SEARCH SURVEY AND CHARACTERIZATION EFFORT OF EXTRA-SOLAR PLANETS FROM THE HOLOMON ASTRONOMICAL STATION

Panagiotis Ioannidis

Department of Physics

Section of Astrophysics, Astrononomy and Mechanics

Aristotle University

Supervisor: Prof. John H. Seiradakis

To my loving parents ...

Acknowledgements

This Diploma Thesis would not be possible to be completed without the contribution of some people, which I would like to acknowledge. First of all is my professor, supervisor and mentor, John H. Seiradakis, as he was the person who introduced me in astronomy, encouraged me to complete my studies with this thesis and helped me with every way he could to finish it. Also, I want to thank Dimitris Mislis for his precious help, as he would give me his ideas to work with and answer every question without any complains. In addition John Antoniadis is the person who helped me, in my first steps, to understand the basics of observational astronomy, reduction and photometry. Last, but not least, is my fellow-student and girlfriend Chrysa Avdellidou for following me to the observations, letting me share my ideas with her and generally to help me evolve to the person that I am now.

A big thank you deserves Mr. Panourgias, forester of the University Forest of Mt. Holomon and the caretaker of the facilities Mr. Papanikolaou, for taking care of us during the observations. Finally I would like to thank a lot of people who got disturbed from me about my work on this project like Liakos A., Tsiganis K. and other.

Abstract

The first thoughts about the ThReT (Thessaloniki Research for Transits) project started at 2004, when the Aristotle University, participated to the Wasp0 prototype project. After that the first promising trials took place between 2005-2010. This Diploma Thesis is the summary of the writer's effort to organize and run an efficient and competitive survey for detection of big extrasolar planets with the method of transits, using commercial instruments. The project is consisted of two parts. The first one is the instrumental setup and configuration, for high precession photometry. The second one is the development of the relative software for the reduction and photometry of the raw data, followed by the removal of trends and the search for interesting signals. The results of the first two runs are promising enough as a lot of new variable stars were detected. Also, as all of the information are open in public along with the software, they can be used from others to create their own surveys. The above are described in four main chapters. In the introduction there is a sort description of the methods that have been developed from scientists in order to search for extra-solar planets, as a brief explanation of the advantages and disadvantages for each one of them. In the first chapter there is a analysis of the wide field astronomy in combination with the search of extrasolar planets with the method of transits. In the second chapter there is a description of the equipment which have been used, the software development and the efforts that have been done to evolve the project in order to become a competitive survey for extra-solar planets and variable stars. The third chapter consists of the fundamental analysis and the software development. Finally in the last chapter are presented the observations and the results of the first two runs.

Nomenclature

General Symbols

AU	Astronomical Unit 149.6×10^6 km
M_j	Mass of Jupiter
$\mathrm{M}_\oplus$	Mass of Earth
$\mathrm{M}_\odot$	Mass of Sun
R_j	Radius of Jupiter
$\mathrm{R}_\oplus$	Radius of Earth
$\mathrm{R}_\odot$	Radius of Sun
RV	Radial Velocity
ESA	European Space Agency
$NASA$	National Aeronautics and Space Andministration

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Extrasolar planets

From middle 1990's, when the first extra-solar planet detected, the search for such bodies became one of the most interesting projects in astronomy. This kind of reaction was expected as the increasing number of discovered planets, is strictly connected to the answer of the ultimate question; "are we alone?".

1.1.1 Extrasolar

Extra-solar planet or exoplanet is a planet which is not existing around our Sun but it orbits around an other star. As of February 2012, the IAU working definition of an exoplanet, which issued in 2001 and modified in 2003 (Definition of a "Planet"[UNION, 2003]) contains the following criteria:

- Objects with true masses below the limiting mass for thermonuclear fusion of deuterium (currently calculated to be 13 Jupiter masses for objects of solar metallisity) 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.
- Substellar objects with true masses above the limiting mass for thermonuclear fusion of deuterium are "brown dwarfs", no matter how they formed

or where they are located.

- Free-floating objects in young star clusters with masses below the limiting mass for thermonuclear fusion of deuterium are not "planets", but are "sub-brown dwarfs" (or whatever name is most appropriate).

The detection of such bodies is very interesting as one or more of them could fulfill the appropriate conditions for the development of life. The first reference of the phrase "extra-solar planet" was in 1942, when Kaj Strand of the Sproul Observatory, under the direction of Peter van de Kamp claimed that the detected perturbations in the lightcurves of the Cygni 61 binary was the result of an "invisible third component" (van de Bos et al 1943 [van den Bos, 1943]). The case of Cygni 61 haven't resolved yet.

The first exoplanet that have been ever detected was around the pulsar PSR B1257+12 in 1992. This discovery was treated with great suspicion because of the expectation that extra-solar planets should be orbiting only main sequence stars. After that, in 1995, was the first announcement that a planet of 0.47 M_j had been detected, using the radial velocity method, around the 51 Pegasi star. The confirmed existence of a planet around an other main sequence star started a new era for astronomy, where a lot of other planetary systems where to be detected. The size of the most of those planets and the length of their orbit's semi-major axis, however, wasn't even close to our solar system prototype. Planets with about the size of Jupiter was very close to their host star, some times a lot closer than Mercury. So they were named Hot-Jupiters.

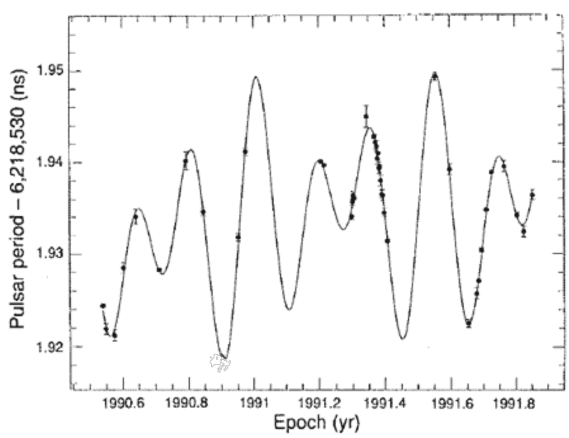

Figure 1.1: Period variations of PSR1257+12. Each period measurement is based on observations made on at least two consecutive days. The solid line denotes changes in period predicted by a two-planet model of the 1257+12 system. [Wolszczan and Frail, 1992]

Some years and almost 30 planets later, the assumption that the flux of the host star should be decreasing slightly as a planet, with inclination around 90^o, crosses the line of site between earth and the star, was confirmed. In 1999, two separated teams, one led from David Charbonneau and the other by Gregory W. Henry detected an 1.7% decrease of the flux (figure 1.2) of the star HD 209458 which caused to the transit of the HD 209458 b, a planet who was earlier confirmed that existed with the RV method.

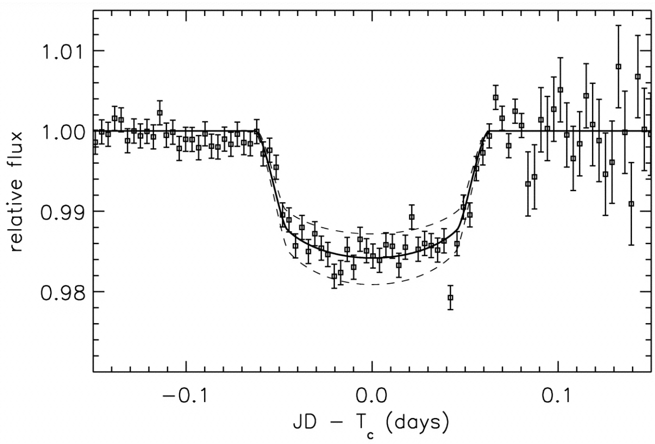

Figure 1.2: HD 209458 first transit detected

In the same year, another great discovery was made. The astronomer J. J. Lissauer and an other group led from P. R. Butler, confirmed that the RV diagram of the main sequence star u Andromeda was containing not only one period but three. Few years later, in 2002, an M type dwarf companion found around this star, so this system became the first known binary star, with main sequence components, which is having planets around it.

Back in 1991 a team of astronomers from Princeton University proposed the use of gravitational microlensing, a part from Einstein's theory for General Relativity, to find exoplanets. After 12 years, the creation of an effective observational technique led to the first success of that method. The combination of the data which were taken from Microlensing Observations in Astrophysics (MOA) and Optical Gravitational Lensing Experiment (OGLE) revealed a planet of, most likely, 1.3 M_j.

Until 2004 about 120 planets were found but none of them was seen directly. In 2004 a group of astronomers, using the ESO's Very Large Telescope Interferometer (VLTI) in Chile, produced the image of 2M1207 b, a planet with mass several times the mass of Jupiter and in a distance of $\sim$46 AU of its host star.

As the scientific interest around extrasolar planets was increasing, a lot of surveys were developed around the world, based on earth and in space. The most successful methods to search planets have been proved to be the radial velocity method and the method of transits[1]. The very big accuracy which is needed in order to detect a planet, is the reason that the future of surveys is in space. There are already two mission ongoing, which are trying to detect planets with the method of transits. These are the CoRoT mission from ESA and the KEPLER mission from NASA, which launched in December of 2006 and March of 2009 respectively. The contribution of those missions is huge. There are over twenty planets found from CoRoT and over thirty planets from KEPLER, which actually has over 2321 planetary candidates after the second data release in February 27^{th}, 2012. In table 1.1 there are the surveys which have found and confirmed at least one planet, in respect with the detection method and the percentage of planets detected with each one.

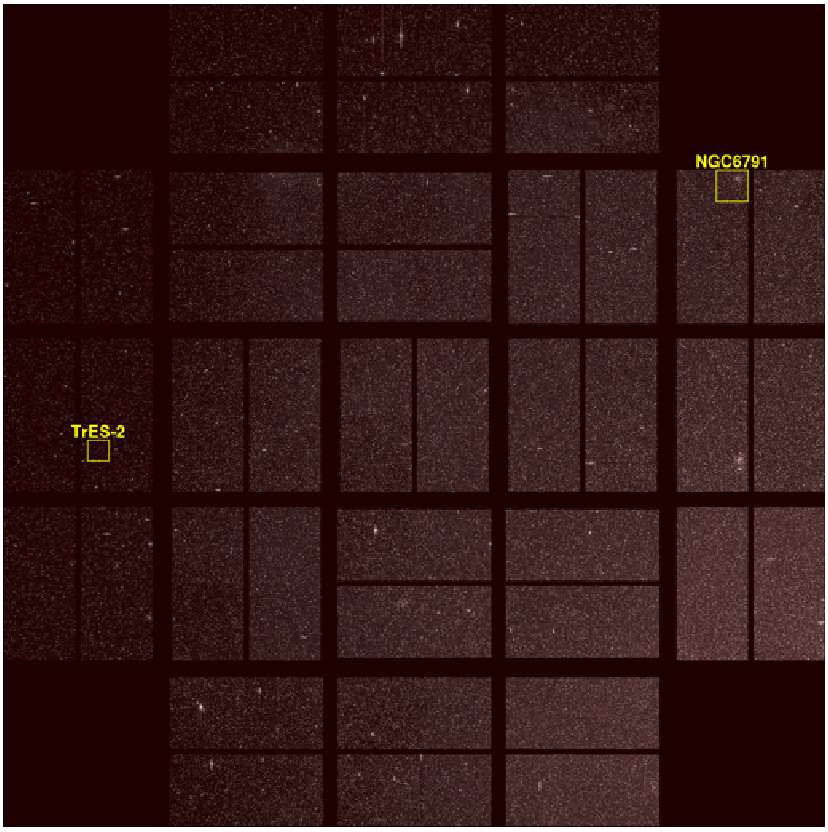

Figure 1.3: Kepler mission full field of view

[1]The reasons are explained in chapter 1.2

Surveys		Method	% of Planets
Earth	Space		
AFOE AAPS CCPS Coralie Elodie ESO-CES HARPS HET McDonald Obs NK2 Sophie		Radial Velocity	61%
HAT QES superWASP TrES XO	CoRoT KEPLER	Transits	31%
		Direct Imaging	4%
MPS MOA OGLE PLANET Robonet		Micro Lensing	2%
PPD		Pulsar Timing	2%

Table 1.1: Planetary Surveys and the percentage of planets detected with each method as of March 8^{th}, 2012.

1.1.2 Planets

Even from the first discoveries of planets, it was obvious that the prototype of our solar system wasn't necessary the dominant model for the rest of the planetary systems. In the other hand, the variety of the planets in our solar system, which covers the most of the possible scenarios of planet sizes, can help us to understand the physiology of the extrasolar planets.

Terrestrial Planets

The name Terrestrial is derived by the Latin name for Earth (Terra). This kind of planets, known also as rocky planets, are primarily composed by silicate rocks or metals. They have metallic core (mostly iron) and silicate mantle. In our solar system the uncompressed density[1] of the Terrestrial planets varies between 2.8 and 5.3 $g \cdot cm^{-3}$. The atmospheres of the Terrestrial planets, is a mix of the primordial and a secondary atmosphere. The primary atmosphere for every terrestrial planet was composed mostly of light gases as H and He that accreted during the initial formation from the protoplanetary nebula. However, the small gravitational field of those planets is incapable to keep the lighter of them but traps the heavier [Pepin, 1991]. Those elements are creating two different groups of matter, rocky materials like iron, olivine and pyroxene and icy materials like $H_2O, CO_2, CH_4, NH_3, SO_2$. If the planet is a small one, cools fast and the icy materials are remaining trapped inside the rocky mantle. In opposition, if the planet has enough mass, the cooling process is slower, with a lot of heat in

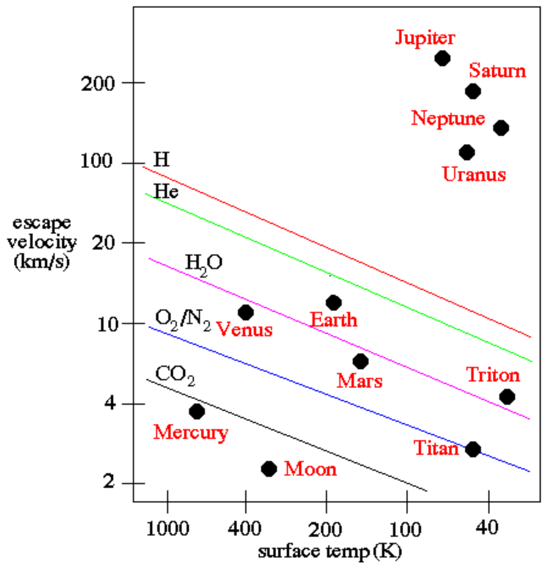

Figure 1.4: Escape velosity vs surface temp of the planet.

[1] Uncompressed density is used rather than true average density because compression within planet cores increases their density

the planet. The result is a large amount of tectonic activity which results the icy materials to escape and form the secondary atmosphere [Lammer et al., 2009]. There are three sub-categories, in respect of the mass of the planets.

The first one is the Subterraneans or Sub-earths with mass distribution between 0 and 0.5 $M_{\oplus}$. This kind of planets, because of their small size, they do not maintain an atmosphere if they are closer to their host star, than the outer edge of the habitable zone.

The second sub-category is the Terrans or the Earth-likes. These planets, with mass distribution between 0.5 and 2 $M_{\oplus}$, are able to hold significant atmospheres with a possibility to maintain liquid water if they live in the habitable zone of their host star. In most of the cases its is believed that over this lower mass limit, the planets are also maintain magnetic fields because of dynamo effect, caused to the melted iron inner core.

The final sub-category of the Terrestrial planets contain the planets with masses between 2 and 10 $M_{\oplus}$. This kind of planets are called Super-earths or Superterrans. It believed that superterrans are able to maintain dense atmospheres with liquid water within the Habitable zone of their star under certain circumstances.

Gas Giant Planets

The Gas giants have completely different structure and properties from the terrestrial planets. They are massive, with masses between 10 and 5000 $M_{\oplus}$. A peculiarity of those planets is the lack of surfaces, in contrast with the terrestrials. These planets are composed of four main layers, the core, the mantle and the main and outer atmosphere. Gas giants are believed to form by the accretion of hydrogen-helium gas around an initial protocore of rock and ice. Because of the pressure in their center, which is the result of their big gravitational momentum, their core is probably molten. It has been calculated that the core of our gas giants contains about 3 to 15 % of the mass of each planet [Althaus, 2004]. The gas giant planets are developing strong magnetic fields. The mechanism of the production of the magnetic field thought is deferent from that in terrestrial planets. It is believed that the dynamo effect of the metallic hydrogen or the ammonia-water seas is responsible for it.There are two sub categories of the gas giants.

In the first category there are planets with mass between the planets which mass waves between 10 and 50 $M_\oplus$. These planets are called Neptunian or Neptunes. They have a molten core and around it there is the mantle which is consisted of ammonia, methane and water oceans. Above the mantle there is the atmosphere which is the initial one and consists mostly of Hydrogen and Hellion. In Neptunians's atmospheres there are also layers of methane. This is the reason that Neptune and Uranus are blue, as the methane absorbs the reddish wavelengths. Finally there is an outer hazy atmosphere. A strange phenomenon that is taking place in both Neptune and Uranus is that the central axis of their magnetic field is significantly tilted, in respect of the rotational axis of the planet, and it is not crosses the center of the planet but there is an offset of $\sim$1/3 R_{planet}.

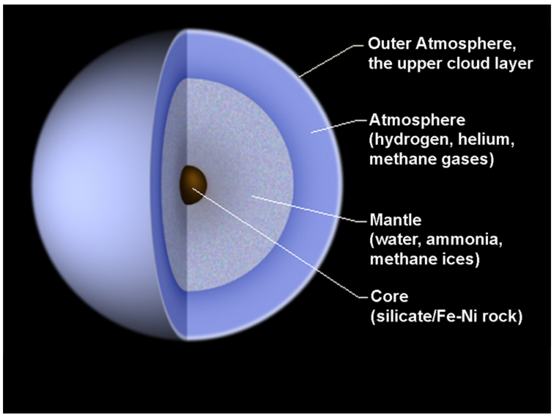

Figure 1.5: The layers of planet Uranus

The second category, the Jovians or Jupiters, contains all the planets with mass over 50 $M_\oplus$ and under the limit of 13M_J. They also have a molten, rocky core in their center but there is one difference. For the planets with the same or bigger the mass of Jupiter, it seems that the rocky core is dissolved, because of the enormous pressure, and rearranged in their mantle. The mantle is consisted of metallic hydrogen. In this state the hydrogen is liquid and reacts like a metal.The nuclei of the atoms are getting very close to each other (it is becoming comparable to the de Broglie wavelength of the electrons) and the electrons are unbound and behave like conduction electrons. Over the mantle there is the main atmosphere which is con-

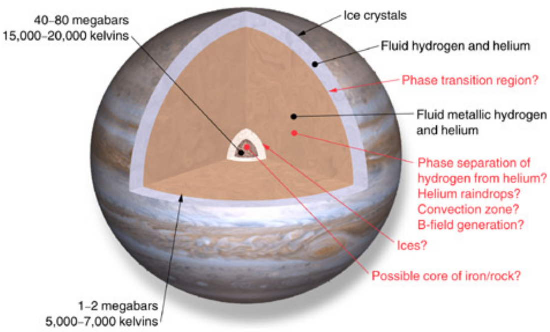

Figure 1.6: The layers of planet Uranus

sisted by Hydrogen and Helium and the outer atmosphere where there are visible clouds of water ammonia and other components which are giving the yellow-brown tones in Jupiter and Saturn.

The most acceptable scenario for the creation of a planetary system is that the protostar is accreted by the cloud of dust and gases. The perturbations inside the protostellar disk are resulting the creation of planetesimals and therefor planets. In distances near the young star the stellar wind is cleaning the region of the gases so the result is the creation of small terrestrial planets. In bigger distances of the star are forming the giants. After their creation the planets are migrating inwards the planetary system [Gomes et al., 2005]. For the terrestrial planets the migration is caused by a slight imbalance of the gravity field created by the spiral density waves of the protostellar disk. When the gravitational force exerted by the outer disk is greater than the one created by the inner disk, the planet looses angular momentum and drifts inwards. However, the migration timescale is very short and that is the reason why people seek for a mechanism able to slow down (or even stop) the migration. Several mechanisms have been proposed like the effect of the magnetic field, tri-dimensional effects, torques at co-rotation (where the planet rotates at the same speed as the disk parti-

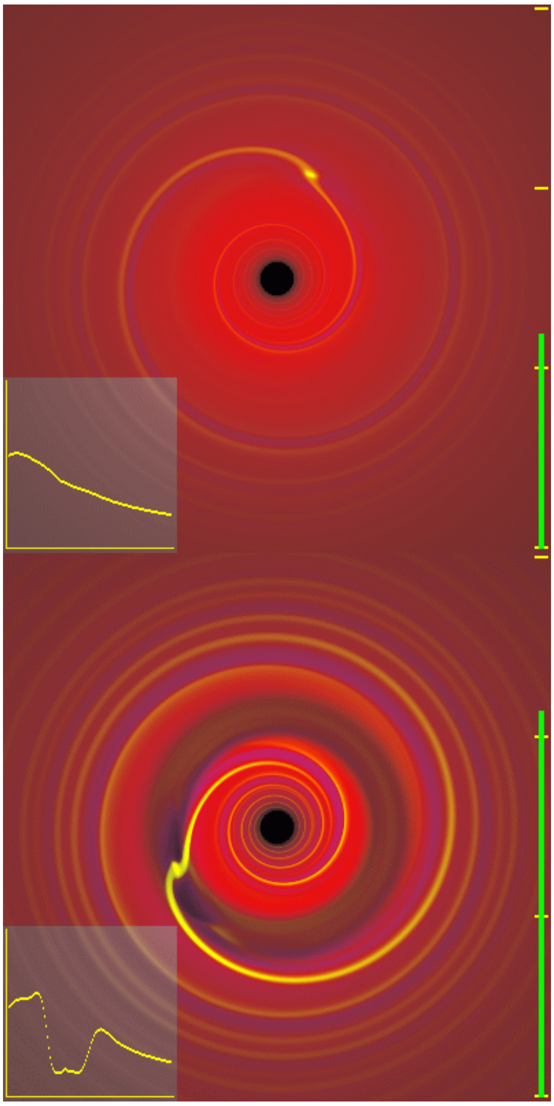

Figure 1.7: Analytical models for planetary creation and migration for
top: low mass planets, type I migration
bottom: high mass planets, gap formation and type II migration
Image Credit: Phil Armitage

cles), eccentricity effects, interaction between several planets, and so on. This is the type I migration. If the planet exceeds 10 $M_{\oplus}$ then it cleans relatively soon its orbit of other planetesimals and stops the type I migration. The formated gap however, continues to get filled by material, moving the planet and gap inward in the accretion disk. This is one hypothesis for how some or most ”hot Jupiters” form [Chambers, 2007].

As their name propagates, hot Jupiters, are planets with the same or bigger mass of the Jupiter. There is a significant difference though. While Jupiter orbits our sun in $\sim$ 5.2 AU, these planets are orbiting their host star in a distance between 0.015 and 0.5 AU. This unique characteristic qualifies them as the perfect candidates for RV and transiting surveys (also see 1.2.1-1.2.2). Also, because of the proximity to their host star, they develop circular orbits and they are tidally locked (always presents the same face to their parent stars). Due to high levels of irradiation they are of a lower density than they would otherwise be. This has implications for radius determination because, due to limb darkening of the planet against its background star, during a transit, the planet's ingress and egress boundaries are harder to determine. As these planets spend their life so close to their host star, the solar wind is striping them from their outer atmospheric layers. It is calculated that a Jupiters mass planet, orbiting at 0.02 AU, may louse 5 to 7 % of its mass during its lifetime because of that. Although, if a planet is closer than 0.015 AU to its parent star, the atmosphere and mantle will probably evaporated leaving only the core. The result is a body called chthonian planet [Hébrard et al., 2004].

Other planetary types and classifications

The properties of the new discovered planetary systems are different than our's, as their creation depends on a lot, deferent factors. So it was necessary the creation of new classification systems which would contain all new characteristics and planet kinds.

The *Sudarsky extrasolar planet classification* system is a theoretical model-based classification system for predicting the atmospheric composition of extrasolar gas giant planets based on their temperature. As the observations of extrasolar planets are not allow the extraction of good established conclusions about their atmospheres, it is not widely used yet. There are five types:

- Class I: Planets in this class have appearances dominated by ammonia clouds. These planets are found in the outer regions of a planetary system as they exist at temperatures less than 150 K. The predicted Bond albedo[1] of a class I planet around a star like the Sun is 0.57
- Class II: Planets in class II are too warm to form ammonia clouds: instead their clouds are made up of water vapor. This type of planet is expected for planets with temperatures below around 250 K. Water clouds are more reflective than ammonia clouds, and the predicted Bond albedo of a class II planet around a sunlike star is 0.81.
- Class III: Planets with equilibrium temperatures between about 350 and 800 K do not form global cloud cover, as they lack suitable chemicals in the atmosphere.These planets would appear as featureless blue globes because of Rayleigh scattering and absorption by methane in their atmospheres. Because of the lack of a reflective cloud layer, the Bond albedo is low, around 0.12 for a class III planet around a sunlike star
- Class IV: Above 900 K, carbon monoxide becomes the dominant carbon-carrying molecule in the planet's atmosphere (rather than methane). Furthermore, the abundance of alkali metals, such as sodium substantially increase, and spectral lines of sodium and potassium are predicted to be

[1]The fraction of the total electromagnetic radiation falling on a non-luminous spherical body that is reflected from the body in all directions; also called bolometric or spherical albedo

prominent in the planet's spectrum. These planets form cloud decks of silicates and iron deep in their atmospheres, but this is not predicted to affect the spectrum of the planet. The Bond albedo of a class IV planet around a sunlike star is predicted to be very low, at 0.03 because of the strong absorption by alkali metals.

- Class V: On the very hot gas giants, with temperatures above 1400 K, the silicate and iron cloud decks are predicted to lie high up in the atmosphere. The predicted Bond albedo of a class V planet around a sunlike star is 0.55, thanks to reflection by the cloud decks. At such temperatures, the planet may glow red from thermal radiation.

The main sequence stars are not the only celestial bodies that have been found to be orbited by planets. With the pulsar timing method have been detected *pulsar planets* orbiting around pulsars. The origin of these planets is not fully understood yet, as the supernovae explosion which created the pulsar should have destroyed the planets around it. One possible scenario is that the planets were formed out of the debris of a destroyed companion star that used to orbit the pulsar. An other one is that the planet is the remaining of a companion of the pulsar, which was almost entirely blasted away by the extreme irradiation from the nearby pulsar.

Almost every planet, which existence is known, belongs to our Milky Way galaxy. There are though three of them which don't. *The extragalactic planets*, as they called, discovered with the only method capable to detect so distant bodies, the microlensing method (see 1.2.3).Their signals however are not repeatable observations, as they are one time chance alignments.

Figure 1.8: An artist's impretion for a rogue planet

An other kind of planets is the free-floating or nomad planets. These are objects which have equivalent mass to a planet but are not gravitationally bound to any star, brown dwarf or other such object, and therefore they orbit the galaxy independently. There are two possible scenarios for the creation of those objects. The first one is that a planetary-mass object formed like a star via gas-cloud collapse, but without sufficient mass to start deuterium fusion like a brown dwarf. These bodies are called *sub-brown dwarfs*. They may form on their own and hence are free-floating. The second case is that they have been ejected from their planetary systems. In that case the planet is called *rogue planet*. There are suggestions that there may be 10^5 compact objects in the mass range 10^{-8} to $10^{-2} M_\odot$ per main sequence star that are unbound to a host star in the Galaxy [Strigari et al., 2012].

1.2 Detection Methods

For someone who understands the capabilities of the telescopes in our days (year 2012), it is obvious that the small non radiate planets is almost impossible to be seen directly. Therefor a variety of methods invented in order to detect them.

