



FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

PHYSIKALISCH-ASTRONOMISCHE FAKULTÄT

ASTROPHYSIKALISCHES INSTITUT UND

UNIVERSITÄTS-STERNWARTE



Die Suche nach Transitzeitvariationen mittels bodengebundener Beobachtungen

—DISSERTATION—

zur Erlangung des Akademischen Grades

Doctor Rerum Naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Physikalisch-Astronomischen Fakultät
der Friedrich-Schiller-Universität Jena

von Dipl.-Phys. MARTIN SEELIGER
geboren am 27. Mai 1985 in Pößneck

Gutachter:

1. Gutachter: Prof. Dr. Ralph Neuhäuser
Friedrich-Schiller-Universität Jena
Astrophysikalisches Institut und Universitäts-Sternwarte
2. Gutachter: Prof. Dr. Stefan Dreizler
Georg-August-Universität Göttingen
Institut für Astrophysik
3. Gutachter: Prof. Dr. Jürgen Schmitt
Universität Hamburg
Hamburger Sternwarte

Tag der Disputation: 04.06.2015

Inhaltsverzeichnis

1	Einleitung	1
1.1	Die Entdeckung extrasolarer Planeten	1
1.2	Die Charakterisierung von Transitplaneten durch Lichtkurven	4
1.3	Transitzeitvariationen	9
1.3.1	Planetare Begleiter als Ursache für TTVs	9
1.3.2	Alternative Quellen für TTV Signale	12
1.4	Ziel der Arbeit	15
2	Datenaufnahme	17
2.1	Auswahl der Beobachtungsobjekte	17
2.2	Beobachtungsstrategie	19
2.2.1	Datenaufnahme an der Universitäts-Sternwarte Jena	21
2.2.2	Datenaufnahme an Partnersternwarten	22
2.2.3	Beobachtungen	24
3	Grundlagen der Datenauswertung und -modellierung	27
3.1	Datenauswertung	27
3.1.1	Reduktion der Rohdaten	28
3.1.2	Erstellung der Lichtkurve mittels differentieller Photometrie	30
3.1.3	Anpassen eines Modells an die Lichtkurve	33
3.1.4	Zeitkorrektur	35
3.2	Modellierung eines hypothetischen Drei-Körper-Systems	37
3.2.1	Mehrkörpersimulation	37
3.2.2	Stabilität des Systems	39
4	Das HAT-P-32 System	41
4.1	Transitbeobachtungen	43
4.2	Modellierung	47
4.3	Simultane Beobachtungen	47

4.4	Transit-Zeit-Variationen	51
4.5	Einschränkungen für hypothetische Begleiter	53
4.5.1	Einschränkungen aus der Radialgeschwindigkeitsschwankung	56
4.5.2	Einschränkungen aufgrund der Beobachtungsabdeckung	58
4.5.3	Zusammenfassung der Systemeinschränkungen	59
4.6	Neubestimmung der Systemparameter	61
5	Weitere Ergebnisse	67
6	Zusammenfassung und Ausblick	75
6.1	Implikationen für die Suche nach TTV-Signalen	77
	Literaturverzeichnis	79
A	Transitlichtkurven	89
A.1	Transitlichtkurven von HAT-P-18b	89
A.2	Transitlichtkurven von HAT-P-19b	90
A.3	Transitlichtkurven von HAT-P-27b/WASP-40b	91
A.4	Transitlichtkurven von HAT-P-32b	92
A.5	Transitlichtkurven von WASP-21b	96
B	Quellcodes	97
B.1	Programme zur Berechnung von Transitzeitpunkten	97
B.2	Programme zur N-Körper Simulation	114
B.3	Programme zur Bestimmung der Stabilität	133
	Danksagung	147
	Ehrenwörtliche Erklärung	149

Abbildungsverzeichnis

1.1	schematische Darstellung möglicher Entdeckungsmethoden für extrasolare Planeten nach Perryman (2000)	2
1.2	Illustration der Radialgeschwindigkeitsmethode aus der ESO Pressemitteilung vom 25. April 2007.	3
1.3	Schematische Darstellung einer Transitlichtkurve	4
1.4	Schematische Darstellung zur Bestimmung des Impaktparameters b	6
1.5	Auswirkungen der Mitte-Rand-Verdunklung auf eine Transitlichtkurve	7
1.6	Schematische Darstellung der durch einen inneren Begleiter induzierten TTV-Signale	10
1.7	Vergleich der resultierenden TTV-Signale verschiedener Systemkonfigurationen	11
1.8	Auswirkungen verschiedener Mondpositionen während eines Transiterignisses auf die Lichtkurve.	13
2.1	Grafische Übersicht der am Projekt beteiligten Observatorien	24
3.1	Illustration zur Messung der Helligkeit mittels Aperturphotometrie	31
3.2	Die integrierte Modellierungsumgebung von TAP	34
4.1	Die präziseste Lichtkurve, die innerhalb des Projektes aufgenommen wurde	45
4.2	Lichtkurven von HAT-P-32b	45
4.3	Das ursprüngliche O–C Diagramm für HAT-P-32b	51
4.4	Das O–C Diagramm mit neu bestimmter Periode	53
4.5	Das Resultat der Mehrkörpersimulation	55
4.6	Anzahl der close encounter	55
4.7	Phasenabdeckung der Beobachtungen	58
4.8	Resultierende Einschränkungen für zusätzliche Begleiter	60
4.9	Ergebnis der Stabilitätsanalyse für HAT-P-32b	61
4.10	Das Ergebnis der Neubestimmung der Systemparameter	62

4.11	Das Ergebnis für das Radienverhältnis für HAT-P-32b	64
5.1	Lichtkurven für HAT-P-18b und HAT-P-19b	68
5.2	Lichtkurven für HAT-P-27b/WASP-40b und WASP-21b	69
5.3	Ergebnisse der O–C Diagramme	73
5.4	Ergebnisse für das Verhältnis a/R_S	73
5.5	Ergebnisse für das Radienverhältnis	74
5.6	Ergebnisse für die Inklination	74

Tabellenverzeichnis

2.1	Tabellarische Übersicht der am Projekt beteiligten Observatorien	23
2.2	Liste aller Transitbeobachtungen innerhalb des Projektes für HAT-P-18b, HAT-P-19b und HAT-P-27b/WASP-40b	25
2.3	Liste aller Transitbeobachtungen für HAT-P-32b und WASP-21b	26
4.1	Systemparameter von HAT-P-32 und HAT-P-32b	42
4.2	Liste der partiellen Transits von HAT-P-32b	44
4.3	Die Ergebnisse der simultanen Beobachtungen für HAT-P-32b	48
4.4	Einzelergebnisse der Modellierung aller Transitbeobachtungen	50
4.5	Vergleich der Resultate mit Literaturwerten	52
4.6	Vergleich aller bisherigen Ergebnisse der Systemparameter	63
5.1	Vergleich dieser Studie mit Literaturwerten	71
5.2	Vergleich der simultanen Transitbeobachtung vom 3.6.2013	72

Quellcodeverzeichnis

B.1	Hilfsprogramm zur Definition von Klassen, Importanweisungen und Funktionen. Die Berechnungen beruhen zum Teil auf Gleichungen von Montenbruck (1984)	97
B.2	Programm zur Berechnung von Transitzeitpunkten	103
B.3	Hilfsprogramm zur Definition von Klassen und Funktionen	114
B.4	Programm zur Berechnung der theoretischen TTV Signale	118
B.5	Programm zur Extraktion der Positionsdaten aus den MERCURY6-Daten . .	128
B.6	Hilfsprogramm zur Sortierung der Daten	129
B.7	Programm zur Darstellung der Mindestmasse in Abhängigkeit des Periodenverhältnisses	130
B.8	Programm zur Darstellung der Anzahl von close encounters	132
B.9	Hilfsprogramm zur Definition von Klassen und Funktionen	133
B.10	Programm zur Berechnung der Stabilität	136
B.11	Programm zur Sortierung der Stabilitätsergebnisse	144

Verwendete Größen und Einheiten

Definition der in dieser Arbeit verwendeten Größen und Nicht-SI Einheiten

Formelzeichen / Nicht-SI-Einheit	Bedeutung	Wert in SI-Einheiten
c	Lichtgeschwindigkeit	$2,998 \cdot 10^8 \text{ m s}^{-1}$
$1 R_{\oplus}$	Erdradius	$6,38 \cdot 10^3 \text{ m}$
$1 R_{\text{J}}$	Jupiterradius	$7,15 \cdot 10^6 \text{ m}$
$1 R_{\odot}$	Sonnenradius	$9,72 \cdot R_{\text{J}}$ $6,96 \cdot 10^8 \text{ m}$
1 AE	Astronomische Einheit	$1,50 \cdot 10^{11} \text{ m}$
$1 M_{\oplus}$	Erdmasse	$5,97 \cdot 10^{27} \text{ g}$
$1 M_{\text{J}}$	Jupitermasse	$1,90 \cdot 10^{30} \text{ g}$
$1 M_{\odot}$	Sonnenmasse	$1,99 \cdot 10^{33} \text{ g}$
JD	Julianisches Datum	Anzahl der Tage, die seit dem 1. Januar 4713 v. Chr. 12:00 Uhr vergangen sind.

1 Einleitung

1.1 Die Entdeckung extrasolarer Planeten

Seit jeher schauen Menschen in den Himmel und fragen sich, ob sie allein im Universum sind. Während die Überlieferungen früher Hochkulturen, wie zum Beispiel der sumerische Schöpfungsmythos Enūma eliš (Vgl. Sitarz, 1983), die Götter in den Raum zwischen den Sternen erhoben, prägte das 20. Jahrhundert der Gedanke, dass hoch entwickelte außerirdische Wesen fremde erdähnliche Welten bewohnen.

Neben der Religion und der Mythologie stellte sich im letzten Jahrhundert mit stetig wachsendem Interesse auch die Wissenschaft die Frage, ob Planeten auch um andere Sterne existieren. Bereits 1944 berechnete K. Strand aus astrometrischen Beobachtungen von 61 Cyg, dass in diesem Doppelsternsystem ein bis dato unbekanntes Objekt mit einer Masse von $m_c \sim 0,016 M_\odot$ existieren könnte⁽¹⁾. Auch Struve (1952) vertritt den Standpunkt, dass “viele Objekte mit planeten-artigem Charakter in der Galaxie” existieren könnten, allerdings die damaligen technischen Möglichkeiten weit davon entfernt seien, um Planeten mit der Größe und Masse des Jupiter zu entdecken. Struve verweist in seiner Arbeit auf zwei mögliche Nachweisverfahren: zum einen könnten mithilfe der Radialgeschwindigkeitsmessung bereits bei einer Genauigkeit von $\pm 2 \text{ km s}^{-1}$ Planeten mit Massen von $m_p \sim 10 M_\oplus$ entdeckt werden, zum anderen würde der Umlauf eines Planeten bei einer Inklination von $\sim 90^\circ$ in regelmäßigen Abständen zu einer Bedeckungen des Sterns mit Helligkeitsabfällen von bis zu 0,02 mag führen.

Der erste erfolgreiche Nachweis der Existenz zweier Objekte planetarer Masse um einen anderen Stern gelang Wolszczan & Frail (1992). Die mittels präziser Messung der Pulsperiode gefundenen Objekte kreiste jedoch um einen Pulsar, PSR1257+12. Drei Jahre später fanden Mayor & Queloz (1995) mithilfe der Messung von Radialgeschwindigkeitsschwankungen den ersten Planeten um den sonnenähnlichen Stern 51 Peg. Seitdem ist die Anzahl der bekannten Planeten und Planetenkandidaten ebenso stark angestiegen, wie die

⁽¹⁾Das von Strand (1944) postulierte Objekt kann bis heute nicht eindeutig nachgewiesen werden. Neuere Arbeiten, z.B. von Gorshanov u. a. (2006), können die Existenz eines massearmen Objektes um 61 Cyg allerdings auch nicht ausschließen.

Vielfalt der möglichen Detektionsverfahren (siehe Abbildung 1.1). Als erfolgreichste Methoden bei der Suche haben sich vor allem die zwei Techniken als besonders erfolgreich herausgestellt, die schon von Struve beschrieben wurden: Die Radialgeschwindigkeitsmethode und die Transitmethode.

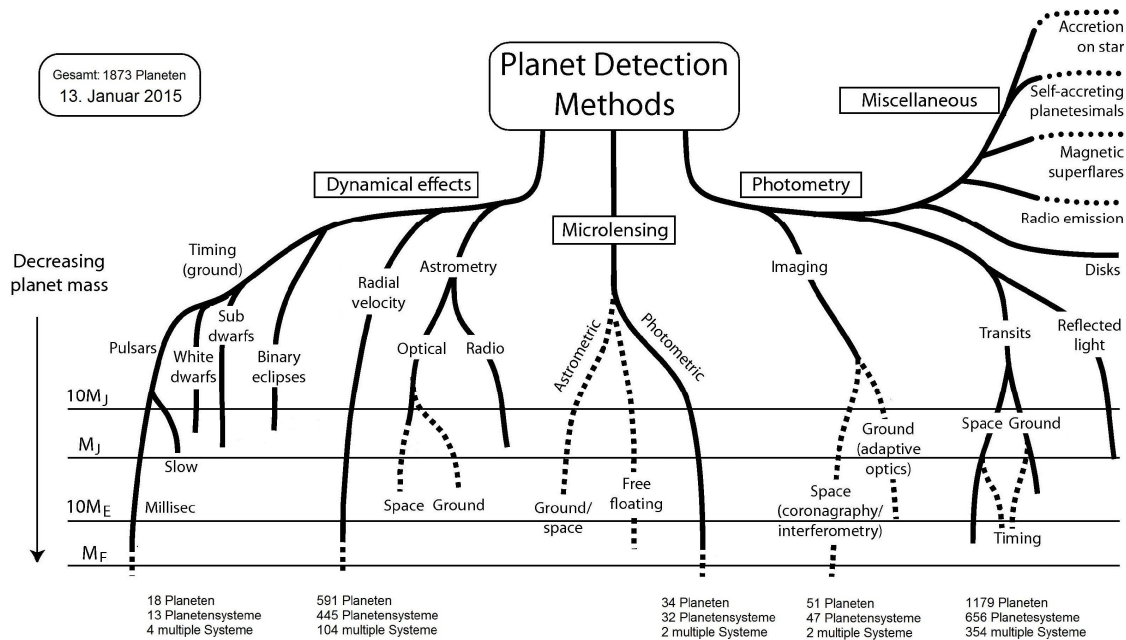


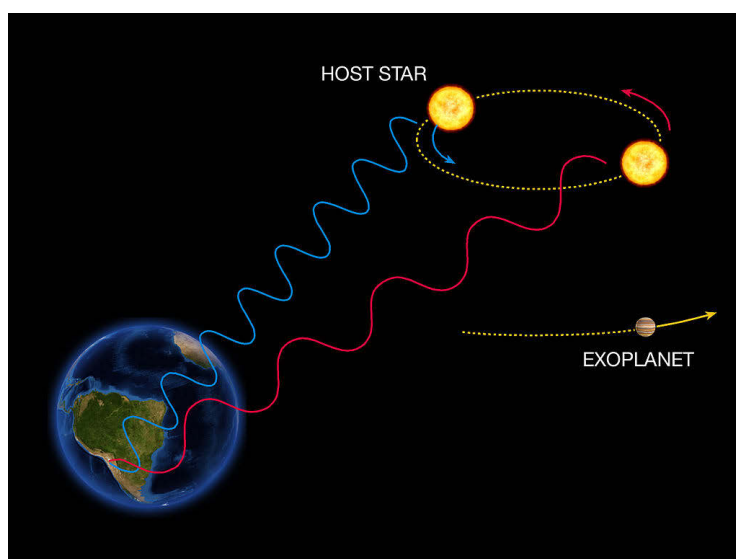
Abbildung 1.1: Eine schematische Darstellung der möglichen Methoden zur Entdeckung extrasolarer Planeten aus Perryman (2000) in der von M. A. C. Perryman aktualisierten Form vom Februar 2010 sowie aktuellen Zahlen von www.exoplanet.eu mit Stand 13. Januar 2015.

Kreisen zwei Körper umeinander, so bewegen sie sich um ihren gemeinsamen Schwerpunkt. Auch im Falle eines Sterns mit planetarem Begleiter ist dies der Fall, wenngleich mit der Besonderheit, dass der Schwerpunkt des Systems durch die Masseunterschiede innerhalb des Sterns liegen kann. Für den Fall, dass die Bahnebene nicht senkrecht zur Sichtlinie zum Beobachter steht, führt diese Bewegung um den Masseschwerpunkt einerseits zu einer Veränderung der Position des Sterns am Himmel, d.h. zu einer transversalen Bewegung. Andererseits ist eine Änderung der radialen Bewegungskomponente messbar. Die Radialgeschwindigkeitsmethode macht sich dabei den Dopplereffekt zunutze. Bewegt sich ein Körper auf einen Beobachter zu, wird die Wellenlänge der von ihm ausgehenden Strahlung verkleinert. Bewegt er sich vom Beobachter weg, so wird die Wellenlänge vergrößert (Abbildung 1.2). Dieser Effekt ist besonders gut an den Absorptionslinien des in der Sternatmosphäre enthaltenen Wasserstoffs sichtbar. Die Wellenlänge dieser Absorptionslinien verschiebt sich periodisch um eine mittlere Wellenlänge. Die Amplitude K

dieser Verschiebung skaliert dabei mit der Orientierung der Bahnebene des Systems relativ zur Sichtlinie (die sogenannte Inklination i), der Masse des Sterns m_S und des Planeten m_p , sowie der Umlaufperiode P :

$$K \propto m_p \cdot \sin i \cdot P^{-1/3} \cdot m_S^{-2/3}$$

Die Bestimmung der Radialgeschwindigkeitskurve erlaubt dabei die Messung der Amplitude K , sowie der Periode P . Bei bekannter Masse des Sterns verbleibt als Unbekannte die Mindestmasse des Begleiters $m_p \cdot \sin i$. Da ohne weitere Studien die Bestimmung der Inklination rein aus der Messung der Radialgeschwindigkeit nicht möglich ist, kann hieraus auch nicht auf die wahre Masse m_p geschlossen werden.



The Radial Velocity Method

ESO Press Photo 22e/07 (25 April 2007)

This image is copyright © ESO. It is released in connection with an ESO press release and may be used by the press on the condition that the source is clearly indicated in the caption.



Abbildung 1.2: Illustration der Radialgeschwindigkeitsmethode aus der ESO Pressemitteilung vom 25. April 2007.

Ist die Inklination des Systems nahe 90° , so wandert der Begleiter einmal pro Umlauf auf der Sichtlinie zwischen Stern und Beobachter am Stern vorbei und verdeckt damit einen Teil der Fläche des Sterns. Dies ist als Einbruch der Helligkeit messbar und wird als Transitereignis bezeichnet. Auch wenn ein Planet zu leuchtschwach ist, um ihn direkt zu sehen, kann man damit durch die Beobachtung des Helligkeitseinbruchs indirekt auf die Existenz des Begleiters schließen. Im Gegensatz zur Radialgeschwindigkeitsamplitude hängt die Größe des Helligkeitseinbruchs lediglich von der Größe der verdeckten Fläche und damit vom Radius des Begleiters ab.

1.2 Die Charakterisierung von Transitplaneten durch Lichtkurven

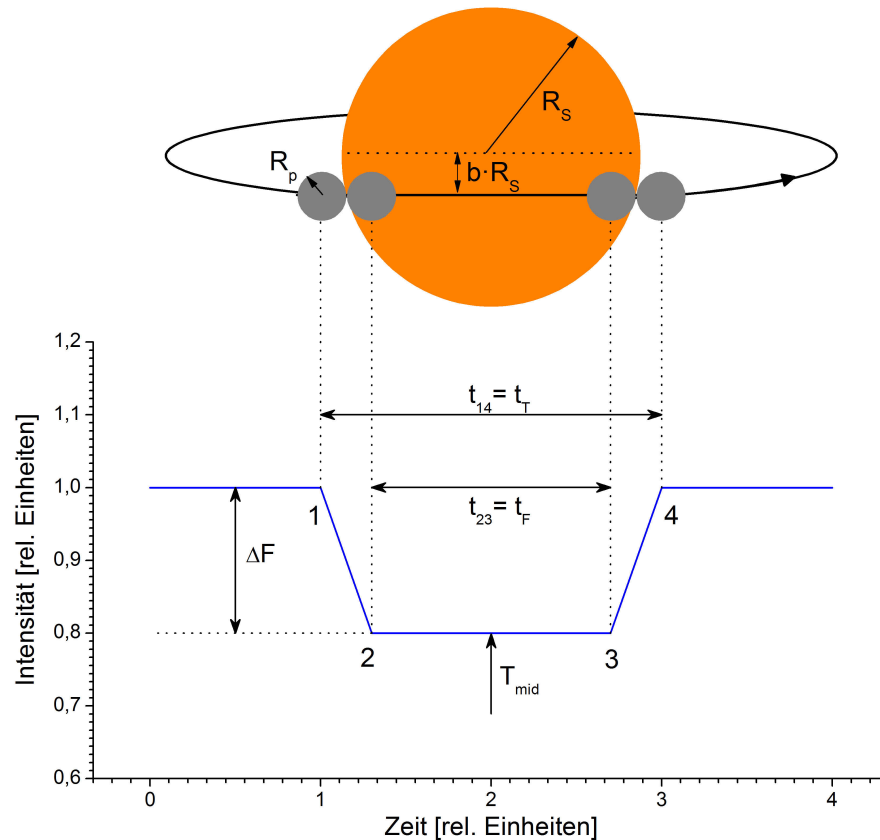


Abbildung 1.3: Eine schematische Darstellung einer Transitlichtkurve bei Vernachlässigung von Mitte-Rand-Verdunklung (frei nach Seager & Mallén-Ornelas, 2003). Die Positionen 1 und 4 markieren den Zeitpunkt des Beginns bzw. des Endes des Transiterignisses. Die Positionen 2 und 3 markieren den Beginn und das Ende der vollständigen Bedeckung.

