features of planet atmospheres. When we observe a transit, not all the light of all wavelengths from the star comes to us. If the planet has got non-negligible atmosphere, its upper part partially absorbs the starlight of some wavelengths. An absorption is dependent on wavelenght which is given by scattering properties of atoms and molecules. Some wavelengths of atoms and molecules are weaker and some stronger at transition. Where the transition is strong, on those wavelengths it is more probable that we can measure the transmission spectrum which tells us something about the composition of the upper part of atmosphere of a planet.

Calculation of the expected signal is rather complicated but we can determine the order of magnitude. For a strong transition of atoms and molecules we can calculate atmospheric scale heights:

$$H = \frac{\mathsf{k}_{\mathrm{B}} T}{\mu_{\mathrm{m}} g}, \tag{5}$$

where k_{B} is Boltzmann's constant, T is the temperature, μ_{m} is the mean molecular mass and g is the local gravitational acceleration. It gives us information about how the effective size of a planet grows. Knowing this, we can estimate how the transit depth (δ) increases due to the absorption of some wavelengths according to Winn (2010):

$$\Delta\delta \approx 2 N_H \delta \left(\frac{H}{R_{\mathrm{p}}}\right). \tag{6}$$

Here N_H is number of scale heights and from this equation we can see that an increase of the transit depth is better detectable for planets with high temperature and low gravitational acceleration, molecular mass and planet radius. For example for Sun-like stars $\Delta\delta$ is about 10^{-6} for an Earth-like planet and 10^{-4} for a "hot Jupiter" type of planet.

2.1.2. Emission photometry

When we observe an occultation of a planet, the depth of the occultation, measured from an obtained light curve, gives us a ratio of fluxes from a measured planet and its host star, so

$$\delta_{\mathrm{occ}} \propto \frac{F_{\mathrm{p}}}{F_*}. \tag{7}$$

During an occultation, the whole planet is hidden and flux coming to us declines. The depth of the occultation is dependent on intensity and radius of the planet and the star:

$$\delta_{\mathrm{occ}} = k^2 \, \frac{I_{\mathrm{p}}}{I_*}, \tag{8}$$

where $k = R_{\mathrm{p}}/R_*$ and I_{p} and I_* are averaged intensities of the disk of the planet and the star, respectively. From the equation 8 is seen how transit and occultation depth are connected, as $k^2 = (R_{\mathrm{p}}/R_*)^2$ (see equation 3). If we do an observation of a secondary eclipse of an exoplanet, a transit observation has been usually done formerly and hence k is known.

There are two sources of planetary radiation—reflected light from the host star and thermal radiation from the planet itself. The thermal radiation emerges at shorter wavelengths than the reflected starlight because the planet is colder than its host star. A wavelength-dependent occultation depth can also be expressed by using the Planck function,

$$B_\lambda(T) \equiv \frac{2\mathsf{hc}^2}{\lambda^5} \frac{1}{e^{(\mathsf{hc}/\lambda \mathsf{k_B} T)} - 1}, \tag{9}$$

where by approximating the both bodies as blackbody radiators

$$\delta_{\mathrm{occ}}(\lambda) = k^2 \frac{B_\lambda(T_{\mathrm{p}})}{B_\lambda(T_*)}. \tag{10}$$

In these equations h is the Planck's constant, c is the speed of light, λ is the wavelength and T is the temperature (T_{p} for the planet and T_* for its star). By integrating the equation 10 over the bandpass we obtain the observed δ_{occ}.

Disks of planets are not uniformly bright and depending on a type of planet (gaseous, rocky, ...) they can have zones which can differ by brightness. By getting a light curve of an occultation we can obtain from its ingress and egress part information how brightness of a planet is distributed.

3. Instruments

3.1. ESO

The European Southern Observatory (ESO) is an astronomical organization which is one of the most important organizations of astronomic research. It was established in 1962 and today it has already got 16 member countries (Austria, Belgium, Brazil, Czech Republic, Denmark, Finland, France, Germany, Holland, Italy, Poland, Portugal, Spain, Sweden, Switzerland and United Kingdom) and Chile as the host state. The headquarters of ESO is located in Garching (near Munich) in Germany. Currently ESO operates three observatories: Chajnantor, La Silla and Paranal. The name of each of the observatories comes from the name of the mountain where the observatory is found. All of them are situated in the Atacama desert in Chile because this area is very suitable for astronomical observations due to the high altitude, an excellent dry climate and a lot of clear nights during a year.

Chajnantor observatory lies at the altitude of 5000 m where Atacama Large Millimeter/submillimeter Array (ALMA) is situated. It consists of 66 giant antennas.

La Silla observatory lies at the altitude of 2400 m and its equipment consists of a few telescopes working in the optical band. Diameters of their mirrors are up to 3.6 m.

Paranal observatory lies at the altitude of 2600 m and it is situated on one of the driest regions in the world. There are four main telescopes which are called Very Large Telescope (VLT). Each of them has got its primary mirror with the diameter of 8.2 m. There are also four Auxiliary Telescopes with main mirrors of 1.8 m in diameter. In an individual use of a Unit Telescope images of celestial objects as faint as 30 mag can be obtained. VLT instrumentation includes large-field imagers, cameras and spectrographs with adaptive optics and so on. It covers a broad spectral region—from deep ultraviolet to mid-infrared wavelengths. All the telescopes can work separately or together to form a giant interferometer.

Currently ESO is constructing European Extremely Large optical/infrared Telescope (E-ELT) which will have a 39-metres primary mirror. It will be the largest telescope in the world for visible and near-infrared band and its launching is planned for 2024.

3.2. Near-Infrared Instruments

When we use infrared electromagnetic radiation (IR) in physics and especially in astrophysics, for our purposes we usually distinguish among a few bands of electromagnetic radiation in dependency on a wavelength. These are roughly following: Near-Infrared (NIR): 1–5 μm; Mid-Infrared (MIR): 5–25 μm; Far-Infrared (FIR): 25–1000 μm. As a visible light we consider wavelenghts between 0.3–1.0 μm. Commonly used Charge-Coupled Devices (CCD) work just on these wavelengths.

Although CCDs and IR detectors use the same physical principle, infrared detectors are very different from CCDs. Since IR radiation corresponds to less energy than a visible light, electrons in IR detector are excited to a conduction band with small energy about 1 eV. This process is very sensitive to temperature as it is generally known about infrared radiation. Every electron has got its energy which is proportional to $\mathbf{k}\boldsymbol{T}$ ($\mathbf{k}$ is the Boltzmann constant and $\boldsymbol{T}$ is the temperature of radiation) and thus electrons can be thermally excited into the conduction band. This, so-called "dark current", has to be decreased by cooling the detector to low temperature.

The most important difference between CCDs and IR detectors is from the view of their architecture. CCDs work as charge-transfer devices when photoelectrons are

collected and from a row to a row then they are read out by a transfer. But IR detectors do not work this way. They utilize hybrid arrays in which photodetection and readout technologies are separated. Each of the pixels utilizes an architecture (called "unit cell") where each pixel has got its own readout amplifier. Another difference between the two types of detectors is a shutter—in contrast to CCDs, there is no shutter at IR detectors, integration is defined electronically (E16).

3.3. High Acuity Wide-field K-band Imager—HAWK-I

HAWK-I, mounted on VLT, is a cryogenic (120 K and detectors at 80 K) near-infrared (0.85–2.5 μm) wide-field imager. It has got a full reflective design. The light passes four mirrors and two filter wheels before hitting a mosaic of four 2048×2048 pixel detectors. The field of view of HAWK-I on the sky is 7.5 arcmin×7.5 arcmin. The four chips are separated by a cross-shaped gap of $\sim$ 15 arcsec and numbered 1–4 anti-clockwise from the bottom left as the first one. The pixel scale is 0.1064 arcsec/pix with distortions < 0.3 % across the field of view. Dark current of the instrument is around 2 electrons/pix/s and the read-out noise is approximately 5 electrons (E12).

Figure 13: High Acuity Wide-field K-band Imager—HAWK-I. Taken from (E12).

Figure 14: The infrared detector of HAWK-I. Taken from (E12).

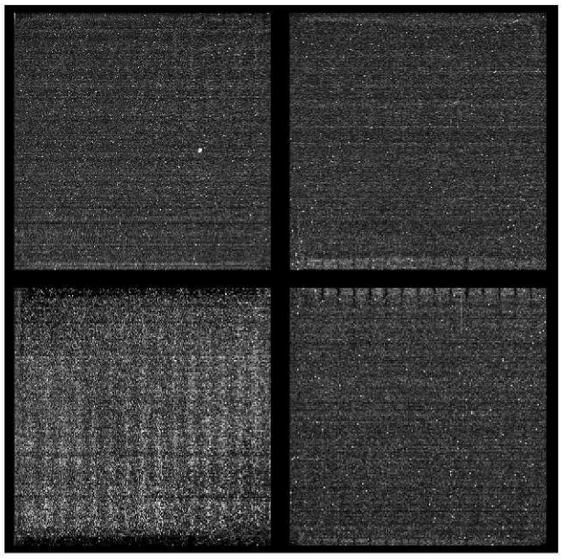


Figure 15: Raw HAWK-I dark frame, filter NB1060 (IR band), integration time 2 s.

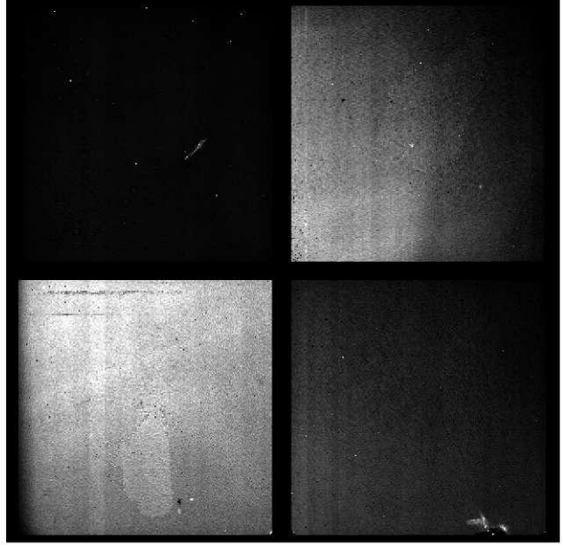


Figure 16: Raw HAWK-I twilight flat field, filter NB1190 (IR band), integration time 2 s.

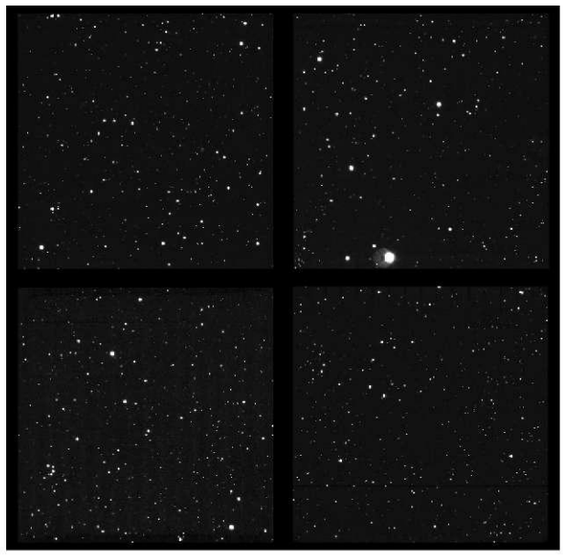


Figure 17: Raw HAWK-I science frame, filter NB1190 (IR band), integration time 2 s.

3.4. Fast Photometry Mode

If a high-cadence observation is required (very useful for objects with fast variability), HAWK-I offers *Fast Photometry Mode* (FPM) or *Windowing*, in HAWK-I user manual called *Fast Jitter Mode*. The main advantage of the mode is that a huge number of single images during one night can be obtained due to its high-cadence. For extrasolar planets observations this mode is very useful because a host star of an exoplanet can quickly change its brightness so we can see those changes in detail.

Each of the detectors of HAWK-I has got a resolution of 2048×2048 px and is read in 16 vertical stripes. It means that the resolution of every stripe is 128×2048 px. Each stripe defines one window, however the locations of the windows are not independent. Since HAWK-I is made of four detector arrays, total number of windows for each frame is 64.

For the fastest FPM observations required only pixels defined by a user are read (hardware windowing). The pixels make windows which are described by four parameters: DET.WIN.STARTX, DET.WIN.STARTY, DET.WIN.NX and DET.WIN.NY. A given value of the first two parameters defines a starting pixel of a window (for an individual stripe of the window) where the value in x-axis increases from left to right and in y-axis increases towards the gap between chips (see figure 18). The second two parameters assign in pixels a size of each stripe for x and y-axis.

The result of detector windowing is a fits file containing cubes. These are formed by slices which are made from the tiled-together images of all windows, so they are spliced together without the gaps. The cubes are actually 3-dimensional images where a value in z axis defines a number of single images of the entire cube. Before a processing, then, a cube must be sliced into particular images (2D).

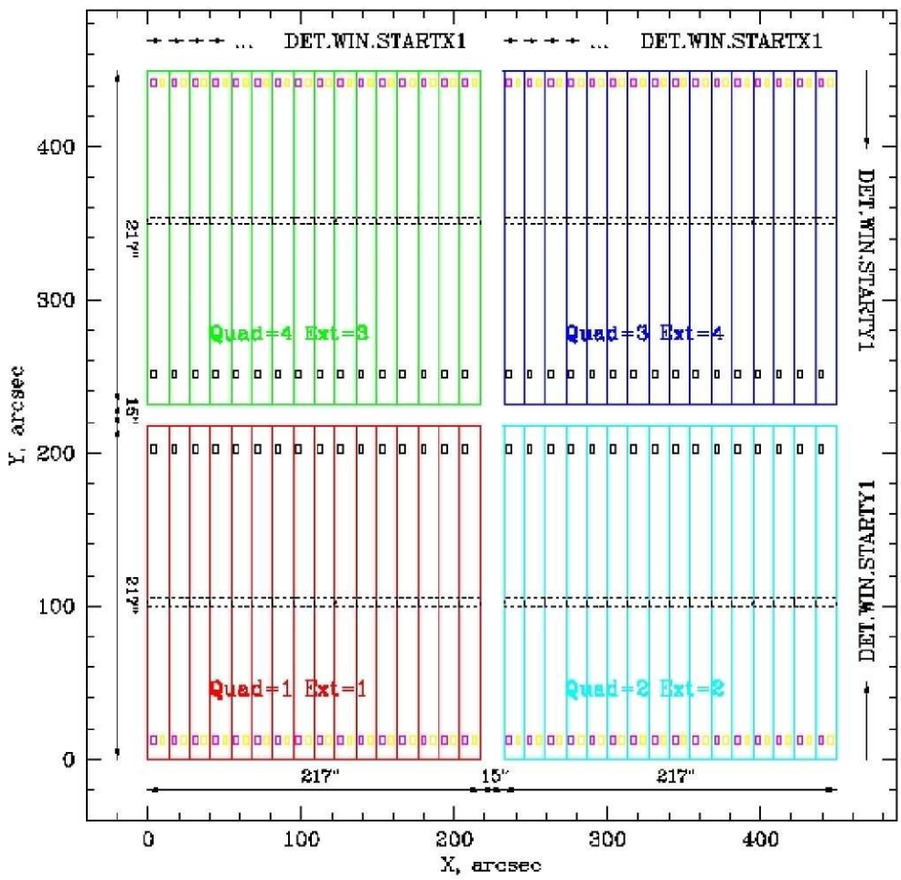


Figure 18: The principle of Fast Photometry Mode of HAWK-I. Each of chips is drawn with a different colour as well as the vertical stripes of them. Four sets are represented here as examples. These are shown by yellow and purple colour and by two types of black lines. Taken from (E12).

4. The data set

The data set, processed in this master thesis, was obtained during the observational night 1.–2. 8. 2013 (ESO proposal *091.C-0488(A), PI: Caceres*). Science images were taken on 02 August 2013 between 0.12 and 6.38 local time. The instrument used for taking the data set has been HAWK-I (see chapter 3.). The file format of all the processed images is Flexible Image Transport System—FITS. Science images were taken in a narrow band infrared filter at the wavelength 2.090 μm (in FITS header marked as "NB2090") with four different integration times by using *Fast Photometry Mode*. Prediction of the occultation of WASP-18b had been from 3.17 to 5.34 local time which corresponds to 7.17–9.34 UTC.

Differences in properties of the four parts of the data set are excerpted in table 2.

τ_{int} [s]	N_{CFF}	N_{FF}	N_{ax1} [px]	N_{ax2} [px]	t_{first} (UTC)	t_{last} (UTC)
2.0000	2	28	4096	2260	04.12:09	04.13:24
1.8432	186	2009	4096	2260	04.17:44	05.41:42
0.5000	98	4053	4096	460	05.52:03	06.32:50
0.3966	655	32,989	4096	460	06.33:23	10.38:04
All	**941**	**39,079**			**t_{total} = 6 h 25 min 55 s**	

Table 2: Properties of each part of the science images data set, separated according to exposure times. τ_{int} is integration time; N_{CFF} is a number of cube fits files; N_{FF} is a number of particular images; N_{ax1} is a length of an image at x-axis; N_{ax2} is a length of an image at y-axis; t_{first} is time of the first image; t_{last} is time of the last image; t_{total} is total time duration covered by the data set.

As seen in the table above, images taken with exposure times 0.5 s and 0.3966 s have got different length of y-axis. As positions of the measured stars during the observation had changed [figures 19 and 20], it was possible and favourable to change this setting and thus take advantage of *Fast Photometry Mode*. The size of every image (x length $\times$ y length) became smaller and this increased the cadence of taking images.

All the single images contain three stars which are situated on the right side of an image. The first star (from left) is the target star, WASP-18, and two remaining fainter stars are comparison stars (comp. 1 and comp. 2).

After the subsection describing the header of images, the following subsections of this chapter describe observing conditions and how they were changing during taking science images. The next subsection shows changes of position of stars on chips for two successive data sets. The last subsection gives details about reduction frames used for data reduction.

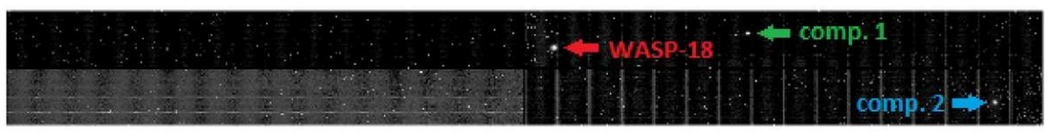


Figure 19: Smaller size of images of the data set—exposure times 0.5 s and 0.3966 s. The three measured stars occupy two chips (2 and 3).