1.2.1 Radial Velocity

The radial velocity method is one of the oldest and the most successful survey method from earth. Radial velocity is the velocity of an object in the direction of the line of sight. The main detection mechanism for extrasolar planets is the spectroscopic radial velocity. If there is a planet of mass m around a star of mass M then the system star-planet will rotate around the center of mass of the system. The light from the star with the substantial relative radial velocity, will be subject to the Doppler effect, so the frequency of the light decreases for objects that were receding (redshift) and increases for objects that were approaching (blueshift) as described by the Doppler's law

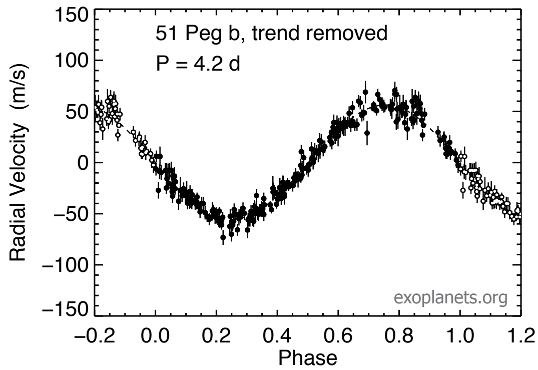

Figure 1.9: The RV diagram of the 51 Peg b. The filled dots are present a full period. (**image credit**: exoplanets.org)

$$f = \left(\frac{c + u_s}{c + u_r} \right) \cdot f_o \tag{1.1}$$

where c is the speed of light, u_s the radial velocity component of the star, u_r the relative speed of the observer (Earth) and f_o is the original frequency. The radial velocity of a star can be measured accurately by taking a high-resolution spectrum and comparing the measured wavelengths of known spectral lines to wavelengths from laboratory measurements. The detection of an extrasolar planet with this method involves taking precise measurements of a stars spectrum with an optical telescope.Each measurement is associated with a specific time. By solving the equation 1.1 as of u_s a plot can be created showing the star's radial velocity as a function of time (Figure 1.10). The result is a sine like curve. The radial velocity semi-amplitude K of this curve is related with the period P of the planet, the inclination i and the eccentricity e of its orbit with the equation:

$$K = \left(\frac{2\pi G}{P}\right)^{1/3} \frac{m \cdot \sin i}{M^{2/3}} \frac{1}{\sqrt{1-e^2}} \tag{1.2}$$

From the radial velocity diagram the K semi-amplitude and the period P can be easily measured. By combining these values and the equation 1.3 the best fitting model is extracted using Monte Carlo simulation, in order to estimate the minimum mass of the planet. It is only possible to estimate the true mass if the inclination of the system is known.

$$m_{true} = \frac{m_{min}}{\sin i} \tag{1.3}$$

One of the great advantages of the spectroscopic radial velocity is that it can find a wide range of planets with few limitations from the geometry of the system. More precisely, this method is capable to find planets with mass as low as a super-earth and it is not applicable only in systems with small values of inclination. One big disadvantage is the need for high precision spectrographs as the radial velocity of a star, which is caused to a planet, is smaller than 1300 m/s. Also the use of big telescope is mandatory in order to achieve the appropriate signal to noise ratio. An other disadvantage is that it is impossible, by this method alone, to determine the real mass of the planet, but the minimum possible one.

An other use of the motion of the star-planet system around the center of the mass is the Astrometric radial velocity which is the radial velocity as determined by astrometric observations. f precisely measuring a star's position in the sky and observing how that position changes over time. It is one of the oldest methods used to find extrasolar planets. Although because of the atmospheric and systematic distortions it is impossible to reach the accuracy that it is needed in order to successfully find as small body as planets. Only space telescopes as HST are able to be so accurate but until now it can only make follow up observations of already known systems. An example is the case of Gliese 876 b [Althaus, 2004].

1.2.2 Transits

The second most successful extrasolar planets detection method is the method of transits. As a planet passes in front of its parent star disk, the observed visual luminosity of the star drops by a small amount. The amount the star dims depends on the relative sizes of the star and the planet.

$$\frac{\Delta L}{L_{star}} = \left(\frac{R_{planet}}{R_{star}}\right)^2 \tag{1.4}$$

The luminosity drop, for a Gas Giant, is less than 2%.

One big advantage of the transit detection method is determination of the planetary radius and, in combination with radial velocity measurements, of the real planetary mass. This in turn constrains the planet's density. As the planet passes in front of its parent star, absorption of starlight passing through the planet's atmosphere during transit, revealing the composition and scale height of the planet atmosphere [Charbonneau et al., 2002]. Modulation of the combined light of the system during secondary eclipse, the pass of the planet behind the star, provides a direct detection of the planet's emergent spectrum [Deming et al., 2005]. Since the planet's emergent radiation peaks in the infrared (IR) spectral region, secondary eclipse measurements are primarily focused on the IR. The different geometries of transit and secondary eclipse allow localization of atmospheric knowl-

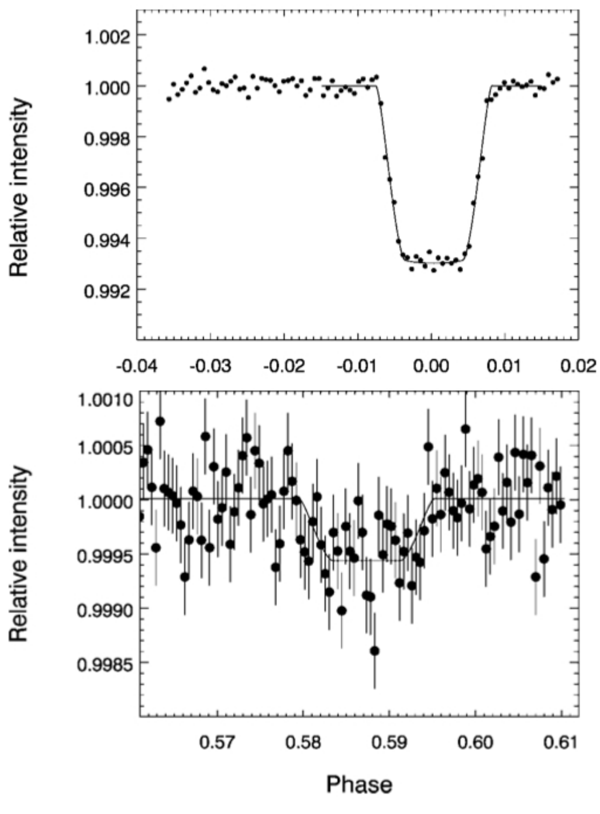

Figure 1.10: The transit and secondary eclipse of GJ 436b using the SPITZER space telescope [Deming et al., 2007]

edge: transit spectroscopy probes the atmospheric interface between the day and night hemispheres of a tidally-locked planet , whereas secondary eclipse measurements probe the emergent spectrum of the dayside [Knutson et al., 2008]. Moreover, measurements of transiting systems have been extended well beyond the times of transit and eclipse, to include observations in the combined light of star and planet at a large range of orbital phases. In instances where the planet's rotation is tidally locked to its orbit, these measurements can be inverted to yield the distribution of emergent intensity versus longitude on the planet. The observational techniques used for transiting systems can also be extended to non-transiting systems, so it is valuable to consider a generalization of the transit technique, namely exoplanet characterization in combined light. Without a transit, the planet radius cannot be measured directly, and that is a significant limitation. Nevertheless, much can be learned, for example from observing fluctuations in IR intensity that are phased to the planet's known radial velocity orbit [Harrington et al., 2006].

A very accurate analytical model for a planetary transit is described by [Pál, 2008]. As described in this paper the relative apparent flux of an eclipsed star with a quadratic limb darkening can be written as f = 1 - Δf (assuming a unity flux out of the transit), where flux decrease Δf can be calculated using the equation

$$\Delta f = W_o F_o + W_2 F_2 + W_1[F_1 + F_K K(k) + F_E E(k) + F_\Pi \Pi(n, k)]. \qquad (1.5)$$

In this equation the quantities Wi (i=0,1,2) are functions of the limb darkening[1] coefficients. In equation 1.5 the terms $F_o, F_1, F_K, F_E, F_\Pi$ and F_2 are functions of the occultation geometry, which are the relative planetary radius $p = R_p/R_*$ and the normalized projected distance z between the center of the star and the center of the planet. The functions K, E and Π denote the complete elliptic integrals of the first, second and third kind, respectively. The variation in the occultation geometry yields 12 distinct cases of obscuration, which are represented in Figure 1.11 and summarized in Table 1.2. The subscripted cases can only occur in case of planetary radius equal (T) or greater (G) than 1/2 of

[1]The phenomenon which refers to the diminishing of intensity of a stellar disk as one moves from the center to the edge or "limb"

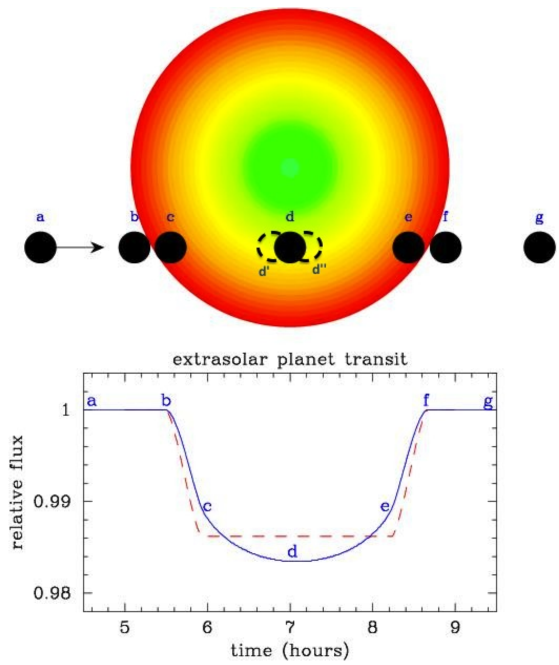

Figure 1.11: The diagram of a planetary transit, with the different cases of the obscuration noticed.

the star radius, except case A_G where the planet radius is equal or bigger than star's. In that case the planet is not supposed to be a planet but another star. The model of a real transit and the determination of its physical parameters is possible by using the eq. 1.5 with the appropriate values for $F_i(i = 0, 1, K, E, \Pi \ \& \ 2)$, for $W_j(j = 0, 1, 2)$ and for the functions K, E and Π for each step, using the Levenberg-Marquardt algorithm.

Relation	Case	Figure
z=0 & p < 1	A	d
z ≤ p - 1	A_G	-
z < p & z < 1 - p	B	d' to d"
z < p & z = 1 - p	B_T	d' to d"
z < p	B_G	d' to d"
z = p & z < 1 - p	C	d' & d"
z = p = 1/2	C_T	d' & d"
z = p	C_G	d' & d"
z < 1 - p	D	c to e
z = 1 - p	E	c & e
z < 1 + p	F	b to f
no shade	G	a to b & f to g

Table 1.2: The steps of different occultation geometries

1.2.3 Gravitational Microlensing

The gravitational microlensing method occurs when there are random alignments between background source stars and foreground stars, which may host planet systems. These background source stars are acting as sources of light that are used to probe the gravitational field of the foreground stars. The relative motion of the source star and lens system allows the light rays from the source to sample different paths through the gravitational field of the foreground system. If a there are planets around the lens stars they are changing total gravitational lens magnification of the source star, by creating caustic lines figure 1.12 .With time that provides the observable gravitational microlensing signal.

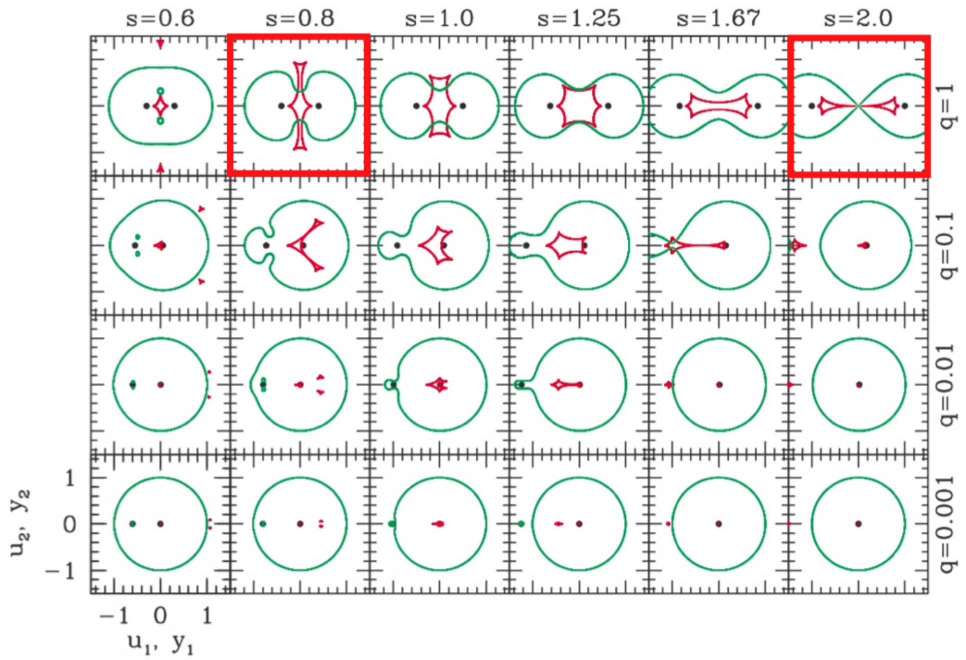

Figure 1.12: Caustics shapes for different mass ratios (q) and seperations (s) between the star and the planet

In 1991, astronomers Shude Mao and Bohdan Paczyński of Princeton University first proposed using gravitational microlensing to look for exoplanets. Successes with the method date back to 2002, when a group of Polish astronomers (Andrzej Udalski, Marcin Kubiak and Michal Szymański from Warsaw, and Bohdan Paczyński) during project OGLE (the Optical Gravitational Lensing Experiment) developed a workable technique.

During one month they found several possible planets, though limitations in the observations prevented clear confirmation. Since then, four confirmed extrasolar planets have been detected using microlensing. As of 2006 this was the only method capable of detecting planets of Earthlike mass around ordinary main-sequence stars.

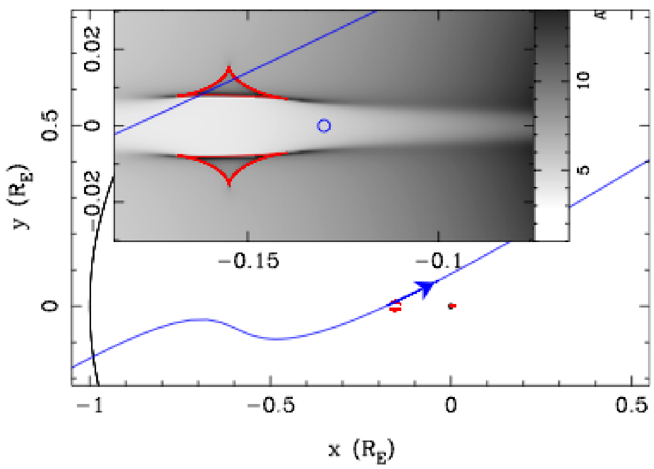

Figure 1.13: The caustics (red shapes) and the path of the lens (blue line) for the event of GLE-2007-BLG-368 [Sumi et al., 2010]

A notable disadvantage of the method is that the lensing cannot be repeated because the chance alignment never occurs again. Also, the detected planets will tend to be several kiloparsecs away, so follow-up observations with other methods are usually impossible. However, if enough background stars can be observed with enough accuracy then the method should eventually reveal how common earth-like planets are in the galaxy.

The microlensing method have unique respects also among extrasolar planet detection methods [Bennett, 2008]. The amplitude of planetary microlensing signals is large (typically $\sim 10\%$) and is almost independent of the planetary mass. Instead, the source-lens alignment necessary to give a detectable planetary signal depends on the planet-star mass ratio q and so the probability of a detectable planetary signal scales as q. This scaling of the probability of planet detection with the mass ratio, q, is shallower than the sensitivity curves for other methods, so microlensing is more sensitive to low-mass planets than other methods that are sensitive to planetary mass as RV. The sensitivity of the microlensing planet search method extends down to $0.1M_{\oplus}$. Microlensing is most sensitive to planets at orbital separations between 1.5-4AU, which corresponds to the vicinity of the Einstein ring radius. This range of separations also corresponds to the ?snow line? where planet formation is most efficient according to the leading core accretion model of planet formation. Thus, microlensing complements the

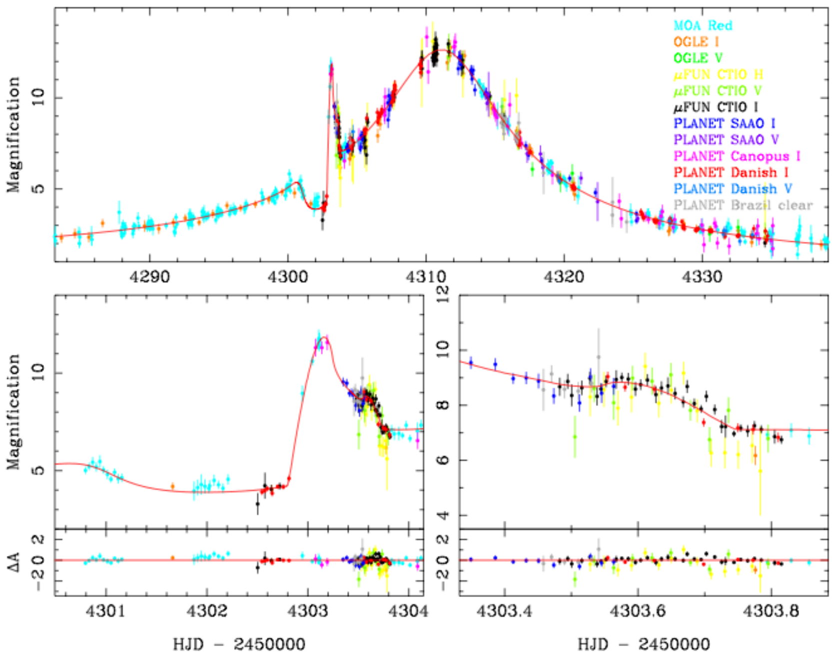

Figure 1.14: The synthetic lightcurve with the fitting model for the event of GLE-2007-BLG-368 and zoom on the planetary event[Sumi et al., 2010]

RV and transit methods, which are most sensitive to planets in very short period orbits. Microlensing is the only planet detection method that is sensitive to old, free-floating planets. Theory predicts that such planets may be quite common, and ground-based microlensing can detect free-floating gas giant planets, while a space-based survey is needed to detect free-floating terrestrial planets. Since the microlensing method doesnt rely upon light from the host star in order to detect its planets, it can detect planets orbiting unseen stars. This can make it difficult to determine the properties of the host stars, but space-based follow-up observations can detect the host stars for most planets discovered by microlensing. A space-based microlensing survey would provide a nearly complete statistical census of extrasolar planets with masses down to $0.1\mathrm{M}_\oplus$ at all separations ≥ 0.5AU. This includes analogs of all the Solar Systems planets, except for Mercury. The only observable feature of a microlensing event is the time variation of the total magnification of all the lens images due to the motion of the lens with respect to the observer and source [Bennett and Rhie, 1996]. The characteristic transverse scale for a lens of mass M is given by the Einstein ring radius which is the radius of the ring image obtained when the source, lens, and observer are colinear. It is given by

$$R_E = 2\sqrt{\frac{GMD}{c^2}} = 4.03\ AU\ \sqrt{\left(\frac{M}{M_\odot}\right)\left(\frac{D}{2\ kpc}\right)} \tag{1.6}$$

where D is the reduced distance defined by $1/D = 1/D_{ol} + 1/D_{ls}$. D_{ol} and D_{ls} are the distances from the observer to the lens and from the lens to the source, respectively. For a point mass lens, the amplification of a microlensing event is given by

$$A = \frac{u^2+2}{u\sqrt{u^2+4}}; u = \sqrt{u_{min}^2 + \left[\frac{2(t-t_0)}{\hat{t}}\right]^2} \tag{1.7}$$

where u is the separation of the lens from the source- observer line of sight in units of R_E and t_0 and $\hat{t}$ refer to the time of peak amplification and the Einstein diameter crossing time, respectively. In a planetary lensing event, the majority of the light curve is described by equation 1.7, but in the region of the planetary deviation we must consider the binary lens case [Rhie and Bennett, 1996].
If u and z denote (in complex coordinates) the source and image positions in the lens plane, the binary lens equation is given by

$$\omega = z - \frac{1-\epsilon}{\bar{z} - \bar{x}_s} - \frac{\epsilon}{\bar{z} - \bar{x}_p} \tag{1.8}$$

where ϵ is the fractional mass of the planet, and x_s and x_p are the positions of the star and planet, respectively. We work in units of the Einstein radius, R_E, of the total mass M. Equation 1.8 has three or five solutions (z) for a given source location, u. The Jacobian determinant of the lens mapping (eq. 1.8) is

$$J = 1 - |\partial_z \bar{\omega}|^2\ ;\ \ \partial_z \bar{\omega} = \frac{1-\epsilon}{(z-x_s)^2} + \frac{\epsilon}{(z-x_p)^2} \tag{1.9}$$

and the total magnification of a point source is obtained by summing up the absolute value of the inverse Jacobian determinant calculated at each image :

$$A = \sum_i |J_i|^{-1} \tag{1.10}$$

The curve defined by J = 0 is known as the critical curve, and the lens mapping (eq. 1.8) transforms the critical curve to the caustic curve in the source plane. By equation (5), a point source that lies on a caustic will have an infinite magnification. The singularity at J = 0 is integrable, so finite sources always have finite magnifications. When the source star is in the region of the caustic curve, the magnification will dither noticeably from the single lens case, allowing a planetary signal to be detected. If the planet mass is of order 1 to 10 $M_{\oplus}$, the total extent of the caustic curves are comparable to the size of the source star, and the point source approximation is not appropriate.

1.2.4 Direct Imaging

The extrasolar planets are extremely faint light sources compared to stars. Because of that the small amounts of light that comes from them is lost in the glare from their parent star. So in general, it is very difficult to detect them directly.

The first step in order to achieve such a difficult detection is the increment of the contrast between the star and the planet. For most of star-planet systems this is impossible. In low temperature stars or in stars that own a circumstellar disk, this is possible if infrared filters are used. Also the use of a coronograph is needed. The planet-to-star flux ratio detectable with a direct imaging high-order adaptive optics coronagraphic system operating at visible or near-infrared wavelengths depends strongly on the atmospheric coherence length and coherence time. Direct planet imaging in the mid-infrared is limited by atmospheric emission rather than site turbulence characteristics. Although large telescopes or interferometers are required to directly image exoplanets in the mid-infrared. The simple models used to predict the signal to noise ratio in different bands, explore the fundamental limitations to direct imaging of exoplanets arising from photon noise alone. The speckle noise component of the uncorrected stellar halo will be significantly larger than the photon noise component. While techniques have been proposed to completely remove speckle noise, it is currently unknown to what extent they will be successful. Any additional contribution to the SNR from speckle noise, however, will be strongly dependent on the atmospheric characteristics.

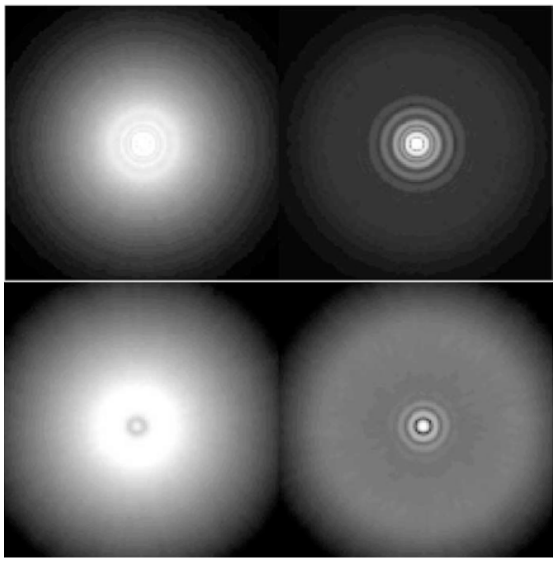

Figure 1.15: top: PSF simulations for low (left) and hi (right) effisiency adaptive optics in H-band using 1000 independent atmospheric phase screen realizations with cell size one 15th of the telescope diameter (3.6-m)
bottom: the same with coronograph [Oppenheimer et al., 2003]

An other very important factor is the maximum resolving power of the telescope. The resolution power of a telescope is given by the relationship

$$A = \frac{12''}{D/1\ cm} \tag{1.11}$$

The angular separation of a star-planet system is given by

$$\tan\theta = \frac{r}{d} \Rightarrow \theta \sim \frac{r}{d} \tag{1.12}$$

where r is the projection of the distance between the star and the planet and d is the distance to the system. If one do the math will find out that the angular separation of a planet, in distance of 1 AU from its star, and 100 Ly away is 0.03". That kind of resolution is achievable from a telescope with diameter larger than 4 m. We mast consider thought that the earth's atmosphere is limiting the threshold of the best angular resolution to 0.5"-1.2" depending on the astronomical seeing of the telescope's construction area. After the development of the adaptive optics systems this limitation is almost overpassed. Thus, even the largest telescopes based on earth, are not capable to see planets which are closer than some tenths of AU to their host star.

Despite those limitations, the direct imaging of an extrasolar planet comes with a grate benefits. Direct detection and direct spectroscopy, have great potential for advancing the understanding of extrasolar planets. In combinations with other methods of planet detection, direct imaging and spectroscopy can allow to eventually: 1) fully map out the architecture of typical planetary systems and 2) study the physical properties of exoplanets (colors, temperatures, etc.) in depth.

1.2.5 Timing variations

Pulsar Timing

The pulsars or neutron stars are the final stage of the life of a main sequence star with mass between 2 and 8 $M_{\odot}$ and because of their physiology they carry very strong magnetic fields. Radiation produced by the neutron star is focused, into two oppositely-directed beams, by the magnetic field. As the star rotates, the beam is swept across the sky and if the beam intercepts the Earth once per rotation, then brief but regular pulses of radiation are seen, much like a lighthouse.

When a planet is introduced, as described in 1.2.1, the pulsar is forced to move around the center of mass of the system. In the case of a pulsar and a planet, the center of mass will lie very close to the pulsar, since it is much heavier than the planet. Therefore, during one orbit the pulsar will move a much lesser distance than the planet. However, even thought the pulsar's 'wobble' is small, it has an effect on the timing of the pulses emitted by it. When the pulsar is moving away from the Earth, the time between each pulse becomes slightly longer; conversely, when the pulsar is moving toward the Earth, the time between pulses becomes slightly shorter. By measuring these periodic changes in pulse timing, it is possible not only to deduce the existence of a planet orbiting around the pulsar, but also to estimate the semi-major axis of the planet's orbit, and place a lower limit on the mass of the planet.

Transit Timing Variations

Transits for a particular extrasolar planet occur at very regular intervals. The examination of a planets transit timing variations (TTVs) it is theoretically capable to detect the effects of gravitational perturbations from other exoplanets in the same system. TTVs are nothing else than how early or late each transit occurs. The efficiency of this method is very high. As a third, forth or Nth body is introduced in the system, its gravitational field affects the motion of the other bodies of the system, resulting, mainly, irregularities in their orbital period. A noteworthy fact is that in 1846, planet Neptune was discovered after its existence was predicted because of discrepancies between calculations and data for the planet Uranus. Astronomers found the new planet almost exactly at the position predicted by their calculations. Great source for data, which are eligible for TTV is the CoRoT and KEPLER missions data.

The confirmation of the existence of an extra component in a planetary system is easy. While one have a lot of presicely timed planetary transits, can produce the O-C (observed-calculated) diagram. If there are more than two components in the system, the middle epoch of every transit will move around a mean epoch forming a sinusoidal curve (figure 3.1). The same technique can be applied to the minima O-C diagrams of the stellar binaries.