Ist der Orbit eines Planeten um einen Stern so orientiert, dass er in regelmäßigen Abständen auf der Sichtlinie zum Beobachter am Stern vorbei zieht, spricht man von einem Transitplaneten. Abbildung 1.3 zeigt schematisch den zeitlichen Ablauf eines Transiterignisses. Am ersten (scheinbaren) Berührungspunkt des Planeten mit dem Stern ist eine Abnahme der Helligkeit des Sterns sichtbar, die ihr Minimum erreicht, sobald der Planet vollständig vor dem Stern ist. Symmetrisch zum Abfall der Helligkeit steigt diese

wieder an, wenn sich der Planet wieder aus der Sichtlinie wegbewegt. Die Tiefe des Helligkeitseinbruches ΔF ist dabei größer, je größer die Fläche des Sterns ist, die der Planet verdeckt. Im Fall eines Transits durch den projizierten Mittelpunkt des Sterns ist dabei die Länge des Transitereignisses am größten. Als Maß für den projizierten Abstand der Passage vom Sternmittelpunkt wurde dabei der Impaktparameter b eingeführt, welcher die Lage relativ zum Sternradius ausdrückt. Je größer der Impaktparameter ist, desto kürzer ist das Transitereignis an sich, während die Dauer des Helligkeitsabfalles und -anstieges ansteigt.

Erstellt man für ein Transitereignis eine Lichtkurve kann man durch eine genaue Analyse der Transitgeometrie die Größe des Helligkeitsabfalls, die Dauer des Transits, sowie den exakten Transitzeitpunkt T_{mid} (definiert im Transitmittelpunkt) bestimmen. Diese Parameter erlauben wiederum die Bestimmung wichtiger Systemparameter (eine umfangreiche Erläuterung liefert z.B. Haswell, 2010). Im Speziellen sind die Orbitperiode P , die Inklination i , das Radienverhältnis zwischen Planet und Stern R_p/R_S und das Verhältnis aus großer Halbachse und stellarem Radius a/R_S von Interesse. Zu Beachten ist dabei, dass rein aus den photometrischen Größen keine absoluten Informationen über den Radius und die große Halbachse des Planeten getroffen werden können, sondern nur bestimmte relative Größen mit dem stellaren Radius als Normierung. Geht man zunächst von einem idealen Transit⁽²⁾ aus, so lassen sich nach Seager & Mallén-Ornelas (2003) Lichtkurvengrößen mit physikalischen Parametern verknüpfen. Abbildung 1.3 zeigt eine schematische Darstellung eines Transits ohne Berücksichtigung einer Mitte-Rand-Verdunklung des Sterns.

Im Falle eines vollständigen Transits, d.h. für den Fall, dass der Begleiter komplett vor seinem Mutterstern vorbei zieht, ist der Fluss des Sterns F_S um den Fluss der vom Planeten verdeckte Fläche F_T vermindert. Diese Flussverringerung ist dabei proportional zum Quadrat ihres Radienverhältnisses.

$$\Delta F/F = \frac{F_T}{F_S} = \left(\frac{R_P}{R_S}\right)^2 \quad (1.1)$$

Der damit verbundene Helligkeitsabfall ergibt sich daher aus der Differenz der Helligkeit

⁽²⁾Bei einem idealen Transit soll davon ausgegangen werden, dass ein nicht selbstleuchtender sphärischer Planet vollständig vor einer sphärischen, gleichmäßig hellen Sternfläche vorbei zieht.

außerhalb des Transits m_{aT} und der Helligkeit während des Transits m_T mit:

$$\begin{aligned} \Delta m = m_{aT} - m_T &= 2,5 \log \left(\frac{F_S - \Delta F}{F_S} \right) \\ &= 2,5 \log \left(1 - (R_p/R_S)^2 \right) \end{aligned} \quad (1.2)$$

Die Beobachtung eines Transitereignisses stellt damit eine nützliche Ergänzung für Radialgeschwindigkeitsmessungen dar. Wurde auf deren Grundlage die Mindestmasse des Begleiters bestimmt, kann im Falle eines Transits durch die auf $i \sim 90^\circ$ festgelegte Inklination auch die wahre Masse ermittelt werden. Zusätzlich erhält man bei bekanntem stellaren Radius aus der Messung der Transittiefe den Radius des Begleiters und in Kombination beider Resultate dessen Dichte.

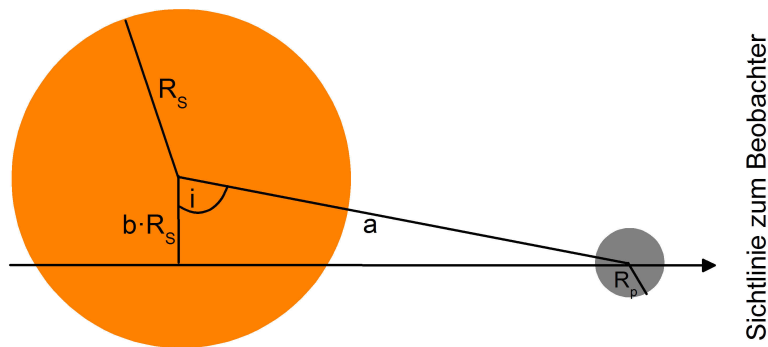


Abbildung 1.4: Ein seitlicher Blick auf ein System während eines Transits. Der Impactparameter b beschreibt den projizierten Abstand des Planeten bezüglich des Mittelpunkt des Sterns.

Der Impactparameter b beschreibt, wie stark die Bahn des Transits von einer Äquatorpassage abweicht. Aus geometrischen Überlegungen kann er durch $b \cdot R_S = a \cdot \cos i$ bestimmt werden (Abbildung 1.4). Aus physikalischer Sicht wird er maßgeblich durch das Verhältnis der totalen Transiddauer t_T (die Zeit, die der Planet benötigt, um vom ersten zum letzten Kontakt am Stern vorbei zu ziehen; Punkte 1 und 4 in Abbildung 1.3) und der Dauer der flachen Transitphase t_F (d.h. der Zeit, in der sich der Planet komplett vor dem Stern befindet; Punkte 2 und 3) bestimmt. Je zentraler der Planet am Stern vorbei zieht, desto kürzer ist die Phase des Anstiegs und Abfalls der Helligkeit, wodurch sich

der Impaktparameter mit

$$b = \frac{a}{R_S} \cdot \cos i \quad (1.3)$$

$$= \left[\frac{\left(1 - \sqrt{\Delta F/F}\right)^2 - \left[\sin^2(t_F\pi/P) / \sin^2(t_T\pi/P)\right] \left(1 + \sqrt{\Delta F/F}\right)^2}{1 - \left[\sin^2(t_F\pi/P) / \sin^2(t_T\pi/P)\right]} \right]^{1/2} \quad (1.4)$$

rein aus den Beobachtungsgrößen $\Delta F/F$, T_T , T_F und der Umlaufperiode P ergibt. Aus der totalen Transiddauer (siehe Gleichung (3) und (8) in Seager & Mallén-Ornelas, 2003) kann dann das Verhältnis

$$\frac{a}{R_S} = \left[\frac{\left(1 + \sqrt{\Delta F/F}\right)^2 - b^2 \left[1 - \sin^2(t_T\pi/P)\right]}{\sin^2(t_T\pi/P)} \right]^{1/2} \quad (1.5)$$

bestimmt werden, so dass aus der Definitionsgleichung des Impaktparameters (1.3) auch die Inklination des Systems folgt.

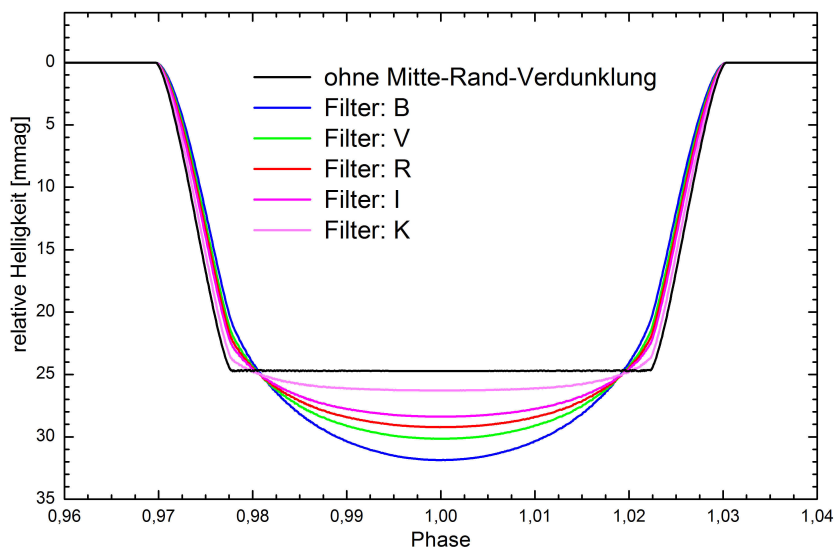


Abbildung 1.5: Die Auswirkungen der Mitte-Rand-Verdunklung am Beispiel von HAT-P-32 in Abhängigkeit verschiedener Filter. Gezeigt ist die Beschreibung des Effekts mittels eines zweiparametrischen quadratischen Gesetzes. Die Lichtkurven wurden mithilfe des Programmes `JKTEBOP` erzeugt. Die Koeffizienten c_1 und c_2 wurden aus den Tabellen Claret & Bloemen (2011) mit dem Code von Eastman u. a. (2013) für die entsprechenden Filter und die stellaren Parameter von HAT-P-32 (siehe Tabelle 4.1) entnommen.

Bei der Analyse realer Lichtkurven muss ein Effekt berücksichtigt werden, der in den genannten Gleichungen nicht berücksichtigt wird, die Mitte-Rand-Verdunklung (*engl.* Limb Darkening, LD). Sie beschreibt die das Phänomen, dass ein Stern aufgrund des Temperaturgradienten der Photosphäre im Zentrum deutlich heller erscheint als am Rand (Schwarzschild, 1906). Die Stärke der Mitte-Rand-Verdunklung hängt im Wesentlichen von Temperatur, Metallizität und Oberflächenschwerkraft des Sterns, sowie des Wellenlängenbereichs ab, in dem der Stern betrachtet wird. Abbildung 1.5 zeigt ihre Auswirkungen auf die Lichtkurve in Abhängigkeit der Beobachtungswellenlänge. Die einzelnen Modelllichtkurven wurden mit dem Programm JKTEBOP (Southworth, 2008) erstellt, wobei in allen Fällen die Systemparameter aus Tabelle 4.1 genutzt wurden. Lediglich die LD-Koeffizienten, ermittelt mithilfe des EXOFAST/LIMBDARK-Algorithmus von Eastman u. a. (2013) durch lineare Interpolation der Tabellen von Claret & Bloemen (2011), variieren je nach modelliertem Filter. Während bei Beobachtungen im Infrarotbereich die Mitte-Rand-Verdunklung weniger stark ins Gewicht fällt, so darf sie im visuellen Bereich nicht vernachlässigt werden. Vorallem hin zu kürzeren Wellenlängen wirkt sie sich stark auf die Form der Lichtkurve und die resultierende Transittiefe aus, so dass hier die einfache Betrachtung aus Gleichung (1.2) nicht angewendet werden kann. Zur mathematischen Beschreibung ihrer Auswirkungen existieren in der Literatur verschiedene Möglichkeiten. Der erste Ansatz eines linearen Gesetzes ist auf Schwarzschild (1906) zurück zu führen, er stellte sich aber im Zuge immer genauer werdender Messungen als zunehmend ungenau heraus. Daher schlug Kopal (1950) eine zweiparametrische Beschreibung in Form einer quadratischen Abhängigkeit der Intensität I an der Stelle μ relativ im Zentrum des Sterns I_0 mit

$$\frac{I_\mu}{I_0} = 1 - c_1(1 - \mu) - c_2(1 - \mu)^2 \quad (1.6)$$

vor, wobei $\mu = \cos \gamma$, mit γ dem Winkel zwischen der Sichtlinie des Beobachters und der Normalen auf der stellaren Oberfläche, gilt. Im Laufe der Jahrzehnte kamen weitere zweiparametrische Gesetze hinzu. So schlugen Klinglesmith & Sobieski (1970) ein logarithmisches Gesetz vor, Diaz-Cordoves & Gimenez (1992) ein wurzelförmiges Gesetz und Claret & Hauschildt (2003) eine exponentielle Abhängigkeit. Für genauere Beschreibungen wurden auch höherparametrische Mitte-Rand-Verdunklungen vorgeschlagen, so etwa ein dreiparametrisches (Sing u. a., 2009) und vierparametrisches Gesetz (Claret, 2000). Welches davon zur Beschreibung verwendet werden sollte, wird in der Literatur stark debattiert. Brown u. a. (2001) konnten zeigen, dass sich das lineare Gesetz nicht für die Analyse von

Transitlichtkurven eignet. Nach Analysen von Mandel & Agol (2002) und Southworth (2008) ist die Beschreibung mittels eines quadratischen Zusammenhangs für Lichtkurven bodengebundener Teleskope mit Spiegeldurchmessern $d \lesssim 2$ m ausreichend. Für genauere Daten, wie etwa Lichtkurven des *Kepler*-Satelliten (Koch u. a., 2010; Borucki u. a., 2011) ist dagegen ein höherparametrisches, nichtlineares Gesetz notwendig (Eastman u. a., 2013).

Die im Rahmen dieser Arbeit erhaltenen Lichtkurven wurden mit bodengebundenen Teleskopen aufgenommen, weshalb hier das quadratische Gesetz für die Mitte-Rand-Verdunklung verwendet wurde. Dieses wird von den zur Modellierung verwendeten Programmen TAP und JKTEBOP (siehe Kapitel 3.1.3) unterstützt.

1.3 Transitzeitvariationen

Die Bahn eines Planeten um einen Stern lässt sich mithilfe der Keplerschen Gesetze beschreiben. Dabei führt der Planet ungestörte, streng periodische Bewegungen aus. Besonders der Transitzeitpunkt sollte sich daher streng periodisch wiederholen, so dass er sich mittels der Umlaufperiode P und der Anzahl der Beobachtungsepochen E nach einer beliebig gewählten Nullepoche mit Zeitpunkt T_0 stets nach

$$T_{\text{mid}}(E) = T_0 + P \cdot E \quad (1.7)$$

berechnen lässt. Diese Größen werden im Allgemeinen als Ephemeriden bezeichnet. Für den Fall, dass sich weitere Objekte im System befinden, muss ein Korrekturterm δT in Gleichung (1.7) eingeführt werden, so dass

$$T_{\text{mid}}(E) = T_0 + P \cdot E + \delta T(E) \quad (1.8)$$

gilt. Dieser Korrekturterm ist notwendig, da zusätzliche Körper zu gravitativen Wechselwirkungen mit dem Stern und dem Planeten führen, die sich auf die Periodizität des Umlaufs auswirken und zu sogenannten Transitzeitvariationen (TTVs) führen.

1.3.1 Planetare Begleiter als Ursache für TTVs

Abbildung 1.6 zeigt diesen Sachverhalt für einen zusätzlichen inneren Planeten schematisch. Für diesen Fall überwiegt der Einfluss der Wechselwirkung zwischen dem Mutterstern und dem inneren Begleiter. Durch die Bewegung des Sterns um das gemeinsame Massezentrum kommt es dabei zu periodischen Variationen des Transitzeitpunktes. Für

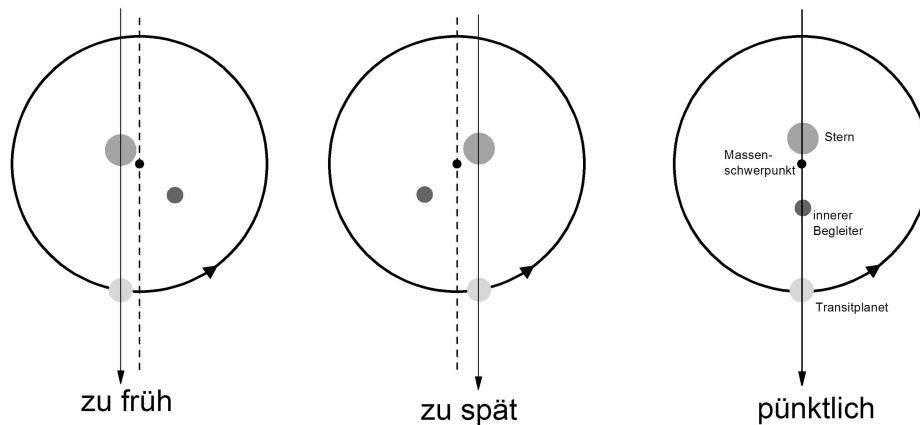


Abbildung 1.6: Eine schematische Darstellung der durch einen inneren Begleiter induzierten TTV-Signale (nach Abb. 1 aus Agol u. a., 2005). Durch die Bewegung des Sterns um den gemeinsamen Schwerpunkt mit dem inneren Begleiter erscheint der Transit im linken Fall zu früh, während er im mittleren Fall später als erwartet stattfindet. Im rechten Bild findet der Transit zum erwarteten Zeitpunkt statt.

äußere Begleiter ist der dominierende Störfaktor die Interaktion der beiden Begleiter untereinander (Agol u. a., 2005).

Zur Analyse der Transitzeitvariationen wird δt in einem sogenannten $O-C$ Diagramm (*engl.*: Observed minus Calculated) als Differenz zwischen dem mittels einer Modellanpassung bestimmten Transitmittelpunktes ($T_{\text{mid}}^{\text{beobachtet}}$) und dem mithilfe der bekannten Parameter Beobachtungsepoche, Periode und Zeitpunkt der Nullepoche berechneten Zeitpunkt ($T_{\text{mid}}^{\text{berechnet}}$) über der Beobachtungsepoche dargestellt.

$$\begin{aligned} O - C \equiv \delta t &= T_{\text{mid}}^{\text{beobachtet}} - T_{\text{mid}}^{\text{berechnet}} \\ &= T_{\text{mid}}^{\text{beobachtet}} - (T_0 + E \cdot P) \end{aligned} \quad (1.9)$$

Ein Wert von $(O - C) = 0$ bedeutet, dass keine Variationen im System vorliegen. In den meisten Fällen zeigt sich bei der Analyse des $O-C$ Diagramms ein linearer Trend, welcher in erster Linie auf eine ungenaue Bestimmung der Periode hindeutet. Bleiben nach Abzug dieses Trends Variationen übrig, so deuten diese auf das Vorhandensein von zusätzlichen Objekten im System hin.

Die Stärke der induzierten Variationen hängt dabei sowohl von der Masse des zusätzlichen Körpers ab, als auch von dessen Abstand zum bekannten Planeten. Holman & Murray (2005) konnten zeigen, dass dabei selbst die Existenz eines Körpers planetarer Masse im System ausreicht, um Abweichungen in der Periodizität zu verursachen. Wenn sich zwei Planeten in einer Bahnresonanz (*engl.* mean motion resonance) zueinander be-

finden, sind die Interaktionen besonders stark (Steffen u. a., 2007). Dabei handelt es sich um Konstellationen, in denen für das Periodenverhältnis zwischen dem bekannten Planeten und dem hypothetischen zusätzlichen Körper $P_b/P_c = m/n$ gilt, wobei m und n kleine ganze Zahlen sind. Innerhalb einer Bahnresonanz können selbst Planeten mit erdähnlicher Masse zu Variationsamplituden von mehreren Minuten führen (Agol u. a., 2005; Holman & Murray, 2005). Mit einer Genauigkeit von wenigen 10 Sekunden (Maciejewski u. a., 2011a) sind bodengebundene Transitbeobachtungen damit bereits heute in der Lage, mittels TTV-Analysen erdähnliche Planeten aufzuspüren, auch wenn eine direkte Transitbeobachtung nicht möglich ist.