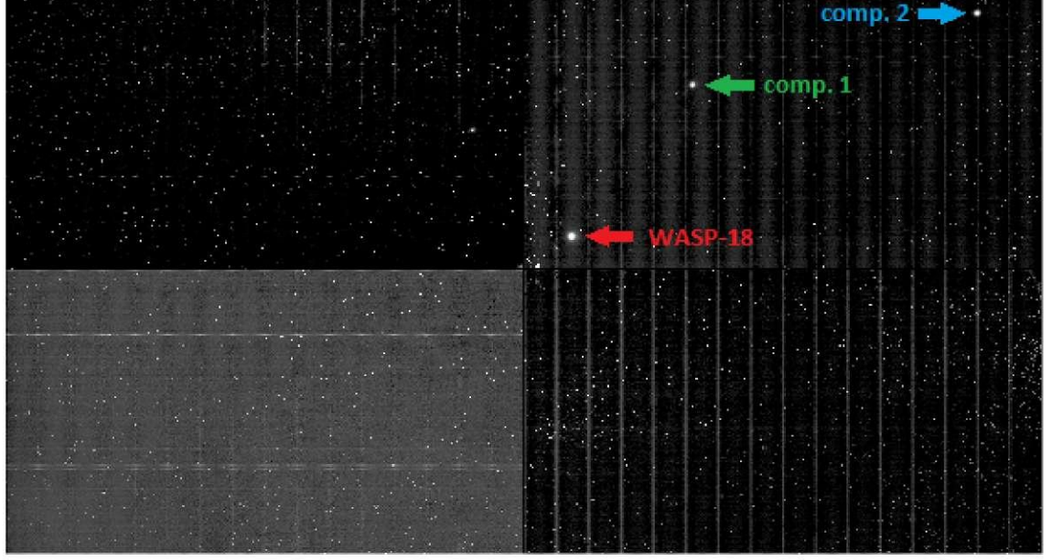


Figure 20: Larger size of images of the data set—exposure times 2 s and 1.8432 s. In contrast to the previous image, here the measured stars occupy only the one chip (3).

4.1. The header

The header of files of HAWKI-I contains a lot of useful information and then we can get from it values of many important characteristics which influence the quality of data. However, the header has also got a disadvantage. All the information are valid only in time of the exposure beginning of the first image in a cube fits file unless different time is specified there explicitly (i. e. "Airmass at **end**"). Due to this, time of exposure of each single image had to be calculated by using a script. There are about 500 rows in every header which equals to the same number of single information about the given image. For working with a header extension "0" is typed in commands while extensions "1"–"4" correspond to individual chips of the instrument.

The header is very important for a developed pipeline because all information about images processed by the pipeline is taken from it.

```
SIMPLE  =                    T / Standard FITS
BITPIX  =                  -32 / # of bits per pix value
NAXIS   =                    3 / # of axes in data array
NAXIS1  =                 4096 / # of pixels in axis1
NAXIS2  =                  460 / # of pixels in axis2
NAXIS3  =                   51 / # of pixels in axis3
EXTEND  =                    T / Extension may be present
ORIGIN  = 'ESO-Paranal'         / European Southern Observatory
DATE    = '2013-08-02T09:59:31.3320' / Date the file was written
TELESCOP= 'ESO-VLT-U4'          / ESO Telescope Name
INSTRUME= 'HAWKI   '            / Instrument used.
OBJECT  = 'No name '          / Original target.
RA      =            24.363839          / 01:37:27.3 RA (J2000) pointing
DEC     =           -45.68250           / -45:40:56.9 DEC (J2000) pointing
EQUINOX =              2000.0           / Standard FK5
RADECSYS= 'FK5     '            / Coordinate reference frame
EXPTIME =            0.3966000 / Integration time
MJD-OBS =      56506.41609641 / Obs start 2013-08-02T09:59:10.729
DATE-OBS= '2013-08-02T09:59:10.7295' / Observing date
UTC     =            35949.000          / 09:59:09.000 UTC at start
LST     =             7331.324          / 02:02:11.324 LST at start
PI-COI  = 'UNKNOWN '          / PI-COI name.
OBSERVER= 'UNKNOWN '          / Name of observer.
CTYPE1  = 'RA---TAN'                    / Pixel coordinate system
CTYPE2  = 'DEC--TAN'                    / Pixel coordinate system
CRPIX1  =           -37.500000          / Ref. pixel in <axis direction>
CRPIX2  =          2161.500000          / Ref. pixel in <axis direction>
CRVAL1  =            24.363839          / Coordinate value of ref. pixel
CRVAL2  =           -45.682505          / Coordinate value of ref. pixel
CD1_1   =        -2.836396E-05          / Translation matrix element.
CD2_1   =        -8.337211E-06          / Translation matrix element.
CD1_2   =        -8.337211E-06          / Translation matrix element.
CD2_2   =         2.836396E-05          / Translation matrix element.
ORIGFILE= 'HAWKI_IMG_OBS_FAST214_0838.fits' / Original File Name
HIERARCH ESO ADA ABSROT END  =   -192.02603 / Abs rot angle at exp end
HIERARCH ESO ADA ABSROT PPOS = 'POS      '  / sign of probe position
HIERARCH ESO ADA ABSROT START=   -191.78649 / Abs rot angle at exp start
HIERARCH ESO ADA GUID DEC    =    -45.57073 / -45:34:14.6 Guide star DEC J2000
HIERARCH ESO ADA GUID RA     =    24.564953 / 01:38:15.5 Guide star RA J2000
HIERARCH ESO ADA GUID STATUS = 'ON       '  / Status of autoguider
HIERARCH ESO ADA POSANG      =    -16.28800 / Position angle at start
HIERARCH ESO DET CHIP GAIN   = 1.00 / Gain in e-/ADU
HIERARCH ESO DET CHIP ID     = 'ESO-Hawaii2RG-chip66' / Detector ID
HIERARCH ESO DET CHIP NAME   = 'HAWKI-MOSAIC' / Detector name
HIERARCH ESO DET CHIP NX     = 4096 / Pixels in X
HIERARCH ESO DET CHIP NY     = 4096 / Pixels in Y
```

Figure 21: A header of a cube fits file made by HAWKI-I which corresponds to the bottom image of the figure 19.

However, as the disadvantage mentioned above, observing time of each of the images had to be calculated.

4.2. Airmass during observations

Airmass (X) is one of important effects on a result of processing data. For plane-parallel atmosphere it is defined as $X = \sec z = \frac{1}{\cos z}$, then airmass is only function of zenith altitude z which is related to altitude h as $z = 90° - h$. The data set begins at airmass 2.5 which corresponds to the altitude about 23.6° and in the highest point the altitude is about 65.4° which corresponds to the least airmass equal to 1.1.

The graph in figure 22 show how airmass has been changing during observation (02 August 2013). Each point describes one cube fits file. Values of airmass are taken from header of cube fits files and for every file there are two values—airmass at start and airmass at the end. All these are plotted in the graph in figure 22.

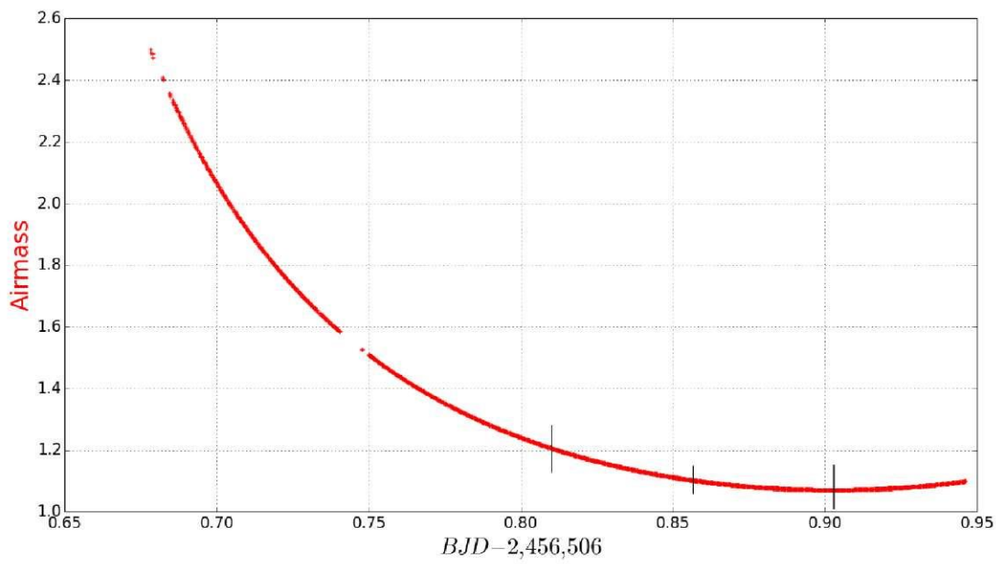
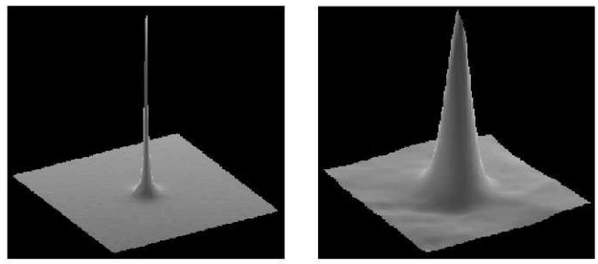


Figure 22: Changes of airmass during the observation of WASP-18. For every cube fits file there are two points close to each other due to a very similar time when they were taken. The vertical black lines show time points of the beginning, the centre and the end of the occultation, respectively. Time points start at 4.12:09 and end at 10.37:44 UTC.

4.3. Seeing during observations

The other important factor affecting the quality of data is seeing. Unlike airmass its value changes coincidentally. This is caused by turbulent motions of particles of the air, so astronomical seeing describes how much images of sky objects are perturbed by atmosphere of the Earth. Hence, the better seeing, the better quality of an image. The seeing is measured in arcsec and it directly relates to Point Spread Function (PSF)—a function describing how a point source is imaged by an optical system. For greater values of seeing PSF is less sharp and begins to flatten [figure 23].

In figure 24 we can see how raw seeing (measured in optical band) and seeing corrected by airmass have been changing during observation (02 August 2013). Values of seeing are



(a) Seeing 0.1". (b) Seeing 0.7".

Figure 23: An example how seeing affects Point Spread Function. Taken from Bouy et al. (2007).

taken from header of cube fits files and for every file there are two values of raw seeing—observatory seeing at start and at the end. The time-seeing dependency is plotted in the graph in figure 24.

As seen, seeing had been excellent when an occultation of WASP-18b had begun and it lasted until its half. But from that moment it got worse and its value became double of the origin value. There are also points with value equal to -1 at the very end of the data set which can not be indeed a value of seeing. This value means that seeing has not been determined. It might be caused by the sky being too bright (after sunrise).

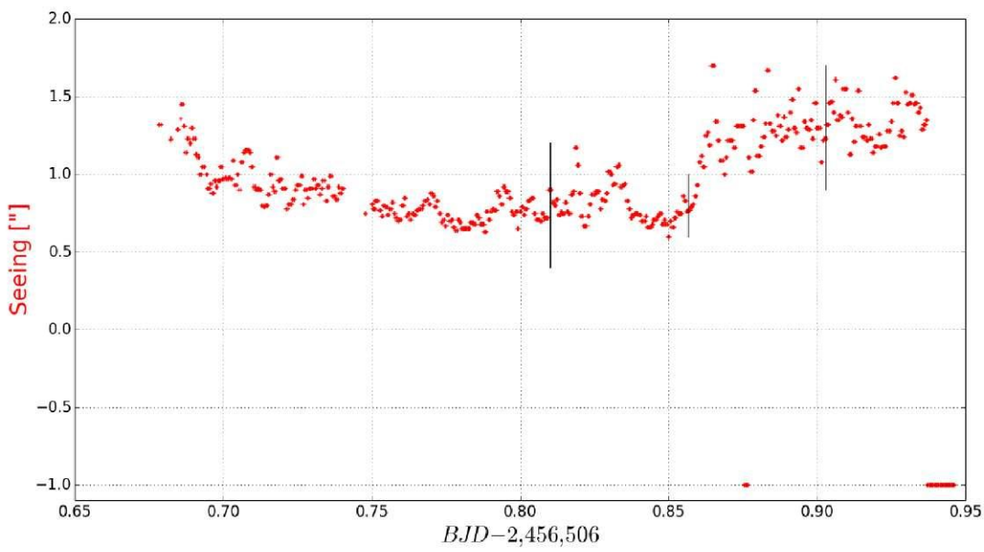
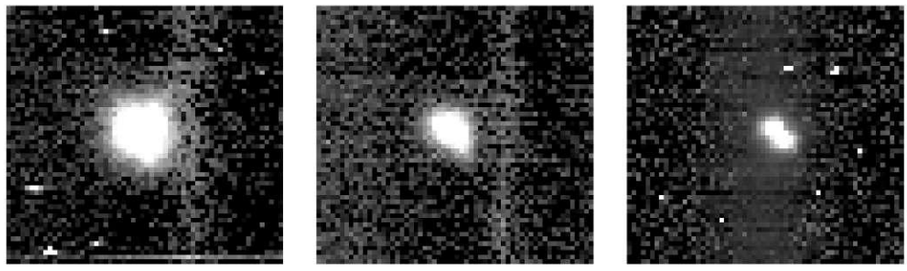


Figure 24: Changes of seeing during the observation of WASP-18 (marked by red). For every cube fits file there are two points close to each other due to a very similar time when they were taken. Changes of delivered seeing corrected by airmass are marked by green. The vertical black lines show time points of the beginning, the centre and the end of the occultation, respectively. Time points start at 4.12:09 and end at 10.37:44 UTC. The mean seeing for Paranal is about 1.0" (E12).

From the second part of the eclipse (roughly 0.85 on x axis) seeing had got worse and this fact had an impact on the images. Stars had become unfocussed [figure 25] and results of photometry were then negatively affected by that.



(a) WASP-18. (b) Comparison star 1. (c) Comparison star 2.

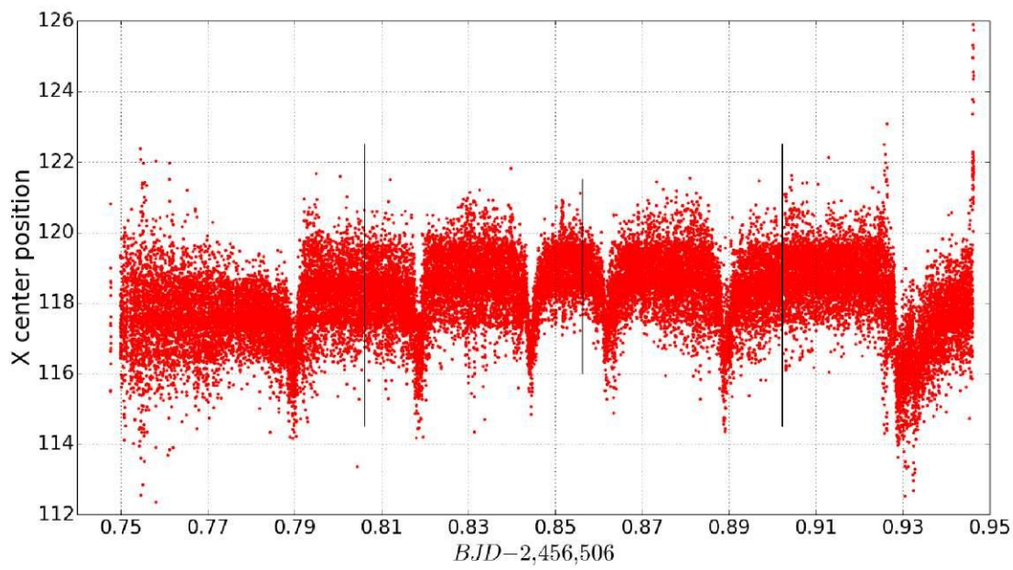
Figure 25: Low quality of seeing had a negative impact on the quality of data.

4.4. Positions of stars on a chip

The pointing of VLT is accurate, however, it is always necessary to monitor eventual movements of stars on the chip due to potential guiding errors. It can be critical if a star is positioned on different pixels on different images.

Positions of measured stars on particular chips, taken from photometry, can also affect the result of a photometric measurement. It can be more apparent if the position is not the same during measurement time. This is a case of the processed data set.

Figures 26 and 27 show changes of centre position of WASP-18b and comparison star 1 both in x-axis and in y-axis during the observation. The graphs correspond to data sets with exposure time 0.5000 and 0.3966 s because positions of the stars on images of these data sets are the same in contrast to the remaining images.