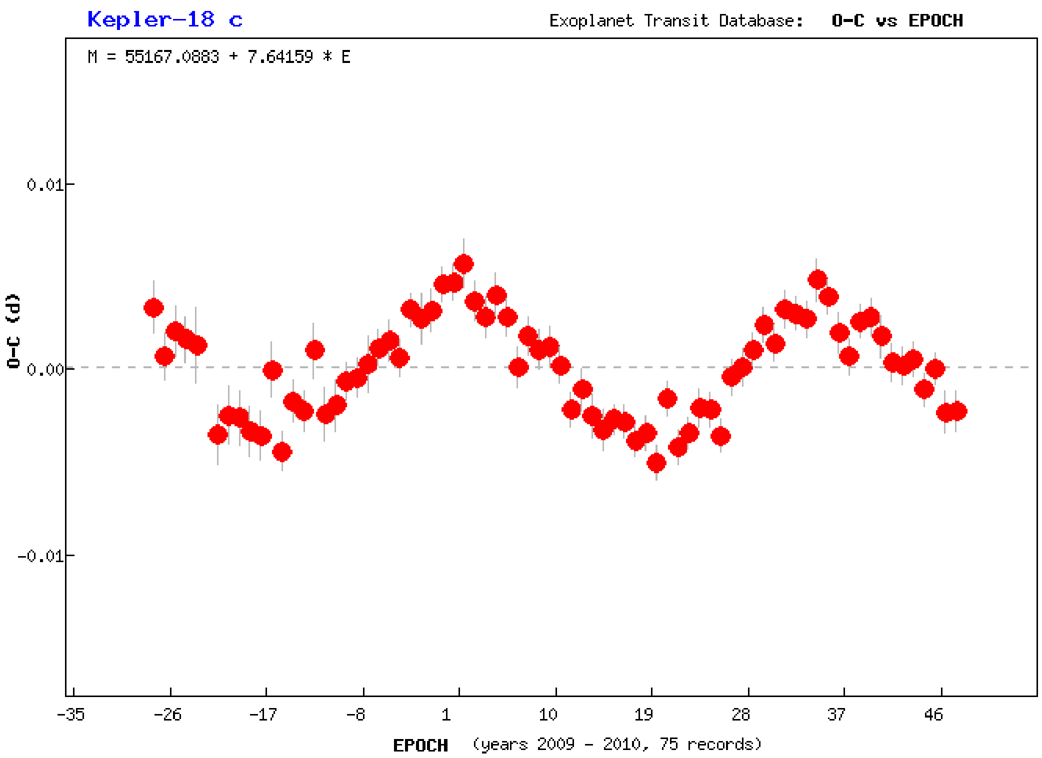

Figure 1.16: The O-C diagram of planet Kepler 18 c (**image credit**: ETD-Extrasolar Transit Database)

Chapter 2

Designing a wide field survey for transient planets

2.1 Preparation

2.1.1 Equipment

The main goal for every survey is to identify and monitor as many objects related to its subject, as possible. In order to achieve this target, it is very important the selection of the equipment to fit some specific requirements. For a wide field survey for transient planets these are:

- The achievement of the widest possible field of view, in order to increase the stars in the sample.
- As a transit of an extrasolar planet is detectable to light curves with less than 1% photometric accuracy, the noise should be reduced by using a cooled CCD with small readout noise.
- For the same reason the placement of the telescope should be in a place with low astronomical seeing and without disturbance of light sources nearby.

The coverage of those needs with commercial, low baget instruments, is not an easy task. For the ThReT survey a Takahashi E180 was selected as telescope and the Fingerlakes PL6303E was selected as the CCD camera.

The Takahashi E180 has diameter of 180 cm and F = 500 mm. In combination with the big CCD array size, 3072 x 2048 pix, a field of 3.5 x 2.5 degrees is achieved. Also the cooling system of the camera, which allows the CCD to be cooled to ΔT = -45 ^{o} C of the environmental temperature, along with the medium quantum efficiency (68%) and the low readout noise (9 e^{-}), making capable the detection of transient extrasolar planet's signal.

The position were the equipment is installed is at Mt. Holomon at Chalkidiki. This place is about 100 km away from Thessaloniki, which is making the light pollution minimum, with affection only to North-West sky, for heights less than 25^{o}. The mean atmospherical seeing of this region, which is measured with the ESO-DIMM method, calculated to be 0.74".

2.1.2 The ThReT pipeline

The second most important factor for a planetary survey is the pipeline which will be used in order to reduce, perform photometry, remove systematics and search for signals.

An older version of ThReT pipeline, which was developed by John Antoniadis, consisted the main body of the new ThReT pipeline which developed from me. In order to make it fast and with high compatibility, a combination of C-Shell and python scripts were used along with Starlink routines.

Starting the Pipeline

When the user run the pipeline for the first time, he is asked to select the raw data files, including the reduction frames. After that the program is testing if there was an older run for those data. If yes then he is capable to select to continue so there is no need to reenter the information needed for the program to run. If no older run existed, the program is separating the reduction frames

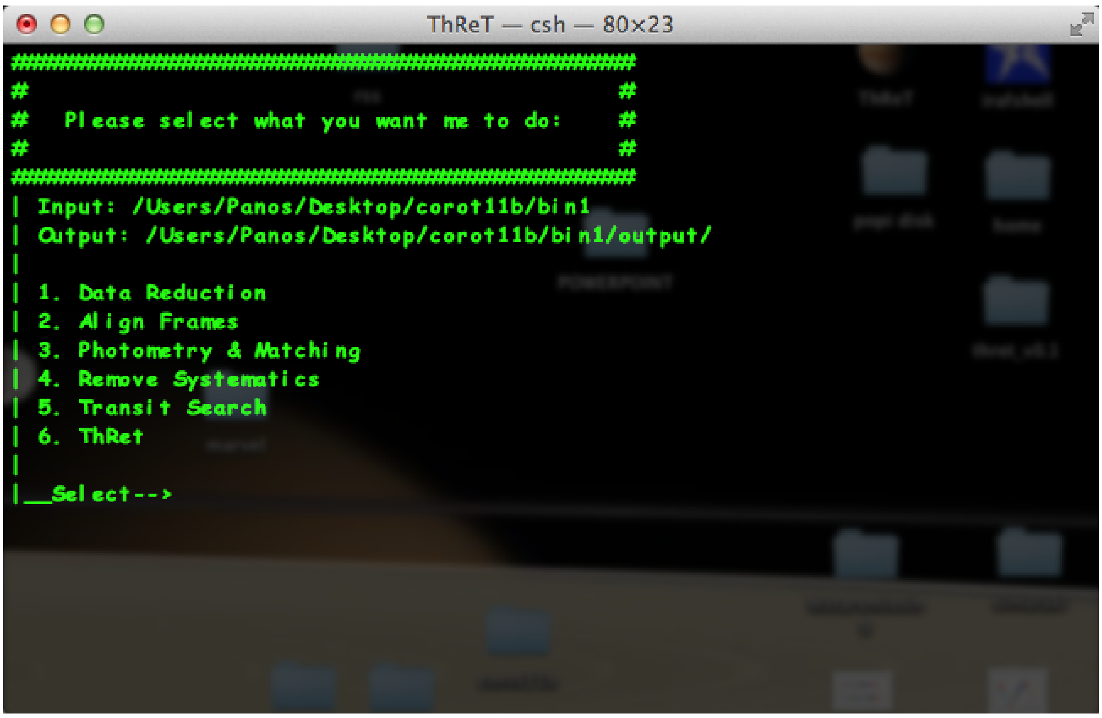

Figure 2.1: The ThReT pipeline main screen

from the raw data. If a reduction frame type is missing or it is incompatible (i.e. different dimensions) to the raw data then there is the option to select a different folder where there are the right reduction data or exit. After this procedure is done the program is presenting the number of raw and reduction data detected. Also it is asking for the readout noise and the gain of the CCD camera along with the RA and Dec of the target field. After that the main screen of the program can be seen (Fig 2.1). Inside the file with the raw data an output file is created in which will be placed all the output frames and files from now on.

Data Reduction

In order to clean the raw frames, the standard reduction method is been used. After the transformation of all the data to the NDF starlink format, the program is performing the masterbias and masterdark subtraction and after that the masterflat division from all the images.

Align Frames

This part of the code is can be used in oder to align the frames, if the data are suffering of severe shift of the field of view. As the matching algorithm (see next paragraph) is very strong, If the shift is not bigger than 5 pixels per frame it is not necessary.

Photometry & Matching

The photometry process is divided to four parts. In the five part, aperture photometry is performed by the Daophot 2 algorithm. A range of different apertures is used, depending to the aperture of the brighter star in the field. In the second part is performed the transformation of the time from Gregorian to Julian Date. In the third part the matching process is performed by using the match algorithm ([Valdes et al., 1995]). For that purpose a reference image should be elected. After matching, the lightcurves are created by using the best aperture. This is possible by checking for which one of the apertures, the lightcurve has the lowest mean internal rms, Z, which is given by the relationship

$$Z = \frac{1}{n-1}\sum_{i=0}^{n-1} m_{i+1} - m_i \tag{2.1}$$

where n is the number of measurements. The name of each lightcurve is containing the median of its points, with the x and y position on the reference frame. Finally the lightcurves are checked. If any of them has less than 50% of n they are rejected. Also, if any points are missing, they are filed with points with the value of the median of each curve, so in the end all the curves have the same number of points. In the same step some statistical values are computed for each lightcurve.

Remove systematics

This is one of the most critical stages of the analysis procedure. Because of the big field of view the lightcurves are suffer from various systematic errors. In order to remove those the DSTL code (by Dimitris Mislis) is used. The procedure of the errors identification is quote simple. The best much of a comparison, for a specific star is determined. After that a 2_{nd} order polynomial relationship is produced and removed, in order to remove any remaining variability.

Variability and Periodicities Search

This is the final stage of the pipeline. Here, every single lightcurve is checked in order to determined if it is contain any variabilities and if so, whether there are periodicities in them or not . The procedure is as follows. For every lightcurve we are calculating the mean internal rms Eq. 2.1 and the standard deviation of the magnitudes for each lightcurve by the relationship

$$s_m = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - \overline{m})^2} \tag{2.2}$$

where m$_i$ is each magnitude measurement $\overline{m}$ is the mean magnitude of the lightcurve and n is the number of measurements. The variability factor is calculated by the division of Eq. 2.2 & 2.1.

$$V_f = \frac{s_m}{Z} \tag{2.3}$$

After the end of this procedure the stars which have $V_f > 1$ are considered to have variabilities. For those stars a Lomb-Scragle [Scargle, 1982] periodogram is produced in order to determine if there are periodicities in their variabilities. As the L-S periodogram produces big spikes if a periodicity is found. The V_f over the maximum value of the produced periodogram is considered as periodicity factor. After that a catalog is produced with some candidates for eye investigation.

2.1.3 Target field selection

In order to select the target field, a transit probability map was created [Antoniadis et al., 2010], based to the ThReT instrumentation properties. The probability for the detection of a planetary transit is a combination of a given star to host an extrasolar planet, the planetary system geometry and the observing conditions.

The probability for a star to host a planet [Valenti and Fischer, 2005], can be derived by the relationship:

$$P_{\exists Planet}([Fe/H]_*) = 0.03 \times 10^{2[Fe/H]_*} \tag{2.4}$$

By theoretical reflections about arbitrary inclinations of the orbital plane with respect to the observers line of sight and including Keplers third law, the geometric transit probability can calculated by:

$$P_{geo} = 23.8\left(\frac{M_*}{M_\odot}\right)^{-1/3}\left(\frac{R_*}{R_\odot}\right)\left(\frac{P}{d}\right)^{-2/3} \tag{2.5}$$

[Gilliland et al., 2000].

The combination of eq. 2.4 and 2.5 is giving the total probability for a planetary transit to occur for each star

$$P_{tot} = P_{geo} \times P_{\exists Planet} \tag{2.6}$$

The application of eq. 2.6 on a full sky survey (in our case the Tycho catalog), is producing a probability map, from where the selection of the most promising regions can be made.

2.1.4 Software tools

Transit Lightcurve Generetor

In order to define what kind of planets, is the survey capable to detect, a simple transit generator code created (with the precious help of Dimitris Mislis). This script is the translation, in python language, of the transit analytical model as described by Pál [2008]. For the resulted lightcurve, it is possible to add random noise, with the desired rms.

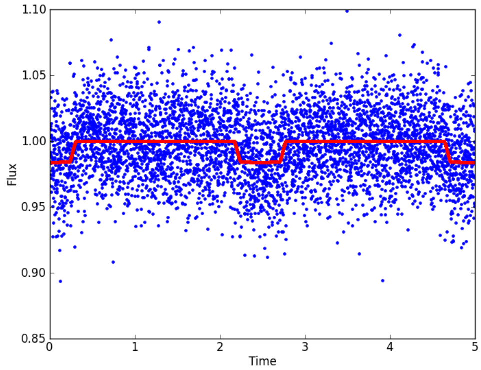

Figure 2.2: The predicted lightcurve of the Tres-2 star, as produced by transit generator, with log(rms) = -1.6, 90 seconds per frame and total time of 2 times its period.

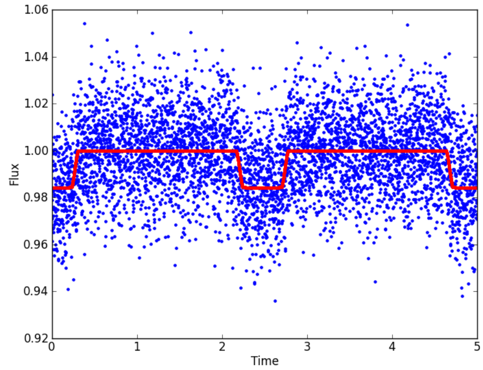

Figure 2.3: Same as figure 2.2, with log(rms) = -1.8

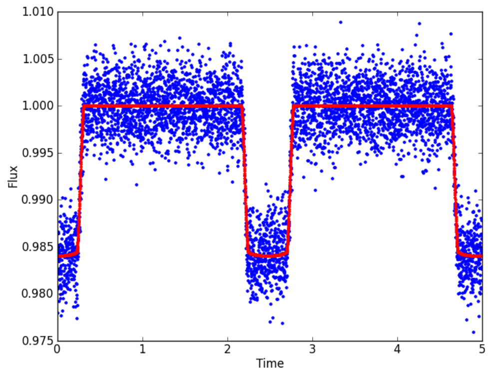

Figure 2.4: Same as figure 2.2, with log(rms) = -2.6 (about the same with CoRoT mission)

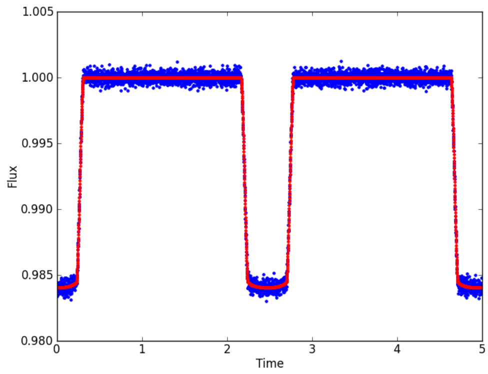

Figure 2.5: Same as figure 2.2, with log(rms) = -3.5 (about the same with KEPPLER mission)

Period Probability Detector

All the earth-based telescope surveys suffer from the day-night alternation. Also, depending on the position of the telescope, there may be night with clouds, precipitation, high humidity, high winds, dust and other phenomena which cause the observations not to happen. Because of those situations it is very hard or even not possible for a survey to detect the complete phase of some certain periods. In order to find out what kind of period is it possible for the survey to detect, a simple sort code created, which can predict the probability to detect the phases of a range of periods. The input is consisted of the observation times list in JD and the desired period range. In figures 2.6, 2.7 and 2.8 is presented the probability to detect periods between 0.3 and 4 days for a survey of 4 and 9 nights, with 90 sec frame exposures, for 6 and 9 hours per night.

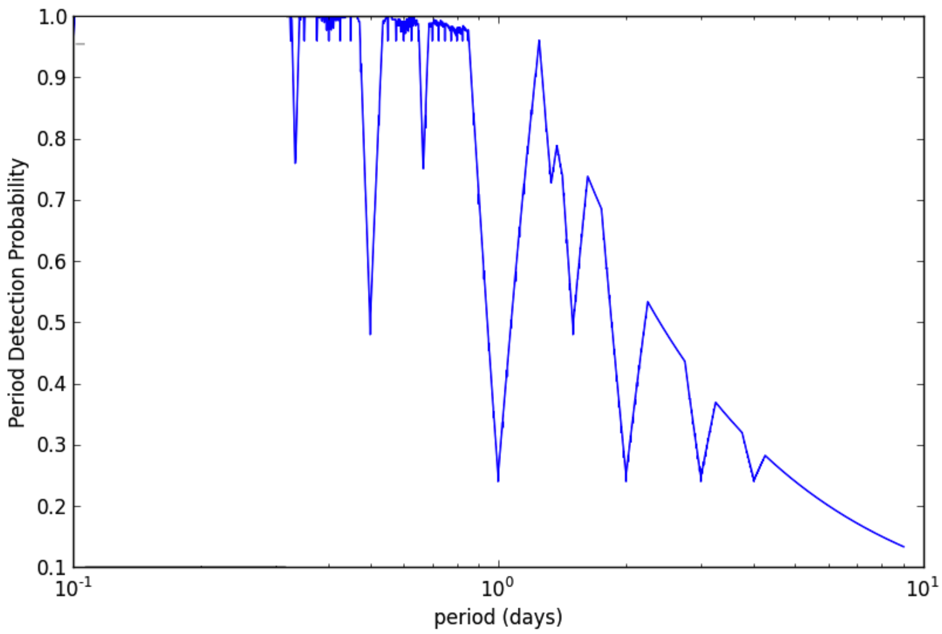

Figure 2.6: The period detection probability for a survey of 4 nights, 6 hours per night

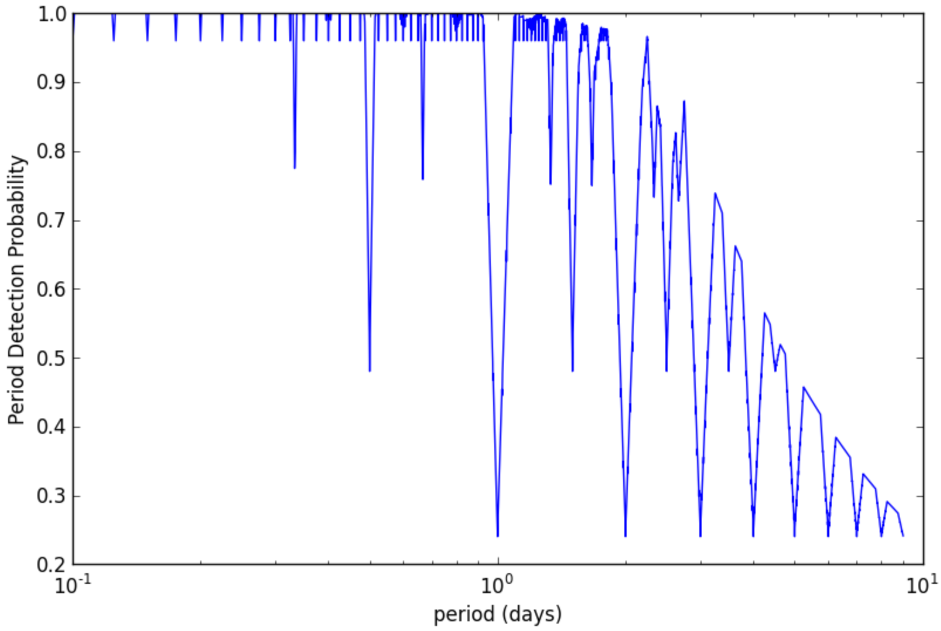

Figure 2.7: The period detection probability for a survey of 9 nights, 6 hours per night

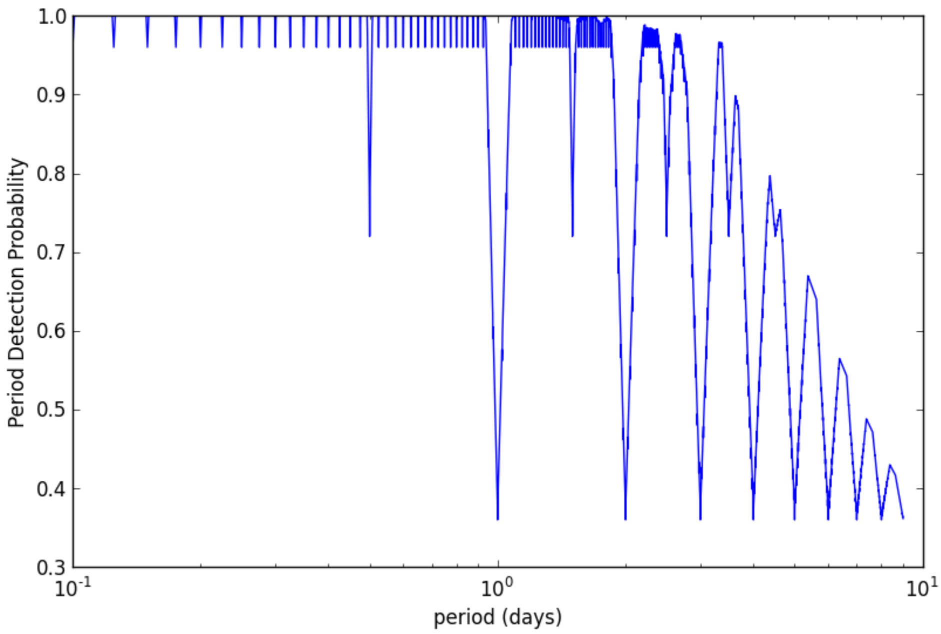

Figure 2.8: Same as figure 2.7 with 9 hours per night

2.2 Test phase

It is very important for a survey, before the actual start of the running section, to perform a test run. This test run should be performed in the same conditions as the survey will be. After the test run all the appropriate adjustments should be made in order to increase the performance of the system.

In order to put in a test the ThReT survey instruments and pipeline, the field of CoRoT-11b was selected. The test performed in dates when it was predicted by ephemeris that the planet would make transit. This is a planet who detected from the CoRoT mission. Its parent star is an F6V star with magnitude of 12.8 and the planet is causing a transit with depth of 0.00114 (table 2.1).

Star Name	CoToT-11
RA	18:42:44
Dec	+05:56:15
T_{eff}	6400 K
Epoch of Transit Center	2454597.679
Transit Duration (days)	0.1042
Transit Depth	0.0114
Orbital Period (days)	2.99433

Table 2.1: CoRoT-11b stellar and transit information

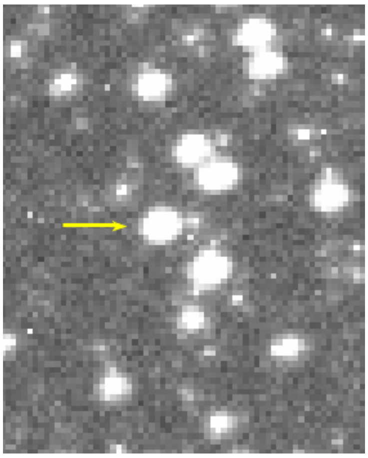

Figure 2.9: A close of CoRoT-11b.

The CoRoT-11b was selected because its transit is of medium depth and his period of 2.99 days allows two days of rest before the transit. The measurements were undertaken between the 3^{rd} and the 8^{th} of July 2011. The weather conditions were not identical because of high humidity during the nights. Also the transit of the planet it wasn't expected to be totally monitored, because of the early dawn that time of the year, which would make it even more challenging to get found of the pipeline. The transit of the CoRoT-11b was expected to occur at 5^{th} and 8^{th} of July.

A wide range of settings was already been tested for the specific instrumentation. The most promising one, which was finally used for the test, was consisted of 90 seconds of exposures, with a 60 seconds space between the captures for the read out. In table 2.2 is presented the full configuration.

Exp time	90 sec
Space between exp	60 sec
Filter	no
binning	1
Frame (pix)	3072 x 2048
deg per pix	0.001016
CCD & mount software	MaximDL

Table 2.2: The test phase settings

The results of the test run with those setting were more than promising. After the end of the observations, the data were processed by the ThReT pipeline. A total number of 28284 stars were detected with more than 4000 stars with photometric accuracy under 1.5% (Figure 2.10).

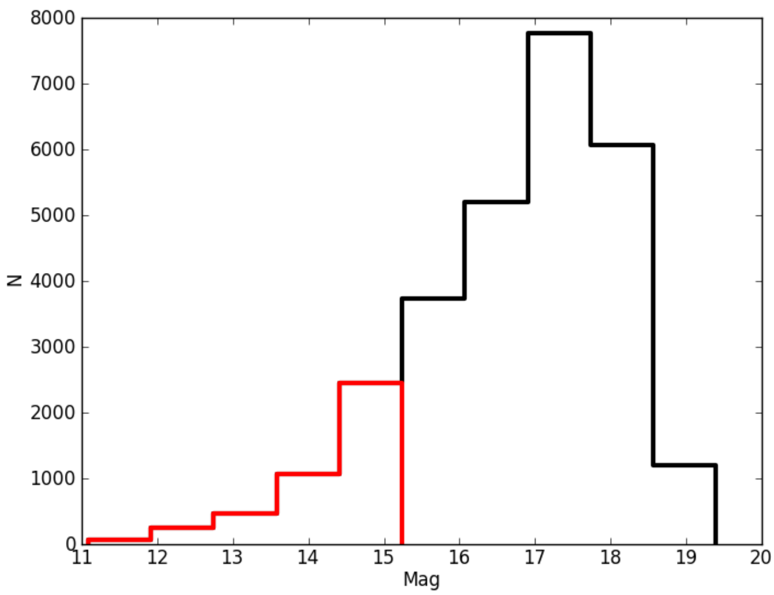

Figure 2.10: The sample of stars around CoRoT-11b with their instrumental magnitudes. The red region is consisted of the high accuracy (<1.5%) stars

The transit of the CoRoT-11b was clearly detected in both of the expected days Figures 2.11 & 2.12. This fact was grate by it self as the 8^{th} of July was the most humid of the observation days and the possibilities to detect the transit in this day's data were low. The best fitting models were calculated for the predicted duration and mid-transit epoch, leaving the depth as the only variable. The biggest success of the test, however, was the relatively high rank of the star, after the search for periodicities by the last part of the pipeline.

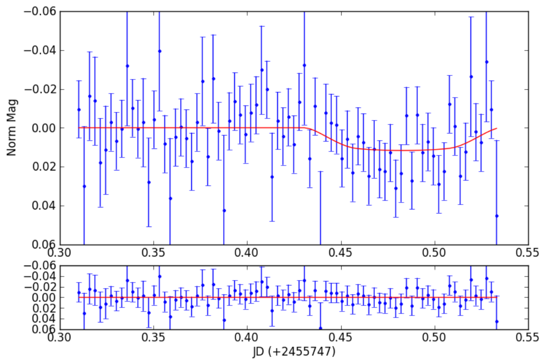

Figure 2.11: CoRoT-11b at 5^{th} of July transit

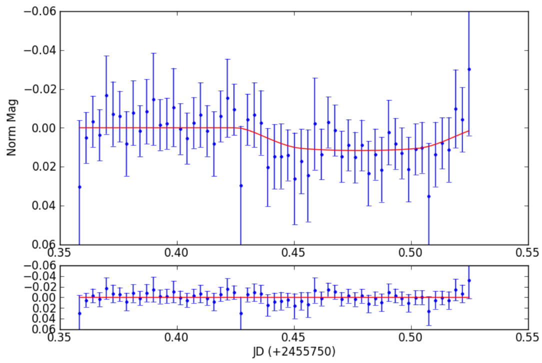

Figure 2.12: CoRoT-11b at 8^{th} of July transit.