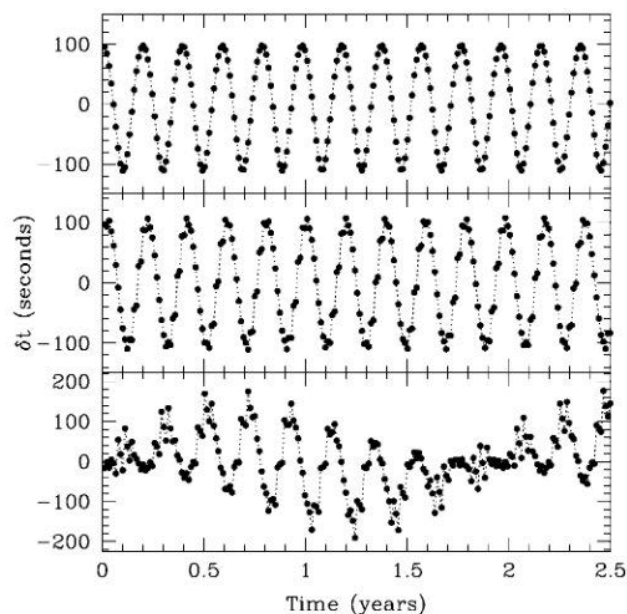


Abbildung 1.7: Ein Vergleich der resultierenden TTV-Signale verschiedener Systemkonfigurationen (Abb. 2 aus Ford & Holman, 2007). Simuliert wurde jeweils ein Planet mit $m_p = 0,5 M_{\oplus}$. Zusätzlich wurde im oberen Diagramm ein Trojaner mit $m_c = 1 M_{\oplus}$ (siehe Kapitel 1.3.2), im mittleren Diagramm ein äußerer Planet mit $M_c = 28 M_{\oplus}$ nahe der 2:1 Resonanz und im unteren Diagramm ein äußerer Planeten mit $m_c = 4,8 M_{\oplus}$ nahe der 3:2 Resonanz im System platziert. Alle drei Konfigurationen zeigen eine ähnliche Periodizität und Signalamplitude.

Nesvorný & Morbidelli (2008) zeigen allerdings, dass die Interpretation von TTV-Signalen kompliziert ist, da es keine eindeutige Zuordnung von einem vorhandenen Signal auf die Eigenschaften des Störkörpers gibt. Vielmehr ist es notwendig, mithilfe von N-Körper Integratoren verschiedene möglichen Systemkonfigurationen zu simulieren und die TTV-Signale der simulierten Systeme mit den beobachteten Signalen zu vergleichen. Abbildung 1.7 zeigt das Ergebnis einer Analyse von Ford & Holman (2007). Die Auto-

ren dieser Studie haben zusätzlich zu einem Planeten mit einer Masse von $m_p = 0,5 M_{\oplus}$ Objekte mit unterschiedlichen Massen auf verschiedene Umlaufbahnen platziert und die resultierenden TTV-Signale verglichen. Dabei stellten sie fest, dass auch komplett unterschiedliche Szenarien ähnliche TTV-Amplituden und Periodizitäten aufweisen können, welche nur durch eine ausreichend große Beobachtungsdauer aufgelöst werden können. Dies deckt sich auch mit dem Ergebnis von Nesvorný & Morbidelli (2008), welche anhand eines Testszenarios überprüften, wie viele aufeinanderfolgende Transitbeobachtungen notwendig sind, um die reale Konfiguration eindeutig zu identifizieren. Wenngleich diese Anzahl vom betrachteten System abhängt, so kann die von Nesvorný & Morbidelli (2008) gefundene Anzahl von ~ 20 benötigten beobachteten Transitereignissen als Richtwert verstanden werden.

Dass durch die Suche nach Transitzeitvariationen zuvor unbekannte Planeten gefunden werden können, zeigen zum Beispiel die Analysen von Ballard u. a. (2011) im Falle des Kepler-9-Systems. Die Analyse der Transitzeitpunkte von Kepler-9b zeigte eine TTV-Amplitude von 5 min. Aufgrund von Simulationen kann dies von einem Objekt mit einer Umlaufperiode von $P < 160$ d und einer Masse von $m < 6 M_{\oplus}$ hervorgerufen werden. Nesvorný u. a. (2012) fanden für Kepler-6b sogar eine TTV-Amplitude von 2 Stunden, welche auf die Existenz eines äußeren Planeten schließen lassen, der aufgrund seiner Inklination allerdings keine Transits verursacht. Desweiteren dient eine TTV-Analyse in Systemen, in denen die Transitereignisse mehrerer Objekte gesehen werden können, oftmals als indirekter Nachweis der planetaren Natur dieser Objekte. Dazu werden die beobachteten TTV-Signale mit theoretischen Berechnungen verglichen, um die Masse der Transitobjekte durch deren gravitativen Einfluss abzuschätzen. Auf diese Weise konnten zum Beispiel die Planeten in den Systemen Kepler-48 bis Kepler-60 mithilfe von TTV-Studien durch Steffen u. a. (2013) bestätigt werden.

1.3.2 Alternative Quellen für TTV Signale

Neben der Erzeugung von TTV-Signalen aufgrund zusätzlicher Planeten auf inneren oder äußeren Umlaufbahnen, gibt es auch andere Ursachen für Transitzeitvariationen. Holman & Murray (2005) weisen darauf hin, dass auch für den ungestörten Fall kleinere Korrekturen der Umlaufperiode notwendig sein können. So kann zum Beispiel aus Präzessionseffekten bei exzentrischen Planetenbahnen oder Gezeitenwechselwirkungen (Sasselov, 2003) eine Änderung der Umlaufperiode resultieren (Miralda-Escudé, 2002).

In unserem eigenen Sonnensystem gibt es neben Planeten (und Kleinplaneten) auch Monde und Asteroiden, welche ebenfalls gravitativ mit den Planeten und der Sonne wech-

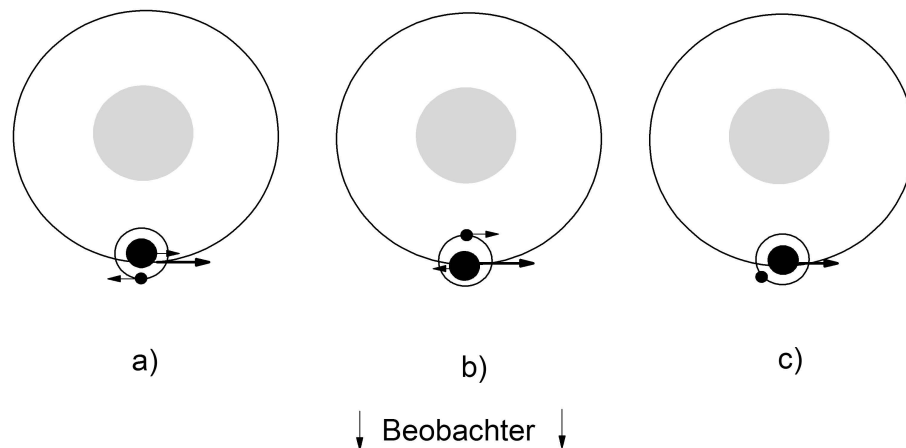


Abbildung 1.8: Auswirkungen verschiedener Mondpositionen während eines Transiterignisses auf die Lichtkurve. In Position a) ist die Transitdauer verringert, in Position b) erhöht. In Position c) führt neben dem Transit des Planeten auch der Transit des Mondes in der Lichtkurve des Sterns zu einem Helligkeitsabfall.

selwirken. Von besonderem Interesse ist die Existenz von Monden um extrasolare Planeten. Diese können, je nach Umlaufperiode, verschiedene Einflüsse auf die Lichtkurve eines Transits haben. Abbildung 1.8 zeigt exemplarisch verschiedene Positionen eines Mondes während eines Transiterignisses. Ausgehend davon, dass sich der Mond in der gleichen Ebene wie die Umlaufbahn des Planeten befindet, führt Position a) durch die Bewegung des Mondes und des Planeten umeinander zu einer geringen Erhöhung der effektiven Umlaufgeschwindigkeit des Planeten, Position b) zu einer Verzögerung beim Umlauf um den Stern. Die dadurch gemessene Transitdauer ist folglich in Position a) kürzer, in Position b) länger. Daneben kann der Mond natürlich auch selbst ein Transiterigniss hervorrufen (Position c)), welches aufgrund der Größe potentieller Monde allerdings mit bodengebundenen Teleskopen nicht signifikant messbar sein würde. Kipping (2009) hat für einige Modellsysteme die erwarteten Transitzeit- (TTV) und Transitdauervariationen (TDV) simuliert und stellen fest, dass TDV-Amplituden durchaus einige wenige Sekunden betragen und auch die Stärke der induzierten TTV-Signale übertreffen kann. Die Autoren weisen allerdings auch darauf hin, dass die Genauigkeit der Bestimmung der Transitdauer wesentlich geringer ist. In einem Projekt zur expliziten Suche nach Exomonden bei *Kepler*-Sternen (eine Projektbeschreibung liefert Kipping u. a., 2013, sowie die fortfolgenden Veröffentlichungen des HEK-Projektes) konnte aktuell⁽³⁾ noch kein Exomond gefunden werden. Der erste mögliche Kandidat um Kepler-90g wurde von Kipping

⁽³⁾Stand 23. Juni 2015

u. a. (2014) verworfen. Bodengebundene Beobachtungen sind dabei derzeit noch nicht in der Lage, hinreichend genaue Messungen zur Suche nach bzw. zum Nachweis von Exomonden zu liefern.

Auch die in Zusammenhang mit Abbildung 1.7 genannten Trojaner, also Objekte auf der gleichen Umlaufbahn wie der bekannte Transitplanet, können TTV-Signale verursachen. Die von Ford & Holman (2007) gezeigten TTV-Amplituden sind mit bis zu 100 Sekunden dabei auch mit heutigen bodengebundenen Beobachtungen messbar. Eine direkte Beobachtung des Transitereignisses eines Trojaners mit bodengebundenen Teleskopen ist aufgrund ihrer Größe (trotz möglicher Massen bis in das Erdmassenregime) oder einer relativ zum Transitplaneten inkliniert Bahn (Morbidelli u. a., 2005) nicht möglich.

1.4 Ziel der Arbeit

Zu Beginn dieses Projektes war die Frage, ob massereiche Planeten, die sich sehr nah an ihrem Zentralstern befinden, in Multiplanetensystemen vorkommen können noch offen. Zum einen war nur eine geringe Anzahl solcher Objekte bekannt, zum anderen konnten aufgrund der vorhandenen Beobachtungsmöglichkeiten nur einzelne Objekte nachbeobachtet werden. Mithilfe moderner Planetensuchprogramme wie der *Kepler*-Mission wurde es im Laufe der Zeit möglich, nach immer kleineren Planeten zu suchen und sogar Planetenkandidaten mit erdähnlicher Masse innerhalb der habitablen Zone ihres Muttersterns zu entdecken. Durch die geringe Tiefe eines solchen Transitereignisses ist dies für bodengebundene Teleskope allerdings nicht direkt möglich.

Abhilfe kann die Messung von Transitzeitvariationen bei bekannten Transitplaneten schaffen. Durch die Stärke der auch von massearmen Planeten induzierten TTV-Signale innerhalb der Resonanzen können selbst mit kleinen bodengebundenen Teleskopen erdähnliche Planeten indirekt nachgewiesen werden.

Daneben ist auch die Entdeckung von Monden oder Trojanern durch TTV-Analysen theoretisch möglich, selbst wenn ein direkter Beweis deren Existenz unterhalb der Nachweisgrenze liegt.

Im Rahmen dieser Arbeit wurden sieben bekannte Transitplaneten ausgewählt. Bei diesen sollte durch die Beobachtung von Transitereignissen und eine exakte Analyse der Transitparameter nach TTV-Signalen gesucht werden, die auf die Existenz bisher unbekannter massearmer Begleiter schließen lassen. Die Auswahlkriterien der Objekte, sowie die Beobachtungsstrategie wird in Kapitel 2 erläutert. Kapitel 3 schildert detailliert die Grundlagen der Datenauswertung von der Reduktion bis zur Erstellung einer Lichtkurve, sowie die Anpassung an eine Modelllichtkurve. Ebenfalls werden die Schritte zur theoretischen Modellierung eines hypothetischen System erklärt, bevor in den Kapiteln 4 und 5 die Ergebnisse präsentiert werden. Zum Schluss gibt Kapitel 6 eine Zusammenfassung, sowie eine Einordnung dieser Studie in die aktuelle Forschung.

2 Datenaufnahme

2.1 Auswahl der Beobachtungsobjekte

Mit der Etablierung des Projektes *TTV@YETI* innerhalb des weltweiten Teleskopnetzwerkes YETI (*engl.* Young Exoplanet Transit Initiative; Initiative zur Suche nach jungen, extrasolaren Planeten; Neuhäuser u. a., 2011) wurde in unserer Arbeitsgruppe⁽⁴⁾ begonnen, systematisch bei ausgesuchten Transitplaneten in potentiellen Mehrplanetensystemen nach TTV-Signalen zu suchen. Dabei spielen vier Auswahlkriterien eine Rolle:

- Die vorhandenen Daten in den Entdeckungsveröffentlichungen der Transitplaneten deuten auf die Existenz eines zusätzlichen Begleiters im System hin.

Ein Hinweis auf die Existenz eines zusätzlichen Planeten kann zum Beispiel eine vorhandene Exzentrizität sein. Diese wird im Zuge der Nachbeobachtung von Transitplanetensystemen durch die Aufnahme von Radialgeschwindigkeitsmessungen bestimmt. Die Zirkularisationszeitskala gibt die Zeit an, in der u.a. aufgrund von Gezeitenkräften ein ursprünglich exzentrischer Orbit eines einzelnen Planeten zirkularisiert wird (Zahn, 1977; Bodenheimer u. a., 2001). Weist der Planet einen exzentrischen Orbit als mögliche Orbitlösung auf, obwohl das Alter des Systems größer als die Zirkularisationszeitskala ist, kann dies auf eine verzögerte Zirkularisation aufgrund eines weiteren Planeten im System hindeuten (Christian u. a., 2009).

Die Aufnahme der Radialgeschwindigkeit von Planetenmuttersternen zeigt in einigen Fällen auch auf, dass neben der periodischen Störung durch den vorhandenen Planeten auch teils große Fluktuationen existieren. Nach Hartman u. a. (2011b) können auch diese Fluktuationen Anzeichen für zusätzliche Begleiter im System sein.

Zuletzt kann die Summe der Einflüsse mehrerer Planeten auf die Radialgeschwindigkeit als exzentrischer Orbit eines Planeten fehlinterpretiert werden. So wurde die Exzentrizität von 55 Cnc b ursprünglich zu $e = (0,051 \pm 0,013)$ bestimmt (Butler

⁽⁴⁾Die Projekthomepage ist unter <http://www.home.umk.pl/~gmac/TTV/doku.php> verfügbar.

u. a., 1997). Eine Neubestimmung nach der Entdeckung und Modellierung weiterer Planeten im System reduzierte diesen Wert schließlich auf $e = (0,014 \pm 0,008)$ (Fischer u. a., 2008)

- Das Objekt ist auf der Nordhalbkugel beobachtbar.

Die durchgeführten Beobachtungen von Transitereignissen wurden zum einen an der Universitäts-Sternwarte der Friedrich-Schiller-Universität Jena in Großschwabhausen durchgeführt, zum anderen konnten auch nationale und internationale Partnersternwarten, zum Großteil im Rahmen des YETI-Netzwerkes, zur Datenaufnahme gewonnen werden. Um die Beobachtung an der Universitäts-Sternwarte Jena, sowie den YETI Mitgliedsteleskopen zu ermöglichen, ist die Auswahl der möglichen Objekte auf bekannte Transitplaneten auf der Nordhalbkugel beschränkt.

- Die Helligkeit des Planetenmuttersterns und die Tiefe des Transitereignisses ist ausreichend hoch.

Erste Messungen der photometrischen Präzision in Abhängigkeit der Helligkeit eines Sterns ergaben, dass mit dem Schmidt-Teleskop der Universitäts-Sternwarte Jena eine Präzision von 5 mmag bei einer Helligkeit im R-Band von bis zu $R \sim 13$ mag erreichbar ist (Neuhäuser u. a., 2011). Diese Präzision ist notwendig, um eine gute Modellierung der Daten zu gewährleisten. Deshalb wurden für das TTV-Projekt nur diejenigen Systeme berücksichtigt, welche bekannte Transitplaneten mit Transittiefen von $\Delta m > 10$ mmag besitzen und deren Planetenmuttersterne Helligkeiten von $R < 13$ mag besitzen. Dies bedeutet, dass maßgeblich Heiße Jupiter und Heiße Saturne, also Planeten mit Massen in der Region von Jupiter und Saturn bei gleichzeitig sehr geringen Umlaufperioden als Zielobjekte in Frage kommen. Letzteres ergibt sich, da eine kleine Umlaufperiode garantiert, dass eine hohe Anzahl von Transitereignissen in kurzer Zeit beobachtet werden kann. Gleichzeitig weisen diese Objekte aufgrund ihrer geringen Distanz zum Stern meist einen erhöhten Radius und damit eine größere Transittiefe auf.

- In der Literatur sind bisher keine TTV-Studien zu dem Objekt vorhanden.

Anhand dieser Kriterien wurden von unserer Gruppe bereits die Objekte WASP-12b (Maciejewski u. a., 2011a, 2013a), WASP-10b (Maciejewski u. a., 2011b), WASP-3b (Maciejewski u. a., 2013b), sowie TrES-2 und WASP-14b (Raetz u. a., 2014) im Detail analysiert. Mit Ausnahme von WASP-12b, welcher eine schwache sinusförmige Modulation

des Transitzeitpunktes zeigt, konnten dabei bei keinem Objekt TTV-Signale gefunden werden.

Im Zuge dieser Doktorarbeit wurden sieben weitere Objekte für die Suche nach TTV-Signalen ausgewählt, welche alle oben genannten Voraussetzungen erfüllen:

- HAT-P-15b (Kovács u. a., 2010)
- HAT-P-18b und HAT-P-19b (Hartman u. a., 2011a)
- HAT-P-27b/WASP-40b (unabhängig von einander durch Béky u. a. 2011, sowie Anderson u. a. 2011 entdeckt)
- HAT-P-32b (Hartman u. a., 2011b)
- WASP-21b (Bouchy u. a., 2010) und
- WASP-38b (Barros u. a., 2011a)

Die Analyse von HAT-P-32b wurde dabei bereits in Seeliger u. a. (2014) veröffentlicht. Die Ergebnisse für HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b, sowie WASP-21b sind Bestandteil von Seeliger u. a. (2015, eingereicht).

2.2 Beobachtungsstrategie

Bodengebundene Teleskope haben zwei gravierende Nachteile gegenüber Weltraumteleskopen: Aufgrund des Tag-Nacht-Rhythmus ist die Beobachtbarkeit eingeschränkt und aufgrund des Wetters ist eine Beobachtung nicht in jeder Nacht möglich. Mithilfe des YETI-Netzwerkes als Zusammenarbeit von Teleskopen auf verschiedenen Längengraden vorwiegend auf der Nordhalbkugel können diese teilweise umgangen werden. Nimmt man als Beispiel HAT-P-32b mit einer Umlaufperiode von $P \sim 2,150008$ d (Hartman u. a., 2011b), so können maximal zwei aufeinander folgende Transitereignisse von HAT-P-32b von der Universitäts-Sternwarte Jena aufgenommen werden. Ein darauf folgender Transit würde bereits bei Tageslicht oder nach dem Untergang des Sterns am Himmel stattfinden. Durch den möglichen Zugang zu Teleskopen westlich von Jena (wie zum Beispiel dem Teleskop des Observatorio Sierra Nevada in Spanien) ist es allerdings möglich, auch diesen Transit zu beobachten.

Da innerhalb des YETI-Netzwerkes eine Überschneidung der Beobachtungskorridore existiert, ist es zum einen möglich, den gleichen Transit von mehreren Teleskopstandorten zu beobachten. Zum anderen kann ein Transit, welcher an einem Teleskop aufgrund

schlechten Wetters nicht beobachtet wird, von einem anderen Teleskop beobachtet werden.

Eine komplette Transitbeobachtung umfasst das Transitereignis an sich, sowie mindestens eine Stunde Normallicht vor und nach dem Transit. Dies ist notwendig, um Deformationen der Lichtkurve aufgrund wechselnder Luftmasse korrigieren zu können. Bei einer Transitdauer von 3,1 Stunden im Fall von HAT-P-32b (Hartman u. a., 2011b) bedeutet dies zum Beispiel eine Beobachtungsdauer von jeweils mindestens 5,1 Stunden. Mithilfe eines in PYTHON selbst geschriebenen Programms⁽⁵⁾ wurde anhand der Position des Planetenmuttersterns und der aus den Entdeckungsveröffentlichungen der Planeten bekannten Ephemeriden vorausberechnet, wann ein Transit an einer der beteiligten Sternwarten stattfindet und diese Beobachtung beantragt. Dadurch konnten im Rahmen dieser Arbeit zwischen Oktober 2011 und Dezember 2014 insgesamt 93 Transitbeobachtungen durchgeführt werden.

Alle betrachteten Objekte sind mit Helligkeiten von $V < 12$ mag so hell, dass im fokussierten Zustand eine Belichtungszeit von wenigen Sekunden dazu führt, dass das Objekt auf dem CCD-Chip saturiert. Daher wird für die Transitbeobachtungen das Teleskop so weit defokussiert, dass eine Belichtungszeit von 30 bis 50 Sekunden erreicht wird, ohne zu einer Saturation des Sterns zu führen. Inklusiv der Auslesezeit der Kameraelektronik kann somit mindestens ein Datenpunkt alle 90 Sekunden gewährleistet werden. Diese in der Literatur häufig verwendete Beobachtungstechnik (siehe zum Beispiel Southworth u. a., 2009; Maciejewski u. a., 2011b; Anderson u. a., 2008) ermöglicht es, verschiedene Störeinflüsse zu reduzieren. Darunter zählen einerseits atmosphärische Effekte, die bei einer längeren Belichtungszeit herausgemittelt werden (Southworth u. a., 2009). Andererseits wird das Signal-Rausch-Verhältnis maßgeblich vom Fluss dominiert. Ein erhöhter Fluss durch längere Belichtungszeiten führt in Folge dessen zu einer Verbesserung der Photometrie.