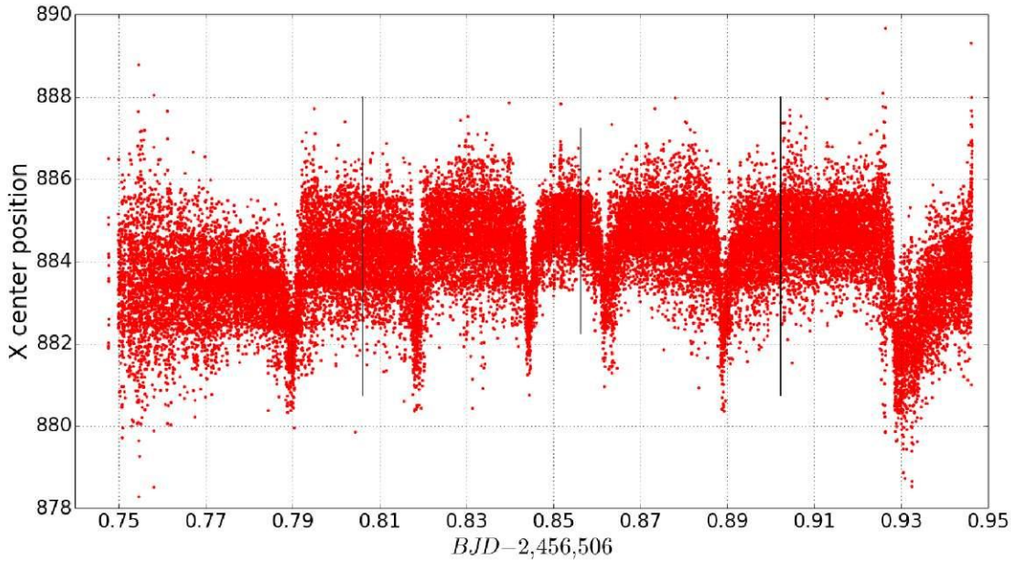
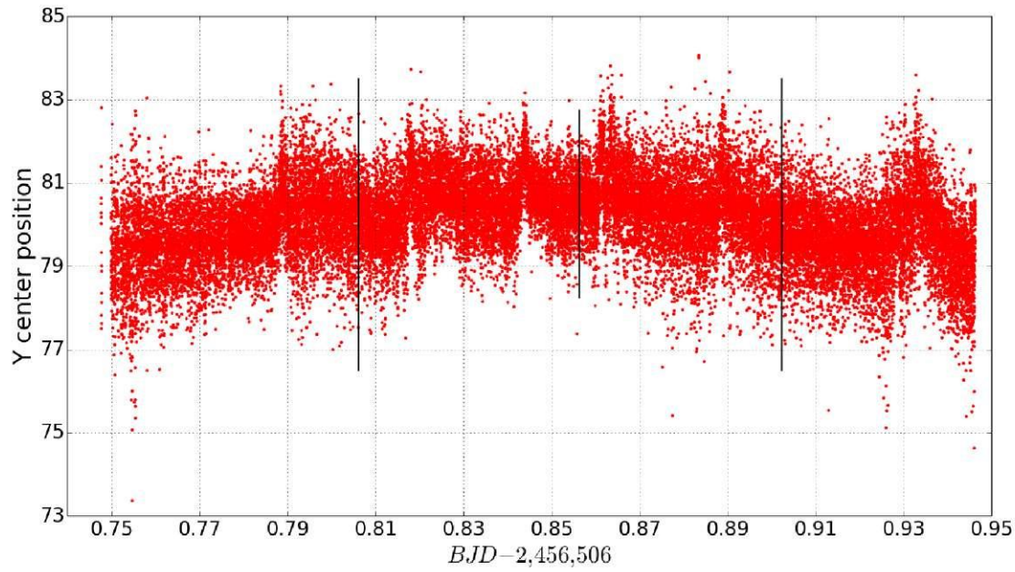


Figure 26: Positions of stars on a chip at x-axis on images with exposure time 0.5000 and 0.3966 s. The bottom graph on the previous page shows dependency on time for WASP-18b, the graph upper for comparison star 1.

From the four graphs there is obvious a small variance in the positions of the stars but at some intervals there are "jumps" from the mean value of the position which are quickly aligned after that. These intervals are not the same, their length changes between 26 and 57 minutes in x-axis and between 25 and 64 minutes in y-axis on a chip. Shifts in the positions of the stars are about 10 pixels in the both axes while remaining on the same chip. Due to good flat fielding this fact seems not to affect the resulting light curve.



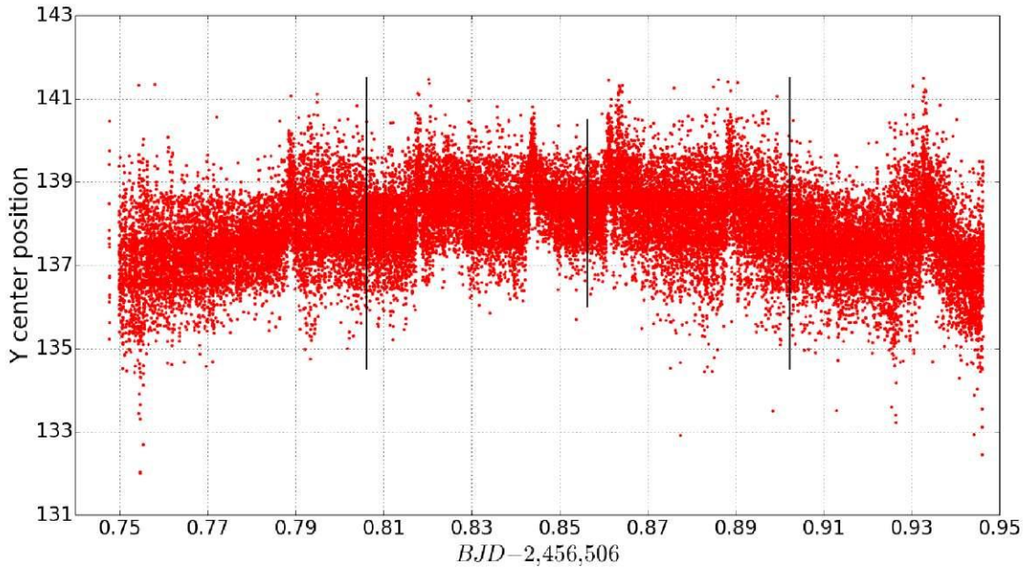


Figure 27: Positions of stars on a chip at y-axis on images with exposure time 0.5000 and 0.3966 s. The bottom graph on the previous page shows dependency on time for WASP-18b, the graph upper for comparison star 1.

4.5. Reduction frames

This section describes properties of dark frames and flat-field frames used for data reduction. A sample of each type of a reduction frame is also shown.

4.5.1. Dark frames

Dark frames were taken with the same exposure times as science frames. However, the used filter is not the same, dark frames were taken in the narrow band infrared filter at the wavelength 0.984 μm (in header "NB0984"). All the images were taken on 02 August 2013 between 7.59 and 8.06 local time in *Fast Photometry Mode*. More information about darks frames is found in chapter 5.2.

τ_{int} [s]	N_{CFF}	N_{FF}	N_{ax1} [px]	N_{ax2} [px]	t_{first} (UTC)	t_{last} (UTC)
0.3966	3	152	4096	460	11.59:48	12.00:33
0.5000	3	153	4096	460	12.00:55	12.01:51
1.8432	3	33	4096	2260	12.02:28	12.03:27
2.0000	3	42	4096	2260	12.03:56	12.05:34
All	**12**	**380**				

Table 3: Properties of each part of the dark frames data set, separated according to exposure times. τ_{int} is integration time; N_{CFF} is a number of cube fits files; N_{FF} is a number of particular images; N_{ax1} is a length of an image at x-axis; N_{ax2} is a length of an image at y-axis; t_{first} is time of the first image; t_{last} is time of the last image; t_{total} is total time duration covered by the data set.

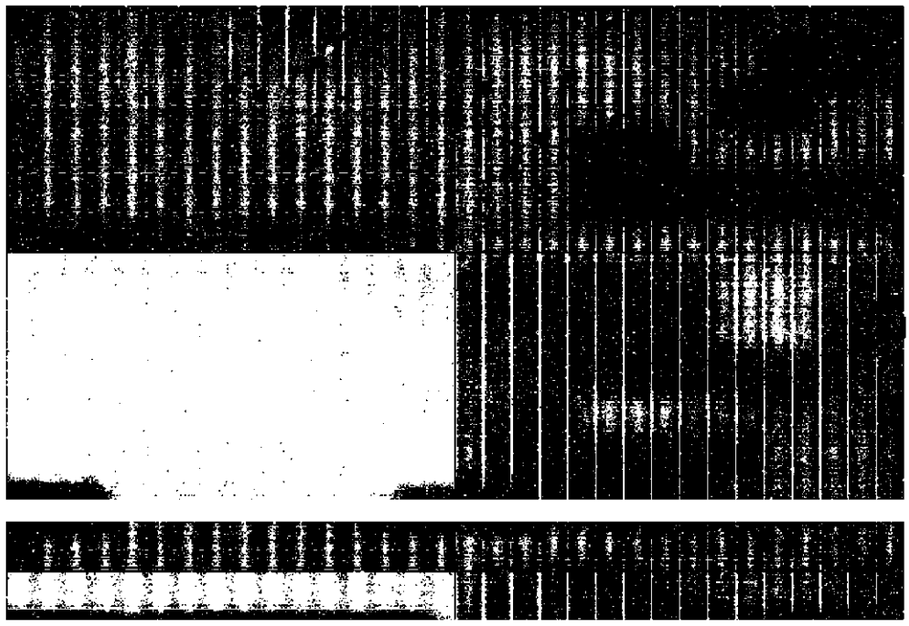


Figure 28: Two sizes of dark frames—top: exposure times 2 s and 1.8432 s; bottom: exposure times 0.5 s and 0.3966 s.

4.5.2. Flat-field frames

Flat-field frames data set consists of images with the only exposure time. All of them were taken in the narrow band infrared filter at the wavelength 2.090 μm (in header "NB2090"), the same as science frames. They were taken on 02 August 2013 between 17.50 and 18.16 local time. HAWK-I relies on twilight flat fields and unlike science and dark frames, flat-field frames are taken in a normal mode. More information about flat-field frames is found in chapter 5.3.

τ_{int} [s]	N_{fr}	$N_{\text{ax1}\times\text{ax2}}$ [px]	t_{first} (UTC)	t_{last} (UTC)
6.0	100	4 × (2048×2048)	21.50:42	22.15:29

Table 4: Properties of flat fields data set. Here τ_{int} is integration time; N_{fr} is a number of flat-field frames; $N_{\text{ax1}\times\text{ax2}}$ is a length at x and y-axis of an image; t_{first} is time of the first image; t_{last} is time of the last image.

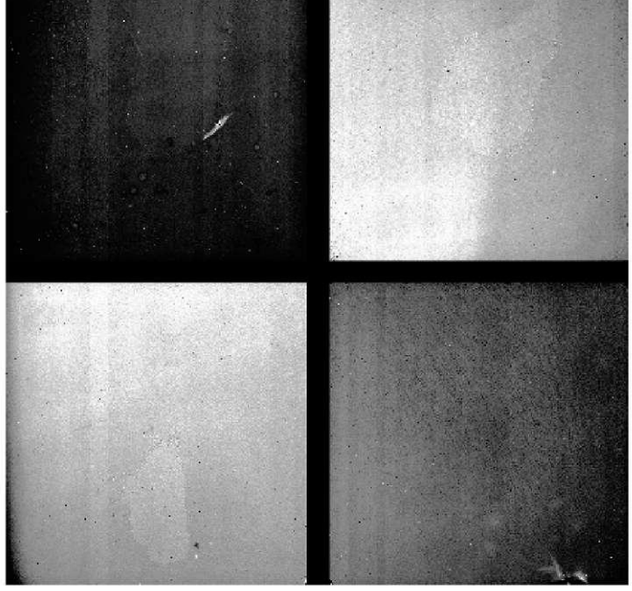


Figure 29: One of used HAWK-I flat fields. It had had to be cut into the same size as science (and dark) images before the data processing.

5. Data reduction

The process of data reduction is necessary to get the best results from data that we have got. The purpose of the reduction is to remove, subtract, and correct the "instrument signature" introduced to a raw image by a detector. The data reduction is an instrument specific, however, NIR imaging follows basic rules like dark frames subtraction and flat fielding therefore a reduction process is required. To make this process easier and faster, a pipeline was developed. It will be described in following sections.

This chapter begins with introductory information about the developed pipeline. The next subsections give details about reduction frames including statistics. Then results of the data reduction are shown and discussed and the end of the chapter is devoted to a bad pixel mask. Furthermore, structures and parts of Python scripts appointed for the data reduction are always appended.

5.1. Developed pipeline

The developed pipeline for HAWK-I *Fast Photometry Mode* consists of twelve scripts written in Python programming language and one configuration text file. Its goal is automatic processing of data and the making of a light curve with basic parameters of photometric quantities. At the end the resulting light curve in a form of a text file is prepared for an analysis by using MCMC code.

The pipeline is started by an introductory script and after executing all commands a next script is launched at its end. This repeats throughout the whole pipeline. Some scripts stop at a given point and require co-operation with the user.

The structure of the pipeline scripts is following:

- `pipeline.py`—the launching script, the option for using dark frames (▲) or not (▼):
  - ▲ `darks.py`—dark frames processing, the making of master dark frames;
  - ▲ `darksStat.py` (optional) doing dark frames statistics;
  - ▲ `flatsD.py` flat-field frames processing, the making of master flat-field frames;
  - ▲ `flatsStat.py` (optional) doing flat-field frames statistics;
  - ▲ `reductFlatsDarks.py` data reduction, the making of resulting images, preparing files for photometry.

  - ▼ `flats.py`—flat-field frames processing, the making of master flat-field frames;
  - ▼ `flatsStat.py` (optional)—doing flat-field frames statistics;
  - ▼ `reductFlats.py`—data reduction, the making of resulting images, preparing files for photometry.
- `photometry.py` doing photometry;
- `aperts.py` choosing the best aperture for a light curve;
- `lightCurve.py` the making of the light curve, preparing a file for fitting routine;
- `std.py` determining accuracy of measurements.

5.1.1. The script `pipeline.py`

A structure of the script:

OPTION WITH DARKS OR WITHOUT
CLEANING THE WORKING DIRECTORY

GETTING INFORMATION FROM HEADER AND MAKING TEXT FILE—3D—ALL FILES
MAKING OF DIRECTORIES
RENAMING OF FILES
GETTING INFO FROM HEADER AND MAKING TXT—3D—ALL FILES—NEW NAMES
Function—MOVING DARK FRAMES IF NOT USED
Function—WITH DARKS OR WITHOUT

A part of the script:

```
#### LOADING LIBRARIES ####
from pyraf import iraf
from numpy import *
import os
from matplotlib import pylab, mlab, pyplot
############################

########################## PARAMETERS ###############################
params    = genfromtxt("config.txt", dtype = "string")
main_dir  = params.take(where(params[:,0] == "main_dir")[0],0)[0,2]
flat_dir  = params.take(where(params[:,0] == "flat_dir")[0],0)[0,2]
dark_dir  = params.take(where(params[:,0] == "dark_dir")[0],0)[0,2]
name      = params.take(where(params[:,0] == "name")[0],0)[0,2]
txt       = params.take(where(params[:,0] == "txt")[0],0)[0,2]

######################## OPTION--WITH DARKS OR WITHOUT ######################
YorN = raw_input("\nDo you want to use dark frames for data reduction ('y' or 'n')? ")
```

5.2. Dark frames

Dark frames (and flat frames as well) are used to correct instrument effects influencing data quality. However, some instruments do not require using dark frames because their thermal noise or read-out noise is very low. This is just the case of HAWK-I (see details in section 3.3.).

Despite this fact, it is recommended to revise the frames, so I worked with them and did some statistics. As HAWK-I consists of four chips, using dark frames allowed me to see sensitivity differences among them.

To make a master dark frame I have used `imcombine` command and a combining method *median*. Since not all processed images are the same size, a number of master dark frames is corresponding to a number of sets of science images differentiated by sizes.

5.2.1. Dark current measurement

To measure a dark current of each of chips I have used all four sets of dark frames (integration times 0.3966, 0.5, 1.8432 and 2 s). By using `imstat` command of IRAF (Image Reduction and Analysis Facility (E17)) I had determined mean flux of each chip of all dark frames. These data then were plotted into a graph to show dependency mean flux on index.

As it is seen on the graphs below, there is linear dependency mean flux (dark current) on an integration time. Another fact is obvious from the graphs—sensitivity of all chips is different. For the same integration time the dark current has not got the same value for each of the chips. The chips 1 and 2 exhibit the highest dark current, respectively.

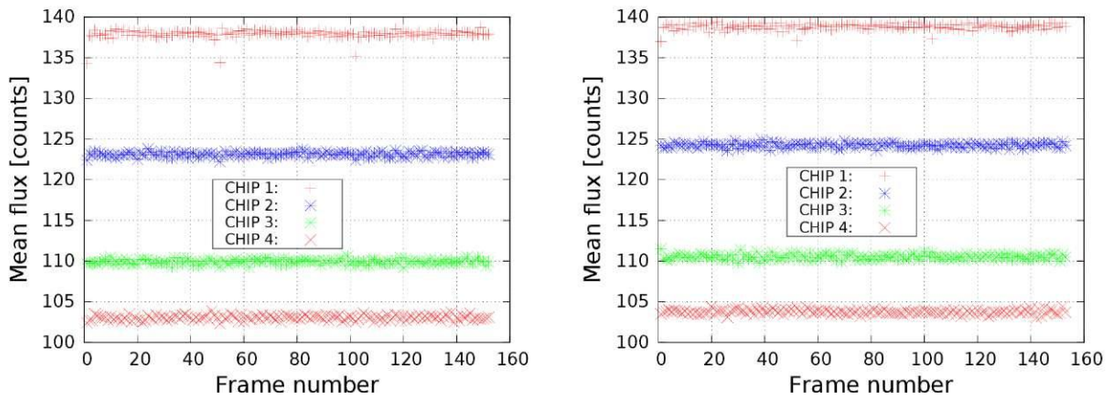


Figure 30: Dependency mean flux on index—dark frames. Size of images: (2048×230) pix, integration times: left—0.3966 s, right—0.5 s. Each colour of a point in the graphs indicates one of the four chips of the detector.

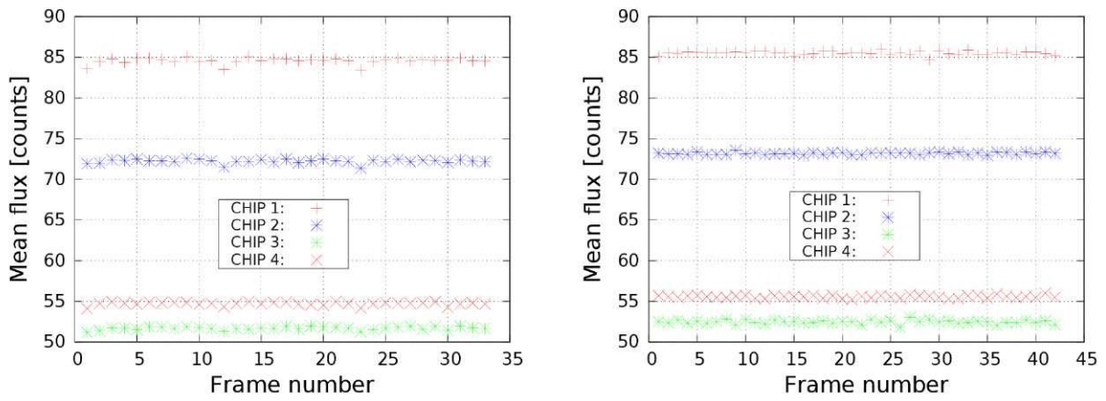


Figure 31: Dependency mean flux on index—dark frames. Size of images: (2048×1130) pix, integration times: left—1.8432 s, bottom right—2.0 s. Each colour of a point in the graphs indicates one of the four chips of the detector.