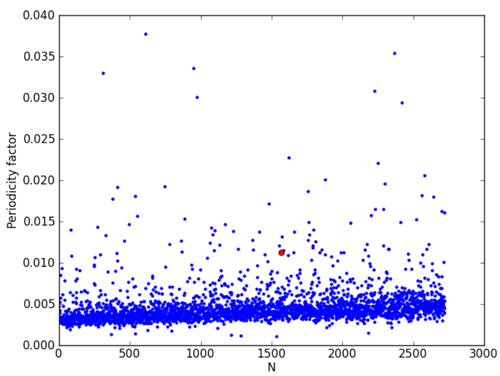

Figure 2.13: The periodicity results for the stars with the bigger variability. The CoRoT-11b star is presented with the red dot.

Chapter 3

The ThReT Survey

After the end of the testing phase, the ThReT survey was ready to run for the first time in a new, unexplored, target field. The settings configuration was exact the same, as in test phase (Table 2.2).

3.1 The first long term run

3.1.1 Observational Strategy

The first long term run of the ThReT survey started at 19^{th} of September and finished at 6^{th} of October. Unfortunately the weather conditions was not ideal. Because of that, only ten of the actual eightteen days are useful. The target field, which selected using the method described in 2.1.3, was the region around R.A. 00h 08m 00s and Dec 33^o 30' 00 in Andromeda constellation. For brevity this field will be referred as TEF2011 (ThReT ExoField 2011). This field was also selected because of its convenient position in respect to the galactic plane. It is close to it, so it contains a lot of stars in it but it is also not overcrowded.

Each night of observations was started with the flat fields capture in the dusk sky twilight. At 17:00 UT was scheduled the start for the observations, while the 01:00 UT was scheduled to end. The early end of the observations for each night was result of the concealment of the field behind some trees in the west. After the end of the observations, a set of dark and bias fields was obtained in order to reduct the raw data.

R.A.	00h 08m 00s
Dec	33^o 30' 00"
Duration (days)	18
Effective days	10
Exp time	90 sec
Space between exp	60 sec
Filter	no
binning	1
Frame (pix)	3072 x 2048
deg per pix	0.001016
CCD & mount software	MaximDL

Table 3.1: The first long term run settings

3.1.2 The TEF2011 sample

The TEF2011 is a medium crowded field without any noticeable nebula or big galaxy. The identification of the extracted objects in the field was one of the first considerations. Because of that the creation of the TEF2011 catalog was mandatory. In order to create this catalog, one of the best frames of the run was astrometried. The extracted sources of this frame were matched with the actuals of USNO-A2.0 catalog.

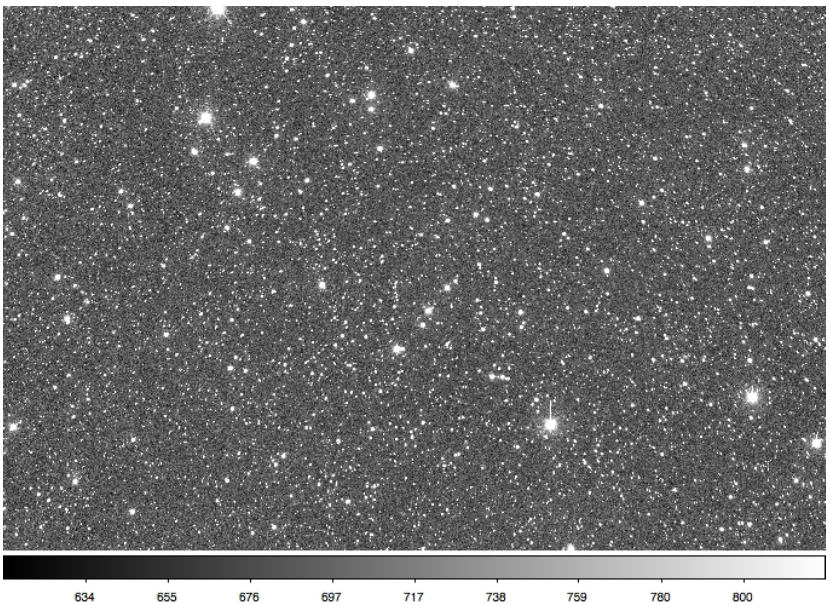

Figure 3.1: The TEF2011 field.

The catalog of the TEF2011 field is consisted of 12134 stars with magnitudes between 10 and 17 mags. After the end of the detrend phase, of the ThReT pipeline, only the 2000 of the stars had accuracy good enough (<2 %) in order to be able to show transits, in case they were occurred.

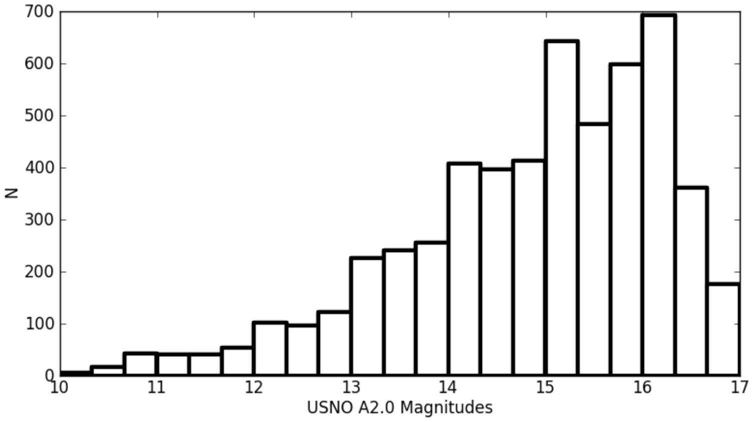

Figure 3.2: The TEF2011 sample.

The sort period of observations is causing the probability of a transit to occur, in the TEF2011 sample, to be very low. Also, for the same reason, the range of the periods that sould have been detected was very small. In figure 3.3 is presented the period probability results detector for the actual times of the run.

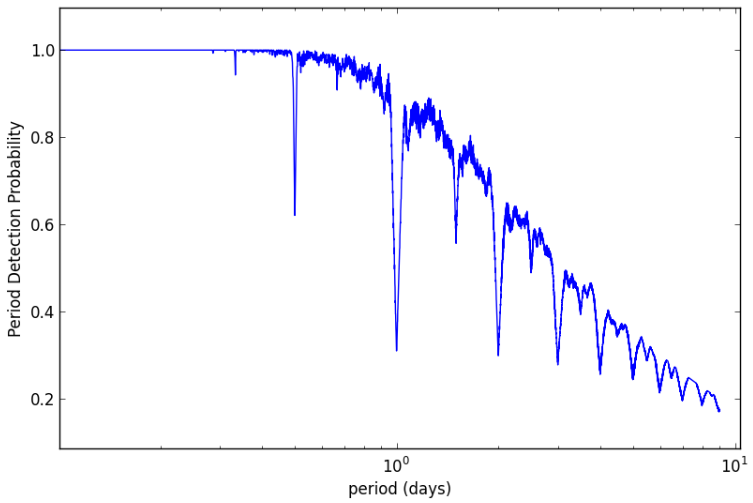

Figure 3.3: The period propability detector results for TEF2011.

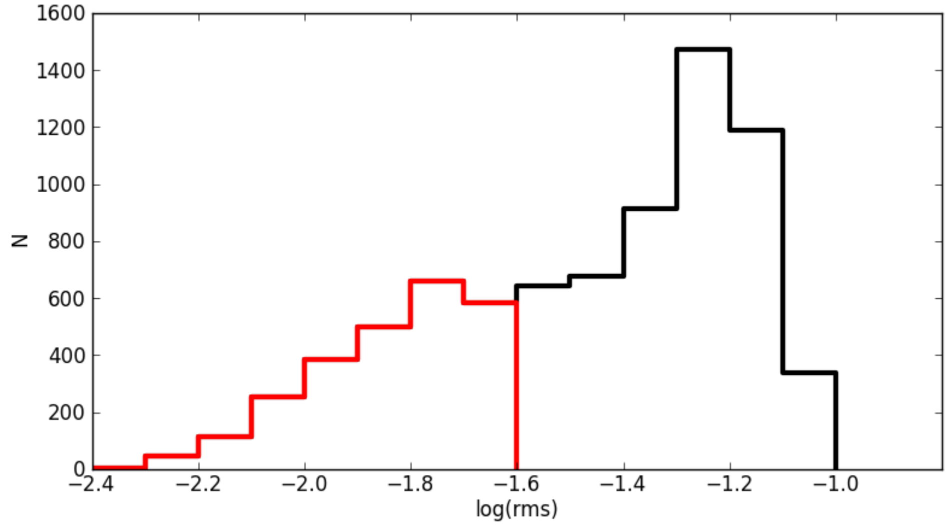

Figure 3.4: The TEF2011 sample rms distribution. The red area is consisted of stars with accuracy good enough for transit detection.

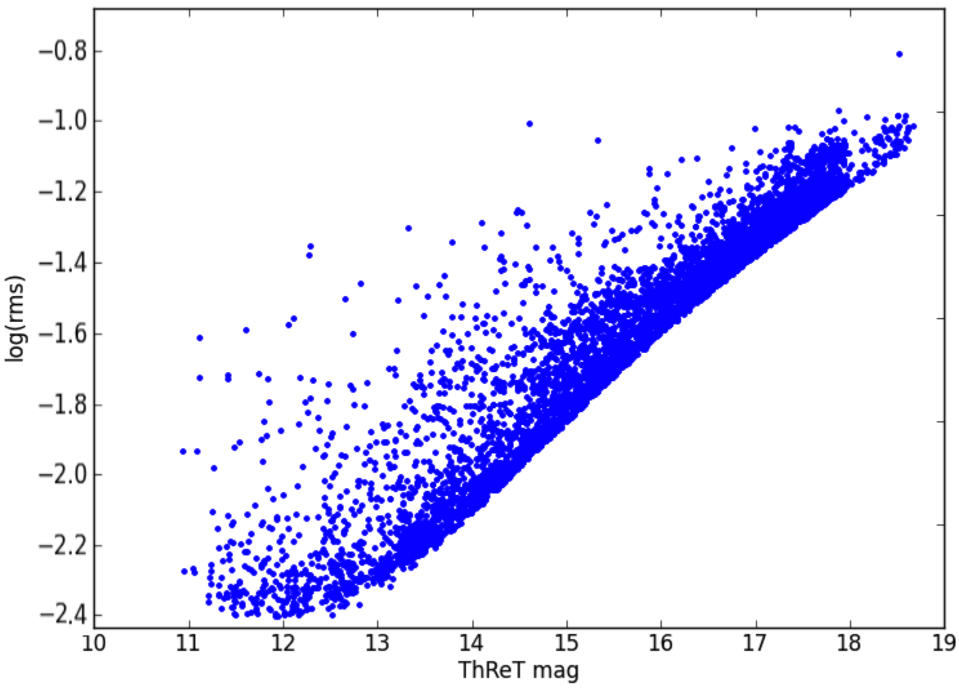

Figure 3.5: The TEF2011 sample plrecition after the trends removal.

3.1.3 TEF2011 results

After the end of the first run the data were analyzed by the ThReT pipeline. Most of the variable star candidates are binaries of the W UMa type with periods between 0.3-0.7 days. There are also two sort period candidates for δ scutti with periods about 0.16 days. Finally there are two signals, in TEF2011-114 and TEF2011-833, which are identical to transit like signals and they will be investigated more.

TEF2011-114

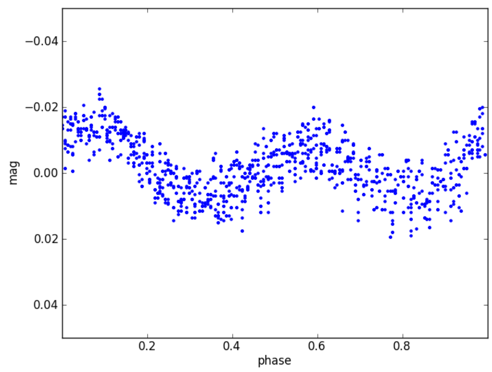

Figure 3.6: A phase folded diagram for TEF2011-114

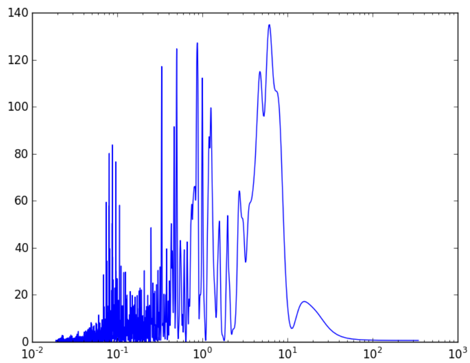

Figure 3.7: The periodogram of TEF2011-114

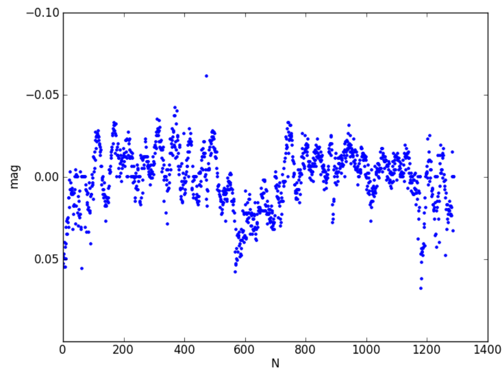

Figure 3.8: The full TEF2011-114 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-306

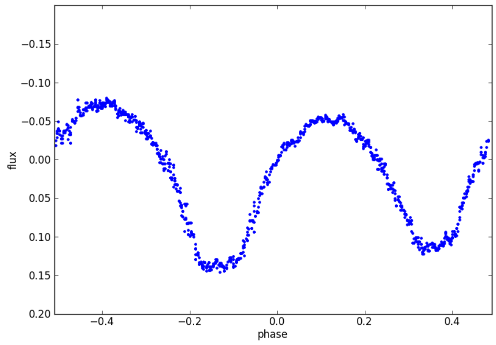

Figure 3.9: A phase folded diagram for TEF2011-306

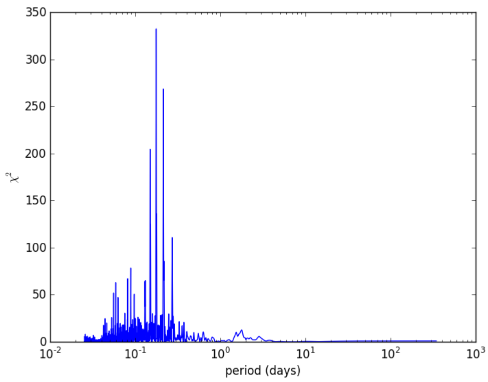

Figure 3.10: The periodogram of TEF2011-306

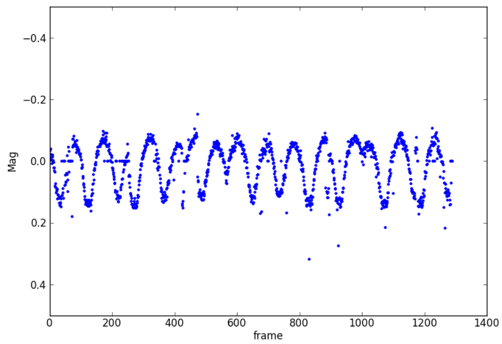

Figure 3.11: The full TEF2011-306 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-536

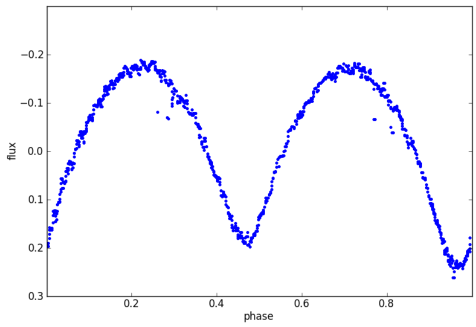

Figure 3.12: A phase folded diagram for TEF2011-536

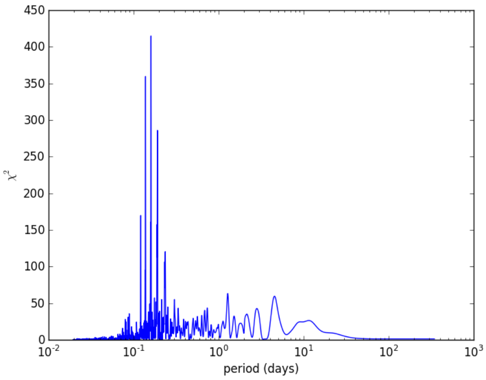

Figure 3.13: The periodogram of TEF2011-536

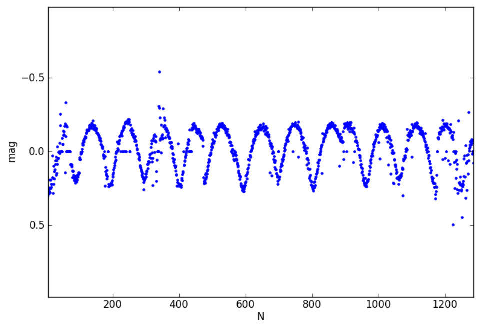

Figure 3.14: The full TEF2011-536 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-556

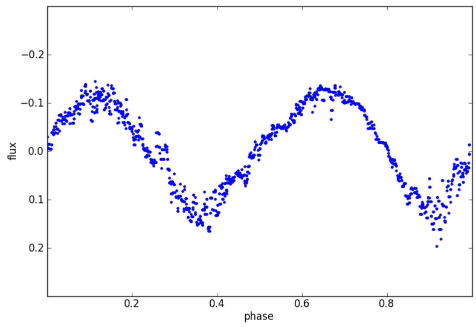

Figure 3.15: A phase folded diagram for TEF2011-556

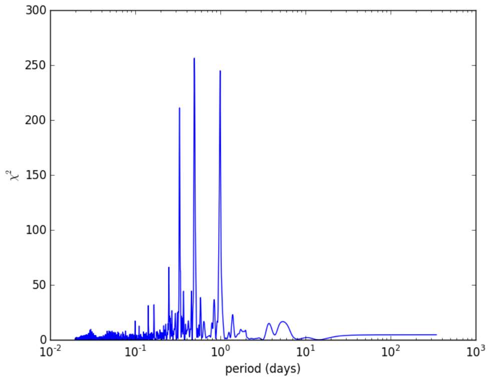

Figure 3.16: The periodogram of TEF2011-556

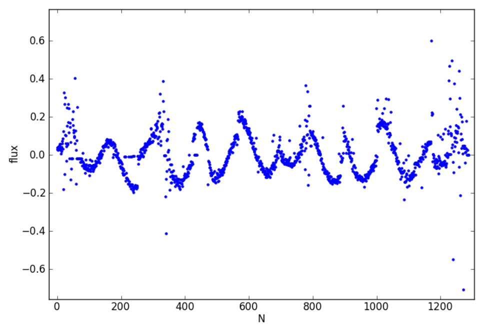

Figure 3.17: The full TEF2011-536 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-833

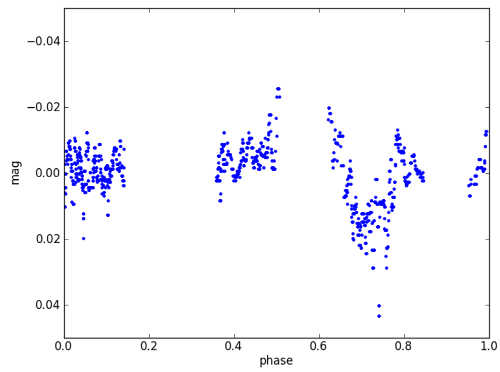

Figure 3.18: A phase folded diagram for TEF2011-833

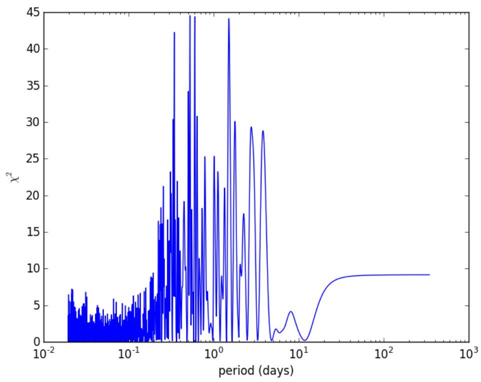

Figure 3.19: The periodogram of TEF2011-833

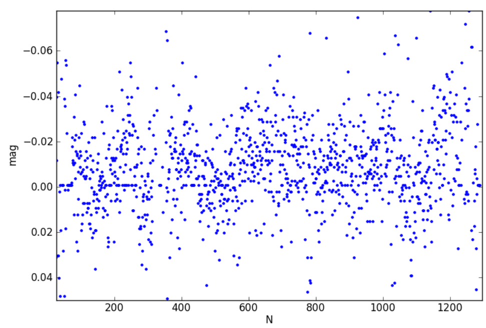

Figure 3.20: The full TEF2011-833 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-855

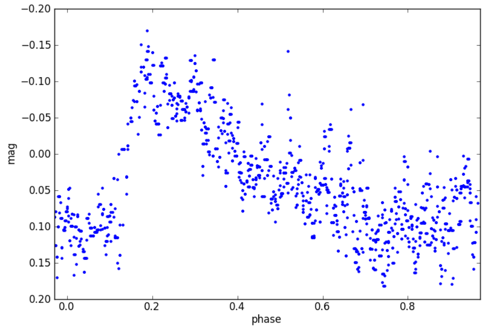

Figure 3.21: A phase folded diagram for TEF2011-855

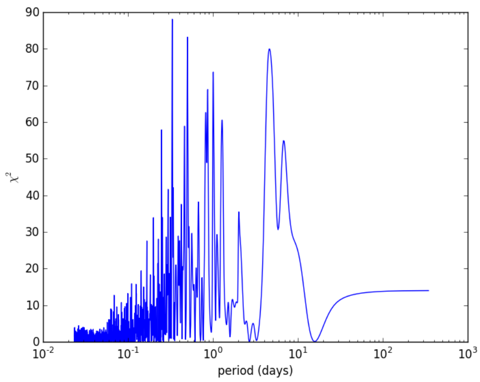

Figure 3.22: The periodogram of TEF2011-855

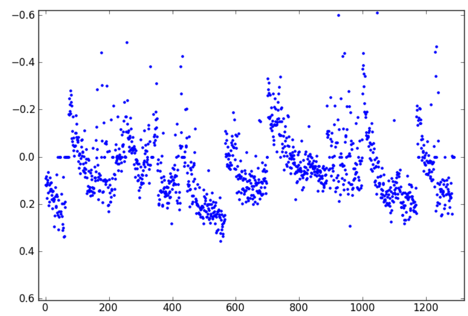

Figure 3.23: The full TEF2011-855 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-856

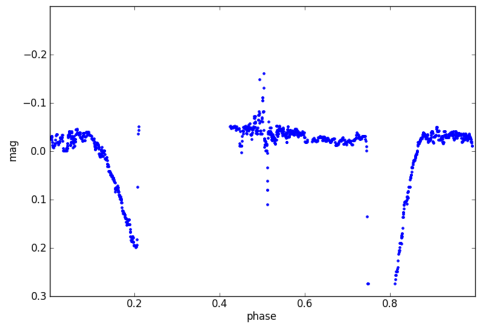

Figure 3.24: A phase folded diagram for TEF2011-856

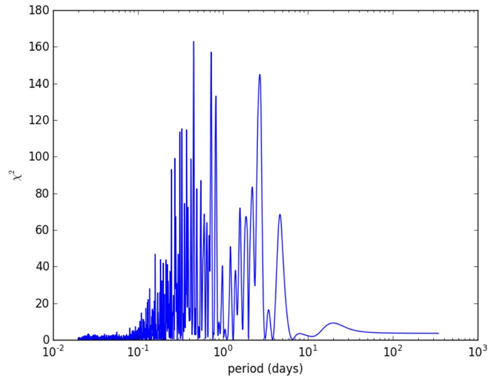

Figure 3.25: The periodogram of TEF2011-856

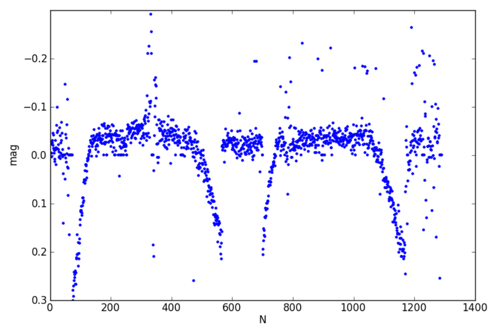

Figure 3.26: The full TEF2011-856 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-942

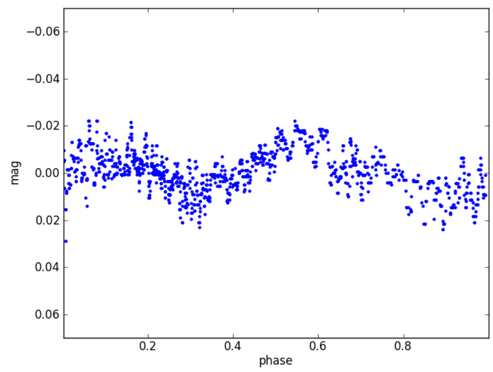

Figure 3.27: A phase folded diagram for TEF2011-942

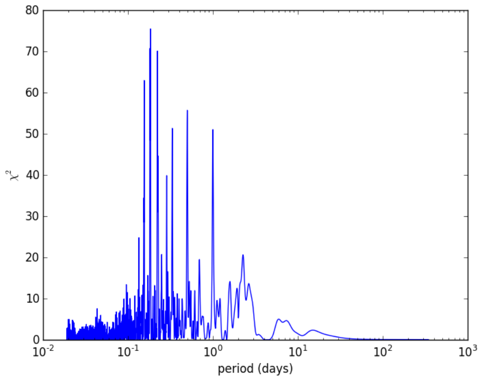

Figure 3.28: The periodogram of TEF2011-942

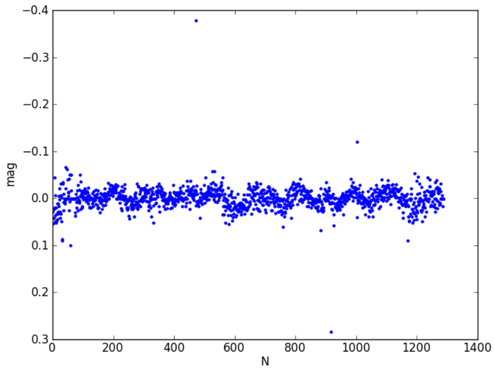

Figure 3.29: The full TEF2011-942 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-1133

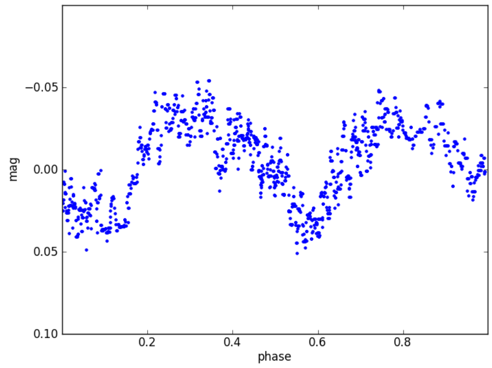

Figure 3.30: A phase folded diagram for TEF2011-1133

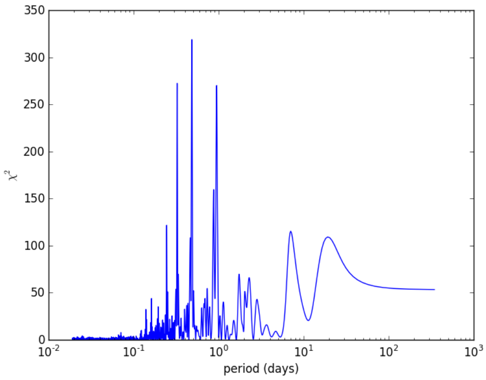

Figure 3.31: The periodogram of TEF2011-1133

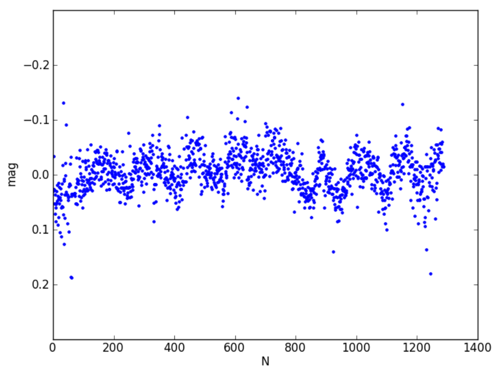

Figure 3.32: The full TEF2011-1133 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-1187