Um dennoch eine genaue Bestimmung des Transitmittelpunktes zu gewährleisten, ist es notwendig, dass vor allem die Flanken des Transitereignissen, also der Abfall und Anstieg der Helligkeit, gut bestimmt werden können, wozu eine genügend hohe Anzahl an Datenpunkten während dieser Phasen benötigt wird. Eine Abschätzung dieser Zeit liefern die Gleichungen (18) und (19) in Winn (2010) und beträgt für die sieben betrachteten Objekte zwischen 20 min und 30 min. Mit unserer Beobachtungsstrategie ist es somit möglich, mehr als 10 Datenpunkte während des Abfalls und Anstieges der Helligkeit zu erhalten und eine gute Anpassung der Daten an ein Modell zu gewährleisten.

⁽⁵⁾Die Quellcodes aller benötigten Programme sind im Anhang aufgelistet.

Alle beteiligten Teleskope können photometrische Aufnahmen im optischen Wellenlängenbereich durchführen und bieten typischerweise die Filter B, V, R und I, sowie in einigen Fällen auch die Möglichkeit ohne Filter zu beobachten an. Da moderne CCD-Detektoren die höchste Quanteneffizienz im R-Band aufweisen, wurde dieser Filter für die meisten Beobachtungen genutzt. Wie bereits in Kapitel 1.2 beschrieben wurde, ist im R-Band auch der Einfluss der Mitte-Rand-Verdunklung geringer als im blauen Bereich des Spektrums (siehe Abbildung 1.5). Zwar kann durch die verwendeten Programme zur Modellanpassung der Daten dieser Effekt korrigiert werden, dennoch führt ihr stärkerer Einfluss auf die Modellierung zu größeren Unsicherheiten, vor allem bei den Parametern $k = R_p/R_S$ und a/R_S . Bei wenigen Beobachtungen wurde aufgrund technischer Limitationen der V-Band Filter genutzt. Auch die Beobachtung ohne Filter wurde in einem Fall getestet. Speziell für kleinere Teleskope bietet sich dadurch die Möglichkeit mehr Licht zu sammeln. Allerdings ergibt sich durch den breiten Spektralbereich ein größerer Einfluss einer sich stark ändernden Luftmasse während der Beobachtung aufgrund von Refraktion, so dass sich keine wesentliche Verbesserung durch den höheren Fluss ergab. In zwei Fällen wurde versucht, ein Transitereignis mit einem schmalbandigem H_α -Filter aufzunehmen, um eine mögliche Farbabhängigkeit des Transitereignisses zu testen, was allerdings ebenfalls einem zu geringen Fluss des Zielsterns zur Folge hatte.

2.2.1 Datenaufnahme an der Universitäts-Sternwarte Jena

Die Beobachtungsstation Großschwabhausen des Astrophysikalischen Instituts und Universitäts-Sternwarte Jena verfügt unter anderem über ein Schmidt-Teleskop mit einem Durchmesser von 90 cm. Im Schmidt-Modus, welcher für photometrische Aufnahmen mit einer CCD-Kamera (STK; Schmidt-Teleskop-Kamera; Mugrauer & Berthold, 2010) ausgestattet ist, ist aufgrund der montierten Schmidt-Platte nur ein effektiver Durchmesser von 60 cm nutzbar. Im Rahmen dieser Arbeit war es dabei möglich, einen Großteil der Beobachtungen selbst durchzuführen. Um eine korrekte Bestimmung des Transitzeitpunktes zu ermöglichen, wurde vor jeder Beobachtung überprüft, ob die Systemzeit des Teleskopcomputers korrekt ist. Dazu wurde die Zeit mit dem Zeitserver der Physikalisch-Technischen Bundesanstalt synchronisiert. Für die zur Reduktion der Daten notwendigen Kalibrationsaufnahmen wurden in jeder Beobachtungsnacht aufgenommen. Da der Detektor mit einer Peltierkühlung ausgestattet ist, ist die Verwendung von Bias-Aufnahmen nicht ausreichend, weswegen Dunkelstrombilder mit der jeweils passenden Belichtungszeit aufgenommen wurden. Die Weißlichtkorrektur erfolgte durch die Aufnahme entsprechender Bilder stets nach einem Transit mit dem gleichen Fokuspunkt mithilfe des in

der Kuppel angebrachten Weißlichtschirmes. Eine ausführliche Beschreibung der Kalibrierung und der dafür benötigten Daten erfolgt in Kapitel 3.1.1. Daneben wurde auch das Cassegrain-Teleskop der Beobachtungsstation Großschwabhausen mit einem Durchmesser von 25 cm für einige Aufnahmen verwendet. Aufgrund der geringen Teleskopgröße wiesen die erhaltenen Daten allerdings eine zu schlechte Qualität auf und konnten so nicht zur weiteren Analyse herangezogen werden.

2.2.2 Datenaufnahme an Partnersternwarten

Neben der Sternwarte in Großschwabhausen waren an der Datenaufnahme auch Teleskope im Rahmen des YETI-Netzwerkes beteiligt, darunter Observatorien aus Deutschland, Bulgarien, Taiwan, Polen, den USA, China, der Slowakei und Spanien. Das YETI-Netzwerk (Neuhäuser u. a., 2011) ist ein Zusammenschluss von vorwiegend auf der Nordhalbkugel befindlichen Teleskopen mit Spiegeldurchmessern von (0,2 . . . 2) m und wurde ursprünglich zur Suche nach jungen extrasolaren Transitplaneten in offenen Sternhaufen gegründet. Um eine möglichst kontinuierliche Beobachtung zu ermöglichen, verfügt das Netzwerk über Teleskope auf verschiedenen geografischen Längengraden. Diese Ausgangslage ist, wie bereits beschrieben, auch für die Beobachtung von Transitereignissen ideal, so dass das Netzwerk auch für die Suche nach Transitzeitvariationen genutzt wird. Erste Ergebnisse dieser Zusammenarbeit innerhalb des Teilprojektes *TTV@YETI* wurden bereits in Maciejewski u. a. (2011b, 2013a) und Raetz u. a. (2014) publiziert.

Zusätzlich konnten auch außerhalb des YETI-Netzwerkes Teleskope aus Deutschland, Spanien und der Türkei für Transitbeobachtungen zur Suche nach TTV-Signalen gewonnen werden. Neben professionellen Sternwarten ist dabei auch die private Sternwarte des Michael Adrian Observatoriums in Trebur beteiligt. Eine Übersicht aller beteiligten Institutionen gibt Tabelle 2.1, eine grafische Übersicht ist in Abbildung 2.1 gezeigt. Die Beobachtungen wurden dabei jeweils von Kollegen vor Ort durchgeführt und die Rohdaten nach der Beobachtung zur Verfügung gestellt. Die einheitliche Reduktion und Auswertung (siehe Kapitel 3) erfolgte dann in Jena. Lediglich bei Transit #41, aufgenommen am 4. Oktober 2011 an der Sternwarte der Universität Ankara wurde die Datenreduktion und Erstellung der Lichtkurve bereits am Teleskop durch Gozde Saral vorgenommen und nur die resultierende Lichtkurve übermittelt.

Tabelle 2.1: Die Auflistung aller Teleskope, die am TTV-Projekt teilgenommen haben. Aufgelistet sind neben den Observatorien die Teleskope, die für die Transitbeobachtung genutzt wurden, sowie die Abkürzung, mit der sie im Verlauf der Arbeit bezeichnet werden, der Teleskopdurchmesser und die Gesamtzahl durchgeführter Beobachtungen (inkl. später nicht weiter verwendbarer Daten). Im oberen Teil der Tabelle befinden sich alle Teleskope, die innerhalb des YETI-Projektes (Neuhäuser u. a., 2011) teilgenommen haben. Zusätzlich teilnehmende Observatorien sind im unteren Teil der Tabelle aufgelistet.

#	Observatorium	Teleskop (Abkürzung)	\varnothing [m]	N_{tr}
1	Universitäts-Sternwarte Jena (Deutschland)	Schmidt-Teleskop (Jena 0,6m)	0,6/0,9	12
		Cassegrain (Jena 0,25m)	0,25	5
2	National Astronomical Observatory Rozhen (Bulgarien)	Ritchey-Chrétien-Coudé (Rozhen 2,0m)	2,0	7
		Cassegrain (Rozhen 0,6m)	0,6	2
3a	Graduate Institute of Astronomy Lulin (Taiwan & USA)	Tenagra II (Tenagra 0,8m)	0,8	5
3b		RCOS16 (Lulin 0,4m)	0,4	2
4	Toruń Centre for Astronomy (Polen)	Cassegrain (Torun 0,6m)	0,6	5
5	Peter van de Kamp Observatory Swarthmore (USA)	RCOS (Swarthmore 0,6m)	0,6	2
6	Gettysburg College Observatory (USA)	Cassegrain (Gettysburg 0,4m)	0,4	2
7	University Observatory Bochum (Cerro Armazones, Chile)	VYSOS6 (Chile 0,15m)	0,15	1
8	Xinglong Observing Station (China)	90/60 Schmidt (Xinglong 0,6m)	0,6/0,9	1
9	Stará Lesná Observatory (Slowakei)	0,5m Reflektor (Stará Lesná 0,5m)	0,5	1
10	Sierra Nevada Observatory (Spanien)	Ritchey-Chrétien (OSN 1,5m)	1,5	8
11	Teide Observatory, Canarian Islands (Spanien)	STELLA-1 (Tenerife 1,2m)	1,2	15
12	Michael Adrian Observatorium Trebur (Deutschland)	Trebur 1Meter Teleskop (Trebur 1,2m)	1,2	11
13	TÜBITAK National Observatory (Türkei)	T100 (Antalya 1,0m)	1,0	8
14	Calar Alto Astronomical Observatory (Spanien)	1,23m Telescope (CA-DLR 1,2m)	1,23	4
15	Ankara University Observatory (Türkei)	Schmidt-Teleskop (Ankara 0,4m)	0,4	1
16	Istanbul University Telescope in Çanakkale (Türkei)	0,6m Teleskop (Çanakkale 0,6m)	0,6	1



Abbildung 2.1: Eine grafische Übersicht der in diesem Projekt beteiligten Observatorien. Die Nummerierung entspricht der in Tabelle 2.1.

2.2.3 Beobachtungen

Aufgrund ihrer individuellen Eigenschaften konnten für zwei Objekte, HAT-P-15b und WASP-38, keine kompletten Lichtkurven aufgenommen werden. Dies resultiert hauptsächlich aus der verhältnismäßig hohen Transitdauer verbunden mit Umlaufperioden von mehr als 6 Tagen, was eine Transitbeobachtung vom Boden schwierig macht. Selbst innerhalb des YETI-Netzwerkes war so keine Beobachtung möglich.

Für die restlichen fünf Objekte konnten insgesamt 93 Transitbeobachtungen mit einer Gesamtbelichtungszeit von 291 Stunden durchgeführt werden, welche in den Tabellen 2.2 und 2.3 aufgelistet sind. Aus verschiedenen Gründen konnten nicht alle durchgeführten Beobachtungen zur weiteren Analyse verwendet werden. Ein Hauptgrund dafür waren sich verschlechternde Wetterbedingungen während der Beobachtung, so dass diese abgebrochen werden musste bevor das Transitereignis vorbei war. Die Bestimmung des genauen Transitmittelpunktes setzt allerdings voraus, dass sowohl der Abfall, als auch der Anstieg der Helligkeit mit hoher Präzision beobachtet werden. Fehlt eine der beiden Flanken in der Lichtkurve, oder ist die Streuung der Datenpunkte aufgrund der Beobachtungsbedingungen sehr groß, so ist der ermittelbare Transitmittelpunkt für eine TTV-Analyse nicht nutzbar. Zur weiteren Auswertung verblieben daher insgesamt 42 Transitbeobachtungen.

Tabelle 2.2: Die Liste aller Transitbeobachtungen, welche im Rahmen dieses Projektes für HAT-P-18b, HAT-P-19b und HAT-P-27b/WASP-40b durchgeführt wurden. Auch Beobachtungen, welche aufgrund der Qualität der resultierenden Lichtkurve aussortiert wurden, sind in dieser Liste mit enthalten. Die Beobachtungen, welche für die weitere Analyse verwendet werden konnten, sind mit einem Sternchen (*) versehen. Die Filterindizes B, C und J bezeichnen die photometrischen Systeme von Bessel, Cousins und Johnson. Die letzte Spalte gibt die Anzahl der Aufnahmen und die jeweilige Belichtungszeit für jede Transitbeobachtung an.

#	Datum	Teleskop	Filter	Aufnahmen
<i>HAT-P-18b</i>				
1*	2011-04-21	Trebur 1,2m	R _B	189 x 90 s
2	2011-05-02	Trebur 1,2m	R _B	123 x 45 s
3*	2011-05-24	Trebur 1,2m	R _B	323 x 60 s
4	2011-06-04	Rozhen 2,0m	R _C	1000 x 10 s
5*	2012-05-05	Rozhen 0,6m	I _C	219 x 90 s
6	2012-06-07	CA DLR 1,23m	B _J	250 x 60 s
7	2013-04-28	Antalya 1,0m	R	214 x 50 s
8	2014-03-30	Torun 0,6m	clear	297 x 40 s
<i>HAT-P-19b</i>				
9*	2011-11-23	Jena 0,6m	R _B	246 x 50 s
10	2011-11-23	Jena 0,25m	R _B	320 x 50 s
11	2011-11-23	Trebur 1,2m	R _B	461 x 30 s
12	2011-12-05	Jena 0,6m	R _B	129 x 60 s
13	2011-12-05	Jena 0,25m	V _B	28 x 300 s
14*	2011-12-09	Jena 0,6m	R _B	290 x 50 s
15	2011-12-09	Jena 0,25m	V _B	118 x 150 s
16	2011-12-09	Trebur 1,2m	R _B	380 x 35 s
17*	2011-12-17	CA DLR 1,23m	R _J	273 x 60 s
18	2014-08-01	Antalya 1,0m	R	148 x 60 s
19	2014-08-05	Antalya 1,0m	R	196 x 40 s
20	2014-08-21	Jena 0,6m	R _B	152 x 50 s
21*	2014-10-04	Jena 0,6m	R _B	280 x 50 s
<i>HAT-P-27b</i>				
22*	2011-04-05	Lulin 0,4m	R _B	166 x 40 s
23*	2011-04-08	Lulin 0,4m	R _B	250 x 40 s
24	2011-05-03	Tenerife 1,2m	H α	180 x 100 s
25*	2011-05-05	Trebur 1,2m	R _B	162 x 70 s
26	2011-05-08	Tenerife 1,2m	H α	190 x 100 s
27	2011-05-21	Tenagra 0,8m	R	141 x 40 s
28	2012-03-07	StaraLesna 0,5m	R	361 x 30 s
29	2012-03-29	Tenagra 0,8m	R	240 x 30 s
30	2012-04-01	Tenagra 0,8m	R	329 x 20 s
31	2012-04-04	Tenagra 0,8m	R	333 x 20 s
32	2012-04-25	Xinglong 0,6m	R	154 x 40 s
33	2012-05-16	Trebur 1,2m	R _B	231 x 70 s
34	2012-05-25	Chile 0,15m	I _J /R _J	220 x 80 s
35	2012-06-13	Tenagra 0,8m	R	223 x 15 s
36*	2013-06-03	Antalya 1,0m	R	156 x 60 s
37*	2013-06-03	OSN 1,5m	R	435 x 30 s
38	2013-06-06	CA DLR 1,23m	R _J	172 x 60 s
39*	2014-06-18	Antalya 1,0m	R	146 x 50 s

Tabelle 2.3: Die Liste aller Transitbeobachtungen, welche im Rahmen dieses Projektes für HAT-P-32b und WASP-21b durchgeführt wurden. Alle Erklärungen sind analog zu Tabelle 2.2.

#	Datum	Teleskop	Filter	Aufnahmen
<i>HAT-P-32b</i>				
40	2011-09-21	Jena 0,6m	R_B	218 x 40 s
41	2011-10-04	Ankara 0,4m	R_C	1988 x 10 s
42	2011-10-17	Torun 0,6m	R	526 x 30 s
43	2011-11-01	Jena 0,6m	R_B	315 x 50 s
44*	2011-11-01	Tenerife 1,2m	r_S	287 x 15 s
45*	2011-11-14	Jena 0,6m	R_B	303 x 40 s
46*	2011-11-29	Tenerife 1,2m	r_S	237 x 15 s
47*	2011-11-29	Rozhen 2,0m	V_C	637 x 20 s
48*	2011-12-01	Rozhen 2,0m	R_C	639 x 20 s
49*	2011-12-14	Rozhen 0,6m	R_C	257 x 60 s
50	2011-12-14	Tenerife 1,2m	r_S	69 x 15 s
51*	2011-12-27	Rozhen 2,0m	R_C	350 x 20 s
52	2011-12-29	Tenerife 1,2m	r_S	158 x 15 s
53*	2012-01-15	Swarthmore 0,6m	R_C	356 x 50 s
54	2012-01-15	Gettysburg 0,4m	R	198 x 50 s
55	2012-01-26	Jena 0,6m	R_B	236 x 50 s
56*	2012-08-15	Rozhen 2,0m	R_C	385 x 25 s
57	2012-08-15	Antalya 1,0m	R	1219 x 3 s
58*	2012-08-18	Tenerife 1,2m	r_S	311 x 15 s
59*	2012-09-12	OSN 1,5m	R_C	437 x 30 s
60*	2012-09-12	Trebur 1,2m	R_B	278 x 50 s
61*	2012-09-14	OSN 1,5m	R_C	463 x 30 s
62	2012-09-14	Tenerife 1,2m	r_S	279 x 10 s
63*	2012-10-10	Trebur 1,2m	R_B	286 x 50 s
64	2012-10-10	Jena 0,25m	R_B	180 x 100 s
65	2012-10-12	Jena 0,25m	R_B	159 x 100 s
66	2012-10-12	OSN 1,5m	R_C	221 x 20 s
67	2012-10-25	Tenerife 1,2m	r_S	329 x 10 s
68	2012-11-10	Tenerife 1,2m	r_S	369 x 15 s
69*	2012-11-22	OSN 1,5m	R_C	514 x 30 s
70	2012-11-22	Tenerife 1,2m	r_S	77 x 20 s
71	2012-11-24	Tenerife 1,2m	r_S	311 x 20 s
72	2012-12-07	Tenerife 1,2m	r_S	283 x 25 s
73	2012-12-22	Tenerife 1,2m	r_S	317 x 25 s
74*	2013-01-04	Tenerife 1,2m	r_S	317 x 25 s
75	2013-03-16	Torun 0,6m	<i>clear</i>	1432 x 10 s
76*	2013-09-06	Jena 0,6m	R_B	317 x 40 s
77*	2013-09-06	Rozhen 2,0m	R_C	390 x 30 s
78*	2013-09-06	Torun 0,6m	<i>clear</i>	1432 x 10 s
79	2013-10-02	Torun 0,6m	<i>clear</i>	1630 x 6 s
80*	2013-10-06	OSN 1,5m	R_C	585 x 30 s
81*	2013-11-01	Rozhen 2,0m	R_C	315 x 25 s
82*	2013-11-03	OSN 1,5m	R_C	529 x 30 s
83*	2013-12-01	OSN 1,5m	R_C	536 x 30 s
84*	2013-12-29	Trebur 1,2m	R_B	298 x 50 s
85*	2014-10-28	Jena 0,6m	R_B	315 x 40 s
86*	2014-12-08	Jena 0,6m	R_B	299 x 50 s
<i>WASP-21b</i>				
87*	2011-08-24	Swarthmore 0,6m	R_B	545 x 45 s
88	2011-08-24	Gettysburg 0,4m	R	230 x 60 s
89*	2012-08-16	Trebur 1,2m	R_B	365 x 40 s
90	2012-10-20	Antalya 1,0m	R	242 x 40 s
91*	2013-09-18	CA DLR 1,23m	R_J	584 x 30 s
92	2013-09-22	Antalya 1,0m	R	208 x 50 s
93	2013-09-22	Çanakkale 0,6m	R_B	163 x 110 s

3 Grundlagen der Datenauswertung und -modellierung

Das Vorgehen bei der Suche nach Transitzeitvariationen ist ein mehrstufiger Prozess. Grundlage der Analyse stellt die Beobachtung verschiedener Transits eines extrasolaren Planeten dar. Die dabei gewonnenen Daten werden reduziert und anschließend mit synthetischen Transitverläufen modelliert. Zusätzlich muss noch eine Korrektur der Zeitbasis erfolgen, die sich aus der Revolution der Erde um die Sonne und den damit verbundenen unterschiedlichen Lichtlaufzeiten ergibt. Erst nach diesem Schritt können die erhaltenen Transitzeiten auf etwaige Variationen untersucht werden.

Parallel zu dieser experimentellen Suche kann auch ein theoretischer Ansatz gewählt werden. Dabei wird die Frage beantwortet, welche Variationen in der Transitzeit bei einer gegebenen Systemkonfiguration zu erwarten sind. Neben den bekannten Eigenschaften des Zentralsterns, sowie den Parametern des im System bekannten Planeten, wird dabei ein weiterer, hypothetischer Körper im System platziert, welcher in Masse und Position variiert wird. Unter Zuhilfenahme von Viel-Körper-Simulationen wird dann die Auswirkung des zusätzlichen Körpers auf das Planetensystem untersucht.

In den folgenden Kapiteln sollen alle Vorgehensweisen detailliert beschrieben werden.