With longer integration time (lower two images) mean flux should be greater than with shorter time but it is not so here. This is because the lower images have got larger size and thereby the mean pixel value is divided by greater number (2,314,240 compared to 471,040 of upper two images). In terms of images of the same size, the mean flux raises with longer integration time as expected.

5.2.2. The script `darks.py`

A structure of the script:

OPTION OF DARK FRAMES STATISTICS
MAKING OF DARKS LIST FILE (INFO FROM HEADER)
SLICING DARK FRAMES TO 2D
GETTING INFO FROM HEADER + MAKING OF TXT—2D
CUTTING OF DARK FRAMES ACCORDING TO CHIPS
GETTING INFO FROM HEADER + MAKING TXT—2D—PARTICULAR CHIPS
LAUNCHING DARK STATISTICS SCRIPT IF REQUIRED

MAKING OF LIST-FILES ACCORDING TO SIZES (X,Y)
MAKING OF MASTERDARKS
COPYING, MOVING, REMOVING FILES

A part of the script:

```
######################## SLICING OF DARK FRAMES TO 2D ##########################
print "\nIMSLICING OF DARK FRAMES.\n"
iraf.imslice("@darks3D.txt", "2D_dark_//@darks3D.txt", "3", verbose="no")
iraf.files("2D_dark*", Stdout="darks2D.txt")

################### GETTING INFO FROM HEADER + MAKING TXT--2D ###################
os.popen("dfits %s |fitsort %s %s %s %s 'HIERARCH ESO DET WIN STARTX1' \
'HIERARCH ESO DET WIN STARTY1' %s %s %s > headerDf2D.txt" \
% ("2D_dark*","OBJECT","NAXIS","NAXIS1","NAXIS2","EXPTIME","MJD-OBS","DATE-OBS"))

################## CUTTING OF DARK FRAMES ACCORDING TO CHIPS ###################
print "CUTTING OF DARK FRAMES.\n"
with open("headerDf2D.txt","r") as darksheader:
    FILE, X, Y = ([] for _ in range(3))
    for line in darksheader:
        column = line.split()
        FILE.append(column[0])
        X.append(column[3]) # = NAXIS1 in header
        Y.append(column[4]) # = NAXIS2 in header

for chip in range(1,5):
    os.popen("mkdir %s/chip%i" % (dark_dir, chip))
    for j in range(1,len(X)):
        Name = FILE[j]
        if chip == 1:
            xy = "["+str(1)+":"+str(int(X[j])/2)+","+str(1)+":"+str(int(Y[j])/2)+"]"
            image = Name + xy
            image_out = Name + "_ch" + str(chip)
            iraf.imcopy("%s" % image, "%s" % image_out)
            iraf.hedit("%s" % image_out, "chip_num", "%i" % chip, add="yes", verify="no")
```

5.2.3. The script darksStat.py

A structure of the script:

Function—CREATION OF A GRAPH
MAKING OF IMAGE STATISTICS
ADDING NEEDED INFO TO IMSTAT TEXT FILES
MAKING OF STATISTICS LISTS ACCORDING TO CHIPS

A part of the script:

```
######################### MAKING OF IMAGE STATISTICS ###########################
print "\nDOING IMAGE STATISTICS."
iraf.imstat("@darks2Dchips.txt", fields="image,npix,mean,midpt,mode,stddev,min,max", \
Stdout="imstat_darks.txt")

print "\nCREATING A GRAPH."
LINE_IMST, IMAGE, EXPT, VALUES, LINE_IMST_new = ([] for _ in range(5))
with open("imstat_darks.txt","r") as imst_darks:
    for line_i in imst_darks:
        LINE_IMST.append(line_i)
```

```
        col_i = line_i.split()
        IMAGE.append(col_i[0])
    with open("headerDf2Dchips.txt","r") as head_darks:
        for line_h in head_darks:
            col_h = line_h.split()
            EXPT.append(col_h[8])
            VALUES.append((col_h[0],col_h[8], col_h[1]))
EXPTset = set(EXPT[1:])

################### ADDING NEEDED INFO TO IMSTAT TEXT FILES ###################
with open("imstat_darks.txt","w") as imst_darks_w:
    row1 = "\t\t\t\t\t%s\t%s\t\t%s" % (LINE_IMST[0][:-1], "EXPTIME", "CHIP_NUM\n")
    imst_darks_w.write(row1)
    for i in range(len(IMAGE)):
        for j in range(len(VALUES)):
            if IMAGE[i] == VALUES[j][0]:
                row = "%s\t%s\t%s\n" % (LINE_IMST[i][:-1], VALUES[i][1], VALUES[i][2])
                LINE_IMST_new.append(row)
                imst_darks_w.write(row)
```

5.3. Flat-field frames

As sensitivity of pixels of a detector is not the same, we remove a sign of this property by using flat-field frames. A flat field frame records the response of a detector, so it is actually a map of pixel sensitivity. After we divide an image by a master flat-field frame, the image is corrected about this sign.

Unlike dark frames, flat fields are not taken in *Fast Photometry Mode* so a flat field is always $4\times[(2048\times2048)$ pix$]$. Therefore for data reduction it is necessary to cut flat fields (or a master flat field) according to size of science and dark frames.

As I have done comparison of sensitivity of the HAWK-I chips in the case of dark frames, I have done it for flat-field frames as well. Again it is seen well that each of the chips has got different sensitivity—the curves begin on various values of the mean flux. The typical shape of the curves on the graph below is given by the fact that the flat-field frames were taken during sunset.

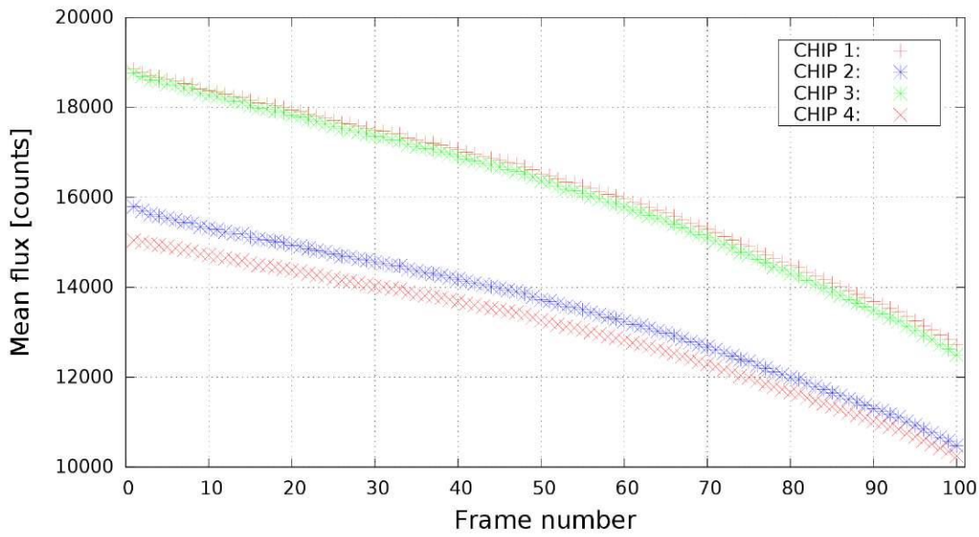


Figure 32: Dependency mean flux on index—flat-field frames.

5.3.1. Scripts `flats.py` and `flatsD.py`

A structure of the script (the same for both scripts):

OPTION OF FLAT FIELD STATISTICS
MAKING OF FLATS LIST FILE (INFO FROM HEADER)
LAUNCHING FLAT STATISTICS SCRIPT IF REQUIRED
MAKING OF MASTERFLATS
EXCHANGING NAMES OF TWO FILES (CHIPS)
ADDING NUMBERS OF CHIPS TO MASTERFLATS HEADERS
COPYING, MOVING, REMOVING FILES

A part of the script:

```
############################ MAKING OF MASTERFLATS ##############################
print "MAKING OF MASTERFLATS."
for chip in range(1,5):
    list_chip = "@flats2D.txt[" + str(chip) + "]"
    image = "mflat_ch"
    iraf.imcombine("%s" % list_chip, "%s %i" % (image, chip), combine="median")
                            # making of master flats

#################### EXCHANGING NAMES OF TWO FILES (CHIPS) ####################
iraf.imcopy("mflat_ch3", "mflat_ch3copy")
iraf.imdelete("mflat_ch3")                    ###> these 6 rows
iraf.imcopy("mflat_ch4", "mflat_ch3")            # deal with HAWKI speciality:
iraf.imdelete("mflat_ch4")                       # extension 4 = chip 3 and
iraf.imcopy("mflat_ch3copy", "mflat_ch4")        # extension 3 = chip 4 <###
iraf.imdelete("mflat_ch3copy")

################ ADDING NUMBERS OF CHIPS TO MASTERFLATS HEADERS ################
for chip in range(1,5):
    mflat = "mflat_ch" + str(chip)
    iraf.hedit("%s" % mflat, "chip_num", "%i" % chip, add="yes", verify="no")
```

5.3.2. The script `flatsStat.py`

A structure of the script:

Function—CREATION OF A GRAPH
CUTTING OF FLATS ACCORDING TO CHIPS + DOING STATISTICS
STATISTICS RESULTS INTO ONE TEXT FILE
WRITING OF STATISTICS RESULTS INTO HEADER TEXT FILE
MAKING OF LISTS FOR GRAPH ACCORDING TO CHIPS+EXPTIMES

A part of the script:

```
############ CUTTING OF FLATS ACCORDING TO CHIPS + DOING STATISTICS ############
l = [[1,1],[2,2],[4,3],[3,4]] # because ext. 3 = chip 4 and ext. 4 = chip 3
line_num = 0
with open("flats2D.txt","r") as flats2d:
    for line in flats2d:
        line_num += 1
        for i in range(4):
            image = line[:-1] + "[" + str(l[i][0]) + "]"
```

```
            list_ch_out = "Flat_" + line[:-1] + "_ch" + str(l[i][1])
            iraf.imcopy("%s" % image, "%s" % list_ch_out)
            iraf.hedit("%s" % list_ch_out, "chip_num", "%i" % l[i][1], add="yes",
                       verify="no")
            iraf.imstat("%s" % image, fields="image,npix,mean,midpt,mode,stddev,
                        min,max", Stdout="%s%i%s%i%s" % ("imstat_flats", line_num,
                        "_ch", l[i][1], txt))

#################### STATISTICS RESULTS INTO ONE TEXT FILE ###################
line_num, ROWS = 0, []
with open("flats2D.txt","r") as flats2d:
    for line in flats2d:
        line_num += 1
        for chip in range(1,5):
            filename_stat = "imstat_flats" + str(line_num) + "_ch" + str(chip) + txt
            with open(filename_stat,"r") as stats:
                ROWS0 = []
                    for line in stats:
                    ROWS0.append(line)
                fromRow2 = ROWS0[1:]
                ROWS = ROWS + fromRow2
                head = ROWS0[0]

with open("imstat_flats_allch.txt","w") as imst_chips:
    imst_chips.write(head)
    for row in ROWS:
        imst_chips.write(row)
```

5.4. Images before and after reduction compared

Data reduction consisted of the following steps:

- creating a master dark frame by taking the median of dark frames (with the same integration time as science images);
- subtracting the created master dark frame from all raw science images [figure 34] to get dark-corrected science images [figure 35];
- creating a master flat field by taking the median of flat fields;
- subtracting the master dark from the master flat field to create a dark-corrected master flat field;
- dividing the dark-corrected master flat field by the median of pixel values of this image to create a normalised master flat field;
- dividing all dark-corrected science images by the normalised master flat field to get resulting images [figure 36].

When not using dark frames, raw science frames are corrected only by a normalised master flat field. However, by using them except for dark current elimination we can eliminate also bad pixels and vertical stripes, typical for *Fast Photometry Mode*. The bad pixels are the ones with extremely different values of counts in contrast to the rest of an image (outside stars). Hot pixels (with high values) can be recognised as small white dots or strips (see figure 33 on the following page).

The overview below shows statistics of an image during data reduction. After subtracting a master dark frame we can see that values of mean, median (MIDPT) and mode are smaller. When a normalised master flat field is used then, the values are changed again.

# IMAGE	NPIX	MEAN	MIDPT	MODE	STDDEV	MIN	MAX
HAWKI.fits	2314240	116.3	89.38	95.97	828.1	-8453.	58446.
HAWKI_darks.fits	2314240	64.6	64.79	65.45	98.9	-5980.	30773.
HAWKI_flats_darks.fits	2314240	65.3	-97.66	-97.75	1893.	-2.234E6	781104.

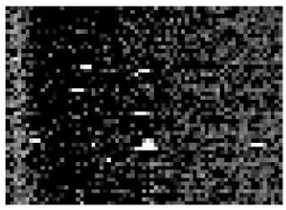


Figure 33: A part of an image with clearly visible hot pixels.

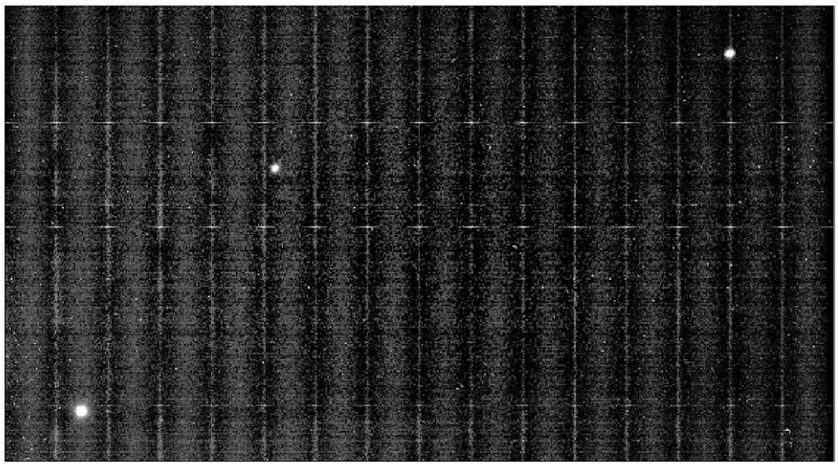


Figure 34: A raw science frame before data reduction.

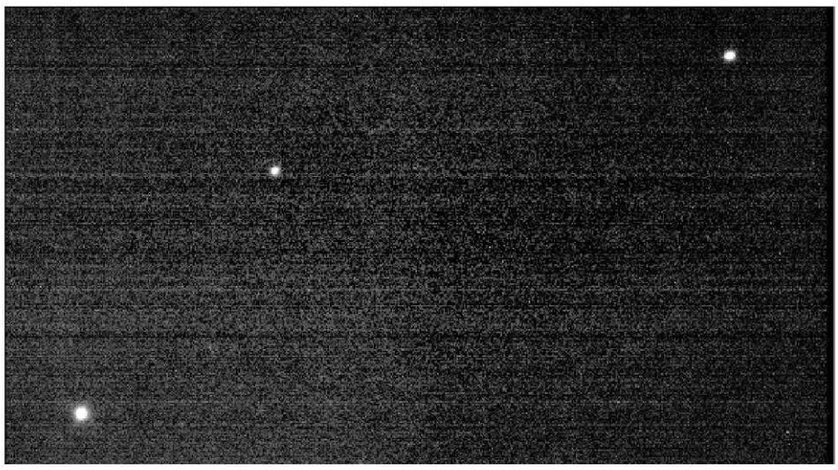


Figure 35: The dark-corrected science frame.

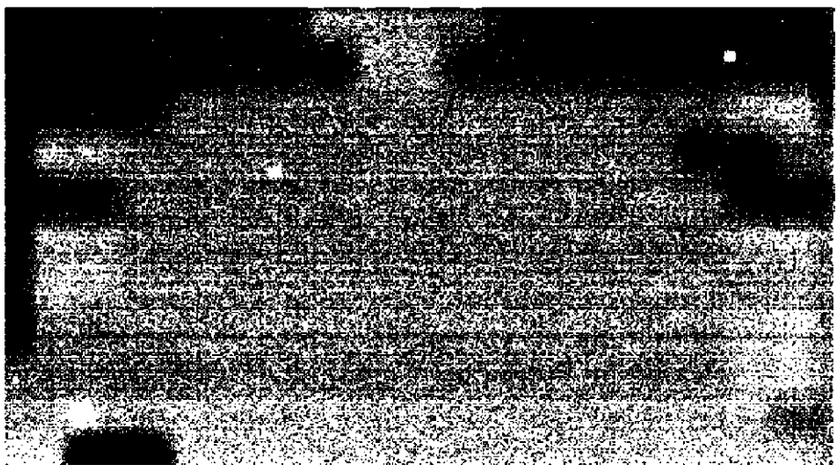


Figure 36: A resulting image the science frame corrected by a master dark frame and a normalised master flat field.