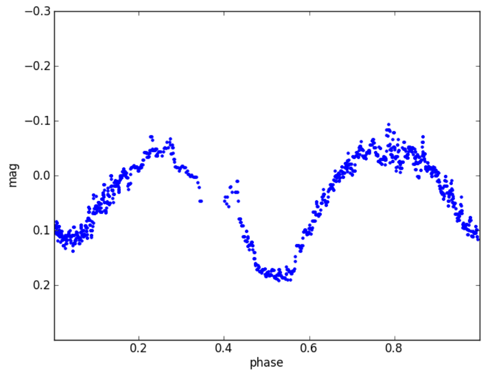

Figure 3.33: A phase folded diagram for TEF2011-1187

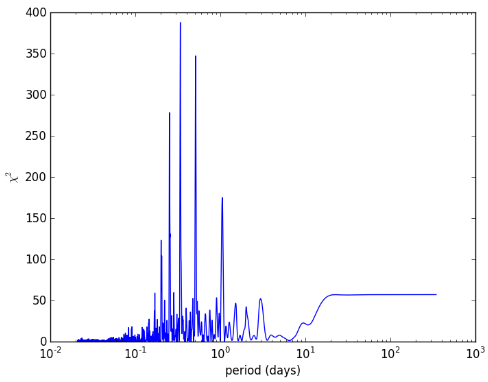

Figure 3.34: The periodogram of TEF2011-1187

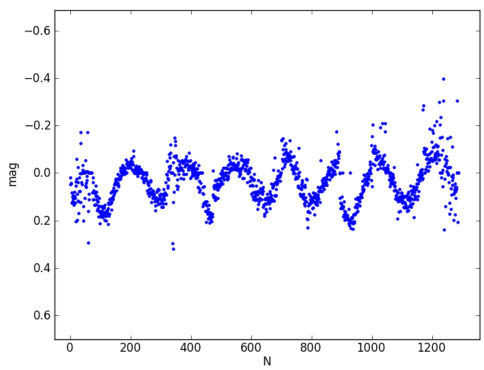

Figure 3.35: The full TEF2011-1187 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-1261

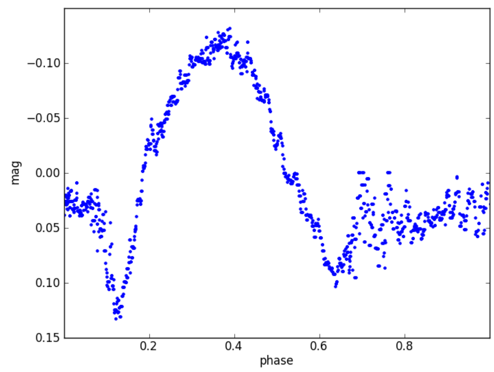

Figure 3.36: A phase folded diagram for TEF2011-1261

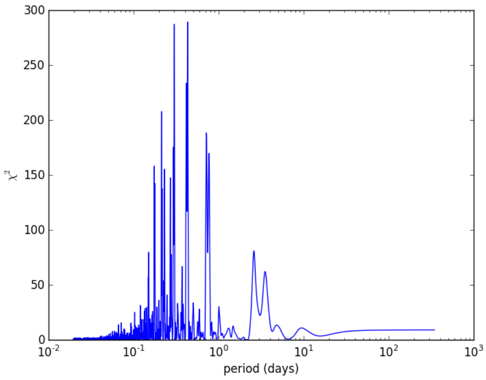

Figure 3.37: The periodogram of TEF2011-1261

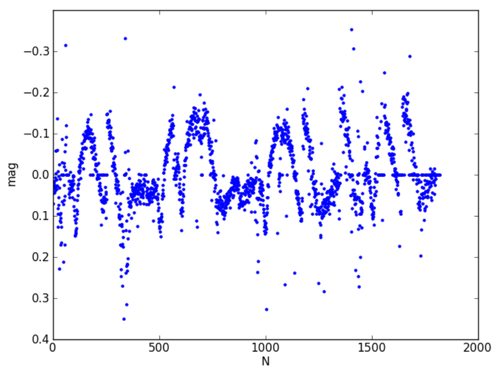

Figure 3.38: The full TEF2011-1261 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-1696

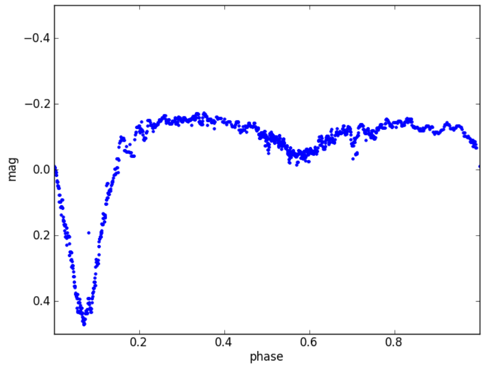

Figure 3.39: A phase folded diagram for TEF2011-1696

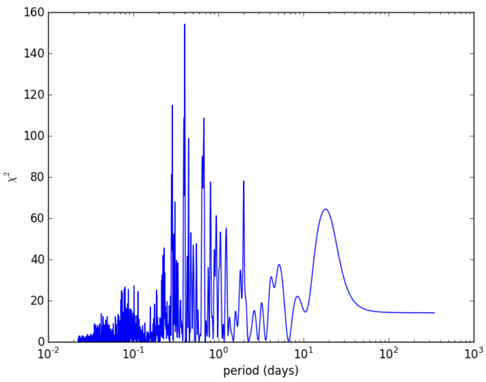

Figure 3.40: The periodogram of TEF2011-1696

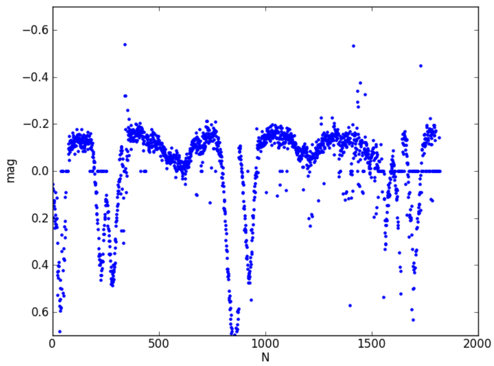

Figure 3.41: The full TEF2011-1696 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-3365

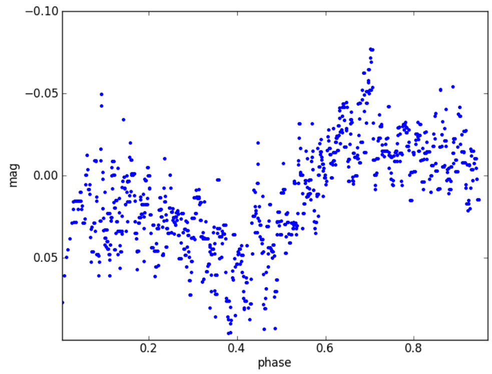

Figure 3.42: A phase folded diagram for TEF2011-3365

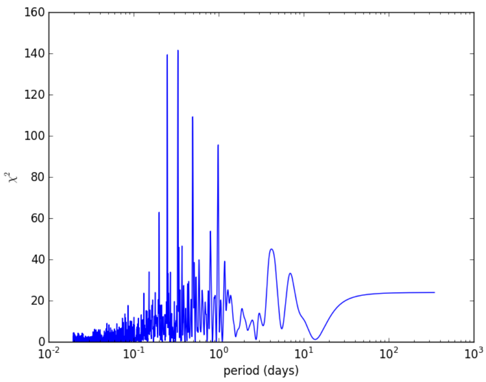

Figure 3.43: The periodogram of TEF2011-3365

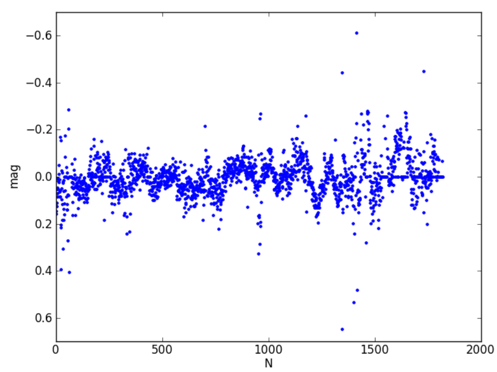

Figure 3.44: The full TEF2011-3365 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-4251

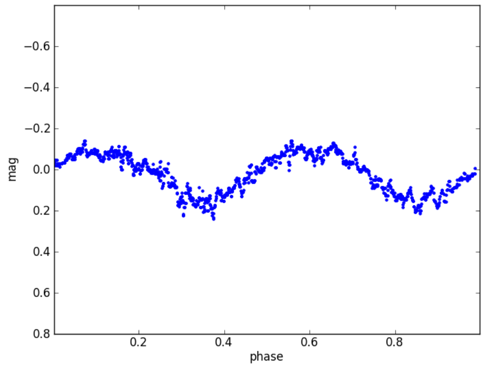

Figure 3.45: A phase folded diagram for TEF2011-4251

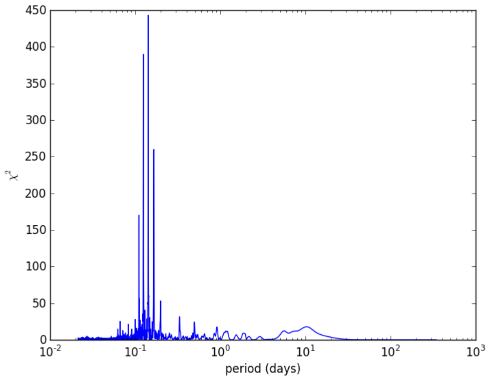

Figure 3.46: The periodogram of TEF2011-4251

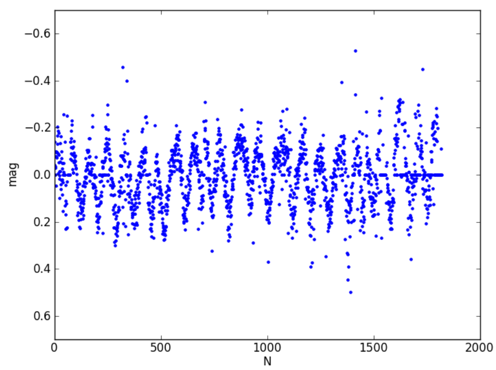

Figure 3.47: The full TEF2011-4251 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-4426

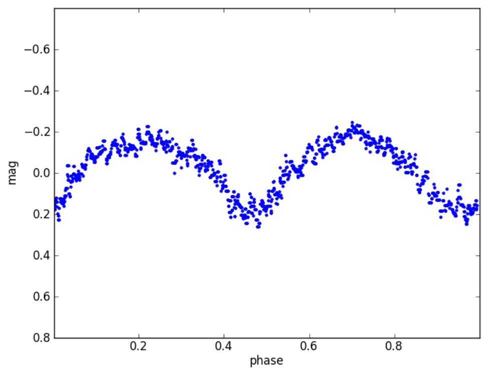

Figure 3.48: A phase folded diagram for TEF2011-4426

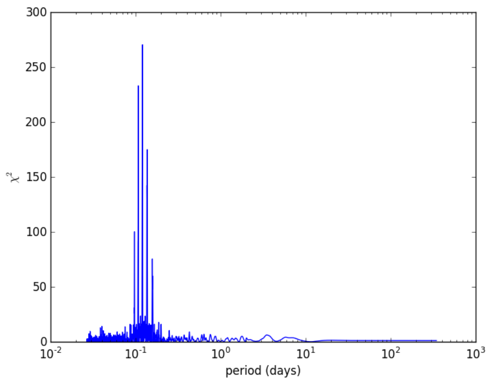

Figure 3.49: The periodogram of TEF2011-4426

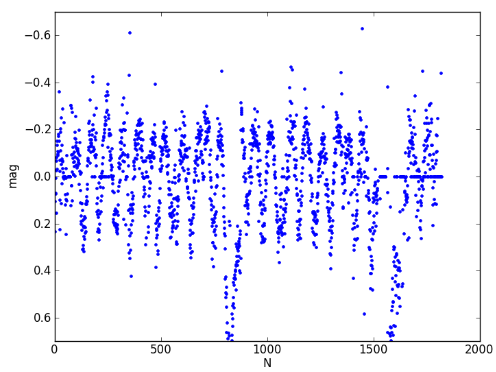

Figure 3.50: The full TEF2011-4426 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-4641

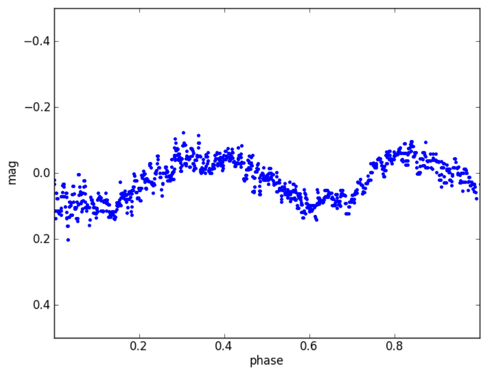

Figure 3.51: A phase folded diagram for TEF2011-4641

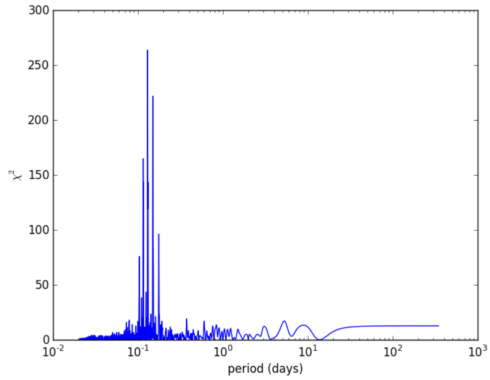

Figure 3.52: The periodogram of TEF2011-4641

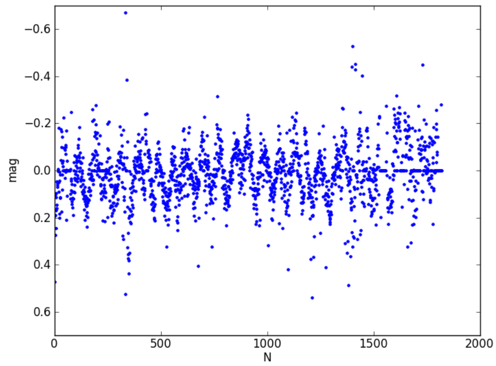

Figure 3.53: The full TEF2011-4641 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-4746

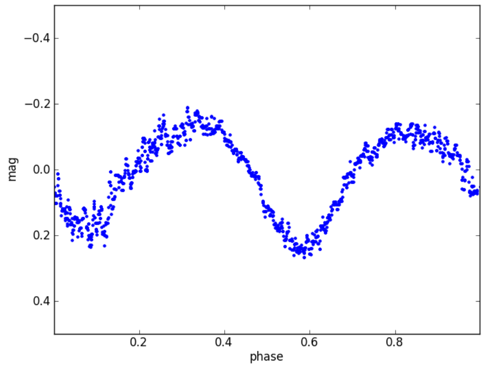

Figure 3.54: A phase folded diagram for TEF2011-4746

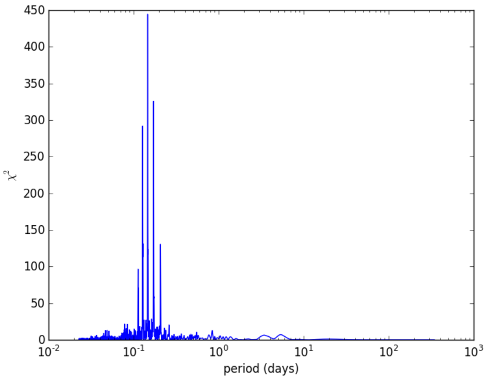

Figure 3.55: The periodogram of TEF2011-4746

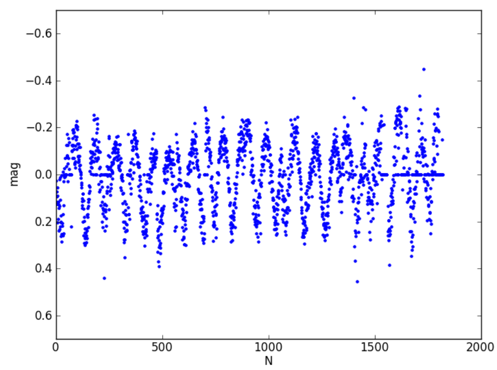

Figure 3.56: The full TEF2011-4746 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-5499

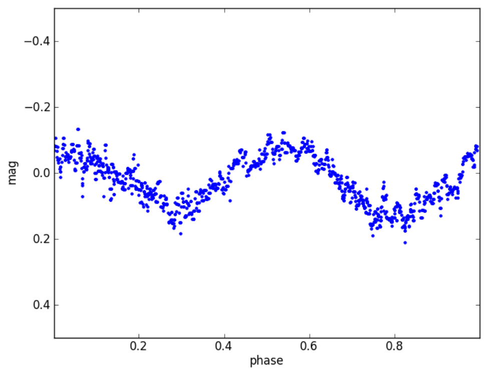

Figure 3.57: A phase folded diagram for TEF2011-5499

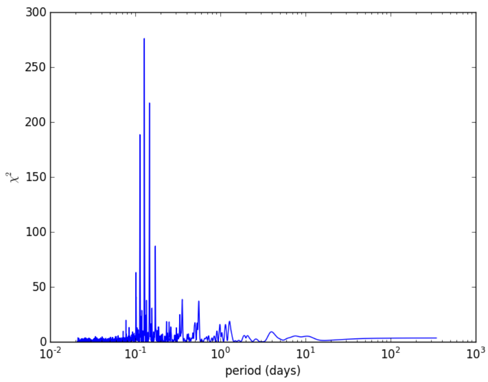

Figure 3.58: The periodogram of TEF2011-5499

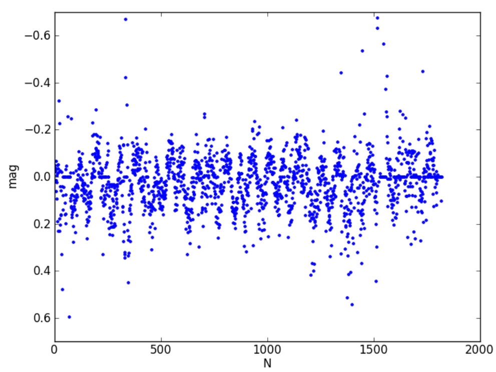

Figure 3.59: The full TEF2011-5499 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-5869

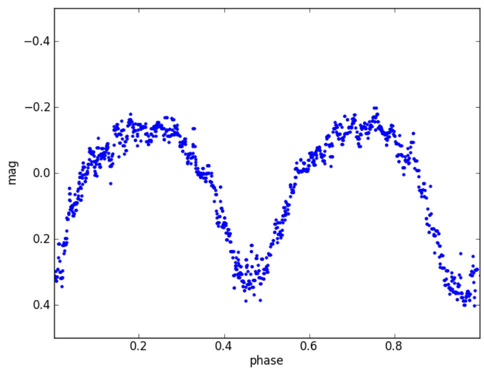

Figure 3.60: A phase folded diagram for TEF2011-5869

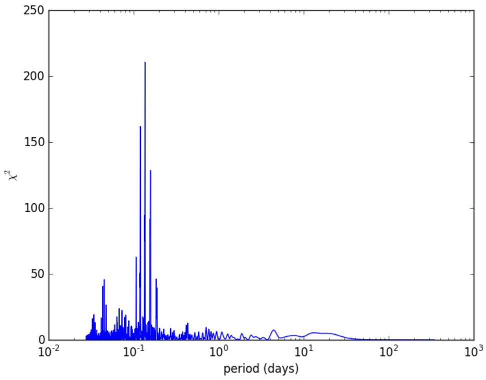

Figure 3.61: The periodogram of TEF2011-5869

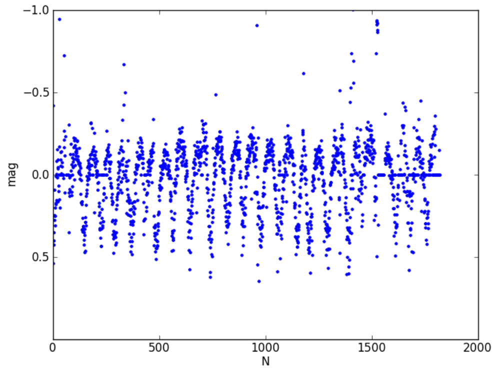

Figure 3.62: The full TEF2011-5869 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-6355

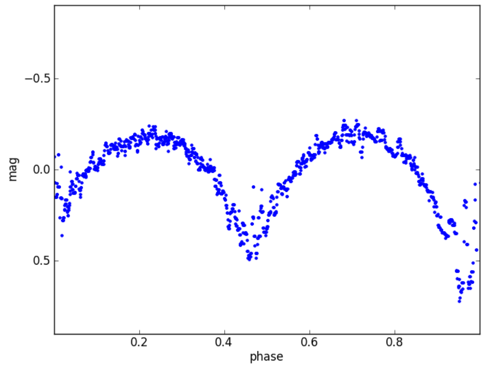

Figure 3.63: A phase folded diagram for TEF2011-6355

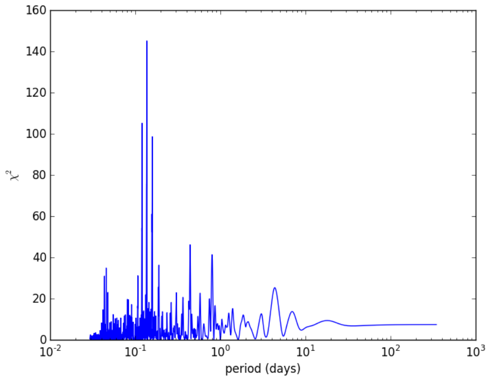

Figure 3.64: The periodogram of TEF2011-6355

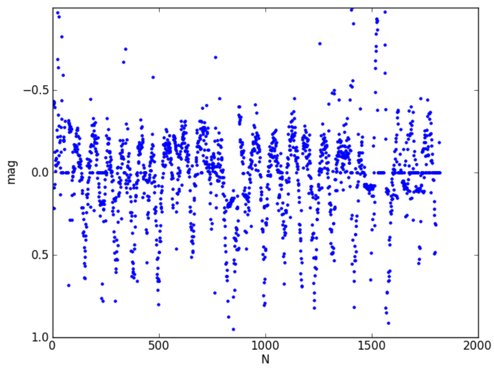

Figure 3.65: The full TEF2011-6355 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-7868

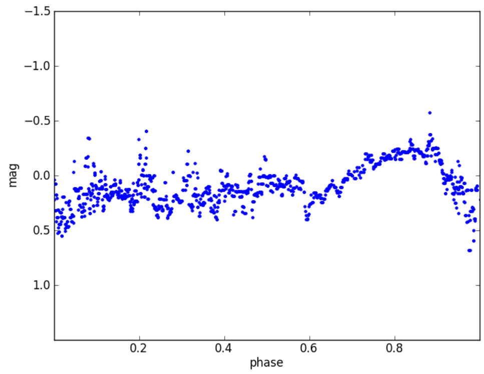

Figure 3.66: A phase folded diagram for TEF2011-7868

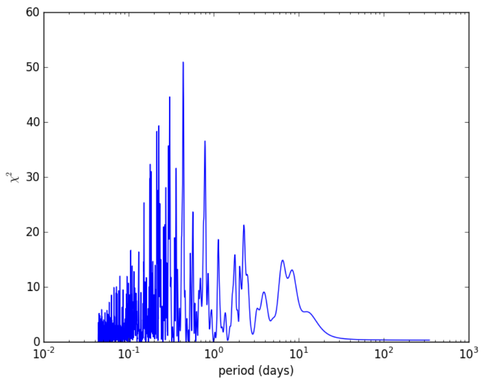

Figure 3.67: The periodogram of TEF2011-7868

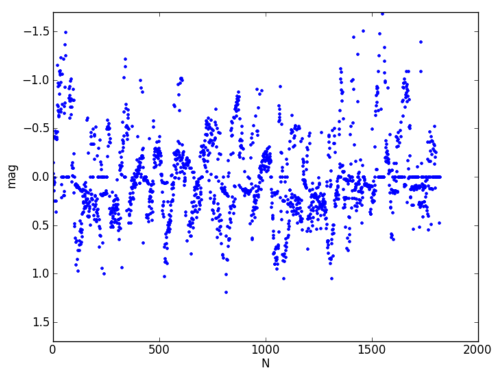

Figure 3.68: The full TEF2011-7868 lightcurve without time in x axis, in order for the variability to be visible.

TEF2011-7875

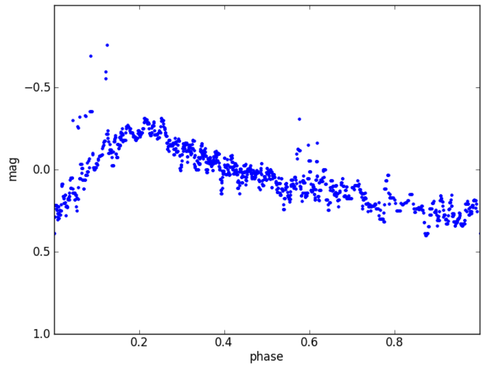

Figure 3.69: A phase folded diagram for TEF2011-7875

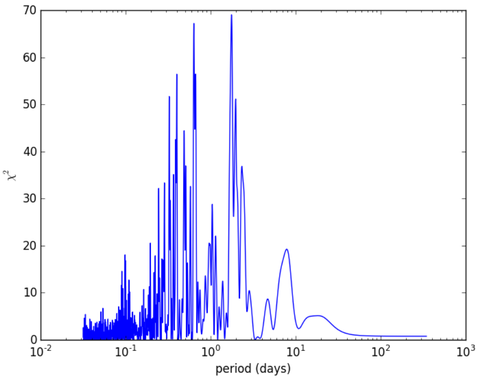

Figure 3.70: The periodogram of TEF2011-7875

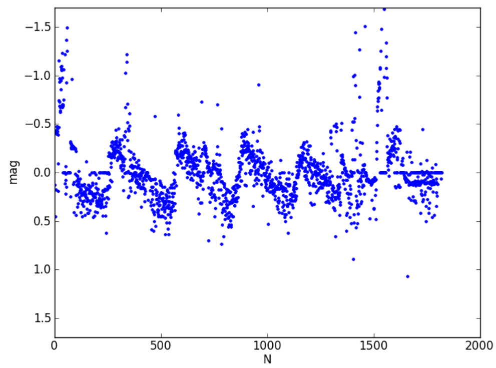

Figure 3.71: The full TEF2011-7875 lightcurve without time in x axis, in order for the variability to be visible.

Chapter 4

Follow up observations on known transits

Despite searching, another part of extrasolar planetary science is the follow up observations of already known planets. There are many things to observe, which could produce a more detailed report for each of planet, such as their spectrum or any differences on their orbital elements. With a small telescope, the most of these observations are not possible to be done because of the small signal to noise ratio. A measurement which is possible to be done, is the search for transit timing variations (paragraph 1.2.5). With the appropriate timing accuracy, after the lightcurve model extraction, it is possible to compute the mid-transit epoch, which can be compared to other mid-transit epoch of the same planet. From Holomon Astronomical Station was observed the transits of three planets: HAT-P-19 b, WASP-33 b and XO-3 b. All the observations were made without filter. The models were produced using the [Pál, 2008] and Levenberg-Marquardt algorithms. The limb darkening coefficient was kept constant with values of u1 = 0.51 and u2 = 0.14 which are referring to 580 nm of the HST, for a star with T_{eff} = 6113 K [Howarth, 2011]. After various tests, it was made clear that the noise in our data was much bigger than the variation, caused to the difference of the limb darkening coefficients, for different filters.