3.1 Datenauswertung

Die Auswertung der Daten kann grob in drei Teile gegliedert werden. Zunächst muss eine Reduktion der aufgenommenen Rohdaten erfolgen. Dabei werden die Bilder im wesentlichen von Einflüssen des Dunkelstroms, des Null-Levels und der unterschiedlichen Sensitivität der einzelnen CCD-Segmente befreit. Die genaue Vorgehensweise wird in Kapitel 3.1.1 erläutert. Die sich anschließende Erstellung der Lichtkurve basiert auf differentieller Aperturphotometrie und wird in Kapitel 3.1.2 geschildert. Anschließend erfolgt die Modellierung des Transits mithilfe der Programme *TAP* (*engl.* Transit Analysis Package; Gazak u. a., 2012) und *JKTEBOP* (Southworth u. a., 2004; Southworth, 2011). Deren

Methodik zur Modellierung der Transitparameter und der Ermittlung des Anpassungsfehlers wird in Kapitel 3.1.3 erläutert. Grundlage der Bestimmung des genauen Transitzeitpunktes bildet die im Dateikopf der einzelnen Aufnahmen gespeicherte Aufnahmezeit der Daten. Diese wird typischerweise in der Zeitbasis der koordinierten Weltzeit (UTC; *engl.* Universal Time, Coordinated) gemessen und anschließend in das korrespondierende Julianische Datum (JD) der Aufnahme umgerechnet. Aufgrund der unterschiedlichen Position der Erde im Sonnensystem zu verschiedenen Zeiten im Jahr und der daraus resultierenden unterschiedlichen Lichtlaufzeiten muss diese Zeit korrigiert werden. Daher ist es erforderlich, den erhaltenen Transitzeitpunkt in eine Zeitbasis zu transformieren, die diese Bewegung berücksichtigt. Kapitel 3.1.4 stellt das dafür verwendete baryzentrische Julianische Datum in der baryzentrisch dynamischen Zeit (BJD_{UTC} ; Eastman u. a., 2010) und die darin berücksichtigten Einflüsse vor.

3.1.1 Reduktion der Rohdaten

Die Vorbereitung der Datenreduktion beginnt bereits mit der Aufnahme der Daten. Neben den eigentlichen Aufnahmen während eines Transits werden in der gleichen Nacht Kalibrationsaufnahmen angefertigt. Alle modernen Kameras zur Aufnahme astronomischer Daten verwenden gekühlte CCD-Detektoren (*engl.* charge-coupled device; ladungsträgergekoppeltes Element). Grund dieser Kühlung ist der Dunkelstrom. CCD-Detektoren bestehen aus einer Vielzahl von gekoppelten Silizium-Fotodioden, in denen eintreffende Photonen ein Elektron aus dem Valenz- in das Leitungsband des Dioden-Halbleiters anheben und dadurch ein Elektron-Loch-Paar erzeugen. Das angehobene Elektron wird anschließend bis zum Auslesen des Bildes in einem Potentialtopf “gespeichert” (für eine spezifische Erklärung des Aufbaus eines CCD-Elements, der sogenannten Pixel, siehe Howell, 2006). Während des Auslesevorgangs erfolgt eine Umrechnung der registrierten Elektronen in eine Zählrate (*engl.* Counts). Der Umrechnungsfaktor, d.h. die Anzahl an Elektronen pro Count, wird durch den sogenannten Gain repräsentiert und wird im Wesentlichen durch die Kapazität der CCD-Pixel, sowie die Bittiefe der Ausleseelektronik. Neben der Möglichkeit, durch den photoelektrischen Effekt Elektron-Loch-Paare zu bilden, kann aufgrund der Boltzmann-Verteilung innerhalb des CCD-Chips auch die thermische Energie ausreichen, um ein Elektron ins Valenzband anzuheben. Da damit auch ohne eine Belichtung eine Spannungsdifferenz aufgebaut wird, spricht man vom sogenannten Dunkelstrom. Dieser Effekt ist umso stärker, je höher die Temperatur der Fotodiode ist, weshalb moderne CCD-Detektoren gekühlt werden, um den Dunkelstrom so gering wie möglich zu halten. In vielen Fällen ist es dabei ausreichend, einen CCD-Detektor auf

etwa -20°C zu kühlen, um eine deutliche Reduktion des Dunkelstroms zu erreichen (siehe zum Beispiel Mugrauer & Berthold, 2010). Diese Temperaturen lassen sich durch Peltier-Elemente relativ einfach erreichen. Für eine weitere Verbesserung müsste eine deutlich bessere Kühlung, zum Beispiel eine Stickstoffkühlung mit Kühltemperaturen von -100°C , benutzt werden. Besonders an kleineren Teleskopen scheitert deren Einsatz aber häufig an einer Kosten-Nutzen-Rechnung.

Die Größe des Dunkelstroms hängt neben der Temperatur auch von der Größe der einzelnen Fotodioden ab, welche fertigungsbedingt leicht schwanken kann, sowie von der Dauer der Belichtung. Ebenso streut sie aufgrund der statistischen Natur der spontanen Absorption der Elektronen um einen gewissen Mittelwert. Daher werden vor oder nach der Aufnahme des Transitereignisses fünf bis zehn Dunkelstromaufnahmen mit geschlossener Blende und gleicher Belichtungszeit wie die astronomischen Aufnahmen angefertigt. Diese werden anschließend gemittelt, um eine gute Abschätzung des Dunkelstroms zu erhalten und von allen wissenschaftlichen Aufnahmen subtrahiert. Bei stickstoffgekühlten CCD-Chips kann in der Regel die Aufnahme eines Dunkelstrombildes entfallen, da die thermische Anregung der Elektronen aufgrund der niedrigen Temperatur von etwa -100°C vernachlässigt werden kann. Es ist dann ausreichend, das sogenannte Biaslevel zu korrigieren.

Das Bias-Level repräsentiert das Null-Level des CCD-Chips. Dazu wird der unbelichtete CCD-Chip bei einer Belichtungszeit von null Sekunden ausgelesen und dadurch das Rauschlevel der Ausleselektronik bestimmt. Aufgrund der statistischen Natur des Rauschens würde man so theoretisch einen Wert von null Photonen zuzüglich einer gewissen Streuung erwarten. Daher wird ein positiver, konstanter Wert aufaddiert, um innerhalb der Streuung negative Werte zu vermeiden (Howell, 2006). Dies wird durch das Anlegen einer konstanten Auslesespannung erreicht, die je nach Verhalten des einzelnen CCD-Elements (Größe, Kontaktierung, Temperatur) leicht schwankt. Unter Umständen kann es vorkommen, dass das Bias-Level zum Beispiel durch einen Temperaturdrift auch während einer Beobachtungsnacht ändern kann. In diesem Fall spricht man von einem Bias-Drift. Daher enthalten astronomische Aufnahmen in der Regel, so auch Aufnahmen an der Universitätssternwarte Jena, einen sogenannten Overscan Bereich. Dies sind zusätzlich an jedem Bild befindliche Pixelzeilen, die nicht belichtet wurden, sondern lediglich als virtuelle Pixel existieren und den aktuellen Bias-Level repräsentieren. Dadurch kann der Bias-Drift während der Beobachtung korrigiert werden. Ist dieser vernachlässigbar, kann bei der Aufnahme von Dunkelstrombildern der richtigen Belichtungszeit auf die Aufnahme eines Bias verzichtet werden, da das Bias-Level auch in jeder Dunkelstromaufnahme enthal-

ten ist. Ist dagegen der Einfluss des Dunkelstroms zu vernachlässigen, muss, wie bereits erwähnt, zwingend ein Bias aufgenommen werden.

Die letzte zu korrigierende Einflussgröße stellt die unterschiedliche Sensitivität der einzelnen Pixel dar. Die fertigungsbedingt variierenden Schichtdicken führen dazu, dass die Quanteneffizienz jedes Pixels leicht unterschiedlich ist. Außerdem kommt es für Pixel im Randbereich des CCD-Chips zu einer Art Abschattungseffekt durch eine unterschiedliche Ausleuchtung (Vignettierung). Für wissenschaftliche Aufnahmen bedeutet dies, dass selbst das Bild eines mit einer gleichmäßig hellen Lichtquelle beleuchteten CCD-Detektors Schwankungen aufweist und in der Regel in der Mitte des CCD-Chips heller ist, als am Rand. Deshalb wird in jeder Beobachtungsnacht der Dämmerungshimmel oder eine gleichmäßig ausgeleuchtete Fläche innerhalb der Teleskopkuppel mit fünf bis zehn Bildern aufgenommen und die Bilder, nach Abzug der entsprechenden Dunkelstrombilder, gemittelt. Durch das so erhaltene Referenz-Weißlichtbild teilt man alle Aufnahmen einer Nacht, wodurch die Empfindlichkeit der unterschiedlichen Bildpixel normalisiert wird und es möglich ist, die Helligkeit der einzelnen Objekte im Bild zu vergleichen. Dieser Prozess der Normalisierung spielt dabei vor allem eine Rolle, wenn die Nachführung des Teleskops nicht perfekt ist. Wandern die Sterne während der Beobachtung über mehrere unterschiedlich empfindliche Pixel hinweg, so werden ohne einen Weißlichtabgleich zusätzliche Helligkeitsvariationen induziert, da sich die scheinbare relative Helligkeit der Sterne untereinander ändert. Bei einer pixelgenauen Nachführung hingegen kann auf die Weißlichtkorrektur verzichtet werden. Da jede Korrektur auch Störeinflüsse auf das Bild hat, kann so unter Umständen eine Verbesserung der resultierenden Lichtkurve erreicht werden (siehe zum Beispiel Maciejewski u. a., 2011b).

3.1.2 Erstellung der Lichtkurve mittels differentieller Photometrie

Die Erstellung der Transitlichtkurven erfolgt durch differentielle Aperturphotometrie. Das bedeutet, dass die absolute Helligkeit der einzelnen Sterne im Gesichtsfeld des Teleskops nicht von Bedeutung ist, sondern lediglich die relative Helligkeit der Sterne untereinander. Die Messungen selbst werden mit der Software *iraf* (*engl.* Image Reduction and Analysis Facility; Tody, 1993) und der darin implementierten *phot*-Routine des DAOPHOT/apphot-Paketes (Davis, 1987) durchgeführt. Die Qualität der resultierenden Lichtkurve hängt damit nicht nur von der Messgenauigkeit der Einzelhelligkeit der zu vergleichenden Sterne ab, sondern auch von deren Stabilität. Die Helligkeit des Transitsterns sollte gegen einen möglichst konstanten Vergleichssterne bestimmt werden, um den Transit als solchen möglichst ohne andere Störgrößen analysieren zu können. Daneben bedeutet

die Verwendung von Aperturphotometrie, dass die instrumentelle Helligkeit eines Sterns bestimmt wird, indem alle Counts innerhalb eines bestimmten Radius, der sogenannten Apertur, um den Stern addiert werden (Abbildung 3.1). Zum Abzug der zugrundeliegenden Himmelshelligkeit werden zusätzlich die Counts innerhalb eines Kreisrings der Fläche A_H gezählt. Der innere Radius dieses Kreisringes wird als Annulus, seine Breite als Dannulus bezeichnet. Anschließend wird die Summe der Counts auf die Aperturfläche A_{Ap} normiert und von den Counts innerhalb der Apertur abgezogen. Die instrumentelle Helligkeit m_{inst} bezüglich einer beliebig wählbaren Nullmagnitude m_0 ergibt sich dann nach Stetson (1987) zu:

$$m_{\text{inst}} = m_0 - 2,5 \cdot \log \left[\left(\Sigma_{\text{Stern}}^{\text{Counts}} - \left(\frac{A_{Ap}}{A_H} \right) \Sigma_{\text{Hintergrund}}^{\text{Counts}} \right) / (\text{Belichtungszeit}) \right] \quad (3.1)$$

Prinzipiell kann nun die Differenz der Helligkeit des Transitsterns mit einem Vergleichs- stern für alle Aufnahmen bestimmt und das Resultat über die Zeit aufgetragen werden, um die Transitlichtkurve zu erhalten.

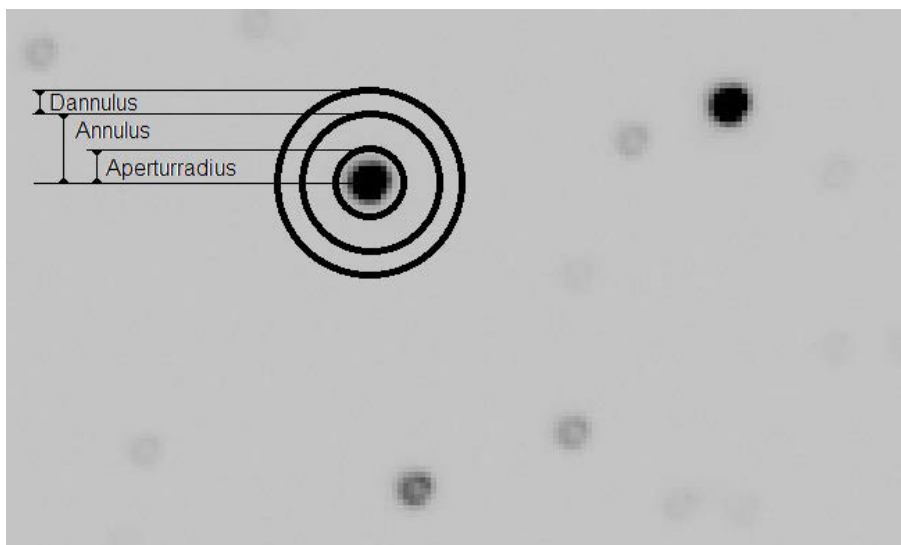


Abbildung 3.1: Illustration zur Messung der Helligkeit mittels Aperturphotometrie. Die Helligkeit des Sterns ergibt sich aus der Summe der Counts innerhalb der Apertur. Die Hintergrundmessung findet in einem Kreisring um den Stern statt. Dessen innerer Radius wird als Annulus, seine Breite als Dannulus bezeichnet. Die Aufnahme ist ein Bildausschnitt aus der Transitbeobachtung vom 08. Dezember 2014 am Jena 0,6m Teleskop. Einige Sterne zeigen die durch die Defokussierung bedingte Ringform.

Die Verwendung von nur einem Vergleichssterne birgt aber die Gefahr, dass Variabilitäten des Vergleichssterne mit Variabilitäten des Transitsterne verwechselt werden. Daher

wird die Methode von Broeg u. a. (2005) zur Erstellung eines künstlichen Vergleichssterne verwendet. Dieser Vergleichssterne ergibt sich aus allen hellen Sternen im Gesichtsfeld, die entweder heller, oder bis zu einer Magnitude dunkler sind als der Transitsterne selbst. Der Anteil eines einzelnen Sternes am künstlichen Vergleichssterne wird dabei je nach dessen Streuung und Variabilität gewichtet. Die initiale Wichtung eines Sternes ergibt sich zunächst aus dem gemessenen instrumentellen Fehler. Helle Sterne werden dadurch bevorzugt, da sie in der Regel einen kleineren Messfehler haben. Anschließend wird die differentielle Helligkeit jedes Sternes bezüglich des künstlichen Vergleichssterne bestimmt und die Standardabweichung ermittelt. Da diese ein gutes Maß für die Variabilität eines Einzelsternes darstellen, wird nun die Größe der Standardabweichung als neue Wichtung der Einzelsterne am künstlichen Vergleichssterne genutzt. Diese Prozedur wird so lange wiederholt, bis sich die neuen Wichtungen im Vergleich zu den alten Wichtungen nicht mehr signifikant ändern.

Von großer Bedeutung bei der Bestimmung der Lichtkurve des Transitereignisses ist die Bestimmung der Messfehler. Neben statistischen ermittelbaren Messfehlern, vor allem Photonenrauschen, spielen in der differentiellen Photometrie Extinktionsunterschiede zwischen den zu vergleichenden Sternen eine Rolle (siehe z.B. Young u. a., 1991). Letztere resultieren daraus, dass die verwendeten Vergleichsterne und auch der betrachtete Transitsterne unterschiedliche Spektraltypen haben und sich so bei der Beobachtung in einem bestimmten Filter unterschiedlich verhalten. Bei meist unbekanntem Spektraltyp der Sterne im Gesichtsfeld ist dieser Fehler allerdings schwierig zu behandeln. Es bietet sich daher an, durch die geeignete Auswahl der Vergleichsterne die Betrachtung des Extinktionsfehlers vernachlässigbar zu machen. Die Methode von Broeg u. a. (2005) bietet durch die eingeführte Wichtung der Sterne ein solches Auswahlverfahren. Als Grundlage für die Bestimmung der Messfehler genügt daher der instrumentelle Messfehler, welcher durch die `phot`-Routine bestimmt wird. Dieser setzt sich nach Davis (1994) im Wesentlichen aus drei Fehleranteilen additiv zusammen: dem Hintergrundrauschen innerhalb der Apertur der Fläche A_{Ap} (bestimmt als Standardabweichung σ des eigentlich als konstant annehmbaren Himmelshintergrundes), dem Photonenrauschen innerhalb der Apertur (aufgrund seiner Poisson-Natur proportional zum Fluss), sowie dem durch die Anzahl der Pixel innerhalb der Apertur (n_{sky}) normierten Ausleserauschen:

$$\Delta m = \frac{2,5}{\ln 10} \cdot \frac{\sqrt{A_{Ap} \cdot \sigma^2 + \text{Fluss}/\text{Gain} + A_{Ap}^2 \cdot \sigma^2/n_{sky}}}{\text{Fluss}} \quad (3.2)$$

Die Größe des Fehlers der Einzelmessungen wird im letzten Schritt der Methode von

Broeg u. a. (2005) skaliert. Die Skalierungsparameter, ein additiver und ein multiplikativer Term, sind dabei für alle Sterne gleich. Ziel der Skalierung ist es, den Fehler so anzupassen, dass er für die konstanten Sterne zu einem $\chi_{\text{red}}^2 \sim 1$ führt. Mit anderen Worten ist der Fehler der Datenpunkte für die konstanten Sterne nach der Reskalierung in der gleichen Größenordnung wie die Streuung der Datenpunkte. Die Berechnung der Fehler, der Wichtungen und damit der Helligkeit und Zusammensetzung des künstlichen Vergleichssterne wird in einem iterativen Prozess so lang durchgeführt, bis der mittlere Fehler für alle verwendeten Vergleichssterne deren Streuung entspricht. Die finale Lichtkurve des Transitsterne bildet anschließend der Vergleich der instrumentellen Helligkeit zur Helligkeit des künstlichen Vergleichssterne.

Eine wesentliche Verbesserung in der Erstellung der Lichtkurve wurde am Astrophysikalischen Institut durch Ronny Errmann eingeführt, der ein Steuerprogramm für den Algorithmus von Broeg u. a. (2005) schrieb. In diesem wird mittels Kreuzkorrelation eine Referenzsternliste für jedes Bild angepasst, um die durch Fehler in der Nachführung hervorgerufenen Positionsänderungen der Sterne automatisch zu berücksichtigen. Desweiteren wird automatisch der beste Aperturradius ermittelt, welcher sich meist beim etwa eineinhalbfachen der Halbwertsbreite befindet (siehe auch Errmann u. a., 2014).

3.1.3 Anpassen eines Modells an die Lichtkurve

Zur Modellierung der Daten wurde das Transit Analyse Packet TAP (*engl.* Transit Analysis Package, Version 2.104; Gazak u. a., 2012) verwendet. Das frei verfügbare Programm wurde für die Analyse von Transitlichtkurven entwickelt und ist innerhalb der IDL-Programmumgebung über eine grafische Oberfläche (Abbildung 3.2) einsetzbar. TAP nutzt dabei den EXOFAST Algorithmus von Eastman u. a. (2013), welcher wiederum das Transitmodell von Mandel & Agol (2002) mithilfe von Markov-Chain-Monte-Carlo Simulationen (MCMC) an eine oder mehrere gegebene Lichtkurven anpasst. Für die hier präsentierten Analysen wurden dabei je 10 MCMC-Ketten mit jeweils 1.000.000 Verlinkungen berechnet. Als Gesetz für die Mitte-Rand-Verdunklung nutzt TAP eine quadratische Abhängigkeit. Die Fehleranalyse von TAP basiert auf der von Carter & Winn (2009) vorgestellten Methode einer Wavelet-basierten Wahrscheinlichkeitsfunktion. Dabei werden für die Fehlerberechnung nicht nur unkorrelierte Fehler mit einbezogen, sondern es wird auch das Auftreten von korrelierten Fehlern berücksichtigt, wie sie bei der Beobachtung von Transitereignissen zum Beispiel durch die Beobachtung bei stetig sinkender oder steigender Luftmasse vorkommen. Für die Details dieser Methode wird aufgrund Ihrer Komplexität an dieser Stelle auf Carter & Winn (2009) verwiesen.