5.4.1. Scripts `reductFlats.py` and `reductFlatsDarks.py`

A structure of the script `reductFlats.py`:

GETTING INFO FROM HEADER + MAKING OF TXT—3D RAW FRAMES
MAKING OF TEXT FILE—3D SCIENCE FRAMES
CALCULATION OF OBSERVING TIME OF EACH RAW SCIENCE FRAME
CREATION OF FILE WITH TIME OF EACH SCIENCE FRAME
SLICING OF RAW SCIENCE FRAMES TO 2D + MAKING OF LIST FILE
ADDING OF CALCULATED OBSERVING TIME TO EACH FRAME'S HEADER
GETTING INFO FROM HEADER + MAKING OF TXT 2D RAW FRAMES
CUTTING OF MASTERFLATS ACCORDING TO SIZES (X,Y) + NORMALISING
CUTTING OF RAW SCIENCE FRAMES ACCORDING TO CHIPS
GETTING INFO FROM HEADER AND MAKING OF TXT RAW SCI., N.M.FLATS
DIVISION RAW SCIENCE FRAMES BY NORM. MASTERFLATS
MAKING OF LISTS OF REDUCED IMAGES ACCORDING TO SIZES AND CHIPS
COPYING, MOVING, REMOVING FILES

A part of the script `reductFlats.py`:

```
############## DIVISION RAW SCIENCE FRAMES BY NORM. MASTER FLATS ###############
print "\nDIVIDING RAW SCIENCE FRAMES BY NORMALISED MASTER FLATS.\n"
with open("headerScRaw.txt","r") as sci_raw:
    SCI, XYC_s = [], []
    for line in sci_raw:
        column = line.split()
        SCI.append(column[0])
        xyc_S = (column[5], column[6], column[3])
        XYC_s.append(xyc_S)

with open("headerNMf.txt","r") as flatheader:
    MFLAT, XYC_f = [], []
    for line in flatheader:
        column = line.split()
        MFLAT.append(column[0])
        xyc_F = (column[4], column[5], column[2])
        XYC_f.append(xyc_F)

n, t = 0, 0
for s in range(len(XYC_s)):
```

```
    for f in range(len(XYC_f)):
        if XYC_s[s] == XYC_f[f]:
            sci, mflat = SCI[s], MFLAT[f]
            iraf.imarith("%s" % sci, "/", "%s" % mflat, "%s %s" % ("nmf_", sci))
            n += 1
            if n == 1000:
                t += 1
                print "Now", n*t, "images done."
                n = 0

####### MAKING OF LISTS OF REDUCED IMAGES ACCORDING TO SIZES AND CHIPS ########
print "\nMAKING OF INPUT TEXT FILES FOR PHOTOMETRY."
os.popen("dfits nmf_* |fitsort %s %s > header_file_mjdc.txt" % ("MJD-CALC","EXPTIME"))
os.popen("cp %s %s" % ("header_file_mjdc.txt", "cp_header_file_mjdc.txt"))

os.popen("dfits nmf_* |fitsort %s %s %s %s 'HIERARCH ESO DET WIN STARTX1'
'HIERARCH ESO DET WIN STARTY1' %s %s %s %s %s 'HIERARCH ESO TEL AMBI FWHM START'
'HIERARCH ESO TEL AIRM START' %s > header_for_phot.txt" % ("ORIGFILE","NAXIS","NAXIS1",
"NAXIS2","CHIP_NUM","EXPTIME","MJD-CALC","MJD-OBS","DATE-OBS","OBJECT"))
os.popen("cp %s %s" % ("header_for_phot.txt", "cp_header_for_phot.txt"))

with open("header_for_phot.txt","r") as nmf:
    FILE0, XYstXstY0, XYstXstYC0 = [], [], []
    for line in nmf:
        column = line.split()
        FILE0.append(column[0])
        xySTxSTyc = (column[3],column[4],column[5],column[6],column[7])
        XYstXstYC0.append(xySTxSTyc)
```

A structure of the script `reductFlatsDarks.py`:

GETTING INFO FROM HEADER + MAKING OF TXT—3D RAW FRAMES
MAKING OF TEXT FILE 3D SCIENCE FRAMES
CALCULATION OF OBSERVING TIME OF EACH RAW SCIENCE FRAME
SLICING OF IMAGES TO 2D + MAKING OF LIST FILE
ADDING CALCULATED OBSERVING TIME TO EACH FRAME'S HEADER
GETTING INFO FROM HEADER + MAKING OF TXT 2D RAW SCIENCE FRAMES
CUTTING OF MASTERFLATS ACCORDING TO SIZES (X,Y)
SUBTRACTION MASTERDARKS FROM MASTERFLATS + NORMALISING
CUTTING OF RAW SCIENCE FRAMES ACCORDING TO CHIPS
GETTING INFO FROM HEADER AND MAKING OF TXT SCI, N.MDARK
SUBTRACTION MASTERDARKS FROM SCIENCE RAW FRAMES
GETTING INFO FROM HEADER AND MAKING OF TXT SCI, N.M.FLATD.
DIVISION DARK-SUBTR. SCI. FRAMES BY NORM. M.FLATS SUBTR. BY DARKS
MAKING OF LISTS OF REDUCED IMAGES ACCORDING TO SIZES AND CHIPS
COPYING, MOVING, REMOVING FILES

A part of the script `reductFlatsDarks.py`:

```
########### SUBTRACTION MASTERDARKS FROM MASTERFLATS + NORMALISING ############
print "\nSUBTRACTING MASTERDARKS FROM MASTERFLATS, NORMALISING.\n"
os.popen("dfits m*_*_*_* |fitsort %s %s %s %s %s %s %s > headerMdMf.txt" %
("OBJECT","CHIP_NUM","NAXIS","NAXIS1","NAXIS2","EXPTIME","DATE-OBS"))

with open("headerMdMf.txt","r") as mdarks_mflats:
```

```
    MDARKS, MFLATS = [], []
    for line in mdarks_mflats:
        column = line.split()
        if column[1] == "DARK":
            MDARKS.append(column[0])
        elif column[1] == "FLAT":
            MFLATS.append(column[0])

for mflat in MFLATS:
    for mdark in MDARKS:
        if mflat[6:] == mdark[6:]:
            iraf.imarith("%s" % mflat, "-", "%s" % mdark, "%s %s" %
                                ("mflatdark_", mflat[6:]))
            iraf.imstat("%s %s" % ("mflatdark_", mflat[6:]),fields="image,npix,
               mean,midpt,mode,stddev,min,max", Stdout="%s%s" % ("imst_mfd", txt))
            iraf.tbdump("%s %s" % ("imst_mfd", txt), "4", "yes", Stdout="%s%s" %
                        ("MIDPT", txt))
            iraf.imarith("%s %s" % ("mflatdark_", mflat[6:]), "/", "%s %s" %
               ("@MIDPT", txt), "%s %s" % ("n_mflatdark_", mflat[6:])) # norm. m. fl.

############## CUTTING OF RAW SCIENCE FRAMES ACCORDING TO CHIPS ################
print "\nCUTTING OF RAW SCIENCE FRAMES.\n"
with open("headerSc2D.txt","r") as sci_header:
    SCI_IM, X, Y, STx, STy = ([] for _ in range(5))
    for line in sci_header:
        column = line.split()
        SCI_IM.append(column[0])
        STx.append(column[1]) # start X in header
        STy.append(column[2]) # start Y in header
        X.append(column[3])   # = NAXIS1 in header
        Y.append(column[4])   # = NAXIS2 in header
```

5.5. Bad pixel mask

Chips of detectors have usually got a small amount of bad pixels. If necessary, these can be removed from data by creating a bad pixel mask. Bad pixel masks show those pixels which do not respond linearly. The mask can be created by taking appropriate flat field observations.

It is generated by taking flat field exposures at high and low intensity levels, and identifying those pixels where the ratio of the normalised flux is more than 15%. For bad pixel mask generating IRAF commands `imcombine`, `imarith` and `ccdmask` of IRAF are used (E12). Command `fixpix` is then used for an application of the generated bad pixel mask on science images. The mask itself is a special type of an image with non-zero values for bad pixels and zero values for good pixels.

Unfortunately, there was no considerable effect after using a bad pixel mask on processed images. An error could have occured due to the wrong selection of master flats or an inner problem in software. Since the mask had not improved quality of tested images, it was not finally used for data reduction.

Figure 37: Bad pixel mask of the chip 3 of HAWK-I.

6. Data processing

After basic photometric data reduction is done, we can now start the next step of data processing our data set. The first, and the most important step, is to do photometry, it means to measure precisely brightness of stars of our interest. From results of the photometry we then create a light curve—dependency of the brightness of a star on time. The most important step is to analyse the resulting data to obtain physical characterisation of the planetary atmosphere.

The beginning of this chapter describe aperture photometry and how apertures were chosen. The next subsection shows the obtained light curve and the following determines precision of photometric measurements. A fitting process of the light curve is described in the last subsection. As in the previous chapter, structures and parts of Python scripts appointed for the data processing are always appended.

6.1. Aperture photometry

To sum light from a star we use a technique called aperture photometry. To get all the light from the star, we have to determine an *aperture radius* which should be larger than the apparent size of the star. But a measured value of brightness also includes a contribution of the sky. To subtract it, we have to choose an annulus within sky background with no star contribution. The size of the annulus is given by *inner annulus* and *outer annulus* parameters [figure 38].

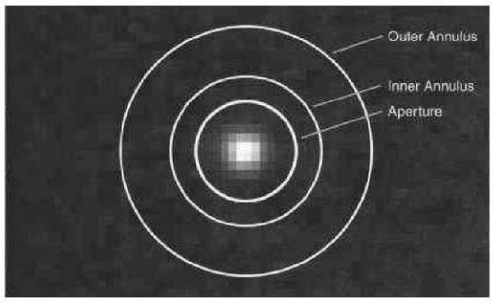


Figure 38: The aperture radius capturing light from the star and the annulus to estimate sky background contribution. Taken from Berry and Burnell (2005).

I have done photometry by using `phot` command in IRAF. An input for the command is a text file containing list of images required to do the photometry and a text file with coordinates of measured stars on given images. We usually need to know fluxes for more aperture radii because we do not know which is the best one and because we measure every star separately. Therefore it is convenient to write all required aperture radii to another text file. For an estimate of the aperture radii it is good to look at a few images to see what pixel area they occupy.

The `phot` command contains various settings and apart from the mentioned text files it is necessary to type sizes of annuli to set width of the sky annulus and its distance from the star. An output of this photometry command is a text file for every single image named according to the image with *mag.#* suffix where we can find all input parameters and especially photometry results such as flux, magnitude, x-centre and y-centre positions, etc.

A selection of an optimal aperture is described in following subsection 6.2.

6.1.1. The script `photometry.py`

A structure of the script:

LOADING NEEDED INFO TO DIFFER SETS
Function CREATION OF 'coords.txt' WITH STARS' COORDINATES
Function CREATION OF 'aperts.txt' WITH APERTURES TO USE
NUMBERS OF HAWK-I CHIPS TO USE IN PHOTOMETRY
SETTING INPUT TEXT FILES FOR PHOTOMETRY
DOING PHOTOMETRY
EXTRACTING INFORMATION FROM mag FILES
WRITING OF PHOT RESULTS AND OTHER INFO INTO TEXT FILES
MAKING OF TEXT FILES WITH THE SAME STAR
COPYING, MOVING, REMOVING FILES

A part of the script:

```
##$##$##$## Function-CREATION OF aperts.txt WITH APERTURES TO USE ##$##$##$##$
def aperts_file(imInfo):
    with open("aperts_"+imInfo+txt,"w") as apertures:
        apert = raw_input("Type apertures which you want to use, separated by comma.\n")
        apertures.write(apert)

################ NUMBERS OF HAWK-I CHIPS TO USE IN PHOTOMETRY #################
CHIPS0, CHIPS = [], []
chips = raw_input("\nWhat chips do you want to use for photometry (type numbers separated
                    by comma)? ")
CHIPS0.append(chips)
for line in CHIPS0:
    for chip in line.split(","):
        CHIPS.append(chip)

################## SETTING INPUT TEXT FILES FOR PHOTOMETRY ####################
ALLINFO, ALLSTARS = [], []
for chip in CHIPS:
    for el in XYXYset:
        i, star, STARS0, INFO = 1, str, [], []
        x, y, STx, STy = str(el[0]), str(el[1]), str(el[2]), str(el[3])
        imInfo = x + "_" + y + "_" + STx + "_" + STy + "_" + chip
        INFO.append(imInfo)
        print "\n" + ">" + 15 * "-" + ">"

        while star != "f": # even if there's no star on an image, 'coords' and
                           'aperts' files must be created (empty)!
            star = raw_input("Write a name of " + imInfo + " set of the " + str(i)
                             + ". star (when finished, type 'f'): ")
            i += 1
            STARS0.append(star)
            ALLSTARS.append(star)
        STARS = STARS0[:-1]
        INFO.append(STARS)
        ALLINFO.append(INFO)
```

6.2. Apertures

After photometry with various apertures had been done, it was necessary to choose one, the best, aperture. It is necessary to select the optimal one. If it is too small, we do not sum all the flux of a star and if it is too large, a lot of sky background is included in the aperture. To determine that aperture it is good to see graphically dependency of an aperture on the flux of a star. The developed pipeline allows the user to display a graph for every image which s/he requires to see. However, the best value of the aperture radius need not be the same for all images, so the user has to specify a criterion for choosing the best one. Another option is to choose more than one aperture and those ones to use for further processing.

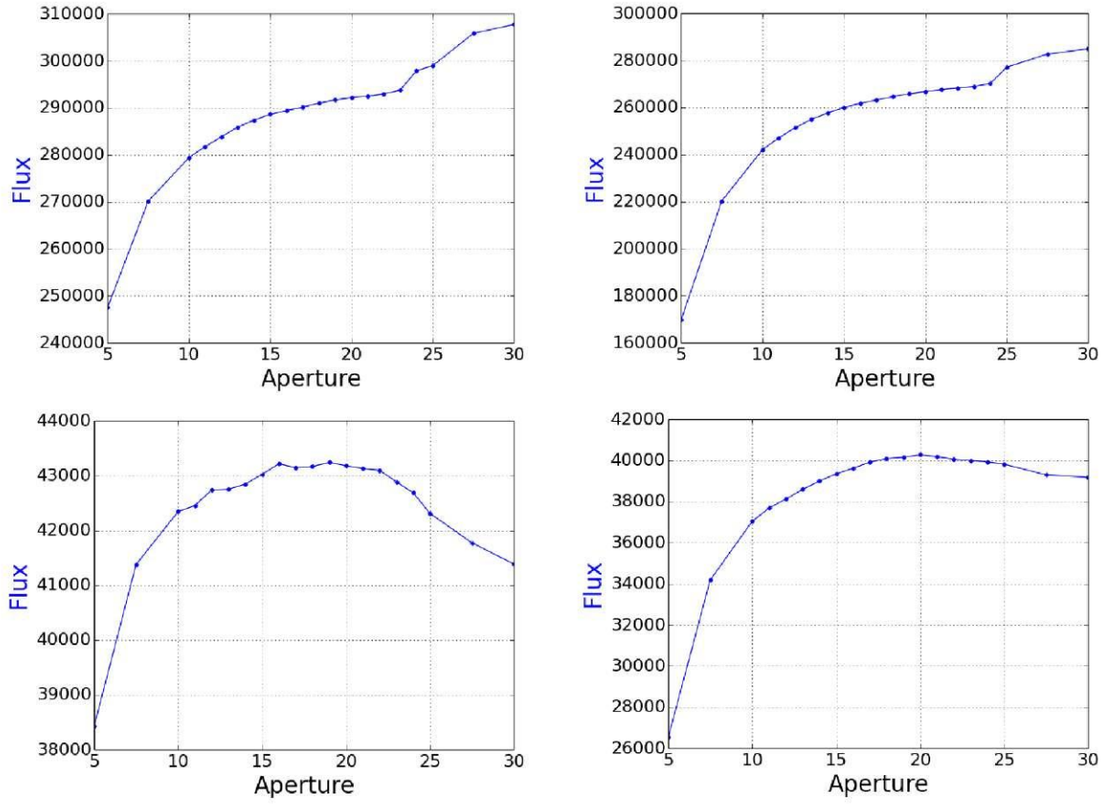


Figure 39: Flux dependency on an aperture for exposure times 0.3966 s: **top**—WASP-18—the first and the last image of the data set; **bottom**—comparison star—the first and the last image of the data set. On images of the comparison stars we can see a drop in flux from the aperture 20. This is because apertures further contain bad pixels of chip with negative values and do not contain any light from the star.

Due to sets of images with different exposure times the size of each star of the set was different. This fact had to be taken into account in choosing the best aperture. I have focused on a set with exposure time 0.3966 s because this set covers a time interval of the whole occultation including sufficient time before and after it.

As primary apertures of the set I have chosen values 15 and 13 for WASP-18 and the comparison star, respectively. By using `txdump` command with easy settings it is possible for further use to extract from *mag.#* files a text file with flux of a star corresponding to a given aperture.

6.2.1. The script `aperts.py`

A structure of the script:

Function—CREATION OF A GRAPH APERTS VS. FLUXES
CHECK FOR POSSIBLE OLD FILES
Function—CHOICE OF THE BEST APERTURE
CHOICE OF STARS FOR A LIGHT CURVE AND BEST APERTURE GRAPH
CHOICE OF STARS' APERTURES FOR THE LIGHT CURVE
COPYING, MOVING, REMOVING FILES

A part of the script:

```
######### CHOICE OF STARS FOR A LIGHT CURVE AND BEST APERTURE GRAPH ###########
LC_STARS = []
for i in range(1,3):
    lc_star = raw_input("Write a name of the " + str(i) + ". star for a light curve
                        (1st / 2nd): ")
    LC_STARS.append(lc_star)

ask_ap = raw_input("\nDo you want to show a graph of apertures vs. fluxes (if not,
                    type 'n')? ")
if ask_ap != "n":
    best_apert(LC_STARS)

################### CHOICE OF STARS' APERTURES FOR THE LIGHT CURVE ####################
check_files()
LINES, ALL, star_count = [], [], 0
for star in LC_STARS:
    star_count += 1
    ap_filename = phot_out + star + txt
    FLUXES_LIST, APS_LIST = [], []
    with open(ap_filename,"r") as lc_ap:
        for line in lc_ap:
            row = line.split()
            num_aps = (len(row) - 6) / 2
            ALLAPS, APS, APS0 = [], [], row[-num_aps:]

            for a in APS0:
                APS.append(float(a))

            num_fluxes_end = ((len(row) - 6) / 2) + 6
            FLUXES, FLUXES0 = [], row[6:num_fluxes_end]
```

6.3. Light curve

In this section, after data reduction, photometry and the best aperture selection, the pipeline creates a light curve, the most important result. To get the final light curve, a few steps preceded. They will be described below.

As observation time of images written in their header is in Modified Julian Date (MJD), to convert MJD to Barycentric Julian Date of Barycentric Dynamical Time (BJD TDB) I had recalculated it to Julian Date (JD) and then I used an online utility (E18) which converted JD of observations to BJD.

A creation of light curves consisted of a few steps:

- image by image to divide flux of a target star by flux of a comparison star (1 or 2) to make a ratio of fluxes;
- from a set of calculated ratios of all images exclude those belonging to images obtained during an eclipse;
- take the median of fluxes ratios of remaining images of the set;
- to divide fluxes ratios of all the images by the median to calculate normalised ratio of fluxes;
- to plot a light curve—a graph showing dependency of the normalised ratio of fluxes on time.

Figure 40 shows light curves af the target star—WASP-18, and comparison star 1. During the whole observation a size of images had been changed as well as exposure times. The change of the size is visible on the light curves as a break between 0.74–0.75 on the time axis. Another effect of this change is that a cadence of taking images increased. From 0.75 the exposure time had been also changed and from 0.775 again, always to shorter times. That is why raw flux of images of particular sets dropped.