4.1 WASP-33 b

The Wasp-33 b is one of the two planets confirmed around an A5 main sequence star. For most cases, the big radius of those stars, makes it very difficult to detect transits of possible planets around them. In addition Wasp-33 is a δ scuti, very fast rotating, variable star with radius of 1.44 $R_\odot$ while the planetary radius is 1.49 R_J and the Δm of the transit is announced to be 11.36 mmag. The star's apparent magnitude is M_V=8.3. The weather during that observation was humid and a thin cloud passed over the field in about the middle of the observation. From the model fit the Δm of the observation was calculated to be 15.6 mmag and the O-C difference 0.0023 d.

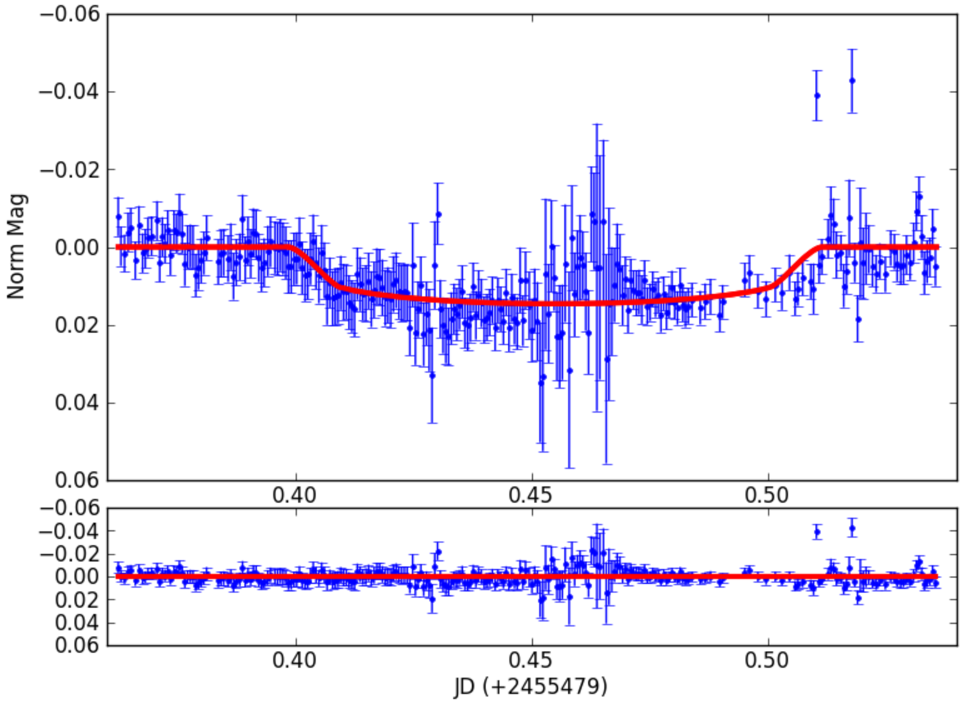

Figure 4.1: The wasp-33 b transit at Oct 9^{th}, 2010

4.2 HAT-P-19 b

The HAT-P-19 b planet is a hot Jupiter, with mass 0.29 M_J and radius 1.13 R_J, around a K dwarf star, with 0.84 $M_\oplus$ and 0.82 $R_\odot$. The star's apparent magnitude is M_V=12.9. The Δm of the transit is 20.1 mmag. The orbital period is 4.01 days with semi-major axis 0.047 AU. The very good weather during the night of the observation contributed to capture an almost perfect transit of the planet. We calculated that the transit depth was 24.1 and the O-C difference was 0.0018 d.

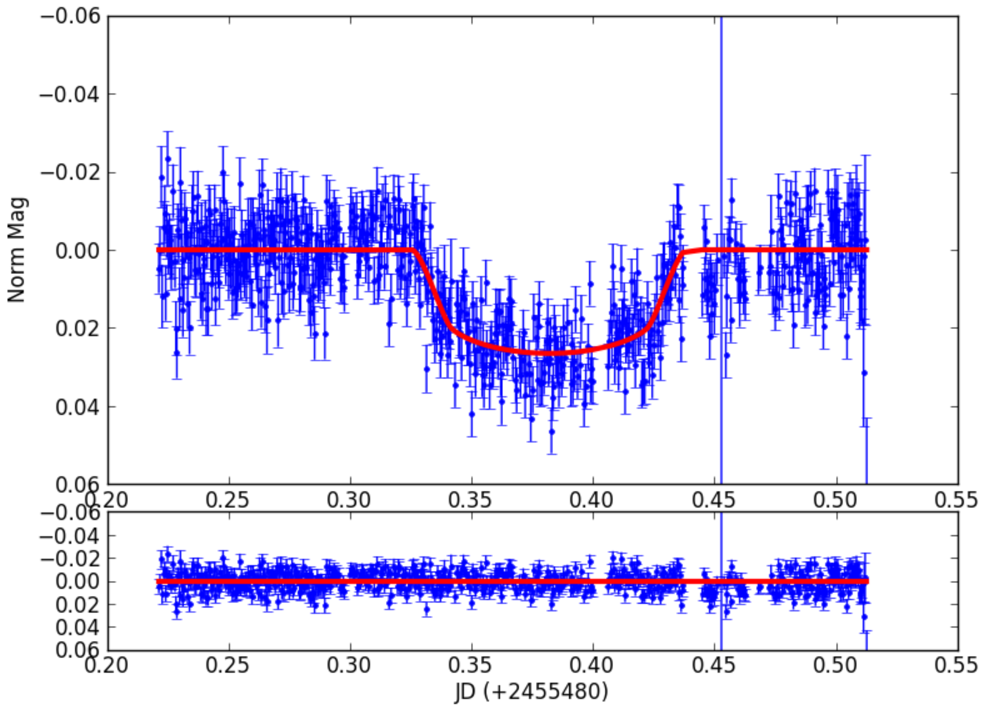

Figure 4.2: The HAT-P-19 b transit at Oct 10^{th}, 2010

4.3 XO-3 b

The XO-3 b planet is a massive hot Jupiter, with radius of 1.217 R_J, which have some peculiar characteristics about its orbit. It is the first extrasolar planet known to have a highly inclined orbit relative to the equatorial plane of its parent star and there is also misalignment between the planetary orbital axis and the stellar rotation axis of -37.3 deg. The host star is an F5 main sequence star with radius of 1.37 $R_\odot$.The star's apparent magnitude is M_V=9.8.

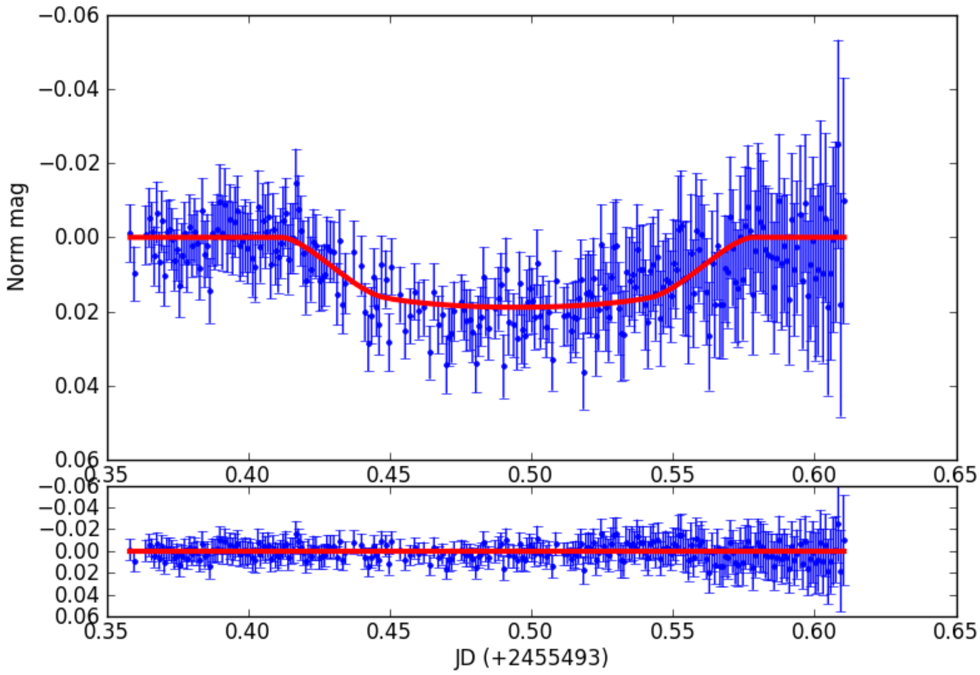

Figure 4.3: The XO-3 b transit at Oct 23^{th}, 2010

Chapter 5

Conclusions and Discution

About thirty variable stars and some transit candidates is the result of the first long term run of the ThReT project. The promising results of the first long term run of the ThReT project are showing that it is not impossible to set up a survey for extrasolar planets, with non expensive, amateur instruments. There are, thought, some points that should be kept in mind to do so.

The first consideration is the place which is being selected as the station of the telescope. It should be in high altitude, with low mean humidity and cloudiness. Also as the search for as small variabilities, as those which are produced by the transit of a planet, needs very low dispersion of the lightcurve in order to be detected, the astronomical seeing must be low.

After the selection of the location, the telescope and the CCD must be selected carefully. A "fast" focal length has to be selected to result a big field of view in combination to a big enough CCD, witch will be able to capture a lot of stars without having extreme vignetting at the edges. In addition, for the same reason that the seeing must be low, the camera must be cooled in temperatures lower than -30^o in order to keep the dark current in low levels. Finally the quantum efficiency must be over 65% and the readout noise lower than $10e^-$.

As the hardware for the survey is set, the next thing to do is to discover the limits of the equipment. There is a big question which has to be answered before the start of the survey. How has the exposure to be set in order to have a big sample without having the very bright and therefore with more accurate lightcurves stars saturated. An other question is whether or not to use any filters having in mind the same as before. All those questions may be answered after some testing in various fields, with deferent configurations.

Despite that the equipment is very critical for a survey, it is only the half of the project. The other half is the software for the reduction, photometry, detrend and search. It is very important to use a pipeline which is fast and accurate, which utilizes to the maximum the data and the computer's capabilities, in order to detect transit like signals.

In that stage, as the equipment and the software have been found, the survey is ready to go. It would be very useful, thought, the first of the fields which will be selected to search, to contain one already known planet, with a host star within the range of stars that the survey is tuned to detect. The pipeline should be able to detect the transit of the planet without any external interference from the user.

After the success of the test phase is time for the selection of a new, unknown field of the sky to be explored. The selection of this field must obey to some rules, so the search could bring some results. First of all, as the probabilities to find anything are increasing with respect to the size of the sample, the selected part of the sky, should content a lot of stars. This can be achived for field near to the Galactic plane. There are limitations thought which are related to the telescope focal length, it's diameter and the seeing of the place. The result of these factors is that if the field is overcrowded there will be a lot of blending, which could cause to loss of accuracy or false alarms.

The selection of the target field or fields is the last step before the actual run of the survey. The planets are objects with a wide range of periods. The hot planets may have from less than to some days, depending to their semi-major axis, while there are planets with periods of thousands of years. This fact the survey should be running for long time, in order to increase the chances to detect transits.

The ThReT project, which is obeying to all of the specifications described above, will continue to search for planets in the future. One of the first improvements, regarding to the searching strategy, is the creation of an input catalog for every one of the fields before the beginning of the run. This will improve the photometric aperture selection for each one of the stars. Also it will allow better results regarding to the predictions about the transit detection probability, as the number of the very luminous main sequence or giant stars will be known, in which a transit is more difficult to be detected. In addition the most interesting cases of the already found stars will be followed up, in order to resolve their cases.

Concluding, there are a lot of potentials for surveys related to the detection of extrasolar planets with the method of transits, using small, low cost instruments. If they manage to overcome problems such as location and the need for long-term measurements could contribute to the discovery of several large extrasolar planets.

Appdx A

The source codes of the python and csh scripts.

.1 ThReT pipeline

.1.1 thret.py

```
#!/usr/bin/env python
from __future__ import with_statement
help = '''
Data reduction and photometry pipeline for the ThReT project V 0.9


Packeges to run this pipeline:
  Python 2.5 or newer
  Numpy, Scipy, Pylab and Pyfits
  Starlink

Usage: ./thret.py -f 'data_files' [OPTION...]


 -f : list of data fits. i.e. '/home/project/data*.fit'


 -o : output directory i.e. '/home/project/output'. If blank an 'output' dir
         will be created inside the data file.

 -r : reduciton frames file i.e. '/home/project/reduction/'. If the same with
         light frames leave blank. Please keep all the reduction frames in the\
same file.
```

```

for any qestions contact : pioannid@auth.gr
'''

chnge_log='''
@ Change log :
  v0.9 at 15/3/2011 : Start of translating the thret pipeline from csh to python
  v0.9.5 at 10/8/2011 : Replacement of daophot with extractor for photometry
  v1.0 at 25/9/2011 : Transformation to GUI enviroment
'''
import tkFileDialog
import itertools
import getopt
import os
import glob
import shutil
import sys
from Tkinter import *
from numpy import *
from scipy import *
from pyfits import *
import string

sys.path.append('./')


def mainmenu(output,maindir):
  ans=[]
  while ans != 'q':
    os.system('./pickfunction.csh intel64')
    funcs = []
    seq = glob.glob('sequense.txt')
    if not seq:
      shutil.rmtree(maindir+'temp')
      sys.exit()
    for line in open('sequense.txt').readlines():
      funcs.append(line)
```

```
    funcs.sort()
    errs=[]
    i=0
    if os.path.exists(maindir+'check.txt'):
      i = int(open('check.txt').readline())
      os.remove(maindir+'check.txt')
    while i < len(funcs):
      os.system('./%s intel64' %(funcs[i].split()[1]))
      if output+'temp/err' in glob.glob(output+'temp/*'):
        while i < len(funcs):
          errs.append(funcs[i].split()[1])
          i=i+1
        break
      i=i+1
    os.system('clear')
    print '\n \n \n \n'
    for i in range(len(funcs)):
      if funcs[i].split()[1] in errs:
        print '\t \t \t '+funcs[i].split()[1]+' ................ended with\
errors'
      else:
        print '\t \t \t '+funcs[i].split()[1]+' ................ended ok'

    msg = '\n \n \n Continue with ThReT (y) or Exit (q)? \n --> '
    ans = raw_input(msg).strip()

  shutil.rmtree(maindir+'temp')
  return()

def thret(dlist, output='', ccd='' ,rfile='' ,coordsset='' ):

  sys.path.append('./')



  #Check values.
  if output == '':
```

```
    output = os.path.dirname(dlist[0]) + '/output/'
  if os.path.exists(maindir+'temp'):
    ans = []
    while not ans:
      msg = 'It seems that ThReT terminated with inappropriate way. Do you\
want to continue the last acction? [Y/N] \n --> '
      ans = raw_input(msg).strip()
      if ans == 'N' or ans == 'n':
        shutil.rmtree(maindir+'/temp')
      elif ans == 'Y' or 'y':
        os.chdir(maindir)
        mainmenu(output, maindir)
        sys.exit()

  if os.path.exists(output):
    ans = []
    while not ans:
      msg = 'There is an older output file. Do you want to continue your job\
witn it or not? [Y/N] \n --> '
      ans = raw_input(msg).strip()
      if ans == 'N' or ans == 'n':
        shutil.rmtree(output)
        os.makedirs(output)

      else:
        os.makedirs(maindir+'temp')
        open(maindir+'temp/output.txt', 'w').write(output)
        open(maindir+'temp/temp.txt', 'w').write(output+'temp/')
        open(maindir+'temp/input.txt', 'w').write(os.path.dirname(dlist[0]))
        os.chdir(maindir)
        mainmenu(output, maindir)
        sys.exit()


  os.makedirs(maindir+'temp')
  os.makedirs(output+'temp')
  open(maindir+'temp/output.txt', 'w').write(output)
```

```
open(maindir+'temp/temp.txt','w').write(output+'temp/')
open(maindir+'temp/maindir.txt','w').write(maindir)
open(maindir+'temp/input.txt','w').write(os.path.dirname(dlist[0]))

###################################################################################
os.chdir(maindir)
headerin = open('headerinfo.dat').readlines()
htypes=[]
for i in headerin:
  if len(i) > 1:
    htypes.append(i.split()[0])


binniKey = headerin[0].split()[2]   # binning Key
xaxisKey = headerin[1].split()[2]   # X axis dimention Key
yaxisKey = headerin[2].split()[2]   # Y axis dimention Key
exposKey = headerin[3].split()[2]   # Exposure time Key
imageKey = headerin[4].split()[2]   # image type KeyA
timerKey = headerin[5].split()[2]   # UT time of observation
daterKey = headerin[6].split()[2]   # Date of observation
os.chdir(output)


############################  light Frames Check #################################
os.system('clear')

print '\n \n \n \n \n \t \t ********* Checking data frames ********* '
print '\t \t \t ', len(dlist), 'data frames found'

binch=0
xaxisch=0
yaxisch=0
expch=0
i=0
light = open(output+'/temp/light.txt','w')
for frames in dlist:
  if os.path.isdir(frames) != True:
```

```
      bin = getheader(frames)[binniKey]
      binch = bin + binch
      xaxis = getheader(frames)[xaxisKey]
      xaxisch = xaxis + xaxisch
      yaxis = getheader(frames)[yaxisKey]
      yaxisch = yaxis + yaxisch
      exp = getheader(frames)[exposKey]
      expch = exp + expch
      i = i + 1
      light.write(frames+'\n')
  light.close()

  # Setting the coordinates of data


  ans = []
  while not ans:
    msg = 'Set r.a. (hh mm ss) \n  --> '
    ans = raw_input(msg).strip()
    if len(ans) != 8 or float(ans[0]) >= 3 or float(ans[0]) == 2 and \
float(ans[1]) > 4 or float(ans[3]) > 6 or float(ans[6]) > 6 :
      print 'Invalid r.a. form. Please insert --> hh mm ss'
      print ans[0]
      ans=[]
    else:
      open(output+'/temp/ra.txt','w').write(ans)

  ans = []
  while not ans:
    msg = 'Set dec. (i.e. +45 00 00) \n --> '
    ans = raw_input(msg).strip()
    if len(ans) != 9 or float(ans[1]) == 9 or float(ans[4]) > 6 or \
float(ans[7]) > 6 :
      print 'Invalid dec. form. Please insert --> +dd mm ss'
      ans=[]
    else:
      open(output+'/temp/dec.txt','w').write(ans)
```

```
 # Setting the others

 open(output+'/temp/xaxis.txt','w').write(str(xaxis))
 open(output+'/temp/yaxis.txt','w').write(str(yaxis))


 if bin != binch / i:
   print '\t \t Different binning light frames detected... \n Exiting'
   shutil.rmtree(maindir+'/temp')
   shutil.rmtree(output)
   sys.exit()
 elif xaxis != xaxisch / i:
   print '\t \t Different dimentions light frames detected... \n Exiting'
   shutil.rmtree(maindir+'/temp')
   shutil.rmtree(output)
   sys.exit()
 elif yaxis != yaxisch / i:
   print '\t \t Different dimentions light frames detected... \n Exiting'
   shutil.rmtree(maindir+'/temp')
   shutil.rmtree(output)
   sys.exit()
 elif exp != expch / i:
   msg= '\t \t  Different exposure light frames detected... \n Do you want to\
continue? (y/n)'
   ans = raw_input(msg).strip()
   if ans == 'n' or ans == 'N':
     shutil.rmtree(maindir+'/temp')
     shutil.rmtree(output)
     sys.exit()
 else:
   print '\n \n \t \t Ligth frames ..... ok'

 bias = []
 flat = []
 dark = []
```

```
  biasw = open(output+'/temp/biasframes.txt','w')
  darkw = open(output+'/temp/darkframes.txt','w')
  flatw = open(output+'/temp/flatframes.txt','w')


############################ Reduction Frames Check ############################


  while len(bias) == 0 or len(dark) == 0 or len(flat) == 0:

    #Reduction frames directory

    if rfile=='':
      rflist = glob.glob(os.path.dirname(dlist[0]).strip() + '/*.fit')
    else:
      rflist = glob.glob(rfile + '/*.fit')

    #Compatibility test


    for frame in rflist:
      try:
        rtype = getheader(frame)[imageKey]
        rbin = getheader(frame)[binniKey]
        rxaxis = getheader(frame)[xaxisKey]
        ryaxis = getheader(frame)[yaxisKey]
      except KeyError:
        print 'Check the FITs Header and make corrections to the options \
file.'
        shutil.rmtree(maindir+'/temp')
        sys.exit()



      if rtype.split()[0] == 'Bias':
        if rbin == bin and rxaxis == xaxis and ryaxis == yaxis:
          bias.append(frame)
```

```
          biasw.write(frame+'\n')
      if rtype.split()[0] == 'Flat':
        if rbin == bin and rxaxis == xaxis and ryaxis == yaxis:
          flat.append(frame)
          flatw.write(frame+'\n')
      if rtype.split()[0] == 'Dark':
        if rbin == bin and rxaxis == xaxis and ryaxis == yaxis:
          dark.append(frame)
          darkw.write(frame+'\n')

    #Availebility test

    if len(bias) == 0:
      print '\n \n \n \n \n \t \t \t Bias problem... Please check'
    elif len(bias) != 0:
      print '\t \t \t Bias frames ..... ok'
    if len(dark) == 0:
      print '\n \n \n \n \n \t \t \t Dark problem... Please check'
    elif len(dark) != 0:
      print '\t \t \t Dark frames ..... ok'
    if len(flat) == 0:
      print '\n \n \n \n \n \t \t \t Flat problem... Please check'
    elif len(flat) != 0:
      print '\t \t \t Flat frames ..... ok'

    #In case of an error

    if len(bias) == 0 or len(dark) == 0 or len(flat) == 0:
      darkw.close()
      flatw.close()
      biasw.close()
      ans = []
      msg = 'Some reduction files are missing or they are not compatible \
with the light frames or they do not have the right fitsheader key.\
\n 1 to set them by name \
\n 2 to select an other directory with reduction frames \
\n q to exit \
```

```
\n --> '
        ans = raw_input(msg).strip()
        if string.lower(ans) == 'q' or string.lower(ans) == '':
          shutil.rmtree(maindir+'/temp')
          print 'Exiting \n'
          sys.exit()

        #Insert reduction name by hand

        elif int(ans) == 1:
          biasw = open(output+'/temp/biasframes.txt','w')
          darkw = open(output+'/temp/darkframes.txt','w')
          flatw = open(output+'/temp/flatframes.txt','w')
          ans=[]
          msg = 'Please set the bias name (i.e. Bias*.fit): \n'
          ans = raw_input(msg).strip()
          bias = glob.glob(rfile+'/'+ans)
          if len(bias) == 0:
            print 'Please check your data again \n Exiting...'
            shutil.rmtree(maindir+'/temp')
            shutil.rmtree(output)
            sys.exit()
          else:
            print '\n \n \n \t \t \t Bias frames ..... ok'
            for i in bias:
              biasw.write(rfile+'/'+i+'\n')
          ans=[]
          msg = 'Please set the dark name (i.e. Dark*.fit): \n'
          ans = raw_input(msg).strip()
          dark = glob.glob(rfile+'/'+ans)
          if len(dark) == 0:
            print 'Please check your data again \n Exiting...'
            shutil.rmtree(maindir+'/temp')
            shutil.rmtree(output)
            sys.exit()
          else:
            print '\t \t \t Dark frames ..... ok'
```

```
          for i in dark:
            darkw.write(rfile+'/'+i+'\n')
        ans=[]
        msg = 'Please set the flat name (i.e. Flat*.fit): \n'
        ans = raw_input(msg).strip()
        flat = glob.glob(rfile+'/'+ans)
        if len(flat) == 0:
          print 'Please ceck your data again \n Exiting...'
          shutil.rmtree(maindir+'/temp')
          shutil.rmtree(output)
          sys.exit()
        else:
          print '\t \t \t Flat frames ..... ok'
          for i in flat:
            flatw.write(rfile+'/'+i+'\n')

      #select an other directory

      elif int(ans) == 2:
        ans = []
        while not ans:
          msg = 'Please enter the reduction files directory or type exit\
to quit: \n'
          ans = raw_input(msg).strip()
          if string.lower(ans) == 'exit':
            print 'Exiting \n'
            shutil.rmtree(maindir+'/temp')
            shutil.rmtree(output)
            sys.exit()
          else:
            rfile=ans

  darkw.close()
  flatw.close()
  biasw.close()

###################################### CCDs ######################################
```

```

  if ccd == '':
    ans = []
    while not ans:
      msg = 'Set Gain \n --> '
      ans = raw_input(msg).strip()
      open(output+'/temp/gain.txt','w').write(ans)
    ans = []
    while not ans:
      msg = 'Set Read Noise \n --> '
      ans = raw_input(msg).strip()
      open(output+'/temp/rn.txt','w').write(ans)
  else:
    ccd=ccd.replace(',',' ')
    gain=ccd[0]
    rn=ccd[1]


###########################################################################################

  os.chdir(maindir)
  mainmenu(output,maindir)






###########################################################################################
###########################################################################################
if __name__=='__main__':

  data_f = '' # file list of light frames.
  output_d = '' # output directory.
  cam = '' #ccd
  red_f = '' #reduction file
  rade = '' #coordinates set manualy or auto
```

```
  #Gui thret starts
  if len(sys.argv) == 1:
    root = Tk()
    root.withdraw()

    #Insert the data directory
    dirname = tkFileDialog.askdirectory(parent=root,initialdir="$HOME",\
title='Please select a directory')
    if len(dirname ) > 0:
      data_dir=dirname
    maindir = '/Users/Panos/Desktop/thret_v1/ThReT_x64/main/'
    os.chdir(maindir)
    data_f=''

    #Insert the data
    while not data_f:
      os.system('./light.py intel64')
      if os.path.exists(maindir+'light'):
        lightnames=open('light').read().strip()
        if len(glob.glob(data_dir+'/'+lightnames)) < 2:
          if lightnames == 'exit':
            os.remove('light')
            sys.exit()
          os.system('./lighterr.py intel64')
          os.remove('light')         #check the type of name given
          continue
        data_f=data_dir+'/'+lightnames
        dlist = glob.glob(data_f)
        os.remove('light')
      if os.path.exists(maindir+'lights'):
        lightnames=open('lights').readlines()
        if len(lightnames) < 2:
          os.system('./lighterr.py intel64')
          os.remove('lights')
          continue
        data_f=lightnames
        dlist = data_f
```

```
            os.remove('lights')
#     print help
#     sys.exit()

######################## Read Command Line Options ############################
  try:
    optlist, args = getopt.getopt(sys.argv[1:], 'h:f:o:c:r:p')
  except getopt.GetoptError, err:
    print os.getcwd()
    print help
    sys.exit()

  for o, a in optlist:
    if o in ('-h'):
      print help
      sys.exit()
    elif o in ('-f'):
      data_f = a
      dlist = data_f
    elif o in ('-o'):
      output_d = a
    elif o in ('-c'):
      cam = int(a)
    elif o in ('-p'):
      red_f = a
    else:
      continue



  os.system('clear')
# maindir = os.getcwd()+'/main/'
  maindir = '/Users/Panos/Desktop/thret_v1/ThReT_x64/main/'
# if os.path.isdir(data_f) == True:
#     print '\n \n \n \n I need files not a directory (i.e."home/project/*.fit")\
        \n Exiting ... '
#     sys.exit()
```

```
  if len(dlist) == 0:
    print 'There is no data files in : ' + data_f

    sys.exit()
  dlist.sort()
# run main thret routine
  thret(dlist, output=output_d, ccd=cam ,rfile=red_f)
```