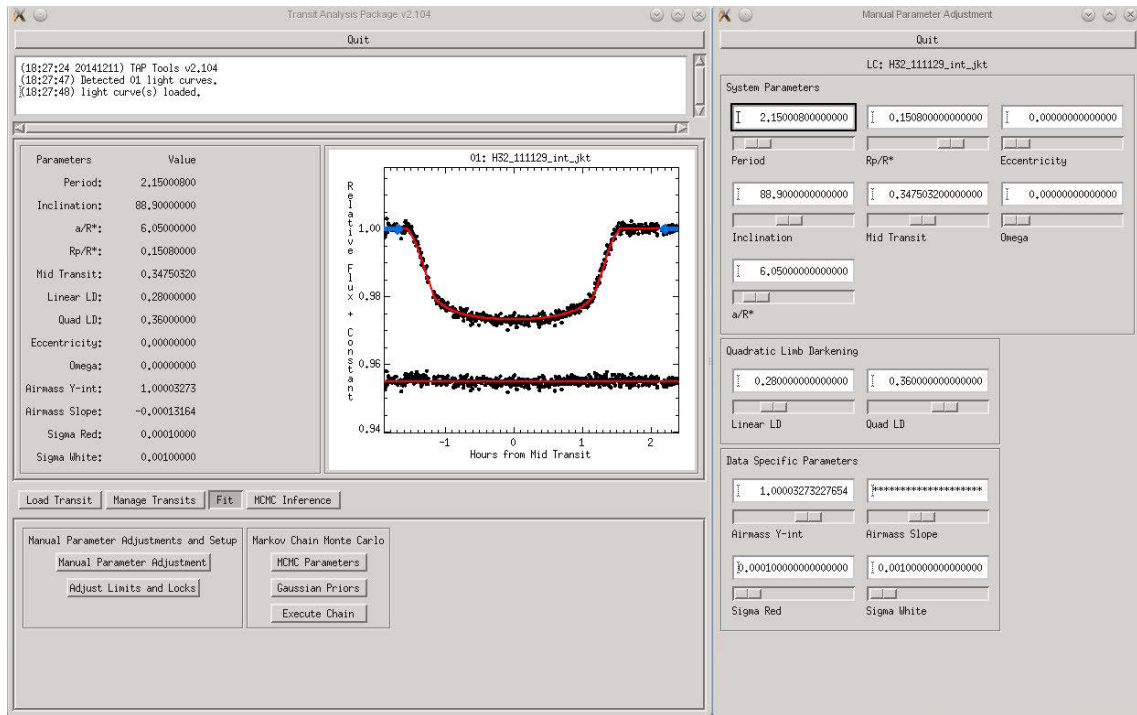


Abbildung 3.2: Die integrierte Modellierungsumgebung von TAP, Version 2.104. Die grafische Oberfläche erlaubt es, Auswirkungen von Änderungen der Systemparameter direkt in der Lichtkurve zu sehen

Neben TAP wurde auch JKTEBOP (Version 25) genutzt, um die Modellparameter für den Transit zu ermitteln. JKTEBOP (siehe zum Beispiel Southworth u. a., 2004; Southworth, 2011, sowie Referenzen darin) wurde auf Grundlage des Codes EBOP (*engl.* Eclipsing Binary Orbit Program; Nelson & Davis, 1972; Popper & Etzel, 1981; Etzel, 1981) von John Southworth weiterentwickelt, um den Spezialfall einer exoplanetaren Bedeckung zu betrachten. Das Programm ermöglicht es außerdem, verschiedene Arten der Mittele-Rand-Verdunklung zu berücksichtigen. Der Nutzer kann selbst entscheiden, ob ein lineares, quadratisches, logarithmisches oder wurzelartiges Gesetz zur Beschreibung des Effekts genutzt wird. Wie bereits erwähnt, zeigte sich in vielen Versuchen, dass ein lineares Gesetz bei der erreichbaren Präzision in der modernen Astronomie keine zufriedenstellende Beschreibung darstellt (Gazak u. a., 2012). Die Verwendung eines quadratischen Gesetzes liefert in den meisten Fällen die besten Resultate (zum Beispiel in Raetz u. a., 2014). Um Unterschiede zwischen den verschiedenen nicht-linearen Gesetzen zu sehen, wird eine Genauigkeit benötigt, die derzeit nur von Lichtkurven von Weltraumteleskopen zu erwarten ist. Zur Bestimmung der Modellfehler bietet JKTEBOP die Möglichkeit, drei verschiedene Verfahren zu verwenden: eine Monte-Carlo-Simulation, ein sog. Bootstrap-

Algorithmus, welcher die Datenpunkte der Lichtkurve innerhalb ihres Messfehlers variiert, sowie eine Residuen-Verschiebung, welche die Behandlung von korreliertem Rauschen berücksichtigen kann.

An jede erhaltene Lichtkurve wurden sowohl mit TAP, als auch mit JKTEBOP Modelle angepasst, wobei als freie Parameter die Inklination i , das Radienverhältnis zwischen dem stellaren Radius R_S und dem planetaren Radius R_P mit $k = R_P/R_S$, der inverse, mit der großen Halbachse normierte stellare Radius a/R_S , sowie der Transitzeitpunkt gewählt wurden. Die Periode wurde auf die jeweils bekannte Umlaufperiode fixiert. Ebenso wurde die Exzentrizität aufgrund der Annahme, dass eine eventuelle exzentrische Orbitlösung lediglich auf die Existenz von zusätzlichen Körpern im System zurückzuführen ist, auf $e = 0$ fixiert. Verwendet wurde zum einen die aus der Photometrie erhaltene Lichtkurve, zum anderen wurden mittels eines gewichteten Mittelwertes jeweils drei Datenpunkte der Lichtkurve zu einem Datenpunkt zusammengefasst. Dieses als Binning bekannte Verfahren kann dabei die Streuung innerhalb der Datenpunkte reduzieren dies allerdings auf Kosten einer schlechteren Zeitauflösung. Desweiteren wurden die Koeffizienten der Mittele-Rand-Verdunklung einmal auf ihre theoretischen Werte von Claret & Bloemen (2011) fixiert. In einem zweiten Durchlauf wurde es beiden Parametern erlaubt, innerhalb eines Bereiches von $\pm 0,2$ zu variieren, um zu sehen, in wie weit eine kleine Variation in diesen Parametern die Genauigkeit der Anpassung ändert. Die Resultate werden in Kapitel 4 näher erläutert. Die Ergebnisse der Modelle wurden anschließend bezüglich der genannten freien Parameter gemittelt. Ebenso wurden die Fehler gemittelt und mit der Streuung der Parameter innerhalb der verschiedenen Modelle verglichen. Als kombinierter Fehler wurde dann der größere von beiden Werten (gemittelter Fehler, bzw. Streuung des angepassten Parameters) verwendet.

3.1.4 Zeitkorrektur

Bei der Datenaufnahme an den im Projekt beteiligten Observatorien ist es entscheidend, vor dem Start der Beobachtung die Computerzeiten zu synchronisieren. Dies geschieht typischerweise durch den Abgleich der Computerzeit mit Zeitservern, welche die aktuelle koordinierte Weltzeit (UTC) bereitstellen. Die Aufnahmezeit wird dann innerhalb der Datenreduktion in das korrespondierende Julianische Datum (JD) umgerechnet. JD_{UTC} gibt die Zeit im Bezugssystem Erde an. Es ist jedoch notwendig, die Zeit auf das Massezentrum des Sonnensystem, das Baryzentrum, zu beziehen. Zur Umrechnung der Zeit wurde dabei der Code von Eastman u. a. (2010) genutzt, welcher neben der Umrechnung des Julianischen Datums in das baryzentrische Julianische Datum auch relativisti-

sche Effekte berücksichtigt. Insgesamt werden vier verschiedene Effekte korrigiert, die nach Eastman u. a. (2010) eine Genauigkeit von $t \sim 30$ ms ermöglichen, wobei diese Genauigkeit weit über die derzeit erreichbare Genauigkeit bodengebundener Beobachtungen hinausgeht. Die einzelnen Korrekturbeiträge sind die Rømer-Verzögerung $\Delta_{R\odot}$, die Korrektur der Zeitbasis Δ_C , die Shapiro-Verzögerung $\Delta_{S\odot}$, sowie die Einstein-Verzögerung $\Delta_{E\odot}$ (Eastman u. a., 2010, Gleichung 1), welche im Folgenden kurz erläutert werden:

$$\text{BJD}_{\text{TDB}} = \text{JD}_{\text{UTC}} + \Delta_{R\odot} + \Delta_C + \Delta_{S\odot} + \Delta_{E\odot} \quad (3.3)$$

Die Revolution der Erde um die Sonne führt dazu, dass die Lichtlaufzeit von einem entfernten Stern zu unterschiedlichen Zeiten im Jahr unterschiedlich lang ist. Im Extremfall muss der unterschiedliche Lichtlaufweg vom Perihel zum Aphel, d.h. eine Distanz von etwa zwei Astronomischen Einheiten betrachtet werden, welcher mit $t \approx 2 \text{ AE}/c = 16,63$ Minuten deutlich als künstliches periodisches Signal in den Daten sichtbar wäre. In Anerkennung der Leistungen Ole Rømers zur Endlichkeit der Lichtgeschwindigkeit (Rømer, 1676) wird dieser Effekt auch als Rømer-Verzögerung (*engl.* Rømer Delay) bezeichnet. Auch wenn durch Einschränkungen in den Beobachtungen dieser Extremfall nur selten erreicht wird, kann je nach Lage des Objektes am Himmel somit eine Verschiebung um mehrere Minuten sichtbar sein (Eastman u. a., 2010), welche ein künstliches, periodisches TTV-Signal erzeugt.

Als zweite, wesentliche Konvertierung findet eine Umrechnung der Zeitbasis von UTC auf die baryzentrische dynamische Zeit TDB (*engl.* Barycentric Dynamical Time) statt. Obwohl UTC keine kontinuierliche Zeitbasis darstellt, da in unregelmäßigen Abständen Schaltsekunden als Korrektur eingefügt werden müssen, hat sie sich auch in der Astronomie als Zeitstandard durchgesetzt. Diese Diskontinuität macht sie aber für relative Zeitabstände unbrauchbar. Seit der Einführung von UTC im Jahre 1961 wurden insgesamt 34 Schaltsekunden eingefügt. Neben einem konstanten Offset von 32,184 s macht dies den größten Teil der Konvertierung aus. Δ_C berücksichtigt weiterhin die Einstein-Verzögerung durch die gravitative Rotverschiebung durch die Sonne. Mit einer Amplitude von unter 2 ms (Irwin & Fukushima, 1999) ist dieser Effekt aber bereits vernachlässigbar für die Analyse des Transitzeitpunktes bodengebundener – und derzeit auch noch weltraumgebundener – Beobachtungen.

Ebenfalls von geringerer Bedeutung sind die verbleibenden zwei relativistischen Korrekturterme $\Delta_{S\odot}$ und $\Delta_{E\odot}$. Die Shapiro-Verzögerung bezeichnet die Verzögerung, die durch die gravitative Wirkung schwerer Körper (vor allem der Sonne) auf das Licht selbst verur-

sacht wird (Shapiro, 1964). Bereits bei einem Abstand des beobachteten Objektes von 30° von der Sonne ist $\Delta_{S_\odot} \lesssim 20 \mu\text{s}$. Die Einstein-Verzögerung schließlich misst den Einfluss der Bewegung der Uhr, bzw. des Beobachters auf die gemessene Zeit und ist noch einmal deutlich geringer als Δ_{S_\odot} , weshalb beide Effekte für die hier betrachteten Vorgänge vernachlässigbar sind.

3.2 Modellierung eines hypothetischen Drei-Körper-Systems

Neben der Frage, ob Variationen des Transitzeitpunktes durch die Beobachtung gefunden werden können, kann eine Betrachtung auch aus theoretischer Sicht erfolgen. Dabei steht im Mittelpunkt die Frage, welche Systemkonfigurationen zu welchen Transitzeitvariationen führen. Ebenfalls von Bedeutung ist, welche Konfigurationen zu stabilen Systemen führen. Dazu zählen alle Konfigurationen, die über einen Zeitraum stabil sind, der dem Alter des Systems entspricht. Diese Annahme gilt natürlich nicht für junge Systeme mit einem Alter von wenigen Millionen Jahren, da hier noch Planetenentstehungsprozesse vorhanden sein können. Da die in dieser Arbeit betrachteten Systeme allerdings bereits mehrere Milliarden Jahre alt sind, ist sie berechtigt.

Bereits zu Beginn der Beobachtungen zeichnete sich ab, dass für HAT-P-32b die meisten Beobachtungsdaten aufgenommen werden können. Daher wurde die Modellierung maßgeblich für dieses System durchgeführt. In den folgenden Kapiteln werden die Programme, welche zur Berechnung der theoretischen TTV-Signale, sowie zur Bestimmung der Stabilität des System verwendet wurden, erläutert. Die Ergebnisse der Simulationen werden dann in Kapitel 4 präsentiert.

3.2.1 Mehrkörpersimulation

Zur Modellierung verschiedener hypothetischer Systeme wurde der N-Körper Integrator MERCURY6 (Chambers & Migliorini, 1997; Chambers, 1999) verwendet. Mit diesem ist es möglich, die zeitliche Entwicklung einer gegebenen Systemkonfiguration berechnen zu lassen. Im Rahmen der hier vorgestellten Analyse wurde der Zentralstern HAT-P-32, der bekannte Planet HAT-P-32b, sowie ein weiterer Körper koplanar im System platziert und die Entwicklung des Systems betrachtet. Für den zusätzlichen Planeten wurden dabei 73 verschiedene Massen zwischen einer Erdmasse und neun Jupitermassen, sowie 1745 verschiedene Entfernungen zum Zentralstern zwischen $a_{c,\min} = 3 \cdot R_{\text{Stern}} = 0,017 \text{ AE}$ und

$a_{c,\max} = 3 \cdot a_b = 0,1 \text{ AE}$ angenommen.⁽⁶⁾ Auch wenn beide Grenzen keine absoluten Limits für zusätzliche Planeten darstellen, ergeben sie sich aus den Nachweisgrenzen für TTV-Signale. Weiter innen liegende Planeten führen aufgrund einer kleinen Masse oder Inklinationen weit unter 90° zu deutlich geringen Wechselwirkungen, würden im Falle höherer Masse und damit größerem Radius bereits durch direkte Transitbeobachtung aufgefallen sein, oder gar innerhalb des Roche-Limits und damit in einer instabilen Region liegen. Gleiches gilt, mit Ausnahme des letzten Punktes, auch für Objekte außerhalb des maximal simulierten Abstandes. Insgesamt ergeben sich damit 127.385 unterschiedlichen Konfigurationen.

Zusätzlich wurde für den Planeten eine nicht-exzentrische Umlaufbahn angenommen. Dies hat vorrangig den Grund, dass die benötigte Rechenzeit reduziert wird, da eine Exzentrizität ungleich null den Periapsisabstand ω als weiteren freie Parameter bedingt. Auch um eine grundsätzliche Aussage über mögliche Begleiter zu treffen, ist eine Analyse nicht-exzentrischer Begleiter ausreichend und kann im Falle eines konkreten Signals mit kleinerem Aufwand verfeinert werden.

Zur Berechnung der Position des bekannten, sowie des hypothetischen Planeten wurde in MERCURY6 der Bulirsch-Stoer-Algorithmus (Stoer & Bulirsch, 1980) gewählt, welcher nach Chambers (1999) als sehr stabil bei N-Körper-Problemen gilt. Mit diesem wurden die Orbitalelemente beider Planeten in einem Zeitintervall von jeweils 10 Minuten für je 5.000 Umläufe des bekannten Planeten berechnet. Für HAT-P-32b mit einer Umlaufperiode von $P = 2,150008 \text{ d}$ (Hartman u. a., 2011b) bedeutet dies eine Gesamtsimulationsdauer von 30 Jahren. Obwohl dieses Vorgehen durch die sehr kleinen Zeitschritte eine hohe Rechenzeit benötigt, ist die Verwendung von konstanten kleinen Zeitintervallen notwendig, um eine genaue Bestimmung der Transitzeitpunkte zu ermöglichen. Anschließend wurden die Orbitparameter des bekannten Planeten in kartesischen Koordinaten (x,y,z) mithilfe des zu MERCURY6 gehörenden Programms ELEMENT6 zu jedem Zeitpunkt t ausgegeben. Damit kann der genaue Zeitpunkt jedes der 5.000 Transitereignisse bestimmt werden. Dazu wurde der Transitzeitpunkt willkürlich an die Position $x = 0$ gesetzt. Ein Transit zwischen den Simulationspunkten $j - 1$ und j wird erkannt, wenn sich das Vorzeichen von x von $\text{sgn}(x_{j-1}) = -$ zu $\text{sgn}(x_j) = +$ ändert. Da während des Transits die Bewegung bezüglich der x -Achse als linear angenommen werden kann, ergibt sich der

⁽⁶⁾Die benötigten Werte $R_S (1,219 \pm 0,016) R_\odot$ und $a_b = (0,0343 \pm 0,0004) \text{ AE}$ stammen aus Hartman u. a. (2011b).

genaue Transitzeitpunkt T durch lineare Interpolation:

$$T = t[j] - x[j] \cdot \frac{t[j] - t[j-1]}{x[j] - x[j-1]}. \quad (3.4)$$

Von allen damit erhaltenen 5.000 Transitzeitpunkten wurde im Folgenden eine durch lineare Regression erhaltene Ausgleichsgerade abgezogen. Die Parameter der linearen Regression ergeben dabei die Nullepoche T_0 , sowie die Periode P des Systems, so dass die resultierenden Werte einem O–C Diagramm entsprechen. Das Maximum der verbleibenden Residuen spiegelt dann die maximale Transitzeitvariation wieder.

Zur Durchführung der einzelnen Schritte wurde ein Steuerprogramm mittels der Programmiersprache PYTHON geschrieben (siehe Anhang Kapitel B.2). Das Ausführen einer einzelnen MERCURY6-Simulation dauerte auf den am Astrophysikalischen Institut verfügbaren Computern etwa zehn Minuten. Der gesamte beschriebene Algorithmus würde daher für alle 127.385 Konfigurationen etwa drei Jahre benötigen. Zentrale Aufgabe des Steuerprogramms war es daher, simultan auf verschiedenen Computern unter Nutzung von zeitweise bis mehrerer Kerne die Gesamtrechenzeit zu verkürzen, so dass die gesamte Simulation nur 3 Monate in Anspruch nahm.

3.2.2 Stabilität des Systems

Neben der Frage, welche Systemkonfiguration zu welchem TTV-Signal führt, sollte auch die Frage beantwortet werden, ob die simulierten Systeme auf längeren Zeitskalen stabil sind. Dazu wurde der bereits beschriebene Programmcode derart geändert, dass ein grobes Raster von 14.986 Konfigurationen für eine Dauer von jeweils 10^8 Umläufen von HAT-P-32b, d.h. knapp 600.000 Jahre simuliert wurden. Die Zeitschritte in MERCURY6 wurden dabei vergrößert, so dass initial zwischen zwei berechneten Orbitelementen 10 Tage liegen. Das Programm wurde beendet, sobald eine Kollision der Planeten erfolgt, oder aber ein Körper aus dem System geschleudert wird. Die Beschränkung der Simulationsdauer wird dabei bedingt durch die benötigte Rechenzeit, welche pro Konfiguration 11 Stunden beträgt. Auch hier diente daher ein Steuerprogramm zur Parallelisierung und damit zur Reduzierung der Gesamtrechenzeit von 18.8 Jahren auf 8 Monate.

4 Das HAT-P-32 System

Der extrasolare Planet HAT-P-32b wurde durch das ungarische Teleskop Netzwerk HAT-NET (*Hungarian-made Automated Telescope Network*; Bakos u. a., 2004) von Hartman u. a. (2011b) gefunden. Mit einer Masse von $m_p \sim 0,9 M_{\text{J}}$ sowie einem Radius von $R_p \sim 1,8 R_{\text{J}}$ ist er einer der größten jupiterartigen Exoplaneten die bisher gefunden wurden⁽⁷⁾. Bereits in der Entdeckungsveröffentlichung stellten die Autoren fest, dass die gemessenen Radialgeschwindigkeiten neben einer nicht-exzentrischen, auch eine exzentrische Lösung besitzen, auch wenn die Zirkularisationszeitskalen wesentlich kürzer sind als das Alter des Systems. Zusätzlich finden sie eine starke Schwankung der Radialgeschwindigkeit, welche Hartman u. a. (2011b) als mögliche Anzeichen für einen zusätzlichen Begleiter im System sehen. Dies macht HAT-P-32b zu einem idealen Kandidaten für die Suche nach TTV-Signalen.

Während der Durchführung der hier präsentierten Analyse wurde der Stern und sein Begleiter von weiteren Arbeitsgruppen unter verschiedenen Aspekten untersucht. So wurden unter anderem durch Sada u. a. (2012) drei weitere Transitereignisse am Kitt Peak National Observatory aufgenommen und für eine Neubestimmung der Transitparameter genutzt. Gibson u. a. (2013) führten eine Transitbeobachtung simultan in mehreren schmalbandigen Filtern durch und stellten fest, dass sich keine relevanten wellenlängenabhängigen Unterschiede finden lassen.

Unter Zuhilfenahme von adaptiver Optik fanden Adams u. a. (2013) ein Objekt im Abstand von $2,9''$, welches später von Zhao u. a. (2014) als M-Zwerg identifiziert werden konnte und sich mutmaßlich mit HAT-P-32 mitbewegt. Da noch keine veröffentlichten Resultate zu dieser Eigenbewegungsanalyse vorliegen, wird der Transitplanet im Folgenden weiterhin als HAT-P-32b bezeichnet. Bei positiver Bestätigung des M-Zwerges als zum System zugehörig, würde eine konsequente Bezeichnung, wie Zhao u. a. (2014) dies bereits tun, als HAT-P-32Ab notwendig sein.