Since quality of photometry results of comparison star 2 had been inconvenient, for further data processing only comparison star 1 was used (see figures 19 and 20).

The length of the eclipse is (8,010±60) seconds which is in light curves marked by black vertical lines. These data have been taken from Nymeyer et al. (2010).

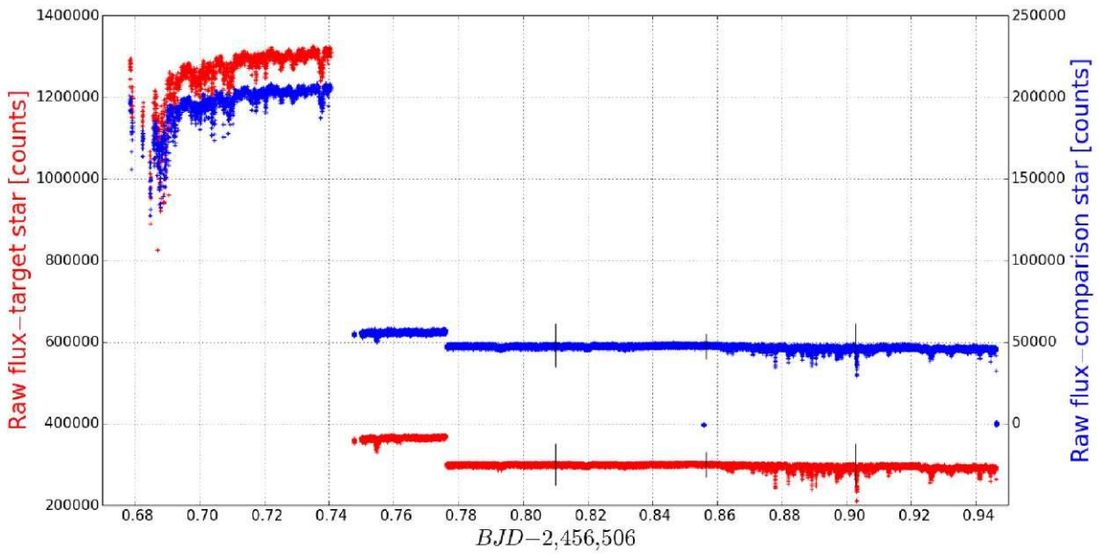


Figure 40: Light curves of WASP-18 (red) and the comparison star (blue) of all the data set (39,079 points). Same as the following images of light curves, the vertical lines show the beginning, the middle and the end of the eclipse.

The flux of the target star had been divided by the flux of the comparison star and then normalised. The resulting differential light curve is shown in figure 41 on the following page.

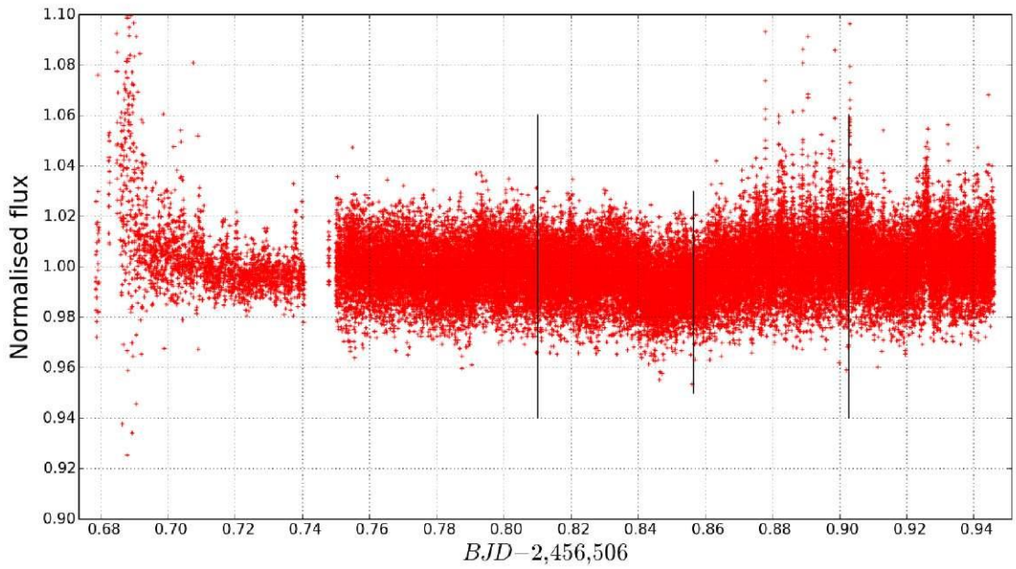


Figure 41: The differential light curve of all the data set (39,079 points)—WASP-18 / comparison star. Density of points increased after size of images had been reduced. The gap between BJD 0.74 and 0.75 is due to the change of the size of the images (and exposure time).

Sets of images with exposure time 2 s, 1.8432 s and 0.5 s (the time from the beginning up to 0.775) were obtained relatively long time before the eclipse, so it was not necessary to use them for next steps of light curve processing.

For further processing only the data set with exposure time 0.3966 s was used. Light curves of the target and the comparison star of the set are shown in figure 42, the differential light curve is shown in figure 43 on the following page.

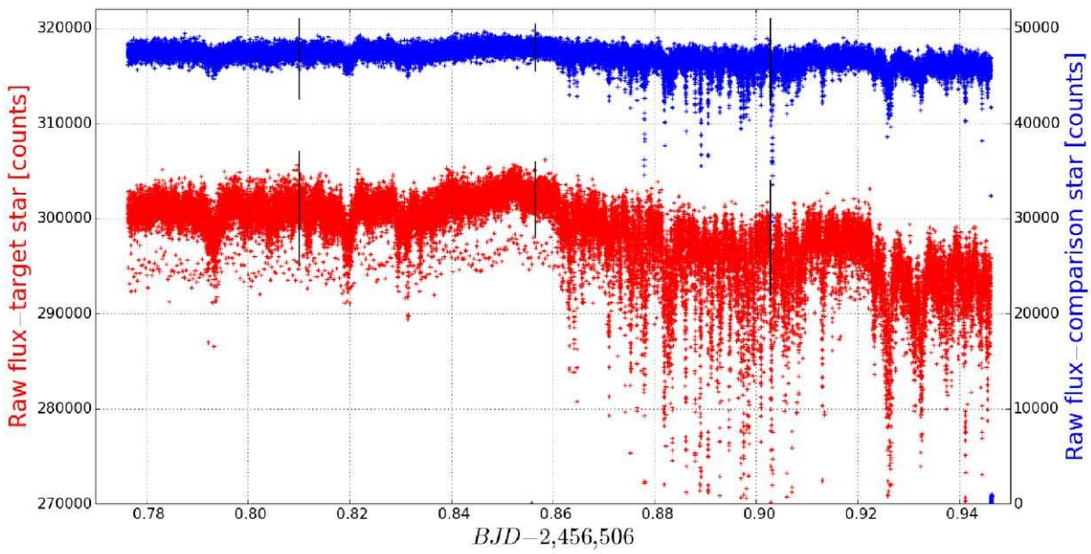


Figure 42: Light curves of WASP-18 (red) and the comparison star (blue) of the data set with exposure time 0.3966 s.

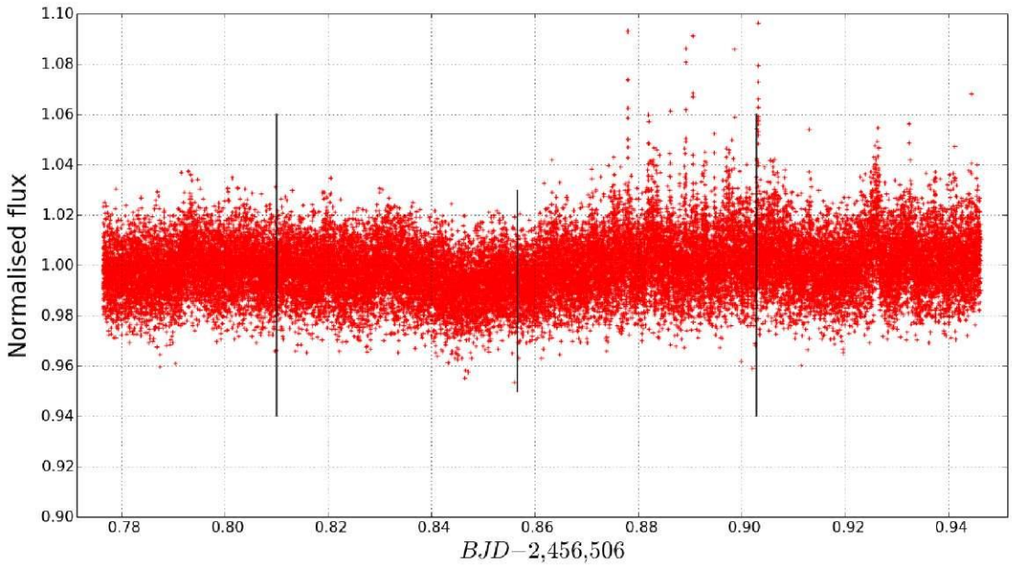


Figure 43: The differential light curve of the data set with exposure time 0.3966 s (32,989 points).

6.3.1. The script `lightCurve.py`

A structure of the script:

Function—CREATION OF A LIGHT CURVE
Function—MAKING OF LC TEXT FILE
Function—MAKING OF OWN TIME SPAN FOR LC
Function—MAKING OF TEXT FILE FOR FITTING →
→ CALCULATION OF ERRORS
→ LOADING AIRMASS + SEEING FROM HEADER
→ MAKING OF TEXT FILE FOR MCMC FIT
BEGINNING, CENTER, END OF A TRANSIT/ECLIPSE
LOADING TEXT FILES WITH MEASURED STARS
LOADING NEEDED INFORMATION OF THE STARS
CALCULATIONS FOR MAKING OF THE LIGHT CURVE
OPTION OF USER'S OWN TIME SPAN FOR LC
OPTION + MAKING OF USER'S OWN TIME SPAN FOR FITTING TXT
COPYING, MOVING, REMOVING FILES

A part of the script:

```
################ BEGINNING, CENTER, END OF A TRANSIT/ECLIPSE ##################
mjd_beg  = raw_input(float("\nType time (in MJD) of the beginning of the transit/
                           eclipse: "))
mjd_cent = raw_input(float("\nType time (in MJD) of the centre of the transit/
                           eclipse: "))
mjd_end  = raw_input(float("\nType time (in MJD) of the end of the transit/
                           eclipse: "))
```

```
################### LOADING TEXT FILES WITH MEASURED STARS ###################
os.popen("ls %s > %s%s" % (main_dir, "files", txt)) # writes names of files and
                                                      folders to a text file
with open("files"+txt,"r") as lc_files:
    for line in lc_files:
        if line[:7] == "1st_lc_":
            File1 = line[:-1]
        elif line[:7] == "2nd_lc_":
            File2 = line[:-1]
os.popen("rm %s%s" % ("files", txt))

################## LOADING NEEDED INFORMATION OF THE STARS ###################
LINES1, LINES2, EXPT1, EXPT2, MJD1, MJD2, FLUX1, FLUX2, FLUX_ratio,\
Xcent1, Xcent2, Ycent1, Ycent2 = ([] for _ in range(13))
with open(File1,"r") as lc1:            # target star
    for line1 in lc1:
        star1 = File1[7:-4]
        LINES1.append(line1)
        col1 = line1.split()
        EXPT1.append(col1[0])
        MJD1.append(col1[1])
        FLUX1.append(float(col1[2]))
        Xcent1.append(col1[3])
        Ycent1.append(col1[4])
    with open(File2,"r") as lc2:        # comparison star
        for line2 in lc2:
            star2 = File2[7:-4]
            LINES2.append(line2)
            col2 = line2.split()
            EXPT2.append(col2[0])
            MJD2.append(col2[1])
            FLUX2.append(float(col2[2]))
            Xcent2.append(col2[3])
            Ycent2.append(col2[4])
```

6.4. Standard deviation

To investigate quality and stability of a light curve, a statistical measure called *standard deviation* is used. To calculate the standard deviation of a measurement, we use the following formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

Here x_i are observed values, $\overline{x}$ is the mean value of all observations and N is the number of values.

I have calculated standard deviations for a few data sets. All sets are of exposure time 0.3966 s. The first one is the set with all points (used for the light curve in figure 43). The second set (marked as "binned-s") contains points limited by known seeing (see figure 45 and details in subsection 6.5.). The third set results from the second set but data are binned by 30 seconds. The last set is the same but binned by 2 min (see figure 46). For all the data sets I have have calculated the deviations for a few time intervals to see how it changes with time. The points implied in the calculations must not cover any part of the occultation so that results would not be affected by a change of flux caused by the occultation. Table 5 compares the calculated standard deviations for the mentioned data sets.

Binning	Time interval				
	30 s	1 min	2 min	5 min	10 min
unbinned	0.0092	0.0095	0.0094	0.0094	0.0094
unbinned-s	0.0048	0.0062	0.0059	0.0087	0.0101
30 s	0.0038	0.0044	0.0048	0.0051	0.0061
120 s	—	—	0.0023	0.0019	0.0034

Binning	Time interval				
	15 min	20 min	30 min	40 min	all data
unbinned	0.0096	0.0096	0.0100	0.0101	0.0113
unbinned-s	0.0105	0.0099	0.0104	0.0112	0.0135
30 s	0.0064	0.0064	0.0072	0.0077	0.0084
120 s	0.0037	0.0037	0.0043	0.0046	0.0051

Table 5: Calculated standard deviations. *Unbinned*: all points; *unbinned-s*: points limited by known seeing; *30 s*: the same points as the previous, binned by 30 s; *120 s*: the same, binned by 2 min.

As the analysis shows, the precision is in order of millimagnitudes and it is better if binning of data is used. The best stability of the light curve is reached on bin 2 minutes over an interval of 5 minutes with its value of 0.0019. In addition, the light curve is slightly corrected by a polynomial during fitting. We can also notice that with longer time intervals the precision becomes worse. Quality of the data had worsened and the variance increased. It is probably caused by repeated changes of seeing which became boldly worse during the occultation (see figure 24).

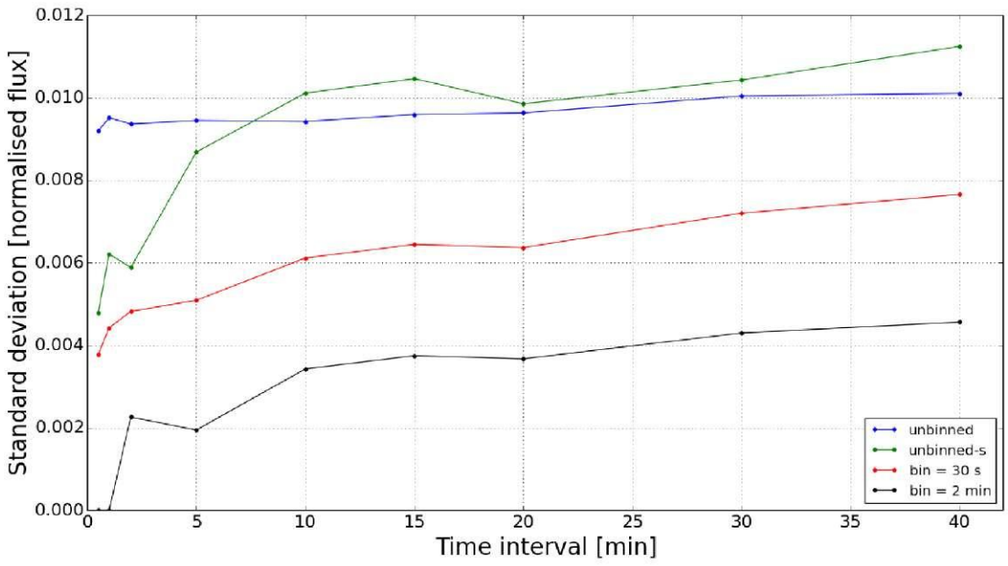


Figure 44: Standard deviation for different time intervals calculated for the unbinned data (blue), unbinned data with the seeing limitation (green), the same data binned by 30 s (red) and by 2 min (black).

A larger bin width means better signal-to-noise ratio, however, it also contributes to a larger value of red noise. This makes precision and stability of a light curve worse. The impact of the red noise of our data is shown on graph 49 in section 6.5. If there is only white noise, the graph would be a line but the red noise deviates the curve from the straight line.

6.4.1. The script `std.py`

A structure of the script:

LOADING NEEDED LC FILE + INFO FROM THAT
SETTING OF INTERVALS + STARTING POINT FOR CALCULATION
MAKING OF LISTS FOR CHOSEN INTERVALS
CALCULATION OF A STANDARD DEVIATION
MAKING OF TEXT FILE WITH RESULTS
PRINTOUT OF A TABLE WITH RESULTS
COPYING, MOVING, REMOVING FILES

A part of the script:

```
############ SETTING OF INTERVALS + STARTING POINT FOR CALCULATION ##############
TM, MINS, SIGmed, SIGmean = ([] for _ in range(4))
time = raw_input("Type in minutes (separated by comma) intervals to determine
                  standard deviation: ")
TM.append(time)

start_ask = raw_input("\nDo you want to start from the first point of the data
                        set or type your own time (if own, type 'o'): ")
if start_ask == "o":
    start_point = float(raw_input("\nType your MJD time: "))
    start = start_point
else:
    start = min(Pair)

#################### MAKING OF LISTS FOR CHOSEN INTERVALS ######################
step = 1.0 / 86400            # step per one second
for line in TM:
    for minute in (line.split(",")):
        MINS.append(minute)
        FLUX = []
        t = float(minute) * 60              # time in seconds
        Range = t * step                    # range of data for RMS
        end = start + Range                 # ending point in MJD

        if end > max(Pair):
            print "\nThe last point is on " + str(max(Pair)) + "."

        for date in Pair:                   # 1st item of each pair in dict.
            if start <= date <= end:
                FLUX.append(Pair[date])
```

6.5. Fitting

For fitting of the resulting data I used Markov Chain Monte Carlo (MCMC) fitting method. MCMC methods in Bayesian statistics (called after English statistician) are algorithms to sample from a probability distribution. It is based on stochastic simulations and on constructing a chain (called Markov after Russian mathematician) with the desired distribution. After a number of steps the state of the chain is then used as a sample of the desired distribution. The higher the number of steps the better quality of the sample. Since this case involves a planet orbiting a star, the model used for fitting is based on Keplerian orbit about a centre of mass of theirs. The used routine has been kindly provided by Michael Gillon. It is closely described in (Gillon et al., 2009). Before an eclipse if being modelled, by using polynomials with low order the light curve is detrended. They can be also used to remove atmospheric extinction, correct positions of the star on the detector or eventually seeing.