.1.2 pickfunction.csh

```
#! /bin/csh
if (-e sequense.txt) then
  rm sequense.txt
endif

set input=`cat temp/input.txt`
set output=`cat temp/output.txt`
restart:
clear
#set alldata=`cat temp/light.txt`
#set allbias=`cat temp/bias.txt`
#set alldark=`cat temp/dark.txt`
#set allflat=`cat temp/flat.txt`

echo
echo "###############################################"
echo "#                                             #"
echo "#    Please select what you want me to do:    #"
echo "#                                             #"
echo "###############################################"
echo "| Input: $input"
echo "| Output: $output"
#echo "| Data: $alldata"
echo "|"
echo "| 1. Data Reduction"
echo "| 2. Align Frames"
```

```
echo "| 3. Photometry & Matching"
echo "| 4. Remove Systematics"
echo "| 5. Transit Search"
#echo "| 6. Some of these or all of them"
echo "| 6. ThRet"
#echo "| 8. Asteroid Light-Curve Extraction"
#echo "| 9. Variable Star Search"
#echo "| c. Change input directory"
#echo "| o. Options"
echo "|"
echo -n "|__Select--> "

set fans=$<
#if ($fff == 1 && $fans == 'c') then
#goto restart
#else if ($fff == 1 && $fans == 'p') then
#goto restart
#else

#endif
if ($fans == 1) then
  echo '1. convert.csh' > sequense.txt
  echo '2. biascheck.csh' >> sequense.txt
  echo '3. reduce.csh' >> sequense.txt
else if ($fans == 2) then  #! if fans=2 align->here
  echo '1. aligner.csh' > sequense.txt

else if ($fans == 3) then   #! if fans=3 photsettings->allstar->match->here
  echo '1. allstartak.csh' > sequense.txt
# echo '1. photom.csh' >> sequense.txt
  echo '2. timer.csh' >> sequense.txt
  echo '4. matcher.csh' >> sequense.txt
  echo '5. curverer.py' >> sequense.txt
  echo '6. sigma.py' >> sequense.txt

else if ($fans == 4) then
#   echo "|"
```

```
#    echo "|"
#    echo "| You have selected to remove systematics. "
#    echo "| Please choose with which routine:"
#    echo "| 1. Ensemble "
#    echo "| 2. Dstl "
#    echo "|"
#    echo "|"
#    echo -n "|__Select--> "
#    set ch=0
#    while ($ch=0)
#    set sysans=$<
#       if ($sysans == 1) then
           echo '1. dstl2.py' > sequense.txt
           echo '2. sigmadstl.py' >> sequense.txt
           clear
           echo
           echo
           echo
           echo
           echo
           echo
           echo " Please wait for dstl to start"
#          set ch=1
#       else if ($sysans == 2) then
#          echo '1. pdt.py' > sequense.txt
#          set ch=1
#       else
#          set ch=0
#       endif
#    end
else if ($fans == 5) then
  echo 'Under construction'
  goto restart
####################### Answer Some of them #############################

else if ($fans == test) then    #! if fans=4
  clear
```

```
echo "############################################################"
echo "#                                                          #"
echo "#   Please select which of them you want me to execute     #"
echo "#   and then type ok:                                      #"
echo "#                                                          #"
echo "############################################################"
echo "|"
echo "| 1. Data Reduction"
echo "| 2. Align Frames"
echo "| 3. Photometry & Matching"
echo "| 4. Remove Systematics"
echo "| 5. Transit Search"
echo "|"
pickerror1:
echo "|"
echo -n "|__Select --> "
set rans=$<
set rannum=1
set next1=0
set next2=0
set next3=0
set next4=0
set next5=0
# set fdir1=0
# set fdir2=0
# set fdir3=0
# set fdir4=0
# set fdir5=0
while ($rannum < 5)
  if ($rans == Ok || $rans == OK || $rans == ok) then #! if you type ok then
    if ($?next1 == 0) then                  #! you will end the prosses
      echo "|"                       #! unless you havn't pick something
      echo "|__You haven't seleced something!"
      goto pickerror1
    else
      echo "| You ended the selection prosses"
      goto combend
```

```
      endif
    else if ($rans <= 5 ) then
      @ ranst = $rans + 1500                    #! The answer is the expected
      if ($ranst == 1500) then  #! if you just press enter you redirected back ..
        @ rannum = $rannum - 1
        goto pickerror
      else
      endif
      set tyyans=`grep "$rans." Functions.txt`
      #set fdir$rannum=`grep "$rans" fincdirectories.dat`
      set next$rannum=$rans
      echo $rans >> next.txt
      if ($tyyans[1] == '1.') then
        echo '1. convert.csh' >> sequense.txt
      else if ($tyyans[1] == '2.') then
        echo '2. align.csh' >> sequense.txt
      else if ($tyyans[1] == '3.') then
        echo '3. allstar.csh' >> sequense.txt
      else if ($tyyans[1] == '4.') then
        echo '4. dstl.csh' >> sequense.txt
      else if ($tyyans[1] == '5.') then
        echo '5. eebls.csh' >> sequense.txt
      endif
      echo "|"
      echo "|__You selected $tyyans"
      echo "|"
      pickerror:
      echo "|"
      echo -n "|__next--> "
      set rans=$<
      if ($rans == $next1 || $rans == $next2 || $rans == $next3 \
|| $rans == $next4 || $rans == $next5) then
        echo "|"
        echo "| You have allready choosen that! Pick again..."      #! Errors
        echo "|"
        echo "|__next--> "     #! in case you select an allready selected routine
        goto pickerror
```

```
      else
      endif
    else if ($rans > 6 || $rans < 10) then
      set tyyans=`grep "$rans." Functions.txt`
      echo "|"
      echo "| The $tyyans is a complete routine and it can't be combined."
      echo "| If you want to run it type reset, otherwise press next."
      wrongty1:
      echo "|"
      echo -n "|__Select--> "
      set errselect=$<
      if ($errselect == reset) then
        goto restart
      else if ($errselect == next) then
        goto pickerror
      else
        echo "|"
        echo "| What? I can't understand you! Please type it again! "
        goto wrongty1
      endif
    else if ($rans == 6) then
      echo "|"
      echo "| What? I can't understand you! Please type it again! "
    else if ($rans > 9) then
      echo "|"
      echo "| Nothing there..."
      echo "|"
      echo -n "|__Next--> "
      set rans=$<
    else
      echo "|"
      echo "| What? I can't understand you! Please type it again! "
      goto pickerror
    endif

  @ rannum = $rannum + 1
```

```
    end

    set tyyans=`grep "$rans." Functions.txt`
    #set fdir$rannum=`grep "$rans" fincdirectories.dat`
    set next$rannum=$rans
    echo $rans >> next.txt
    echo "|"
    echo "|__You selected $tyyans"

    combend:

      if ($?tyyans == 0) then
      echo "|"
      echo "|"
      echo "|__You haven't seleced something! Restarting..."
      sleep 3
      goto restart
      else
      endif

#  echo $next2 >> sequense.txt
#  echo $next3 >> sequense.txt
#  echo $next4 >> sequense.txt
#  echo $next5 >> sequense.txt


      if ($next1 == 4 || $next2 == 4 || $next3 == 4 || $next4 == 4 || \
$next5 == 4) then
#     echo "|"
#     echo "|"
#     echo "| You have selected to remove systematics. "
#     echo "| Please choose with which routine:"
#     echo "| 1. Dstl "
#     echo "| 2. PDT "
#     echo "|"
#     syseror:
```

```
#    echo "|"
#    echo -n "|__Select---> "
#    set sysans=$<
#    @ sysanst = $sysans + 1500
#       if ($sysanst == 1500) then
#       goto syseror
#       else if ($sysans == 1) then
        sed -i -e 's/systematic/ensemble.csh/g' sequense.txt
#       rm -f *-e
#       else if ($sysans == 2) then
#       sed -i -e 's/systematic/TFA\/pdt.csh/g' sequense.txt
#       rm -f *-e
#       endif
    endif

# source photsettings.csh

############################ Answer threat #################################

else if ($fans == 6) then
  echo '1. convert.csh' > sequense.txt
  echo '2. biascheck.csh' >> sequense.txt
  echo '3. reduce.csh' >> sequense.txt
# echo '4. aligner.csh' > sequense.txt
  echo '31. allstartak.csh' > sequense.txt
# echo '4. photom.csh' >> sequense.txt
  echo '5. timer.csh' >> sequense.txt
  echo '7. matcher.csh' >> sequense.txt
  echo '8. curverer.py' >> sequense.txt
  echo '9. sigma.py' >> sequense.txt
  echo '91. dstl2.py' >> sequense.txt
  echo '92. sigmadstl2.py' >> sequense.txt

########################### Answer Asteroids ################################

else if ($fans == 8) then
```

```
  source asteroid.csh

########################### Answer Var Search ###############################

else if ($fans == 9) then

  source variables.csh

########################## Quit ###############################################

else if ($fans == quit || $fans == q) then
  exit
else if ($fans == o) then
  echo '1. headeropt.py' > sequense.txt
  exit
else if ($fans == c) then
  exit

else if ($?fans == 0) then
  goto restart
else
  goto restart
endif








exit
```

.1.3 convert.csh

```
#! /bin/csh

```

```
###################################################################
set stardir=`cat starlinkdir.txt`
source $stardir/etc/login
source $stardir/etc/cshrc
kappa>junk
figaro>junk
pisa>junk
ccdpack>junk
extractor>junk
convert>junk
rm -f junk
echo '0' > check.txt
set main=`pwd`
set dirt=`cat temp/temp.txt`
set dir=`cat temp/output.txt`
set inp=`cat temp/input.txt`
cd $dirt
echo '' > err
set bias=`cat biasframes.txt`
set dark=`cat darkframes.txt`
set flat=`cat flatframes.txt`
set data=`cat light.txt`
cd $dir
if (-e ndfdata) then
  rm -r ndfdata
endif
mkdir ndfdata
cd $inp
###################################################################



#make sure that all past lists have been deleted
if (-e bias.list)then
rm -f bias.list
endif
if (-e dark.list)then
```

```
rm -f dark.list
endif
if (-e flat.list)then
rm -f flat.list
endif
if (-e data.list)then
rm -f data.list
endif

if (-e sflat.list)then
rm -f sflat.list
endif


#convert images to starlink format
clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                    CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                    BIAS FRAMES................................      "

foreach image($bias)
set im=$image:r
rdfits $image $im \\
end

clear
echo
echo
```

```
echo
echo
echo
echo
echo
echo
echo
echo
echo "                  CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                  BIAS FRAMES................................DONE"

clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                  CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                  DARK FRAMES................................     "
foreach image($dark)
set im=$image:r
rdfits $image $im \\
echo $im >>dark.list
end

clear
echo
echo
echo
echo
echo
echo
```

```
echo
echo
echo
echo
echo "                 CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                 DARK FRAMES................................DONE"

clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                 CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                 FLAT FRAMES................................     "
foreach image($flat)
set im=$image:r
rdfits $image $im \\
echo $im >>flat.list
end
clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                 CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
```

```
echo "                 FLAT FRAMES................................DONE"

clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                 CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                 DATA FRAMES................................     "
foreach image($data)
set im=$image:r
rdfits $image $im \\
echo $im >>data.list
echo $im >>sflat.list
end
clear
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo
echo "                 CONVERTING IMAGES TO STARLINK FORMAT PLEASE WAIT"
echo "                 DATA FRAMES................................DONE"

##################################################
cd $dirt
```

```
rm -f err
cd $main
echo '1' > check.txt
###################################################
exit
```

.1.4 biascheck.csh

```
#! /bin/csh

#bias quallity check script
###################################################################
set stardir=`cat starlinkdir.txt`
source $stardir/etc/login
source $stardir/etc/cshrc
kappa>junk
figaro>junk
pisa>junk
rm -f junk
echo '1' > check.txt
set main=`pwd`
set dirt=`cat temp/temp.txt`
set dir=`cat temp/output.txt`
set inp=`cat temp/input.txt`
cd $dirt
echo '' > err
set bias=`cat biasframes.txt`
set dark=`cat darkframes.txt`
set flat=`cat flatframes.txt`
set data=`cat light.txt`
cd $inp
###################################################################

#this section puts the Mean values of each bias frame
#onto a .list file
foreach file($bias)
```

```
set image=$file:r
set mean=`istat $image xstart=min ystart=min xend=max yend=max|grep Mean`
echo $mean[3] >>biascheck.list
end
unset mean

#summarise all mean values and set it into a variable
awk '{sum +=$1} END{print sum}' biascheck.list >sum
set sum=`cat sum`

#obtain the number of frames and set it into a variable
awk 'END{print NR}' biascheck.list >total
set total=`cat total`

#calculate the total mean value of all frames
set meanv=`calc $sum/$total prec=_double`

#bias quality check subrutine
set mi=`calc $meanv-3`
set ma=`calc $meanv+3`
set min=`calc "nint($mi)"`
set max=`calc "nint($ma)"`
foreach file($bias)
set f=$file:r
set mean=`istat $f xstart=min ystart=min xend=max yend=max|grep Mean`
set x=`calc "nint($mean[3])"`
@ a= ( $min <= $x && $x <= $max )
if ($a == 1)then
echo $f >>bias.list
endif
end

#cleanup
rm -f total
rm -f biascheck.list

#continue with the next script
```

```
#####################################################
cd $dirt
rm -f err
cd $main
echo '2' > check.txt
#####################################################
exit
```

.1.5 photom.csh

```
#! /bin/csh

############################################################
set stardir=`cat starlinkdir.txt`
source $stardir/etc/login
source $stardir/etc/cshrc
extractor>junk
rm -f junk
set main=`pwd`
setenv mainthretdir $main
set dirt=`cat temp/temp.txt`
set dir=`cat temp/output.txt`
set inp=`cat temp/input.txt`
cd $dirt
echo '' > err
set gain=`cat gain.txt`
set rn=`cat rn.txt`
cd ..
if (-e photometry) then
rm -r photometry
endif
mkdir photometry
cd photometry
clear
#############################################################

```

```
set ref1='0'
foreach file (../ndfdata/*.sdf)
clear
echo
echo
echo
echo
echo
echo
echo "     Photometry of file $file"


echo $file > junk
set fileroot=`sed 's/\// /g' junk`
set fileroot=$fileroot[3]:r
# Extract the stars from the sdf frame, photometry them and create the lists

extractor $file config=$main/default.sex > extr.log
awk 'NR>13' test.cat > $fileroot.list

# Election of the best reference file (more stars)

set ref=`awk 'END {print NR}' test.cat`
if ($ref > $ref1) then
cp $fileroot.list referance
set ref1=$ref
endif



# Remove files
rm -f extr.log
rm -f check.sdf

#  End of loop.
end

```

```
###################################################
cd $dirt
rm -f err
cd $main
echo '1' > check.txt
###################################################


exit
```

.1.6 apps.py

```
#!/usr/bin/env python
from numpy import *

m = open('test.cat').readlines()
m=m[9:]
fwhm = []
for i in m:
 fwhm.append(i.split()[8])

fwhm = array(fwhm,float)
mefw = median(fwhm)
maxfwhm = median(fwhm)*5.
a5 = (1.5 * mefw) / 2.
a2 = (2 * mefw) / 2.
a3 = (2.5 * mefw) / 2.
a4 = (3 * mefw) / 2.
a1 = (3.8 * mefw) / 2.
a6 = (4.7 * mefw) / 2.
a7 = (5.6 * mefw) / 2.
a8 = (6.5 * mefw) / 2.
a9 = (7 * mefw) / 2.
a10 = (mefw) / 2.
iss = mefw * 7.
os = mefw * 18.
```

```
open('sets','w').write(str(round(mefw,1))+'\t'+str(round(maxfwhm,1))+'\t'+ \
str(round(a1,1))+'\t'+str(round(a2,1))+'\t'+str(round(a3,1))+'\t'+ \
str(round(a4,1))+'\t'+str(round(a5,1))+'\t'+str(round(a6,1))+'\t'+ \
str(round(a7,1))+'\t'+str(round(a8,1))+'\t'+str(round(a9,1))+'\t'+ \
str(round(a10))+'\t'+str(round(iss))+'\t'+str(round(os)))
```

.1.7 timer.py

```
#!/usr/bin/env python

import os, glob, time
from numpy import *
from scipy import *

ra=open('ra.txt').readline()
rahour=float(ra.split()[0])
ramin=float(ra.split()[1])
rasec=float(ra.split()[2])

dec=open('dec.txt').readline()
decdeg=float(dec.split()[0])
decmin=float(dec.split()[1])
decsec=float(dec.split()[2])

ut=open('ut.txt').readline()
hour=float(ut.split()[0])
min=float(ut.split()[1])
sec=float(ut.split()[2])

date=open('date.txt').readline()
year=float(date.split()[0])
month=float(date.split()[1])
day=float(date.split()[2])


lat = 40.43277          #latitude Lat/lon ....
```

```
lon = −23.50527          #longintute ( − ) means east ( + ) means west

#Find Coordinates in decemical form
ra = rahour + (ramin / 60) + (rasec / 3600)
if 0 > decdeg:
  decdeg =  abs(decdeg)
  dec = −(decdeg + (decmin / 60) + (decsec/3600))
else:
  dec = decdeg + (decmin / 60) + (decsec/3600)



#Compute UT time − Universal Time
ut = hour +( min/60) + (sec/3600)

#Compute Julian Date
jd = 367∗year − int((( year + int((month + 9) / 12 ))) ∗ 1.75)
jd = jd + int((month) ∗ 30.55555 ) + day + 1721013.5 + (ut / 24)

#Compute GMST − Greenwich Mean Sidereal Time
jdo = jd − (ut/24)
d = jdo − 2451545
gmst = 6.69738 + (0.0657 ∗ d) #for H = 0

#Compute LST − Local Sidereal Time
lst = (gmst − (lon∗0.0666666666666667)) + (1.00274 ∗ ut) #The ∗0.6 is to \
convert lon to hours

#Compute Hour Angle
hourangle = lst − ra
hourangle = hourangle ∗ 15

#Compute Air Mass
la = lat ∗ 0.01745329   #Convert to radians #same for la
de = dec ∗ 0.01745329   #Convert to radians #de is a new variable for leaving \
dec value without change
hourangle = hourangle ∗ 0.01745329
airmass = (sin(la)∗sin(de))+(cos(la)∗cos(de)∗cos(hourangle))
```

```
airmass = 1 / airmass

#Compute Correction for low altitudes
a = airmass -0.0018167*(airmass -1)-0.002875*(airmass -1)**2
airmass = a - 0.0008083 * (airmass - 1 )**3

open('jdates.txt','w').write(str(jd))
open('airmass.txt','w').write(str(airmass))
```

.1.8 ref.py

```
#!/usr/bin/env python
import os,glob
from numpy import *

refs = []
for file in glob.glob('*.ap'):
 m=open(file).readlines()
 m=m[4:]
 del m[2::3]
 i=0
 m=array(m)
 list1 = []
 while i < len(m):
  k = column_stack((m[i].split()[-10:],m[i+1].split()[-10:]))
  k = concatenate(k)
  z = m[i].split()[:3]
  z = z + list(k)
  list1.append(z)
  i=i+2
 refs.append((file.replace('ap','list'),len(list1)))
 w=open(file.replace('ap','list'),'w')
 for ii in list1:
  for j in ii:
   w.write(str(j)+'\t')
  w.write('\n')
```

```
w.close()
```

.1.9 matcher.csh

```
#!/bin/csh

#pipe line code
##########################################################
set main=`pwd`
echo '2' > check.txt
set main=`pwd`
set dirt=`cat temp/temp.txt`
set dir=`cat temp/output.txt`
set inp=`cat temp/input.txt`
cd $dirt
echo '' > err
cd $dir/photometry
###########################################################
echo "choose your reference file"

set referr=$<

cat $referr>referance

foreach file(*.list)

echo
echo
echo
echo "        Doing file $file"
echo
set f=$file:r

match $file 1 2 3 referance 1 2 3 outfile=$f id1=0 id2=0 cubic \
trirad=0.0001 matchrad=10 nobj=100
clear
```

```
end

##################################################
cd $dirt
rm -f err
cd $main
echo '3' > check.txt
##################################################
```

.1.10 curverer.py

```
#!/usr/bin/env python

import os, glob, time
from numpy import *
from scipy import *

# Definition of the sfm function which is calculating the sum
# of the absolute diferance between the elemnents of the given array

def sfm(aray):
 if not aray:
  return 1000
 if mean(aray[:-1]) == 0 or mean(aray[:1]) == 0:
  return 1000
 aj=aray[:-1]/mean(aray[:-1])-aray[1:]/mean(aray[1:])
 ak=sum(abs(aj))
 return ak

# Determination of the main and output directory
main=os.getcwd()
open('check.txt','w').write('3')
dir=open('temp/output.txt').readline()
os.chdir(dir+'/temp')
open('err','w').write(' ')
os.chdir(dir+'/photometry')
```

```

# Timer start
i=1
to=time.time()
tkkm=[]

# Searching for identical matces for stars between the reference image and the
# others in the output files of match algorithm

ref=open('referance').readlines()
attr=[]
mtb=[]
mta=[]
ptr=[]
for z in glob.glob('*.mtB'):
 tim=open(z.replace('mtB','jd')).readline().split()[0]
 air=open(z.replace('mtB','air')).readline().split()[0]
 attr.append(tim+' '+air+' '+z)
 mt=[]
 tt=[]
 pt=[]
 for kl,tl in zip(open(z).readlines(),open(z.replace('mtB','mtA')).readlines()):
  mt.append(kl.split()[0])
  tt.append(tl)
 for pl in open(z.replace('mtB','list')).readlines():
  pt.append(pl)
# mtb.append(z)
 mtb.append(mt)
# mta.append(z)
 mta.append(tt)
# ptr.append(z)
 ptr.append(pt)

#Seting the coordinates reference image
refim=z.replace('mtB','sdf')
os.system('cp ../ndfdata/%s .' %(refim))
os.system('mv %s refim.sdf' %(refim))
```

```
# Curve creation for each star
while i < len(ref)+1:
# check = os.path.isfile(str(i)+'.lc')
# if check == True:
#   i=i+1
#   continue
    t=time.time()              #Timer stats
    tkm=t-to                   #
    to=time.time()             #
    tkkm.append(float(tkm))    #
    rest=mean(tkkm)*(len(ref)-i)  #
    hh=rest/3600               #
    mm=(hh-floor(hh))*60       #
    ss=(mm-floor(mm))*60       #
    lc = []
    ap1 = []
    er1 = []
    ap2 = []
    er2 = []
    ap3 = []
    er3 = []
    ap4 = []
    er4 = []
    ap5 = []
    er5 = []
    ap6 = []
    er6 = []
    ap7 = []
    er7 = []
    ap8 = []
    er8 = []
    ap9 = []
    er9 = []
    ap10 = []
    er10 = []
    x=open('referance').readlines()[i-1].split()[1]
    y=open('referance').readlines()[i-1].split()[2]
```

```
  for zk,gk,tk,pk in zip(attr,mtb,mta,ptr):
   z = zk.split()[2]
   tim = zk.split()[0]
   air = zk.split()[1]
   if str(i) in gk:
     ind = int(gk.index(str(i)))
     kr=tk[ind].split()[0]
     lck = pk[int(kr)-1].split()
#      if float(lck[9]) > 30.:          # remove bad measurements for Daophot
#        continue
#      x=float()
#      y=float(gk[ind].split()[2])
#      x=floor(x) #Coordinates of the star
#      y=floor(y) #
     lc.append(tim+' '+air)
#      if lck[3] != '99.999':
     ap1.append(float(lck[3]))
     er1.append(float(lck[4]))
#      if lck[5] != '99.999':
     ap2.append(float(lck[5]))                  # remove bad measurements for Daophot
     er2.append(float(lck[6]))
#      if lck[7] != '99.999':
     ap3.append(float(lck[7]))
     er3.append(float(lck[8]))
#      if lck[9] != '99.999':
     ap4.append(float(lck[9]))
     er4.append(float(lck[10]))
#      if lck[11] != '99.999':
     ap5.append(float(lck[11]))
     er5.append(float(lck[12]))
#      if lck[13] != '99.999':
     ap6.append(float(lck[13]))
     er6.append(float(lck[14]))
#      if lck[15] != '99.999':
     ap7.append(float(lck[15]))
     er7.append(float(lck[16]))
#      if lck[17] != '99.999':
```

```
      ap8.append(float(lck[17]))
      er8.append(float(lck[18]))
#      if lck[19] != '99.999':
      ap9.append(float(lck[19]))
      er9.append(float(lck[20]))
#      if lck[21] != '99.999':
      ap10.append(float(lck[21]))
      er10.append(float(lck[22]))



# Check which one of the photometry apertures seems to be better
  if not lc:
   i=i+1
   continue
  po = array((sfm(ap1)/(len(ap1)+1.),sfm(ap2)/(len(ap2)+1.), \
sfm(ap3)/(len(ap3)+1.),sfm(ap4)/(len(ap4)+1.),sfm(ap5)/(len(ap5)+1.), \
sfm(ap6)/(len(ap6)+1.),sfm(ap7)/(len(ap7)+1.),sfm(ap8)/(len(ap8)+1.), \
sfm(ap9)/(len(ap9)+1.),sfm(ap10)/(len(ap10)+1.)),dtype=float64)
  ppo = min(po)
  if ppo == nan or ppo == 1000:
   i=i+1
   continue
  ppr = list(po).index(ppo)+1
  if ppr == 1:
   lap = ap1
   ler = er1
  elif ppr == 2:
   lap = ap2
   ler = er2
  elif ppr == 3:
   lap = ap3
   ler = er3
  elif ppr == 4:
   lap = ap4
   ler = er4
  elif ppr == 5:
   lap = ap5
```

```
    ler = er5
  elif ppr == 6:
    lap = ap6
    ler = er6
  elif ppr == 7:
    lap = ap7
    ler = er7
  elif ppr == 8:
    lap = ap8
    ler = er8
  elif ppr == 9:
    lap = ap9
    ler = er9
  elif ppr == 10:
    lap = ap10
    ler = er10

# lap = ap1
# ler = er1
  lcc=[]
# x=[]
# y=[]
  for zr,zzr,zzzr in zip(lc,lap,ler):
    if zzr != 99.999:
      lcc.append(zr.split()[0]+'\t'+str(zzr)+'\t'+str(zzzr)+'\t'+zr.split()[1]+'\n')
  lcc.sort()
  lcname = str(round(mean(ap1),2))+'_'+x+'_'+y+'.lc'
  lct=open(lcname,'w')
  for ziyr in lcc:
    lct.write(ziyr)
  lct.close()
  print lcname ,len(lc),'  ',int(floor(hh)), 'h', int(floor(mm)), 'm', \
round(ss,2),'s'
  i=i+1

os.remove('../temp/err')
os.chdir(main)
```

```
open('check.txt','w').write('4')
```