⁽⁷⁾Die hier angegebenen Größen beziehen sich stets auf die nicht-exzentrische Lösung von (Hartman u. a., 2011b) mit den exakten Werten $R_p = (1,789 \pm 0,025) R_{\text{J}}$ und $m_p = (0,860 \pm 0,164) M_{\text{J}}$. Die exzentrische Orbitlösung weist einen geringfügig größeren Wert für Masse und Radius auf. Stand: 23. Juni 2015

Tabelle 4.1: Die von Hartman u. a. (2011b) ermittelten Parameter für den Exoplaneten HAT-P-32b. Als Ergänzung wurden die Parameter der Mitte-Rand-Verdunklung des Muttersterns aus Claret & Bloemen (2011) mit aufgeführt, welche mithilfe des EXOFAST/QUADLD-Codes von Eastman u. a. (2013) durch lineare Interpolation bezüglich der Effektivtemperatur, der Schwerebeschleunigung und der Metalizität ermittelt wurden.

Parameter	Formelzeichen [Einheit]	Wert
Summe der Radien	$r_p + r_s$	$0,1902 \pm 0,0013$
Radienverhältnis	R_p/R_s	$0,1508 \pm 0,0004$
Inklination	$i [^\circ]$	$88,9 \pm 0,4$
inverser normierter stellarer Radius	a/R_s	$6,05 \pm 0,04$
große Halbachse	$a [AE]$	$0,0343 \pm 0,0004$
Massenverhältnis des Systems	m_p/M_S	$0,0007 \pm 0,0002$
Exzentrizität	e	0
Orbitperiode	$P [d]$	$2,150008 \pm 0,000001$
Effektivtemperatur	$T_{\text{eff}} [K]$	6207 ± 88
Schwerebeschleunigung	$\log g [cgs]$	$4,33 \pm 0,01$
Metallizität	$[Fe/H] [dex]$	$-0,04 \pm 0,08$
Mitte-Rand-Verdunklungsgesetz	LD	quadratisch
linearer LD-Koeffizient R-Band		0,28
nichtlinearer LD-Koeffizient R-Band		0,35
linearer LD-Koeffizient V-Band		0,37
nichtlinearer LD-Koeffizient V-Band		0,34

Durch eine neue Messung der Radialgeschwindigkeit des Sterns konnten Knutson u. a. (2014) einen Trend feststellen, welcher auf die Existenz eines Begleiters auf einem sehr weit entfernten Orbit hindeutet. Aufgrund der Eigenschaften dieses Trends konnten sie allerdings ausschließen, dass es sich dabei um den stellaren Begleiter handelt, sondern postulierten die Existenz eines weiteren, bisher unentdeckten Objektes im System. Das zusätzliche Objekt soll sich in einer Entfernung von $a = (3,5 \dots 21) AE$ befinden und wäre damit aufgrund des resultierenden großen Periodenverhältnisses ($P_c/P_b > 50$) trotz dessen Massen von $m \sin i = (5 \dots 500) M_{\oplus}$ nicht durch TTV-Signale von HAT-P-32b nachweisbar. Auch die Schwankungen in der Radialgeschwindigkeit können nach Knutson u. a. (2014) nicht von dem entdeckten stellaren Begleiter herrühren.

Tabelle 4.1 führt alle für die Analyse der Transits von HAT-P-32b notwendigen Parameter aus der Entdeckungsveröffentlichung von Hartman u. a. (2011b), sowie die zur Modellanpassung benötigten Mitte-Rand-Verdunklungskoeffizienten (LD-Koeffizienten) aus

Claret & Bloemen (2011) auf. Für den Transitplaneten HAT-P-32b konnten im Rahmen dieses Projektes die meisten Daten aufgenommen werden, so dass hier eine vollständige Analyse der Daten möglich war und im Folgenden vorgestellt werden soll. In Kapitel 4.1 wird dabei kurz auf die für HAT-P-32b gewonnenen Beobachtungsdaten eingegangen. Kapitel 4.2 erläutert im Detail das Vorgehen bei der Modellierung der einzelnen Transitereignisse und die Bestimmung der finalen Transitparameter. Anschließend wird der Nutzen von simultanen Beobachtungen betrachtet, bevor Kapitel 4.4 speziell auf die Suche nach Transitzeitvariationen im HAT-P-32 System eingeht. Eine ausführliche Beschreibung möglicher Begleiter im System gibt Kapitel 4.5. Abschließend zeigt Kapitel 4.6 die Ergebnisse der Bestimmung der Transitparameter a/R_S , i und R_p/R_S .

4.1 Transitbeobachtungen

Wie bereits in Abschnitt 2.2.3 erwähnt, wurden im Rahmen der Untersuchung des Transitplaneten HAT-P-32b 47 Beobachtungen von Transitereignissen durchgeführt. Insgesamt haben sich dabei zehn verschiedene Teleskope an den Beobachtungen beteiligt, von denen die meisten dem YETI Netzwerk angehören.

Die erhaltenen photometrischen Daten von 21 Beobachtungen konnten nicht zur weiteren Analyse verwendet werden. Als Indikator für die Qualität einer Lichtkurve kann die photometrische Rauschrate pnr (*engl* photometric noise rate, Fulton u. a., 2011) verwendet werden. Diese ist definiert als Verhältnis der photometrischen Präzision, d.h. die Wurzel der Summe der quadratischen Abweichung (rms) vom Transitmodell, zur Wurzel der Anzahl der Datenpunkte pro Minute Γ .

$$pnr = \frac{rms}{\Gamma} \quad (4.1)$$

Es wurden für die Analyse nur Daten mit $pnr < 4$ verwendet, da sich zeigte, dass dies mit einer Genauigkeit der Mittelpunktbestimmung von $\Delta T_{mid} \sim 1$ min korreskondiert. Eine der wenigen Ausnahmen stellt zum Beispiel die Transitbeobachtung in Epoche 820 mit dem OSN 1,5m Teleskop dar, bei der aufgrund von Schwankungen im Transit die Streuung der Datenpunkte relativ hoch ist, gleichzeitig die Flanken des Transits sehr gut genau beobachtet und somit eine genauere Bestimmung des Mittelpunktes möglich war. Eine Auflistung aller nicht weiter verwendbaren Beobachtungen mit dem jeweiligen Grund für den Ausschluss aus der weiteren Analyse findet sich in Tabelle 4.2.

Damit verbleiben 26 nutzbare Transitbeobachtungen, von denen Abbildung 4.1 exem-

Tabelle 4.2: Die Liste der partiellen oder nicht weiter nutzbaren Beobachtungen von Transitereignissen des extrasolaren Planeten HAT-P-32b. Die Epoche bezieht sich auf die ursprünglich von Hartman u. a. (2011b) publizierten Ephemeriden.

#	Epoche	Teleskop	Bemerkungen
1	654	Jena 0,6m	aufkommendes schlechtes Wetter
2	660	Ankara 0,4m	große photometrische Fehler
3	666	Torun 0,6m	schlechte Wetterbedingungen
4	673	Jena 0,6m	aufkommendes schlechtes Wetter
5	693	Tenerife 1,2m	schlechtes Wetter zu Transitbeginn
6	700	Tenerife 1,2m	schlechtes Wetter zu Transitbeginn
7	708	Gettysburg 0,4m	schlechte Wetterbedingungen
8	713	Jena 0,6m	aufkommendes schlechtes Wetter
9	807	Antalya 1,0m	technische Probleme
10	821	Tenerife 1,2m	aufkommendes schlechtes Wetter
11	833	Jena 0,25m	schlechte Wetterbedingungen
12	834	Jena 0,25m	aufkommendes schlechtes Wetter
13	834	OSN 1,5m	aufkommendes schlechtes Wetter
14	840	Tenerife 1,2m	große photometrische Fehler
15	848	Tenerife 1,2m	aufkommendes schlechtes Wetter
16	854	Tenerife 1,2m	schlechtes Wetter zu Transitbeginn
17	855	Tenerife 1,2m	große photometrische Fehler
18	861	Tenerife 1,2m	schlechte Wetterbedingungen
19	867	Tenerife 1,2m	schlechte Wetterbedingungen
20	906	Torun 0,6m	aufkommendes schlechtes Wetter
21	1001	Torun 0,6m	große photometrische Fehler

plarisch eine dieser Aufnahmen zeigt. Im Speziellen handelt es sich dabei um eine Transitbeobachtung vom 21. November 2011 am Nationalen Bulgarischen Observatorium in Rozhen. Für die Beobachtung wurde ein Schmidtteleskop mit einem Spiegeldurchmesser von 2,0 m genutzt. Die Wetterbedingungen waren in der entsprechenden Nacht sehr gut, so dass eine photometrische Präzision von 1,0 mmag erreicht werden konnte. Der Transitmittelpunkt wurde auf 14 s genau bestimmt. Diese Messung stellt damit die präziseste Transitmittelpunktmessung im Projekt dar.

Die verbleibenden 25 Transitlichtkurven sind als Übersicht in Abbildung 4.2 gezeigt. Zusätzlich sind in der Literatur die Ergebnisse weiterer Transitbeobachtungen von HAT-P-32b verfügbar. Sada u. a. (2012) untersuchten innerhalb eines Beobachtungsprojektes am Kitt Peak National Observatory die Transits unterschiedlicher extrasolarer Planeten, um deren Eigenschaften näher zu charakterisieren. Dabei konnten sie auch zwei verschiedene Transits von HAT-P-32b beobachten. Eines dieser Transitereignisse konnte mit zwei Teleskopen simultan beobachtet werden. Auch die Ergebnisse der Analysen von Knutson u. a. (2014), Adams u. a. (2013) und Zhao u. a. (2014) wurden, soweit verwendbar, zur Verbesserung der in dieser Arbeit erhaltenen Systemparameter genutzt.

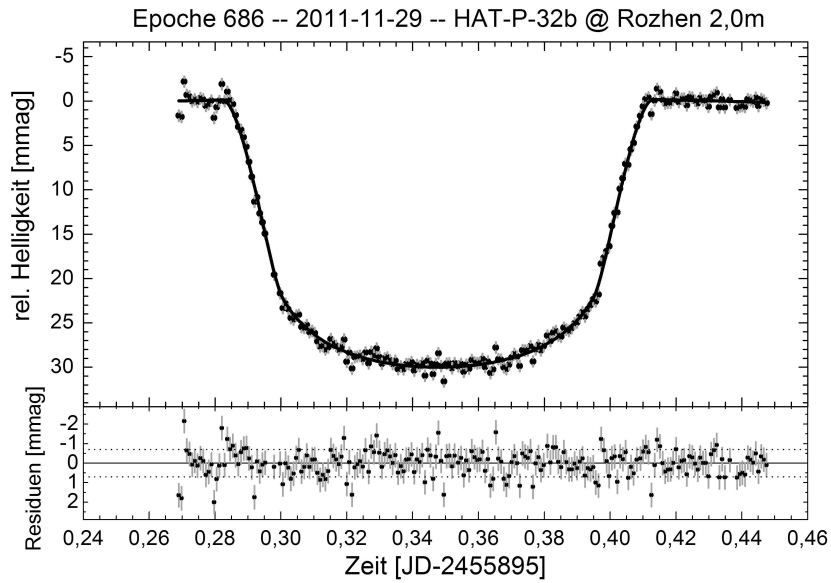


Abbildung 4.1: Die präziseste Lichtkurve, die innerhalb des Projektes aufgenommen wurde. Sie stammt vom Rozhen 2.0m Teleskop und weist eine photometrische Genauigkeit von nur $rms = 0,8$ mmag sowie $pnr = 0,68$ mmag auf. Der Transitmittelpunkt konnte auf 14 s genau bestimmt werden.

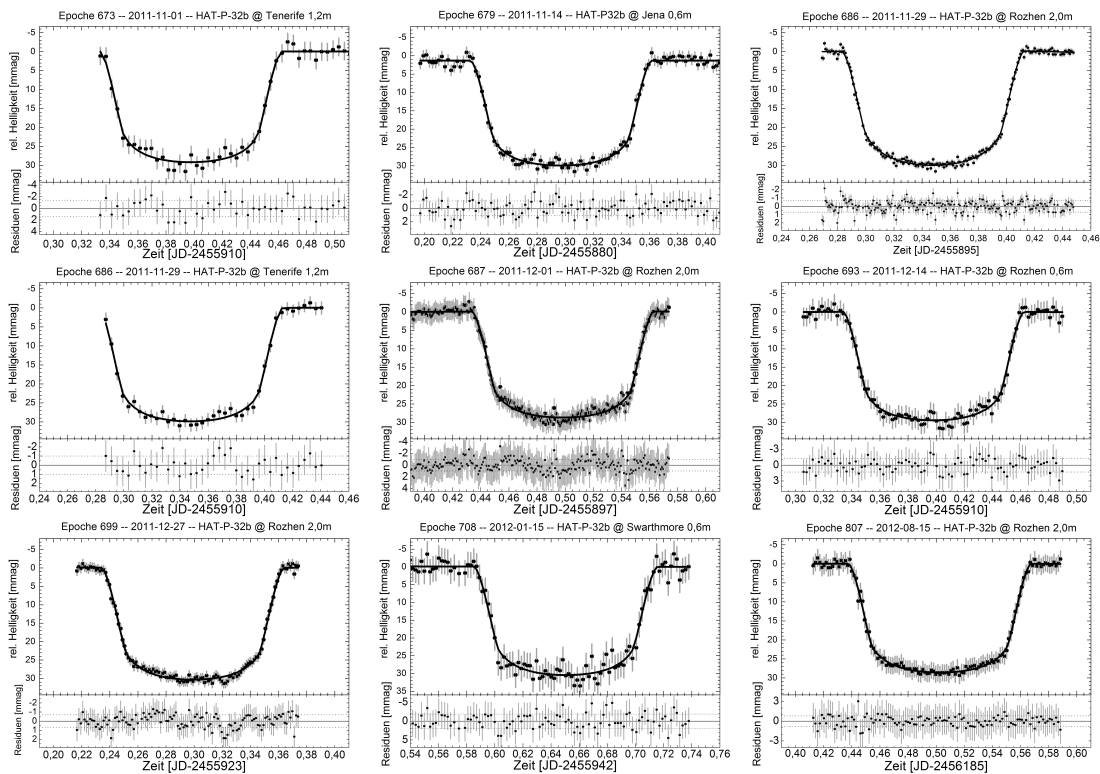


Abbildung 4.2: Neun der 26 verwendbaren Transitlichtkurven von HAT-P-32b. Anhang A zeigt alle Transitlichtkurven noch einmal in einer größeren Darstellung.

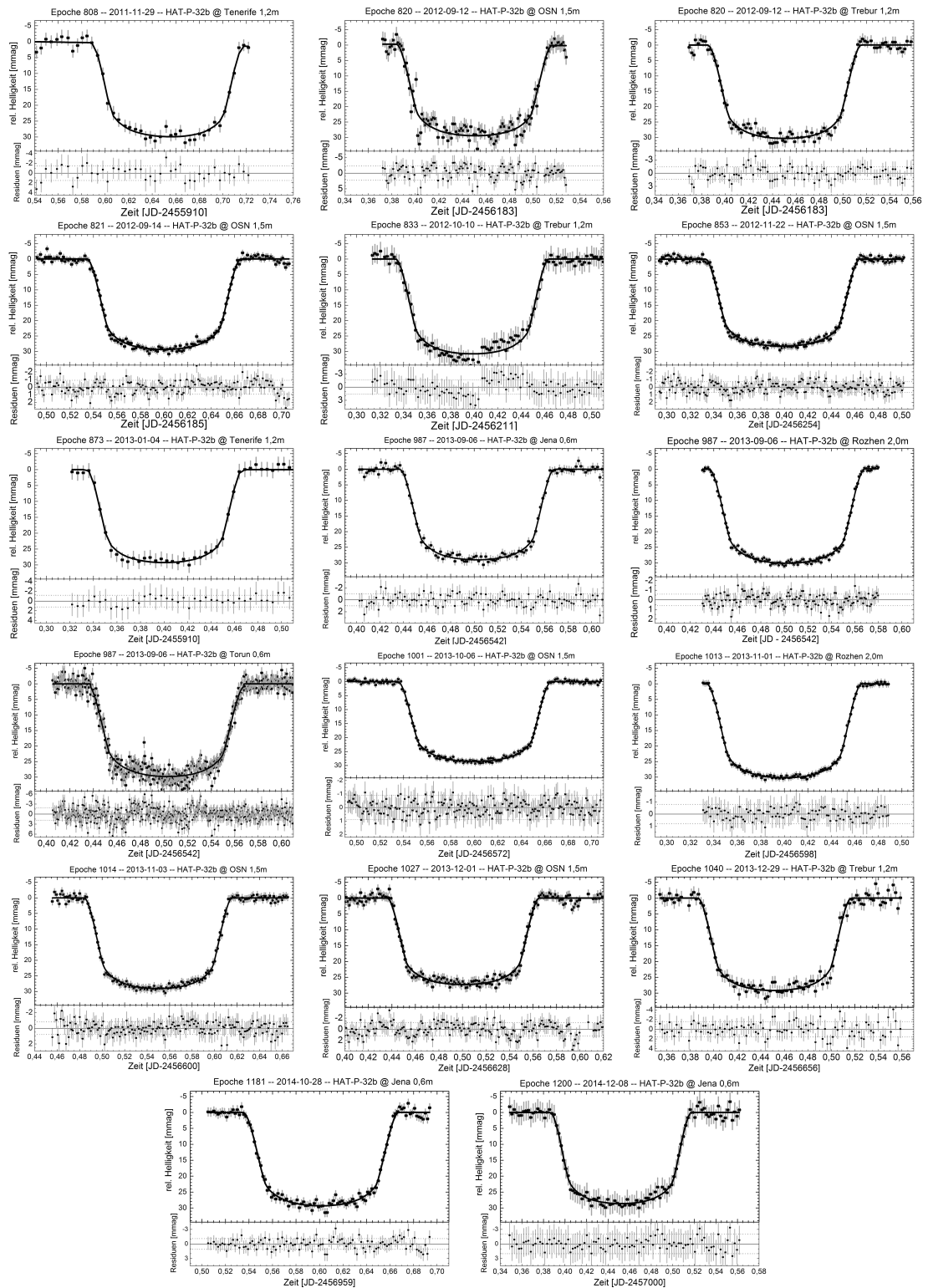


Abbildung 4.2 (fortgesetzt): Die verbleibenden 17 Lichtkurven der verwendbaren Transitbeobachtungen von HAT-P-32b.

4.2 Modellierung

Die Modellierung der einzelnen Transits mittels TAP und JKTEBOP zeigte, dass in allen Fällen die Unterschiede zwischen den Modellen kleiner waren als die ermittelten Modellfehler. Dies betrifft sowohl den Vergleich der Resultate von TAP und JKTEBOP untereinander, als auch den Vergleich der Modellierung mit festen bzw. variablen LD-Koeffizienten und gebinnten bzw. ungebinten Daten (Vgl. Kapitel 3.1.3) innerhalb eines Programms. Beide Programme liefern folglich vergleichbare Resultate. Ebenfalls zeigte die Analyse, dass die Fehlerbalken der von JKTEBOP ermittelten Modellparameter stets kleiner waren, als die von TAP ermittelten Fehler. Dieses Resultat deckt sich dabei mit dem von Carter & Winn (2009): die von JKTEBOP verwendeten Fehlerbestimmungsmethoden berücksichtigen den Einfluss von korreliertem Rauschen nicht ausreichend, was zu einer Unterschätzung der Fehler führen kann. Auch Gazak u. a. (2012) können für simulierte Daten zeigen, dass die Größe der Fehler bei den von JKTEBOP verwendeten Methoden in einigen Fällen unterschätzt werden. Daher wurden die Fehler von JKTEBOP für die weitere Analyse nicht verwendet, um eine konservative Abschätzung der Fehler zu erhalten.

In allen betrachteten Fällen war die Streuung der Modellparameter kleiner als der mittlere Fehler, so dass als ebenfalls konservative Fehlerabschätzung die Streuung der Modellparameter als endgültiger Fehler verwendet wurde.

Die Auswertung aller 26 Transitereignisse zeigt, dass ein nachträgliches Binning der Lichtkurve keine Verbesserung bei der Bestimmung des Transitmittelpunktes bringt. Dies ist maßgeblich durch die Tatsache begründet, dass das Daten-Binning zwar eine Reduzierung des Rauschens der Lichtkurve bewirkt, gleichzeitig aber die Zeitauflösung verschlechtert. Dadurch sind weniger Datenpunkte innerhalb des Auf- und Abstieges zur Bestimmung der Modellparameter vorhanden. Beide Effekte heben sich in ihrer Wirkung derart auf, dass kein signifikanter Unterschied sichtbar ist.

4.3 Simultane Beobachtungen

Durch den Zugang zum YETI-Netzwerk ist es möglich, einen Transit nicht nur von einem Observatorium aus zu beobachten. Wenn dies aufgrund des Zeitpunkts des Transits möglich war, wurde daher versucht, einen Transit mit verschiedenen YETI-Teleskopen gleichzeitig zu beobachten. Die aus diesen Beobachtungen resultierenden Lichtkurven ermöglichen eine Kontrolle auf die Richtigkeit einer einzelnen Beobachtung, da die Teleskope untereinander nicht in Verbindung stehen. Ebenfalls ist die Anzahl der Vergleichssterne

bei der Erstellung der Lichtkurve durch die unterschiedlichen Größen der Gesichtsfelder nicht identisch. Ein Vergleich der Ergebnisse der Modellanpassung von simultan beobachteten Transits kann so die Vertrauenswürdigkeit einer Beobachtung bestätigen.