Since to fit a light curve created of consecutive points of about 0.4 s (the case of all 32,989 points) is very demanding for a computer and it is not necessary to fit such detailed data, some criteria to exclude a lot of points was needed to determine. One of entries of an input file for MCMC method of fitting is seeing. As previously mentioned, a value of seeing is known only for the first and the last image of each of cube fits files. This was a good criterion to choose points for the light curve fitting. I have chosen the first image of every cube of the data set covering the eclipse. It consisted of 654 points.

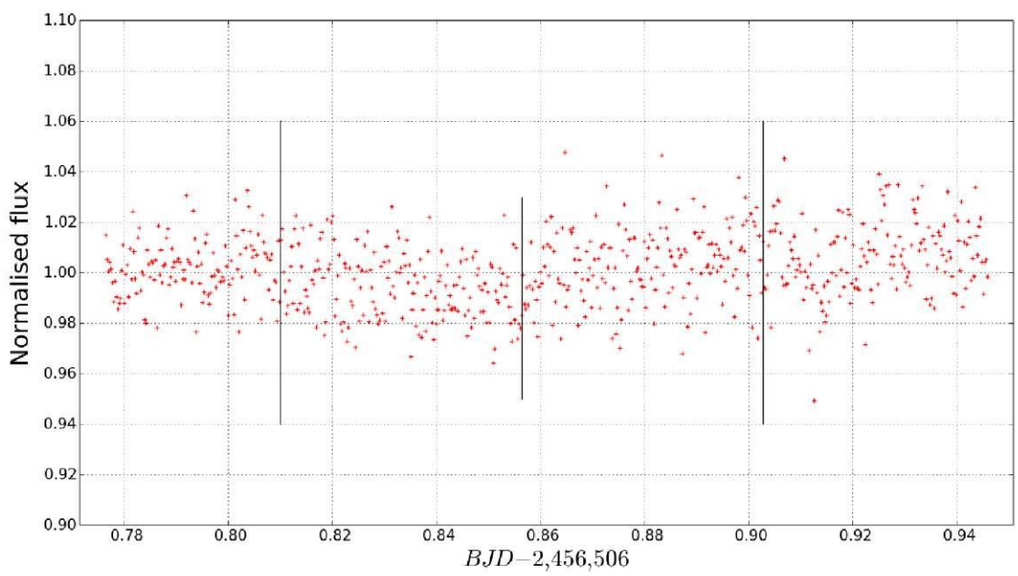


Figure 45: The normalised light curve of the data set with exposure time 0.3966 s created of the first images of each of cubes (654 points).

The input text file for fitting by MCMC code requires 7 columns with respectively these entries: time, a normalised ratio of fluxes, an error of the fluxes, a position of the target star on x-axis, a position of the target star on y-axis, airmass, seeing.

Before fitting is started, except creating the input file parameters of a system and other settings have to be set (in a file *mcmc.dat*). This model of MCMC fitting method allows to fit various physical characteristics but the interest was just to estimate the

eclipse depth, which is equal to a ratio of flux of the planet above flux of the star, so this only fit option was set to **yes**.

I launched the fitting procedure with various settings and with different input files (unbinned with 654 points or binned with smaller number of points). The fitting best converged when binning of 2 minutes was used. This corresponds to 120 points (figure 46). The setting of the *mcmc.dat* file was following:

```
|------------------------------------------------------------------------|
|MARKOV-CHAINS:                                                          |
|------------------------------------------------------------------------|
|NUMBER_OF_CHAINS:                          3                            |
|LENGTH_OF_EACH_CHAIN:                      100000                       |
|BURN-IN_PHASE:                             300                          |
|STAT_LENGTH:                               50                           |
|RED_NOISE_TIMESCALES_(MIN):                10                           |
|BIN_SIZE_OF_GLOBAL_FOLDED_LC_(MIN):        10                           |
|DESIRED_SUCCESS_RATE:                      0.25                         |
|SERIES_OR_NUMERICAL_SOLUTION_TO_KE(s_or_n)?  n                          |
|PRECISION_OF_KE_NUMERICAL_SOLUTION:        1.E-8                        |
|GIBBS_SAMPLER(y_or_n)?                     y                            |
|GLOBAL_ADAPTATION_OF_JUMPS(y_or_n)?        y                            |
|MAXADAPT(min=1)                            1.E3                         |
|GELMAN-RUBIN_TEST?:                        y                            |
|COMPUTATION_OF_MARGINAL_LIKELIHOOD(y_or_n)?  y                          |
|TTV_MINIMIZE(y_or_n)?                      n                            |
|ALLOW_FOR_DDFs(y_or_n)?                    n                            |
|NUMBER_OF_PLANETS:                         1                            |
|MINIMAL_PERIOD_(DAYS):                     0.5                          |
|MAXIMAL_PERIOD_(YEAR):                     0.006                        |
|MINIMAL_PLANET_RADIUS_(R_jup)+SIGMA:       0.001   0.54                 |
|MAXIMAL_PLANET_RADIUS_(R_jup)+SIGMA:       5.3   0.54                   |
|TRANSIT_DILUTION+ERROR_(%):                0.    0.                     |
|------------------------------------------------------------------------|
|DATA:                                                                   |
|------------------------------------------------------------------------|
|NUMBER_OF_PHOTOMETRIC_TIME-SERIES:  1                                   |
|               N      F   A  B_o  C_o  D_o  E  F G TTV Beta WS          |
| 1             120    J+K s  y 1  y 1  n 0  n  n n  n  1.2  n          |
|                           |****************************************|   |
|                           | N=number of points                     |   |
|                           | F=filter [S+uvby,J+UBVRIJHK,ugriz+']   |   |
|                           | A=eclipse nature(p or s)               |   |
|                           | B_o=model extinction?[y/n+order'(1-4)] |   |
|                           | C_o=model trend?[y/n+order(0-4)]       |   |
|                           | D_o=model of X[7]?[y/n+order(0-4)]     |   |
|                           | E=model IRAC/IRS ramp?[y/n]            |   |
|                           | F=model IRAC phase-pixel?[y/n]         |   |
|                           | G=model IRAC WS phase-pixel?[y/n]      |   |
|                           | TTV=fit a TTV for this eclipse?[y/n]   |   |
|                           | DF=fit delta transit depth?[y/n]       |   |
|                           | Beta=Error correction factor[>=1]      |   |
|                           | WS=write baseline PDF?[y/n]            |   |
|                           |****************************************|   |
|OCC: DEPTH + ERROR, DILUTION + ERROR + FIT OPTION[y,n,p]:               |
| 1   0.001   0.1    0.0       0.0     y                                 |
|NUMBER_OF_RV_TIME-SERIES_+_TREND_OPTION[y_or_n+order]      0   n  0     |
|                                    N   JITTER[m/s]  POLYORDER          |
|NUMBER_OF_TRANSIT_TIMINGS:                 0                        n   |
|NUMBER_OF_OCCULTATION_TIMINGS:             1                        n   |
|------------------------------------------------------------------------|
|INITIAL VALUES + ERRORS + FIT OPTIONS[y,n,p]                            |
|------------------------------------------------------------------------|
|DF_+_ERROR:                              0.00917      0.0002       n   |
|IMPACT_PARAMETER_+_ERROR:                0.527        0.052        n   |
|ECLIPSE_DURATION_+_ERROR_(day):          0.0927083    0.000694     n   |
|TO_+_ERROR_(BJD):                        6506.3855    0.2          n   |
|PERIOD_+_ERROR_(day):                    0.9414523    0.0000004    n   |
|EXCENTRICITY_+_ERROR:                    0.0091       0.0012       n   |
|OMEGA_+_ERROR_(degree):                  -91.         3.           n   |
|ERROR_SQRT(E)COSW+SQRT(E)SINW            0.           0.           n   |
|K_+_ERROR_(ms-1):                        0.           0.           n   |
|BETA_+_ERROR_(degree):                   0.           0.           n   |
```

```
|ERROR_VSINICOSB+SINB:                          0.          0.          n |
|-------------------------------------------------------------------------|
|OPTIONS [y/n]                                                            !
!-------------------------------------------------------------------------!
|MODEL_ROSSITER_EFFECT?                         n                         |
|PRIOR_ON_THE_STELLAR_DENSITY?                  n                         |
|FIX_THE_STELLAR_RADIUS_AND_MASS?               n                         |
!PRIOR_ON_VSINI?                                n                         |
!PRIOR_ON_LOGG?                                 n                         |
!USE_TORRES_STELLAR_CALIBRATION?                n                         |
|-------------------------------------------------------------------------|
|LIMB-DARKENING                                                           |
|-------------------------------------------------------------------------|
|LIMB-DARKENING_LAW_[qd,nl,no]                  no                        |
|INTERPOLATE_LD_COEFFICIENTS_FROM_TABLE?_[y,n]  n                         |
|-------------------------------------------------------------------------|
|SPECTROSCOPIC PARAMETERS                                                 |
|-------------------------------------------------------------------------|
|TEFF_+_ERROR_(K):                              6400.0     100.0         |
|LOGG_+_ERROR:                                  4.4         0.15         |
|[FEH]_+_ERROR:                                 0.00        0.09         |
|VTURB_+_ERROR_(kms-1):                         4.8         0.3          |
|VSINI+_ERROR_(kms-1):                          11.0        1.5          |
|-------------------------------------------------------------------------|
|STELLAR PARAMETERS (USED ONLY IF PRIOR ON THE DENSITY)                   |
|-------------------------------------------------------------------------|
|MASS_+_ERROR_(Msun):                           1.28        0.09         |
|RADIUS_+_ERROR_(Rsun):                         1.29        0.16         |
|-------------------------------------------------------------------------|
```

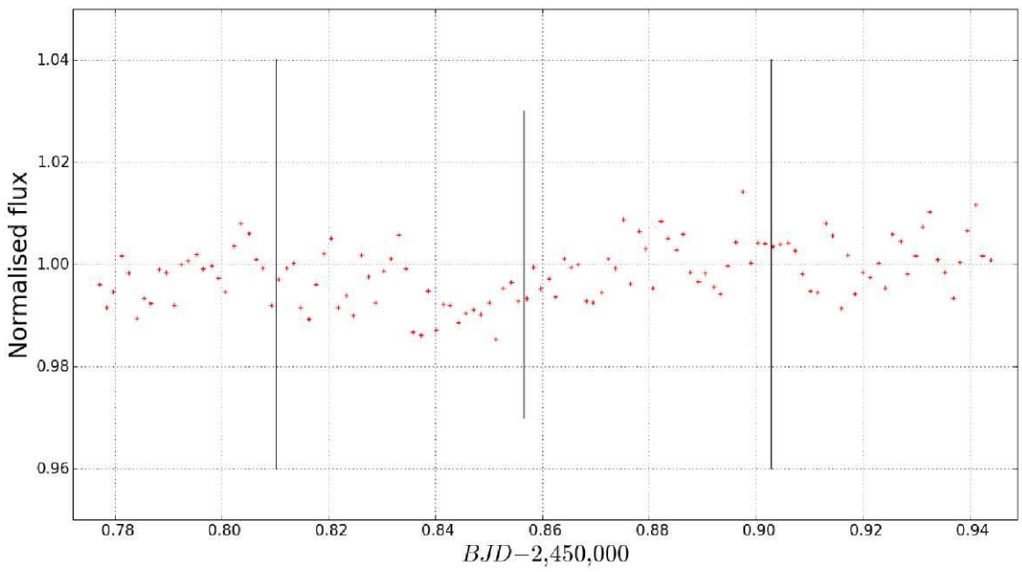


Figure 46: The differential light curve of the data set with exposure time 0.3966 s created from the first images of each of cubes. This curve is binned by 2 min (120 points) and was used for fitting.

The fitting procedure best converged when polynomials to model extinction and to detrend the light curve were of first order. The occultation depth ($\delta_{\rm occ}$) was determined to be $\mathbf{0.0036^{+0.0023}_{-0.0021}}$, in other words, when the planet is hidden behind its host star, the total flux of the system decreases by about 0.36%. In comparison with other observations published in papers (Nymeyer et al. (2010), Iro and Maxted (2013)) the eclipse depth is larger than expected.

The fitted light curve and residuals from the best-fitting model are shown in figures 47 and 48.

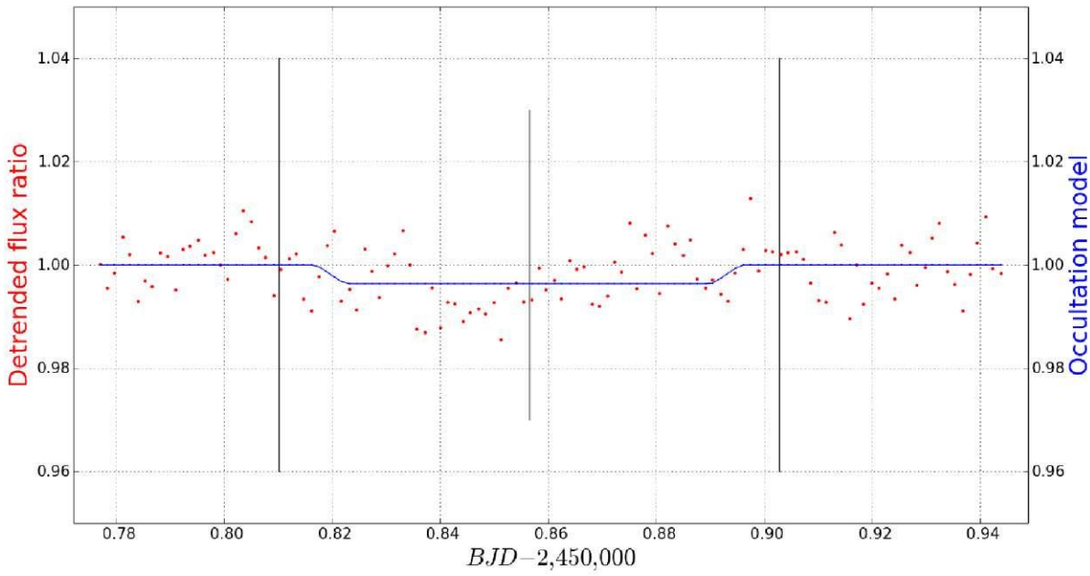


Figure 47: Detrended light curve and the best-fitting model of the occultation of exoplanet WASP-18b. The eclipse depth has been determined to be 0.36 %.

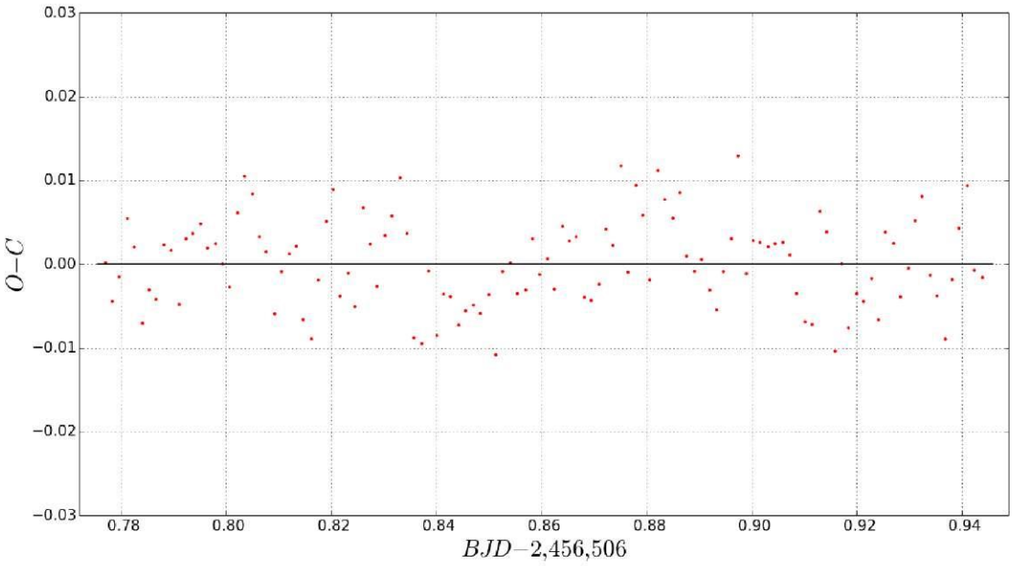


Figure 48: Residuals from the best-fitting model. The standard deviation is 0.0052.

Figure 49 on the following page shows a red noise diagram for the residuals. For various bin widths fitting models had been made. We can see that for larger bin widths, starting at about 20 minutes, the red noise begins to be dominating over the white noise—the curve begins to deviate from the line. This results in the fact that the range of applicable bin widths is limited.

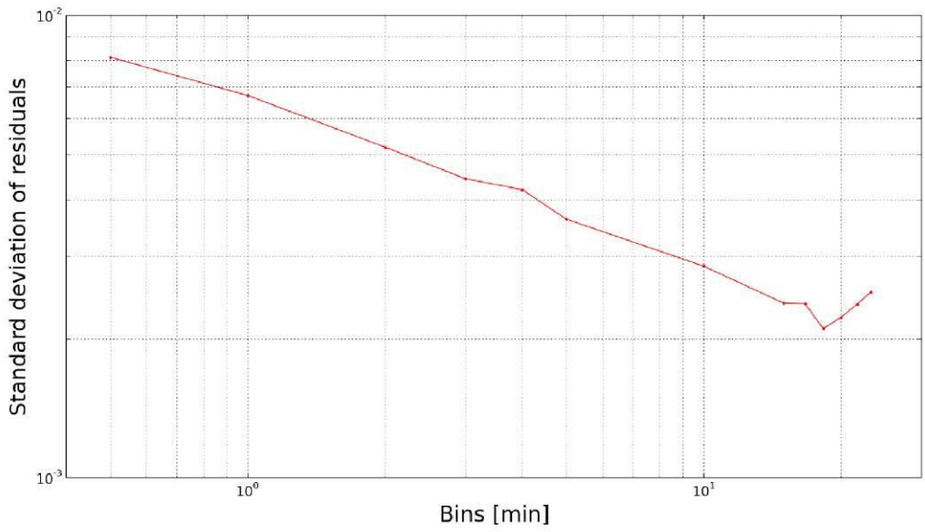


Figure 49: Red noise diagram for the residuals of fitting models. The points on the line represent the standard deviations calculated for a given bin width.