.1.11 sigma.py

```
#!/usr/bin/env python
import os, glob, os.path, math

def col(var,num):
 cool=[]
 for i in var:
  cool.append(i.split()[num])
 return(cool)

main=os.getcwd()
open('check.txt','w').write('4')
dir=open('temp/output.txt').readline()
os.chdir(dir+'/temp')
open('err','w').write(' ')
os.chdir(dir+'/photometry')


os.system('cp %s/scripts/errorf.py .' %(main))
check = os.path.isfile('sigmaplot.txt')
if check == True:
  os.system('rm c*.lc')
  os.system('rm sigmaplot.txt')
check = os.path.isfile('junk')
if check == True:
  os.remove('junk')
time=open('time.txt').readlines()
air=open('air.txt').readlines()
for curve in glob.glob('*.lc'):
  if len(open(curve).readlines()) < 0.5*len(time):
    os.remove(curve)
    os.system('echo %s >> removed' %(curve))
    print "CHECKING NUMBER OF POINTS ON FILE ",curve,"......REMOVED"
```

```
    elif len(open(curve).readlines()) > len(time):
        os.remove(curve)
        os.system('echo %s >> removed' %(curve))
        print "CHECKING NUMBER OF POINTS ON FILE ",curve,"......REMOVED"
    else:
        print "CHECKING NUMBER OF POINTS ON FILE ",curve,"......OK"
    os.system('clear')
s=open('sigmaplot.txt','w')
for curve in glob.glob('*.lc'):
    n=len(open(curve).readlines())
    os.system('echo %s > n' %(n))
    os.system('echo %s > q' %(n))
    os.system("awk '{print $1}' %s > t" %(curve))
    os.system("awk '{print $2}' %s > mag" %(curve))
    os.system("awk '{print $3}' %s > error" %(curve))
    mag=[]
    for line in open('mag').readlines():
        mag.append(float(line))

    os.system('./errorf.py intel64')
    meant=open('mean.txt').readlines()
    sigma=open('sigma.txt').readlines()
    inner=open('inner.txt').readlines()
    pulse=open('pulse.txt').readlines()
    rms=open('rms.txt').readlines()
    lrms=open('logrms.txt').readlines()
    t=open('t').readlines()
    ccurves=open(curve).readlines()
    if len(t) == len(time):
        os.system('cat %s > junkt' %(curve))
    if len(t) < len(time):
        c=open('junkt','w')
        i=0
        k=0
        while i < len(time):
            if time[i] != t[k]:
                c.write(time[i].strip())
```

```
      c.write(' ')
      c.write(meant[0].strip())
      c.write(' ')
      c.write(sigma[0].strip())
      c.write(' ')
      c.write(air[i].strip())
      c.write('\n')
      i=i+1
    else:
      c.write(ccurves[k])
      i=i+1
      k=k+1
      if k >= len(t):
        k=k-1
  c.close()
s.write(meant[0].strip())
s.write('\t')
s.write(sigma[0].strip())
s.write('\t')
s.write(inner[0].strip())
s.write('\t')
s.write(pulse[0].strip())
s.write('\t')
s.write(rms[0].strip())
s.write('\t')
s.write(lrms[0].strip())
s.write('\t')
s.write(curve+'\n')
os.system('mv junkt c%s' %(curve))
os.system('clear')
print
print
print
print
print "            FILE ",curve ,"UNDER ANALYSIS"
print "            CALCULATED MEAN : ",round(float(meant[0].strip()),3)
print "            CALCULATED SIGMA: ",round(float(sigma[0].strip()),3)
```

```
  print "            CALCULATED PULSATION: ",round(float(pulse[0].strip()),3)
s.close()
os.remove('mean.txt')
os.remove('sigma.txt')
os.remove('inner.txt')
os.remove('pulse.txt')
os.remove('rms.txt')
os.remove('logrms.txt')
#os.remove('xycoords.txt')
#os.remove('all.txt')
os.remove('dec.txt')
os.remove('ra.txt')
os.remove('airmass.txt')
os.remove('errorf.py')
os.remove('timer.py')
os.system('rm -f *.mt*')
os.system('rm -f *.un*')
os.remove('t')
#os.remove('x')
#os.remove('y')
os.remove('q')
os.remove('n')
os.remove('mag')
#os.remove('file')
os.remove('error')
os.system('rm -f *.ap')
os.system('rm -f *.cat')
os.remove('../temp/err')
os.system('mkdir uncurves')
os.system('mv 1[1-4]*.lc uncurves/')
os.system('mv 1[4-8]*.lc uncurves/')
os.system('mv [1-9]*.lc uncurves/')
os.system('mv removed uncurves/')
os.system('mkdir lists')
os.system('mv *.list lists/')
os.system('mv referance lists/')
os.system('mkdir time_air')
```

```
os.system('mv *.jd time_air/')
os.system('mv *.air time_air/')
os.chdir(main)
if os.path.exists(main+'/check.txt'):
 os.remove(main+'/check.txt')
```

.1.12 errorf.py

```
#!/usr/bin/env python

import os, glob
from numpy import *
from scipy import *

def stdfa(aray):
 aj=aray[:-1]-aray[1:]
 return aj

mag = []
for i in open('mag').readlines():
 mag.append(i.split()[0])

err = []
for i in open('error').readlines():
 err.append(i.split()[0])

mag = array(mag , dtype = float64)
err = array(err , dtype = float64)


s1 = sum(1/(err**2))
s2 = sum(mag/(err**2))
mea1 = s2/s1
n = float(len(mag))

mssum = sum((mag - mea1)**2.0)

```

```
msigma = sqrt(mssum/n)
minner = mean(abs(stdfa(mag)))

mpulse = msigma/minner
mrms = std(2.5*log(mag))
frms = std(mag)
mlrms = log(mrms)


open('mean.txt','w').write(str(round(mean(mag),3)))
open('sigma.txt','w').write(str(round(msigma,3)))
open('inner.txt','w').write(str(round(minner,3)))
open('pulse.txt','w').write(str(round(mpulse,3)))
open('rms.txt','w').write(str(round(frms,3)))
open('logrms.txt','w').write(str(round(mlrms,3)))
```

.2 Software tools

.2.1 Transit Lightcurve Generetor

```
#!/usr/bin/env python

import numpy
import math
import fileinput
from scipy import integrate
import random
import pylab as pl



pi = 3.1415

def ellk(k,var):
  f1v = 1.0-(k*math.sin(var))**2
  f1k = 1.0/(f1v**0.5)
  return f1k


def elle(k,var):
  f1v = 1.0-(k*math.sin(var))**2
  f1e = f1v**0.5
  return f1e

def ellp(n,k,var):
  f1v = 1.0-n*(math.sin(var))**2.0
  f1k = (1.0-k*k*math.sin(var)*math.sin(var))**0.5
  f1p = 1.0/(f1v*f1k)
  return f1p


def dff(z):               #define the 12 cases as described from Pal[2008]
  flag = 0
  flux = 0.0
```

```
if flag == 0:
    if z == 0.0:
        if p < 1.0:
            case = 1
            flag = 1
if flag == 0:
    if z <= p - 1.0:
        case = 2
        flag = 1
if flag == 0:
    if (z < 1.0 - p) and (z < p):
            case = 3
            flag = 1
if flag == 0:
    if (z < p) and (z == 1.0 - p):
        case = 4
        flag = 1
if flag == 0:
    if z < p:
        case = 5
        flag = 1
if flag == 0:
    if (z == p) and (z < 1.0-p):
        case = 6
        flag = 1
if flag == 0:
    if (z == 0.5) and (p == 0.5):
        case = 7
        flag = 1
if flag == 0:
    if z == p:
        case = 8
        flag = 1
if flag == 0:
    if z < 1.0 - p:
        case = 9
        flag = 1
```

```
    if flag == 0:
      if z == 1.0 - p:
        case = 10
        flag = 1
    if flag == 0:
      if z < 1.0 + p:
        case = 11
        flag = 1
    if flag == 0:
      case = 12
    if case == 1:
      pt = (1.0-p*p)**0.5
      f0 = p*p
      f1 = 2.0*(1.0-pt*pt*pt)/3.0
      f2 = 0.5*p**4
      df = w0*f0+w1*f1+w2*f2
      flux = 1.0-df
    if case == 2:
      pt = (1.0-p**2)**0.5
      f0 = 1.0
      f1 = 2.0/3.0
      f2 = 0.5
      df = w0*f0+w1*f1+w2*f2
      flux = 1.0-df
    if case == 3:
      alp = (p-z)**2
      bet = (p+z)**2
      n = -4.0*p*z/alp
      k = (4.0*p*z/(1.0-alp))**0.5
      ci = 9.0*pi*(1.0-alp)**0.5
      ci = 2.0/ci
      cik = 1.0-5.0*(z**2)+p**2+alp*bet
      cie = (z**2+7.0*(p**2)-4.0)*(1.0-alp)
      cip = -3.0*(p+z)/(p-z)
      f0 = p**2
      f1 = 2.0/3.0
      f2 = 0.5*(p**2)*(p**2+2.0*z**2)
```

```
        fk = ci*cik
        fe = ci*cie
        fp = ci*cip
        rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
        elk = float(rk[0])
        re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
        ele = float(re[0])
        elrp = integrate.quad(lambda var: ellp(n,k,var), 0,1.570796)
        elp = float(elrp[0])
        df1 = w0*f0+w2*f2
        df2 = w1*(f1+fk*elk+fe*ele+fp*elp)
        df = df1+df2
        flux = 1.0-df
    if case == 4:
        pt = (p*(1.0-p))**0.5
        ti1 = (2.0*math.acos(1.0-2.0*p))/3.0*pi
        ti2 = 4.0*(3.0+2.0*p-8.0*p**2)*pt/9.0*pi
        ti = ti1-ti2
        f0 = p**2
        f1 = ti
        f2 = 0.5*(p**2)*(p**2+2.0*z**2)
        df = w0*f0+w2*f2+w1*f1
        flux = 1.0-df
    if case == 5:
        alp = (p-z)**2
        bet = (p+z)**2
        n = (alp-1.0)/alp
        k = ((1.0-alp)/(4.0*p*z))**0.5
        k0 = (p**2+z**2-1.0)/(2.0*p*z)
        k0 = math.acos(k0)
        k1 = (z**2+1.0-p**2)/(2.0*z)
        k1 = math.acos(k1)
        g0 = k0*p**2+k1-(z**2-0.25*(1.0+z**2-p**2)**2)**0.5
        g0 = g0/pi
        cg = 9.0*pi*(p*z)**0.5
        cg = 1.0/cg
        cgk = 3.0-6.0*(1.0-p**2)**2
```

```
    cgk = cgk-2.0*p*z*(z**2+7.0*p**2-4.0+5.0*p*z)
    cge = 4.0*p*z*(z**2+7.0*p**2-4.0)
    cgp = -3.0*(p+z)/(p-z)
    g2 = k1+(p**2)*(P**2+2.0*z**2)*k0
    g2 = g2-0.25*(1.0+5.0*p**2+z**2)*((1-alp)*(bet-1.0))**0.5
    g2 = g2/(2.0*pi)
    f0 = g0
    f1 = 2.0/3.0
    f2 = g2
    fk = cg*cgk
    fe = cg*cge
    fp = cg*cgp
    rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
    elk = float(rk[0])
    re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
    ele = float(re[0])
    elrp = integrate.quad(lambda var: ellp(n,k,var), 0,1.570796)
    elp = float(elrp[0])
    df1 = w0*f0+w2*f2
    df2 = w1*(f1+fk*elk+fe*ele+fp*elp)
    df = df1+df2
    flux = 1.0-df
  if case == 6:
    k = 2.0*p
    f0 = p**2
    f1 = 1.0/3.0
    f2 = (3.0*p**4)/2.0
    fk = 2.0*(1.0-4.0*p**2)/(9.0*pi)
    fe = 8.0*(2.0*p**2-1.0)/(9.0*pi)
    rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
    elk = float(rk[0])
    re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
    ele = float(re[0])
    df1 = w0*f0+w2*f2
    df2 = w1*(f1+fk*elk+fe*ele)
    df = df1+df2
    flux = 1.0-df
```

```
    if case == 7:
        f0 = 0.25
        f1 = (1.0/3.0)-(4.0/9.0*pi)
        f2 = 3.0/32.0
        df1 = w0*f0+w2*f2
        df2 = w1*f1
        df = df1+df2
        flux = 1.0-df
    if case == 8:
        alp = (p-z)**2
        bet = (p+z)**2
        k = 1.0/(2.0*p)
        k0 = (p**2+z**2-1.0)/(2.0*p*z)
        k0 = math.acos(k0)
        k1 = (z**2+1.0-p**2)/(2.0*z)
        k1 = math.acos(k1)
        g0 = k0*p**2+k1-(z**2-0.25*(1.0+z**2-p**2)**2)**0.5
        g0 = g0/pi
        g2 = k1+(p**2)*(P**2+2.0*z**2)*k0
        g2 = g2-0.25*(1.0+5.0*p**2+z**2)*((1-alp)*(bet-1.0))**0.5
        g2 = g2/(2.0*pi)
        f0 = g0
        f1 = 1.0/3.0
        f2 = g2
        fk = (1.0-4.0*p*p)*(3.0-8.0*p*p)
        fk = -fk/(9.0*pi*p)
        fe = 16.0*p*(2.0*p*p-1.0)
        fe = fe/(9.0*pi)
        rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
        elk = float(rk[0])
        re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
        ele = float(re[0])
        df1 = w0*f0+w2*f2
        df2 = w1*(f1+fk*elk+fe*ele)
        df = df1+df2
        flux = 1.0-df
    if case == 9:
```

```
    alp = (p-z)**2
    bet = (p+z)**2
    n = -4.0*p*z/alp
    k = (4.0*p*z/(1.0-alp))**0.5
    ci = 9.0*pi*(1.0-alp)**0.5
    ci = 2.0/ci
    cik = 1.0-5.0*(z**2)+p**2+alp*bet
    cie = (z**2+7.0*(p**2)-4.0)*(1.0-alp)
    cip = -3.0*(p+z)/(p-z)
    f0 = p**2
    f1 = 0.0
    f2 = 0.5*(p**2)*(p**2+2.0*z**2)
    fk = ci*cik
    fe = ci*cie
    fp = ci*cip
    rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
    elk = float(rk[0])
    re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
    ele = float(re[0])
    elrp = integrate.quad(lambda var: ellp(n,k,var), 0,1.570796)
    elp = float(elrp[0])
    df1 = w0*f0+w2*f2
    df2 = w1*(f1+fk*elk+fe*ele+fp*elp)
    df = df1+df2
    flux = 1.0-df
  if case == 10:
    pt = (p*(1.0-p))**0.5
    ti1 = (2.0*math.acos(1.0-2.0*p))/3.0*pi
    ti2 = 4.0*(3.0+2.0*p-8.0*p**2)*pt/9.0*pi
    ti = ti1-ti2
    f0 = p**2
    f1 = ti
    f2 = 0.5*(p**2)*(p**2+2.0*z**2)
    df = w0*f0+w2*f2+w1*f1
    flux = 1.0-df
  if case == 11:
    alp = (p-z)**2
```

```
    bet = (p+z)**2
    n = (alp-1.0)/alp
    k = ((1.0-alp)/(4.0*p*z))**0.5
    k0 = (p*p+z*z-1.0)/(2.0*p*z)
    k0 = math.acos(k0)
    k1 = (z**2+1.0-p**2)/(2.0*z)
    k1 = math.acos(k1)
    g0 = k0*p*p+k1-(z*z-0.25*(1.0+z*z-p*p)**2)**0.5
    g0 = g0/pi
    cg = 9.0*pi*(p*z)**0.5
    cg = 1.0/cg
    cgk = 3.0-6.0*(1.0-p*p)**2
    cgk = cgk-2.0*p*z*(z*z+7.0*p*p-4.0+5.0*p*z)
    cge = 4.0*p*z*(z*z+7.0*p*p-4.0)
    cgp = -3.0*(p+z)/(p-z)
    g2 = k1+p*p*(p*p+2.0*z*z)*k0
    g2v = ((1-alp)*(bet-1.0))**0.5
    g2v = g2v*0.25*(1.0+5.0*p*p+z*z)
    g2 = g2-g2v
    g2 = g2/(2.0*pi)
    f0 = g0
    f1 = 0.0
    f2 = g2
    fk = cg*cgk
    fe = cg*cge
    fp = cg*cgp
    rk = integrate.quad(lambda var: ellk(k,var), 0,1.570796)
    elk = float(rk[0])
    re = integrate.quad(lambda var: elle(k,var), 0,1.570796)
    ele = float(re[0])
    elrp = integrate.quad(lambda var: ellp(n,k,var), 0,1.570796)
    elp = float(elrp[0])
    df1 = w0*f0+w2*f2
    df2 = w1*(f1+fk*elk+fe*ele+fp*elp)
    df = df1+df2
    flux = 1.0-df
  if case == 12:
```

```
    flux = 1.0
  return flux



lliimm = list(i.split()[1] for i in open('input').readlines())
#   Read the 2nd column of the input file which must contain, in the same order:
#   Star mass, planet mass, star radius, planet radius, period, u1 limb
#   darkening coefficient, u2 limb darkening coefficient,
#   inclination, number of days, exposure of each frame (only for
#   time seperation reasons), expected log(rms)
#   i.e. ms     1.2      #star mass Jupiters

lliimm = numpy.array(lliimm,dtype=float)
ms = lliimm[0]
mp = lliimm[1]
rs = lliimm[2]
rp = lliimm[3]
per = lliimm[4]
u1 = lliimm[5]
u2 = lliimm[6]
inc = pi*lliimm[7]/180.0
rms = lliimm[10]
rms = 10**(rms)

step = int(lliimm[9])
tot = int(86400.0*lliimm[8])

semi = ms*(per/365.0)**2
semi = semi**0.333

bb = 214.08*semi/rs
p = 0.10271839*rp/rs

w = 6.0-2.0*u1-u2
w0 = (6.0-6.0*u1-12.0*u2)/w
```

```
w1 = (6.0*u1+12.0*u2)/w
w2 = 6.0*u2/w


eps = 0.0
mean = 0.0
sd = rms

tz=[]
mz=[]
md=[]
file = open('new_model', 'w')
nfile = open('model','w')
pfile = open('phase','w')
for i in range(0,tot,step):
    noise = random.gauss(mean,sd)   #comment this to prevent from makng noise
    tt = i/86400.0
    peri = pi/per
    z1 = bb*(math.sin(peri*(tt-eps)))**2
    z2 = bb*(math.cos(peri*(tt-eps)))**2
    z2 = z2*math.cos(inc)*math.cos(inc)
    z = (z1+z2)**0.5
    klo = 1.0
    if z <= 1.0+p:
        klo = dff(z)
    if klo <= 1.0:
        tz.append(tt)
        mz.append(klo+noise)
        md.append(klo)
        file.write(str('%s %s\n') % (tt, klo+noise))
        nfile.write(str('%s %s\n') % (tt, klo)) # the model without the noise
file.close()

pl.plot(tz,mz,'.')       #plot the resulted light curve
pl.plot(tz,md,'r-')      # with its model

raw_input()
```

```
raise SystemExit
```

.2.2 Period Propability

```
#!/usr/bin/env python

#----- PPD -------

from numpy import *
from pylab import *

ans=0
while ans != 'y' and ans != 'n':
  ans=raw_input('Real time data in jd (y/n):  ')

if ans == 'y':
  ans=raw_input('Please insert the file name:  ')
  t=open(ans).read().split()
else:
  inpt= array(list(i.split()[1] for i in open('dinput').readlines()), float)
#   Read the 2nd column of the input file which must contain, in the same\
    order:
#   Days, Hours per night, exposure of frames in sec
#   i.e. hpn     2      #Hours per night

  days=int(inpt[0])
  hpn=inpt[1]
  expo=inpt[2]
  hpn=hpn/24.
  tt= arange(2455000.375,2455000.375+hpn,expo/(24*3600.))
  for i in range(days):
   tt= append(tt,tt+1)
  t=tt

time = t
```

```
print 'Define period range in days'
low=raw_input('lower:  ')
max=raw_input('maximum:  ')
ps=float(low)
pe=float(max)
pp=0.001
time= array(list(float(str(j)[:11]) for j in time))
timtab= arange(time[0],round(time[-1]),0.001)
timtab=list(str(j)[:11] for j in timtab)
re= zeros(len(timtab))
ma= zeros(len(timtab))
for i in range(len(re)):
    if round(float(timtab[i]),3) in time:
        pos= where(time==float(str(timtab[i])[:11]))[0][0]
        re[i]=float(str(time[pos])[:11])
        ma[i]=1

pertab= arange(ps,pe,pp)
perprob=[]
test=[]
for ij in pertab:
    mas=ma
    l=floor(ij/0.001)
    a=len(mas)%l
    perval= append(mas,zeros(l-a))
    b=len(perval)/l
    perval=perval.reshape(b,l)
    val=perval.sum(axis=0)
    med=perval.transpose()
    zer=float(len(where(val==0)[0]))
    perprob.append(1-zer/len(val))

cla()
perprob=array(perprob)
semilogx(pertab,perprob)
xlabel('period (days)')
ylabel('Period Detection Probability')
```

```
file=open('ppd_out.txt','w')
for i,j in zip(pertab,perprob):
    file.write(str(i)+'\t'+str(j)+'\n')

file.close()
raw_input()

raise SystemExit
```

References

T. Althaus. Book Review: Giant planets of our solar system. Atmospheres, composition and structure / Springer, Heidelberg, 2003, XXVI+362 pp., ISBN 3-540-00681-8. *Sterne und Weltraum*, 43(1):010000, January 2004.

J. Antoniadis, V. Karamanavis, D. Mislis, A. Nitsos, and J. H. Seiradakis. ThReT: A New Survey for Extrasolar Planetary Transits at Mt. Holomon Greece. In K. Tsinganos, D. Hatzidimitriou, and T. Matsakos, editors, *9th International Conference of the Hellenic Astronomical Society*, volume 424 of *Astronomical Society of the Pacific Conference Series*, page 218, July 2010.

D. P. Bennett. *Detection of Extrasolar Planets by Gravitational Microlensing*, pages 47–88. Springer, 2008.

D. P. Bennett and S. H. Rhie. Detecting Earth-Mass Planets with Gravitational Microlensing. *ApJ*, 472:660, November 1996.

J. Chambers. Planet Formation with Type I and Type II Migration. In *AAS/-Division of Dynamical Astronomy Meeting #38*, AAS/Division of Dynamical Astronomy Meeting, July 2007.

D. Charbonneau, T. M. Brown, R. W. Noyes, and R. L. Gilliland. Detection of an Extrasolar Planet Atmosphere. *ApJ*, 568:377–384, March 2002.

D. Deming, J. Harrington, G. Laughlin, S. Seager, S. B. Navarro, W. C. Bowman, and K. Horning. Spitzer Transit and Secondary Eclipse Photometry of GJ 436b. *ApJ*, 667:L199–L202, October 2007.

D. Deming, S. Seager, L. J. Richardson, and J. Harrington. Infrared radiation from an extrasolar planet. *Nature*, 434:740–743, March 2005.

R. L. Gilliland, T. M. Brown, P. Guhathakurta, A. Sarajedini, E. F. Milone, M. D. Albrow, N. R. Baliber, H. Bruntt, A. Burrows, D. Charbonneau, P. Choi, W. D. Cochran, P. D. Edmonds, S. Frandsen, J. H. Howell, D. N. C. Lin, G. W. Marcy, M. Mayor, D. Naef, S. Sigurdsson, C. R. Stagg, D. A. VandenBerg, S. S. Vogt, and M. D. Williams. A Lack of Planets in 47 Tucanae from an HST Search. *ArXiv Astrophysics e-prints*, September 2000.

R. Gomes, H. F. Levison, K. Tsiganis, and A. Morbidelli. Origin of the cataclysmic late heavy bombardment period of the terrestrial planets. *Nature*, 435: 466–469, May 2005.

J. Harrington, B. M. Hansen, S. H. Luszcz, S. Seager, D. Deming, K. Menou, J. Y.-K. Cho, and L. J. Richardson. The Phase-Dependent Infrared Brightness of the Extrasolar Planet υ Andromedae b. *Science*, 314:623–626, October 2006.

G. Hébrard, A. Lecavelier Des Étangs, A. Vidal-Madjar, J.-M. Désert, and R. Ferlet. Evaporation Rate of Hot Jupiters and Formation of Chthonian Planets. In J. Beaulieu, A. Lecavelier Des Etangs, & C. Terquem, editor, *Extrasolar Planets: Today and Tomorrow*, volume 321 of *Astronomical Society of the Pacific Conference Series*, page 203, December 2004.

I. D. Howarth. On stellar limb darkening and exoplanetary transits. *MNRAS*, 418:1165–1175, December 2011.

H. A. Knutson, D. Charbonneau, L. E. Allen, A. Burrows, and S. T. Megeath. The 3.6-8.0 μm Broadband Emission Spectrum of HD 209458b: Evidence for an Atmospheric Temperature Inversion. *ApJ*, 673:526–531, January 2008.

H. Lammer, J. F. Kasting, E. Chassefière, R. E. Johnson, Y. N. Kulikov, and F. Tian. *Atmospheric Escape and Evolution of Terrestrial Planets and Satellites*, page 399. Springer, 2009. URL `http://adsabs.harvard.edu/abs/2009coae.book..399L`.

B. R. Oppenheimer, A. Sivaramakrishnan, and R. B. Makidon. Imaging Exoplanets: The Role of Small Telescopes. In T. D. Oswalt, editor, *Astrophysics and Space Science Library*, volume 289 of *Astrophysics and Space Science Library*, page 155, June 2003.

A. Pál. Properties of analytic transit light-curve models. *mnras*, 390:281–288, October 2008.

R. O. Pepin. On the origin and early evolution of terrestrial planet atmospheres and meteoritic volatiles. *icarus*, 92:2–79, July 1991.

S. H. Rhie and D. P. Bennett. Search for Earth Mass Planets and Dark Matter Too. *Nuclear Physics B Proceedings Supplements, Vol. 51*, 51:86–90, November 1996.

J. D. Scargle. Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data. *ApJ*, 263:835–853, December 1982.

L. E. Strigari, M. Barnabe, P. J. Marshall, and R. D. Blandford. Nomads of the Galaxy. *ArXiv e-prints*, January 2012.

T. Sumi, D. P. Bennett, I. A. Bond, A. Udalski, V. Batista, M. Dominik, P. Fouqué, D. Kubas, A. Gould, B. Macintosh, K. Cook, S. Dong, L. Skuljan, A. Cassan, F. Abe, C. S. Botzler, A. Fukui, K. Furusawa, J. B. Hearnshaw, Y. Itow, K. Kamiya, P. M. Kilmartin, A. Korpela, W. Lin, C. H. Ling, K. Masuda, Y. Matsubara, N. Miyake, Y. Muraki, M. Nagaya, T. Nagayama, K. Ohnishi, T. Okumura, Y. C. Perrott, N. Rattenbury, T. Saito, T. Sako, D. J. Sullivan, W. L. Sweatman, P. J. Tristram, P. C. M. Yock, MOA Collaboration, J. P. Beaulieu, A. Cole, C. Coutures, M. F. Duran, J. Greenhill, F. Jablonski, U. Marboeuf, E. Martioli, E. Pedretti, O. Pejcha, P. Rojo, M. D. Albrow, S. Brillant, M. Bode, D. M. Bramich, M. J. Burgdorf, J. A. R. Caldwell, H. Calitz, E. Corrales, S. Dieters, D. Dominis Prester, J. Donatowicz, K. Hill, M. Hoffman, K. Horne, U. G. Jørgensen, N. Kains, S. Kane, J. B. Marquette, R. Martin, P. Meintjes, J. Menzies, K. R. Pollard, K. C. Sahu, C. Snodgrass, I. Steele, R. Street, Y. Tsapras, J. Wambsganss, A. Williams,

M. Zub, PLANET Collaboration, M. K. Szymański, M. Kubiak, G. Pietrzyński, I. Soszyński, O. Szewczyk, Ł. Wyrzykowski, K. Ulaczyk, OGLE Collaboration, W. Allen, G. W. Christie, D. L. DePoy, B. S. Gaudi, C. Han, J. Janczak, C.-U. Lee, J. McCormick, F. Mallia, B. Monard, T. Natusch, B.-G. Park, R. W. Pogge, R. Santallo, and μFUN Collaboration. A Cold Neptune-Mass Planet OGLE-2007-BLG-368Lb: Cold Neptunes Are Common. *ApJ*, 710:1641–1653, February 2010.

WORKING GROUP ON EXTRASOLAR PLANETS (WGESP) OF THE INTERNATIONAL ASTRONOMICAL UNION. Defintion of a "planet", 2003. URL `http://www.dtm.ciw.edu/boss/definition.html`.

F. G. Valdes, L. E. Campusano, J. D. Velasquez, and P. B. Stetson. FOCAS Automatic Catalog Matching Algorithms. *PASP*, 107:1119, November 1995.

J. A. Valenti and D. A. Fischer. Spectroscopic Properties of Cool Stars (SPOCS). I. 1040 F, G, and K Dwarfs from Keck, Lick, and AAT Planet Search Programs. *ApJS*, 159:141–166, July 2005.

W. H. van den Bos. An extra-solar planet? *Monthly Notes of the Astronomical Society of South Africa*, 2:14, 1943.

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