6.6. Discussion of the atmosphere of WASP-18b

Figure 50 shows a model of the atmosphere of WASP-18b (previously shown at subsection 1.3.1.). The occultation model obtained by the data processed in this thesis was added and it is marked by brown. The expected depth for wavelength 2.09 μm (the used filter) was according to the model slightly over 0.002 but the fitted depth is $\mathbf{0.0036^{+0.0023}_{-0.0021}}$. However, this is the first result. A subject of further study will be thorough check of potential instrument effects.

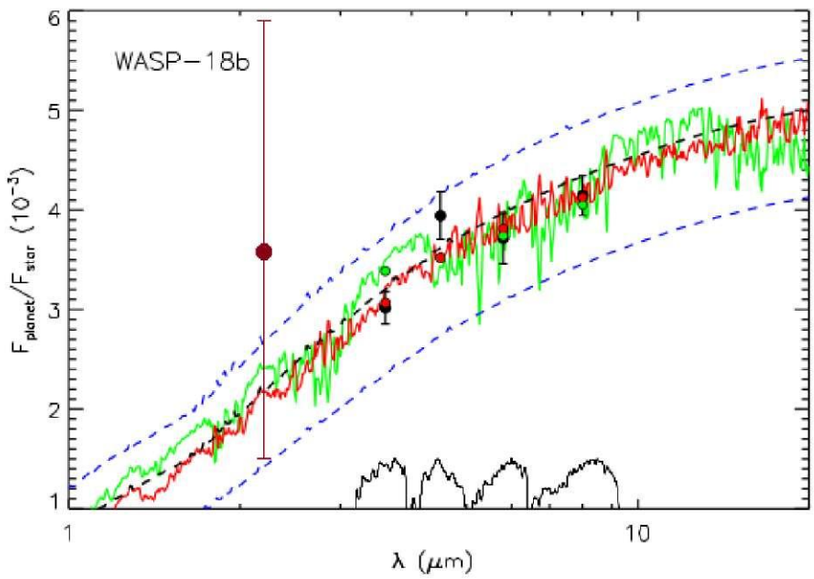


Figure 50: Dayside spectrum of WASP-18b made of observations of *Spitzer's* Infrared Array Camera. It is the same as shown in subsection 1.3.1. but this also includes the results of the fitting process done in this thesis. The brown circle with an error bar is the result which was determined to be 0.0036 but the expected depth had been about 0.002. Adapted from Nymeyer et al. (2010).

The atmosphere of WASP-18b including the heat redistribution was recently studied and published in (Iro and Maxted, 2013). Since the planet is highly irradiated by its host star, one of their input parameters had been temperature inversion on the day-side. The inversion is supposed to be caused by molecules of TiO and VO which in such high temperatures exist in gas phase. The model was calculated for no rotation and for a fast differential rotation of the atmosphere. The rotation model resulted in heat redistribution over the whole planet whereas no rotation model showed higher temperature contrasts between the day-side and the night-side. In the case of no rotation of the atmosphere, the temperature of the day-side is higher than in the rotation case. Then the observed flux of the day-side is higher and consequently the occultation depth as well. No rotation model also matches to the data of the occultation from Nymeyer et al. (2010). Furthermore, when the substellar point (the direction of the star) is exactly facing the Earth, the flux of the planet increases. Another effect affecting the heat redistribution is latitudinal transport of the heat. However, this was not included in the calculations of the model. These mentioned effects may be possible reasons why our occultation depth is larger than expected.

The results of this thesis are preliminary. Further, more detailed processing, enable us to detrend the light curve better which will enhance precision of our measurements.

7. Conclusion

The main aim of this thesis was to develop software able to process photometric data from reduction up to creating a light curve and preparing resulting data for fitting. The developed pipeline is made of a few Python scripts allocated according to their functions. After one script ends, the next one is launched. The pipeline is appointed for *Fast Photometry Mode* where input files/images are 3D cubes. However, after some modifications it could be used to process images not taken by FPM. Some scripts require co-operation with a user, especially those concerning photometry. A detailed description of the pipeline is found in Appendix on page 71.

The software was successfully tested on photometric data of an occultation of system WASP-18, obtained 1.–2. 8. 2013 on ESO Paranal by Very Large Telescope with NIR instrument HAWK-I. The data set consists of four particular sets differing by sizes and especially by exposure times. They contain images with the target star (WASP-18b) a two comparison stars. However, since comparison star 2 could not be used for its inconvenient quality, to create a differential light curve only comparison star 1 could have been used. The first three data sets were not been finally used for processing because they had been taken long time before the occultation. The original number of images had been 39,079. Due to the mentioned reasons and by using a seeing criterion and binning of 2 min, the number of points for the light curve was reduced to 120. After the data reduction and processing had been done, an output of the software was a text file with exactly given structure required by MCMC method of fitting.

Quality of images was affected (among others) by seeing which became worse in about the second half of the eclipse. This is obvious from results of photometry and from the precision analysis. In this stage, an occultation of the planet by its host star has been detected. The detected occultation depth is $\mathbf{0.0036^{+0.0023}_{-0.0021}}$ which is larger than expected when compared to models published in cited publications but it is a clear detection.

In further steps, we need to investigate the nature of such a deep eclipse and try to explain it. Discussion of the photometric precision and more extensive investigation of red noise analysis will be the subject of further analysis along with modelling of an eclipse with a method of Gaussian processes (Gibson et al., 2011). In addition, physical processes such as heat redistribution in the planet surface or potential presence of gasses like TiO and will be taken into account, too. There are various effects on the observed flux of the system during an occultation such as rotation of the atmosphere, position of the day-side of the planet towards an observer, etc. (Iro and Maxted, 2013).

These are first results which will be the subject of further study. After the thorough study is done, the results are intended to be published in a paper.

Bibliography

Berry, R., Burnell, J.: *The handbook of astronomical image processing*. Richmond (USA), Willmann-Bell, Inc., 2005, ISBN 0-943396-82-4. 51

Blažek, M.: *Bakalářská práce: CCD fotometrie tranzitujích extrasolárních planet*. Brno, Faculty of Science of Masaryk University, 2014.

Boss, A. P., et al.: *Working Group on extrasolar planets*. Reports on astronomy, proceedings IAU symposium No. XXVIA, 183–186. International Astronomical Union, 2007. 13

Bouy, H., et al.: *Multi-conjugate adaptive optics images of the Trapezium cluster*. Astronomy & Astrophysics. ESO, 2013. 33

Charbonneau, D., et al.: *Detection of planetary transits across a Sun-like star*. The Astrophysical Journal, volume 529, L45–L48. The American Astronomical Society, USA, 2000. 14, 16

Dragomir, D., et al.: *Rayleigh scattering in the atmosphere of the warm exo-Neptune GJ 3470B*. The Astrophysical Journal, volume 814, number 2, 2015. 19

Gibson, N., P., et al.: *A Gaussian process framework for modelling instrumental systematics: application to transmission spectroscopy*. Monthly Notices of the Royal Astronomical Society, 2011. 67

Gillon, M., et al.: *VLT transit and occultation photometry for the bloated planet CoRoT-1b, 1–10*. Astronomy & Astrophysics, 2009. 61

Hellier, C.: *An orbital period of 0.94 days for the hot-Jupiter planet WASP-18b*. Nature, volume 460, 1098–1100. Letters to Nature, 2009. 18

Huygens, C.: *Cosmotheoros*. Adriaan Moetjens, The Hague, 1698. 12

Iro, N., Maxted, P. F. L.: *On the heat redistribution of the hot transiting exoplanet WASP-18b*. Icarus, volume 226, issue 2, 1719–1723, 2013. 19, 63, 66, 67

Van de Kamp, P.: *Astrometric study of Barnard's star from plates taken with the 24-inch sproul refractor*. The Astronomical Journal, volume 68, number 7, 515–521, USA, American astronomical society, 1963. 12

Knutson, H. A.: *A map of the day-night contrast of the extrasolar planet HD 189733b*. Nature, volume 447, 183–186. Letters to Nature, 2007. 21

Mayor, M., Queloz, D.: *A Jupiter-mass companion to a solar-type star*. Nature, volume 378, issue 6555, 355–359. Geneva observatory, Switzerland, 1995. 12, 16

Nymeyer, S., et al.: *Spitzer secondary eclipses of WASP-18b*. The Astrophysical Journal, 2010. 18, 20, 55, 63, 65

Seager, S.: *Exoplanets*. Arizona (USA), The University of Arizona Press, 2010, ISBN 978-0-8165-2945-2. 12

Seager, S.: *Exoplanet Atmospheres (Physical Processes)*. Princeton–New Jersey (USA), Princeton University Press, 2010, ISBN 978-0-691-14645-4. 19

Seager, S., Deming, D.: *Exoplanet atmospheres*. Astronomy & Astrophysics, volume 48, 631–672. ESO, 2010.

Seager, S., Mallén-Ornelas, G.: *On the unique solution of planet and star parameters from an extrasolar planet transit light curve*. The Astrophysical Journal, 585: 1038–1055. USA, The American astronomical society, 2003. 22

Vidal-Madjar, A., et al.: *An extended upper atmosphere around the extrasolar planet HD209458b*. Nature, volume 422, 143–146. Letters to Nature, 2003. 19

Winn, J. N.: *Transits and occultations*. Massachusetts Institute of Technology, 2010. 21, 23

Wolszczan, A., Frail, D. A.: *A planetary system around the millisecond pulsar PSR1257+12*. Nature, volume 355, 145–147. Letters to Nature, 1992. 12

Internet sources

[E0] http://visibleearth.nasa.gov/

[E1] http://www.iau.org/

[E2] http://www.exoplanet.eu/

[E3] http://exoplanets.org/

[E4] https://corot.cnes.fr/

[E5] http://sci.esa.int/

[E6] http://hatnet.org/

[E7] http://hatsouth.org/

[E8] http://kepler.nasa.gov/

[E9] https://keplerscience.arc.nasa.gov/

[E10] http://ogle.astrouw.edu.pl/

[E11] http://wasp-planets.net/

[E12] http://www.eso.org/

[E13] http://jwst.nasa.gov/

[E14] http://exoplanetarchive.ipac.caltech.edu/

[E15] http://archive.stsci.edu/

[E16] https://www.noao.edu/meetings/gdw/files/Joyce_IR.pdf

[E17] http://iraf.noao.edu/

[E18] http://astroutils.astronomy.ohio-state.edu/

Appendix

A description of HAWK-I Fast Photometry Mode pipeline

This pipeline for processing HAWK-I data obtained using Fast Photometry Mode (FPM) developed in Python programming language uses simple and basic commands. Then, it is easy to understand what each row executes and why even for a user with basic skills in programming.

All scripts in the pipeline co-operate with a configuration file (*config.txt*). All items in the file have to be set as the user requires, especially paths to folders which are used and which will be created by running the program. Some items in the file can be unnecessary for the user, then those rows can be deleted or deactivated for Python reading.

Each of the scripts firstly loads needed libraries and assigns parameters referring to the configuration file. Most information about given images is taken from their fits-header and the rest is calculated and ordinarily writen to the header. Generally, if there is a prompt to choose one of two options and only one option is specified by a letter, the second option is simply Enter. For example, if there is "(if not, type 'n')", a user chooses between typing 'n' (+Enter) or Enter. At the end of most scripts "cleaning" is done which means removing unnecessary files and moving other files to appointed folders. These files could be useful for the user later.

It is supposed that all images meant to be processed are in the same directory and together with all scripts of the pipeline. That directory is a main directory (see "main_dir" in *config.txt*) and all new folders and files made by the pipeline are found here (before the files are moved to a different directory).

The pipeline is started by launching `pipeline.py` script. This starting script cleans the main directory if the pipeline was running previously or there are unnecessary files or folders. Another task of the script is renaming of images prepared for the processing. Sometimes it happens that images belonging together have got different names (not only in numbers) which could cause troubles during data processing. That is why all the images are named according to *origfile* in their header. Then such a problem should be avoided. At the end of the script "darks" and "flats" folders are created and the user chooses if s/he requires dark frames for data reduction or not. If not, they are moved to their folder and not used anymore.

From this point there are two paths for the data reduction using dark frames or without using them.

WITH DARK FRAMES

The introductory script launches a script `darks.py`. In the beginning the user chooses if s/he wants to do dark frames statistics. This script, according to header information, takes dark frames cube fits files of FPM, which are in fact 3D images, and slice them into particular frames (2D). As HAWK-I consists of four chips, which are not uniformly sensitive, all dark frames are then cut into four single images according to those chips. From these images master dark frames are finally created.

If statistics of dark frames is required, a script `darksStat.py` is launched. It does statistics (mean flux) of each of the chips for all dark frames and a graph showing this where an integer on the x-axis indicates dark frame number according to a number of

them and mean flux in counts is on the y-axis. One graph is shown for images of the same exposure time, so if there are various exposure times of the images, more graphs are shown one by one.

The next script, `flatsD.py`, gives to the user in its beginning an option to do statistics of flat-field frames. The script creates, chip by chip, master flat-field frames.

The script named `flatsStat.py` does statistics of flat-field frames if required. Images are cut into single chips and the statistics is done. Then, identically with dark frames, graphs with mean flux of each chip of all the images for every single exposure time are shown.

The last script for the data reduction (using dark frames) is `reductFlatsDarks.py`. It begins with calculations of observing time of each science frame. This necessity is not obvious – observing time written in headers of images of a same cube is the same but this time is real observing time just of the **first image** of the cube.

Science frames are then sliced into 2D and according to their sizes master flats are cut to the same sizes. Since dark frames have to be taken with equal sizes as science frames, now both types of reduction frames are prepared for data reduction. Firstly master dark frames are subtracted from master flat-field frames to create master flat-darks and during this cycle normalised master flat-darks are created as well (by using the median of pixel values). After raw science frames are cut into sets according to chips, master dark frames are subtracted from them to create dark-corrected science frames. These images are then divided by normalised master flat-darks to create resulting images. The data reduction is now done and the script at its end creates text files (lists) according to sizes and chips as input files for photometry.

WITHOUT DARK FRAMES

If the user chooses doing the data reduction without the use of dark frames, after the introductory script `pipeline.py` scripts `flats.py` (+ eventually `flatsStat.py`) and then `reductFlats.py` are launched. A script processing dark frames certainly does not feature here. A structure of the scripts is the same as the structure in scripts using also dark frames.

Scripts `reductFlatsDarks.py` and `reductFlats.py`, at their very end, move, copy and remove most of created files . Here the two previously mentioned paths link back.

DATA PROCESSING

Photometry begins by launching `photometry.py` script. It takes all needed information from a header of reduced science frames to distinguish among different properties of images. For each group of images two text files are needed: *coords_#.txt* with coordinates (X and Y) of measured stars on given images and *aperts_#.txt* with numbers of apertures to use in photometry for each of stars. The user can create these files in advance. The script asks him about them and if they do not exist yet, they are created by the script with a cooperation with the user. The first prompt to the user is to type what chips of processed images s/he wants to use for doing photometry. It means that if there is no star on a chip of an image (or the star is not of his interest), photometry is not done on this chip(s) of all images. This saves a lot of time in the case of high amount of images which is in FPM usual. Then the user is prompted to type names of stars for each individual set of images and also asked if files *coords_#.txt* and *aperts_#.txt* exist. It is very important to type a number of stars as the same as a number of X and Y coordinates in *coords_#.txt*

file and identically an order of the stars in those files. If not so, results of the pipeline will be incorrect.

Variables and all settings for doing photometry are loaded from the configuration file. Photometry is done consecutively on all types of images and after it is finished, required data results from *mag.#* files are saved to *photRes*_set_*star#.txt* files. Certainly the content of that file can be changed by the user by rewriting required data results in `txdump` command. As an output of the script there are text files of particular stars (*photRes*_star#*.txt*) containing photometry results and other information needed for forthcoming processing.

To choose the best aperture for a light curve, the script `aperts.py` is launched as the next one. At the beginning the user is prompted to type names of two stars (they must be the same as typed in `photometry.py`) which will be used for creating the light curve—a target star divided by a comparison star. Then the user can choose if s/he wants to show graphically dependency of aperture radii on fluxes. If so, s/he can select any image (= row in text file of a star) with the given star and the script takes all apertures and corresponding fluxes to plot them. This helps a lot to choose the best aperture. It is possible to display these graphs for as many images as the user requires. After all required graphs of both stars are shown and closed, the user is prompted to type a number of an aperture which will be used for creating the light curve. Files with all needed information of the both stars are then created (*1st_* / *2nd_lc*_star_*ap#.txt*).

The next script is `lightCurve.py`. First of all, the user has to type time of the beginning, the centre and the end of the eclipse which is expected to be seen in the light curve. Then light curves (target / comparison star) are shown consecutively for each set according to exposure times. The last light curve is a normalised one containing all the processed data. Text file with the same values as previously used in the light curves is also created (*lc*_star1_*ap#*_star2_*ap#.txt*). Since the script is focused to be an input for fitting by MCMC code, text file with required content for MCMC code is created as well. There is another option for the user—s/he can set her/his own time span for showing the light curve (and saving it as an image) and the making of the mentioned text files.

The last script of the pipeline, `std.py`, calculates a standard deviation to find out precision of the processed data set. The first values, what the user sets, are time intervals to use for calculating the standard deviation. An own starting (time) point can be also set, otherwise the script starts calculations from the first point. To calculate the standard deviation the script uses the mean as well as the median of given values. Required time intervals with calculated deviations are then printed as a table.

Note: The pipeline is set that all images—both original and created, are saved (and copied to given folders). That is very time-consuming and needs a lot of free space on a hard disc. If the user supposes that s/he will not need original or some created images, s/he should reconfigure scripts to delete all unnecessary images after new ones are created.

An example of data reduction: Raw science frame 3D ⇒ Raw science frames 2D ⇒ Raw science frames 2D cut according to chips ⇒ Dark-corrected science frames ⇒ Flat and dark-corrected science frames = resulting image. Here we can see particular steps of data reduction, so the user can set the scripts to delete all images except for the resulting ones.

My software can be send upon a request. If you would like to use it, please contact me on 393760@mail.muni.cz.