Innerhalb dieses Projekts konnte in sieben Fällen eine Transitbeobachtung simultan an verschiedenen Teleskopen gestartet werden. Leider war es nur in drei Fällen möglich, alle erhaltenen Lichtkurven eines Transits für die weitere Analyse zu verwenden (Epochen 686, 820 und 987). Ein Transit wurde dabei sogar von drei Teleskopen gleichzeitig beobachtet. Drei weitere Transitereignisse konnten zwar von jeweils zwei Teleskopen beobachtet werden, allerdings ergab sich bei jeweils einer der beiden Beobachtungen eine zu große Streuung der Datenpunkte, so dass diese Daten nicht weiter verwendet wurden (Epochen 708, 807 und 833). Sada u. a. (2012) gelang die Beobachtung eines Transits in Epoche 663 mit zwei Teleskopen des Kitt Peak National Observatory.

Tabelle 4.3: Die Ergebnisse der sieben Simultanbeobachtungen für HAT-P-32b. Mit Ausnahme der Beobachtungen in Epoche 807, in welcher eine Beobachtung von starkem Rauschen dominiert wird, stimmen alle Transitmittelpunkte innerhalb des jeweils größeren Fehlerbalkens überein. Die Beobachtungen in Epoche 820 zeigen sogar einen Unterschied von nur 2,5 s.

Epoche	Teleskop	$T_{\text{mid}}[BJD_{\text{TDB}},\text{d}]$	$\Delta T_{\text{mid}} [\text{d}]$
663	KPNO 2,1m	$2\,455\,845,902\,87 \pm 0,000\,24$	0,000 27
663	KPNO 0,5m	$2\,455\,845,903\,14 \pm 0,000\,40$	
686	Rozhen 2,0m	$2\,455\,895,352\,97 \pm 0,000\,16$	0,000 49
686	Tenerife 1,2m	$2\,455\,895,352\,48 \pm 0,000\,80$	
708	Swarthmore 0,6m	$2\,455\,942,652\,87 \pm 0,000\,64$	0,001 08
708	Gettysburg 0,4m	$2\,455\,942,651\,79 \pm 0,001\,13$	
807	Rozhen 2,0m	$2\,456\,155,503\,85 \pm 0,000\,26$	0,001 57
807	Antalya 1,0m	$2\,456\,155,502\,28 \pm 0,001\,07$	
820	OSN 1,5m	$2\,456\,183,453\,64 \pm 0,000\,85$	0,000 03
820	Trebur 1,2m	$2\,456\,183,453\,61 \pm 0,000\,49$	
833	Trebur 1,2m	$2\,456\,211,403\,61 \pm 0,000\,56$	0,000 94
833	Jena 0,25m	$2\,456\,211,402\,67 \pm 0,002\,14$	
987	Rozhen 2,0m	$2\,456\,542,505\,30 \pm 0,000\,18$	0,000 16
987	Jena 0,6m	$2\,456\,542,505\,38 \pm 0,000\,32$	
987	Torun 0,6m	$2\,456\,542,505\,22 \pm 0,000\,52$	

Ein Vergleich der simultanen Beobachtungen bezüglich des ermittelten Transitmittelpunktes zeigt Tabelle 4.3. In den Beobachtungsepochen 663, 686, 820 und 987 stimmt der Transitzeitpunkt innerhalb des jeweils größeren Fehlerbalkens der Einzelmessungen überein. Die Beobachtungen in Epoche 820 liegen sogar nur um 0,00003d ($\sim 2,5\text{s}$) ausein-

ander, während der individuelle Fehlerbalken um mehr als eine Größenordnung höher ist. Auch in den Beobachtungsepochen 708 und 833 zeigt sich, dass die Ergebnisse der Modellierung innerhalb des Fehlerbalkens der verrauschteren Aufnahme überein stimmen. Die kompletten Ergebnisse aller Einzelmodellierungen sind in Tabelle 4.4 aufgelistet.

Tabelle 4.4: Die Einzelergebnisse der Modellierung der 25 guten Transitbeobachtungen (obere Zeilen) für die Modellparameter T_{mid} , a/R_s , $k = R_p/R_s$ und i . Die unteren Zeilen enthalten außerdem die vollständigen Beobachtungen, die aufgrund des zu großen Rauschens für die weitere Analyse ausgeschlossen wurden. Die letzten beiden Spalten geben das r_{rms} und p_{nr} der Beobachtung an. Im Falle der Beobachtungen mit dem Tenerife 1,2m Teleskop resultiert der höhere p_{nr} -Wert aus der Tatsache, dass quasi-simultane Beobachtungen in zwei Filtern gemacht wurden, indem der Filter nach jeder Einzelaufnahme gewechselt wurde und so nur die Hälfte alle Bilder im r_p -Filter vorlagen. Die Aufnahmen im B -Filter wurden aufgrund des geringeren Flusses des Sterns nicht verwendet.

Datum	Epochen	Teleskop	$T_{\text{mid}} - 2,450000 \text{ d}$	a/R_s	$k = R_p/R_s$	$i [^\circ]$	r_{rms} [mmag]	p_{nr} [mmag]
2011/1/1/01	673	Tenerife 1,2m	5867,40301 ± 0,00073	—	—	—	2,3	3,20
2011/1/1/14	679	Jena 0,6m	5880,30267 ± 0,00033	6,13 ± 0,09	0,1493 ± 0,0016	89,3 ± 1,0	1,9	1,97
2011/1/1/29	686	Rozhen 2,0m	5895,35297 ± 0,00016	6,09 ± 0,04	0,1507 ± 0,0010	89,5 ± 0,6	1,0	0,62
2011/1/1/29	686	Tenerife 1,2m	5895,35249 ± 0,00080	—	—	—	1,7	2,36
2011/1/2/01	687	Rozhen 2,0m	5897,50328 ± 0,00033	6,08 ± 0,09	0,1508 ± 0,0016	89,3 ± 0,9	1,5	0,94
2011/1/2/14	693	Rozhen 0,6m	5910,40274 ± 0,00043	6,11 ± 0,13	0,1508 ± 0,0023	89,5 ± 1,1	2,0	3,40
2011/1/2/27	699	Rozhen 2,0m	5923,30295 ± 0,00031	6,03 ± 0,10	0,1536 ± 0,0018	88,5 ± 0,9	1,2	0,96
2012/0/1/15	708	Swarthmore 0,6m	5942,65287 ± 0,00064	6,04 ± 0,18	0,1544 ± 0,0024	88,9 ± 1,4	3,3	3,28
2012/0/8/15	807	Rozhen 1,2m	6155,50385 ± 0,00026	6,01 ± 0,11	0,1496 ± 0,0014	88,5 ± 1,0	1,3	1,11
2012/0/8/18	808	Tenerife 1,2m	6157,65470 ± 0,00072	—	—	—	2,6	3,71
2012/0/9/12	820	OSN 1,5m	6183,45364 ± 0,00085	5,96 ± 0,18	0,1524 ± 0,0027	88,7 ± 1,4	4,3	4,85
2012/0/9/12	820	Trebur 1,2m	6183,45361 ± 0,00049	6,05 ± 0,14	0,1548 ± 0,0022	88,5 ± 1,2	2,1	2,33
2012/0/9/14	821	OSN 1,5m	6185,60375 ± 0,00033	6,01 ± 0,12	0,1509 ± 0,0016	88,2 ± 1,2	1,3	1,07
2012/1/0/10	833	Trebur 1,2m	6211,40361 ± 0,00056	5,98 ± 0,25	0,1554 ± 0,0059	88,1 ± 1,5	2,2	2,18
2012/1/1/22	853	OSN 1,5m	6254,40404 ± 0,00022	6,037 ± 0,062	0,1507 ± 0,0018	89,2 ± 0,8	1,1	0,84
2013/0/1/04	873	Tenerife 1,2m	6542,40397 ± 0,00058	—	—	—	1,9	2,87
2013/0/9/07	987	Jena 0,6m	6542,50538 ± 0,00032	6,04 ± 0,11	0,1497 ± 0,0016	88,7 ± 1,1	1,6	1,57
2013/0/9/07	987	Rozhen 2,0m	6542,50530 ± 0,00018	5,97 ± 0,09	0,1535 ± 0,0012	88,3 ± 0,8	0,9	0,67
2013/0/9/07	987	Torun 0,6m	6542,50522 ± 0,00052	5,89 ± 0,20	0,1515 ± 0,0029	87,9 ± 1,4	3,5	2,33
2013/1/0/06	1001	OSN 1,5m	6572,60532 ± 0,00018	6,11 ± 0,06	0,1465 ± 0,0013	89,2 ± 0,7	0,9	0,63
2013/1/1/01	1013	Rozhen 2,0m	6598,40539 ± 0,00017	6,05 ± 0,06	0,1511 ± 0,0010	88,9 ± 0,8	0,8	0,68
2013/1/1/03	1014	OSN 1,5m	6600,55546 ± 0,00017	6,02 ± 0,05	0,1503 ± 0,0009	89,2 ± 0,8	1,3	1,33
2013/1/2/01	1027	OSN 1,5m	6628,50585 ± 0,00031	6,13 ± 0,09	0,1475 ± 0,0022	89,2 ± 0,9	1,8	1,59
2013/1/2/29	1040	Trebur 1,2m	6656,45533 ± 0,00045	6,07 ± 0,13	0,1509 ± 0,0022	88,8 ± 1,1	2,6	2,74
2014/1/0/28	1181	Jena 0,6m	6959,60681 ± 0,00039	5,92 ± 0,18	0,1514 ± 0,0021	87,7 ± 1,4	2,0	1,87
2014/1/2/08	1200	Jena 0,6m	7000,45692 ± 0,00042	6,02 ± 0,13	0,1492 ± 0,0021	88,6 ± 1,2	2,0	2,04
2011/1/0/04	660	Ankara 0,4m	5839,45347 ± 0,00101	5,9 ± 0,2	0,1448 ± 0,0021	88,1 ± 1,4	4,7	2,22
2012/0/1/15	708	Getysburg 0,4m	5942,65179 ± 0,00113	5,79 ± 0,36	0,1493 ± 0,0054	87,3 ± 1,7	2,5	4,25
2012/0/8/15	807	Tübitak 1,0m	6155,50228 ± 0,00107	5,99 ± 0,19	0,1626 ± 0,0033	88,6 ± 1,5	6,1	3,68
2012/1/0/10	833	Jena 0,25m	6211,40267 ± 0,00214	6,04 ± 0,65	0,1514 ± 0,0089	86,5 ± 2,5	5,3	6,96
2012/1/0/25	840	Tenerife 1,2m	6226,45618 ± 0,00102	—	—	—	3,8	5,05
2012/1/2/22	867	Tenerife 1,2m	6284,50460 ± 0,00100	—	—	—	3,3	5,10

4.4 Transit-Zeit-Variationen

Das Hauptanliegen dieser Arbeit ist die Suche nach möglichen Transitzeitvariationen von HAT-P-32b. Dazu muss zunächst die Umlaufperiode P des Systems mithilfe der Formel

$$T_{\text{mid}} = T_0 + P \cdot E \quad (4.2)$$

neu bestimmt werden. Die jeweilige Beobachtungsepoche $E^{(i)}$ der i -ten Transitbeobachtung wurde dabei mithilfe der ursprünglich von Hartman u. a. (2011b) publizierten Ephemeriden mit einer Periode von $P = 2,150008$ d, sowie dem Zeitpunkt des nullten Transits $T_0[\text{BJD}_{\text{TDB}}] = 2454420,44637$ d durch Umstellen von Gleichung (4.2) ermittelt:

$$E^{(i)} = \text{round} \left[\left(T_{\text{mid}}^{(i)} - T_0 \right) / P \right]. \quad (4.3)$$

Die Differenz zwischen dem durch die Modellanpassung ermittelten Transitmittelpunkt und dem aufgrund der bekannten Ephemeriden berechneten Wert kann anschließend in einem O–C Diagramm über der Beobachtungsepoche aufgetragen werden. Das Resultat ist in Abbildung 4.3 dargestellt. Zur Darstellung und späteren Analyse wurde dabei das Programm ORIGIN⁽⁸⁾ verwendet.

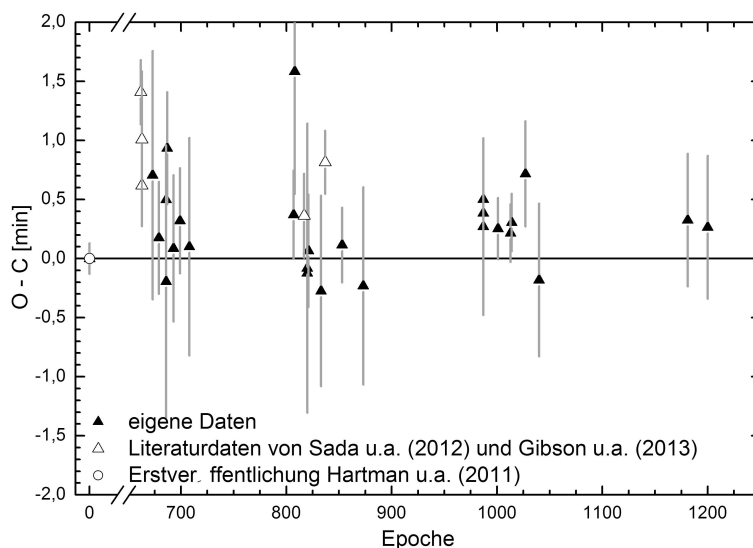


Abbildung 4.3: Das aus den ursprünglichen Ephemeriden von Hartman u. a. (2011b) resultierende O–C Diagramm für HAT-P-32b. Die Abweichung der Datenpunkte nach oben weist darauf hin, dass die bestimmte Periode zu klein ist.

⁽⁸⁾ORIGIN ist ein von ORIGINLAB, Northampton, MA entwickeltes Programm zur Datenanalyse und grafischen Auswertung.

Gut zu sehen ist, dass die meisten Datenpunkte einen positiven Wert aufweisen. Die entsprechenden Transitereignisse finden damit später statt, als durch die bekannten Ephemeriden zu erwarten gewesen wäre, was auf eine zu klein bestimmte Periodendauer hindeutet. Um dies zu korrigieren, wurde daher zunächst eine lineare Anpassung mittels der in ORIGIN zur Verfügung gestellten Methoden durchgeführt, wobei sowohl der Zeitpunkt der Nullepoche, als auch die Periode als freie Parameter dienten. Die zur Verfügung stehenden 31 Datenpunkte – 26 eigene Beobachtungen, sowie drei Datenpunkte von Sada u. a. (2012), zwei Datenpunkte von Gibson u. a. (2013) und ein Datenpunkt von Hartman u. a. (2011b) – verteilen sich auf eine Zeitspanne von fast sieben Jahren und haben damit eine größere zeitliche Abdeckung als frühere Studien. Als Wichtung der einzelnen Datenpunkte während der linearen Anpassung wurden deren individuelle Modellfehler als instrumentelle Fehler verwendet. Im Ergebnis ergibt sich eine Korrektur der Periode um $\Delta P = 0,029$ s.

Tabelle 4.5: Ein Vergleich der Resultate dieser Analyse, sowie der Literaturdaten von Hartman u. a. (2011b), Sada u. a. (2012) und Gibson u. a. (2013). Alle Nullepochen T_0 wurden, wenn nicht bereits so angegeben, zu BJD_{TDB} konvertiert.

	T_0 [d]	P [d]
diese Analyse	2 454 420,446 45 \pm 0,000 09	2,150 008 24 \pm 0,000 000 12
Hartman u. a. (2011b)	2 454 420,446 37 \pm 0,000 09	2,150 008 \pm 0,000 001
Sada u. a. (2012)	2 454 420,446 37 \pm 0,000 09	2,150 010 3 \pm 0,000 000 3
Gibson u. a. (2013)	2 454 942,899 220 \pm 0,000 077	2,150 008 5 \pm 0,000 000 2
Zhao u. a. (2014)	2 454 420,447 12 \pm 0,000 09	2,150 008 05 \pm 0,000 000 09

Wie in Tabelle 4.5 zu sehen ist, stimmt das Ergebnis der Neubestimmung der Periode gut mit den bereits bekannten Werten von Hartman u. a. (2011b) und Gibson u. a. (2013) überein. Da bereits aus der Grafik ersichtlich ist, dass die Transitzeitpunkte von Sada u. a. (2012) stärker von der Nulllinie abweichen ist es nicht verwunderlich, dass dessen Periode innerhalb der Fehlerbalken nicht mit den anderen Werten übereinstimmt.

Betrachtet man das Ergebnis aus Sicht der Suche nach Transitzeitvariationen, so kann im resultierenden O–C Diagramm (Abbildung 4.4) keine signifikante Variabilität festgestellt werden. Mit $\chi^2_{\text{red}} = 1,078$ ist die zweiparametrische lineare Anpassung mit 31 Datenpunkten und folglich 29 Freiheitsgraden die Annahme einer konstanten Periode gerechtfertigt. Die in Abbildung 4.4 verbleibenden Abweichungen von der Nullhypothese zeigen eine Streuung von $\lesssim 1,5$ min und liegen damit in der gleichen Größenordnung wie der typische Fehlerbalken einer Modellanpassung.

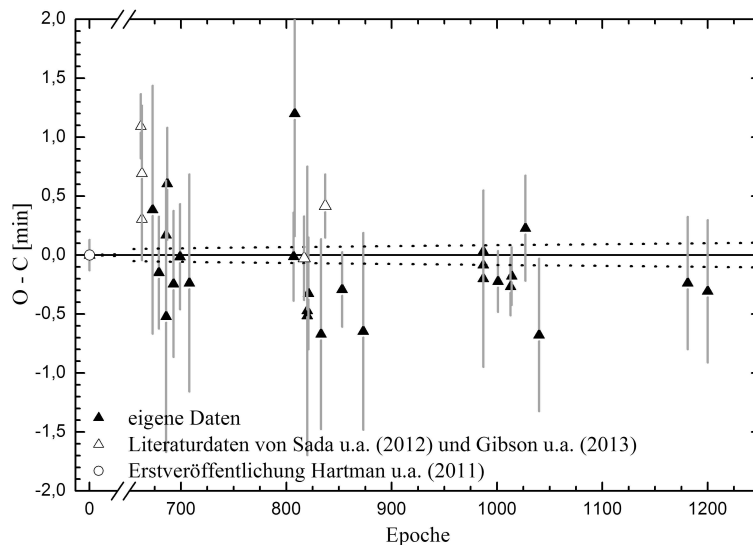


Abbildung 4.4: Das O–C Diagramm für HAT-P-32b unter Zuhilfenahme der neu bestimmten Ephemeriden mit $T_0 = (2\,454\,420,446\,45 \pm 0,000\,09)$ d und $P = (2,150\,008\,24 \pm 0,000\,000\,12)$ d. Die gestrichelte Linie gibt die Unsicherheit der neu bestimmten Periode an.

Daraus resultieren zwei Schlussfolgerungen:

1. Es ist unwahrscheinlich, dass sich um HAT-P-32 ein zusätzlicher Planet befindet, der bei HAT-P-32b eine Transitzeitvariation mit einer Amplitude $\gtrsim 1,5$ min hervorruft.
2. Es ist nicht auszuschließen, dass ein zusätzlicher Körper, welcher eine Transitzeitvariation mit einer Amplitude $\lesssim 1,5$ min erzeugt, vorhanden ist.

Letzteres schließt dabei natürlich auch das von Knutson u. a. (2014) postulierte Objekt auf einem weiten Orbit mit ein.

4.5 Einschränkungen für hypothetische Begleiter

Die mit MERCURY6 durchgeführten Simulationen wurden aufgrund der Ergebnisse des vorherigen Kapitels explizit darauf untersucht, welche Konfigurationen Transitzeitvariationen mit einer Amplitude von $\delta t \sim 1,5$ min erzeugen. Abbildung 4.5 zeigt das Resultat dieser Analyse. Dabei markiert die durchgezogene schwarze Linie im Diagramm die Konfigurationen, welche eine Amplitude von $\delta t = 1,5$ min im O–C Diagramm erzeugen. Alle Konfigurationen oberhalb dieser Linie würden zu einer größeren Amplitude führen

und können daher ausgeschlossen werden. Dies beinhaltet dabei ausdrücklich nicht das von Knutson u. a. (2014) postulierte Objekt, welches für den gefundenen Radialgeschwindigkeitstrend verantwortlich gemacht wird. Trotz dessen relativ großer Masse von bis zu $500 M_{\oplus}$ wäre bei einem Abstand von $a = (3,5 \dots 21)$ AE kein messbares TTV-Signal vorhanden. Innerhalb der Bahnresonanzen, besonders der 1:2 und 2:1-Resonanz, kann ausgeschlossen werden, dass Körper mit Massen von mehr als einigen wenigen Erdmassen existieren. Zusätzliche Planeten auf äußeren Bahnen, vor allem außerhalb der 2:1 Resonanz, führen hingegen selbst mit Massen von mehreren Jupitermassen zu relativ kleinen Störungen der Bahn von HAT-P-32b und können daher nicht ausgeschlossen werden. Die grundsätzliche Form der Kurve ist dabei ähnlich der von Agol u. a. (2005) für HD 209458 oder von Maciejewski u. a. (2011b) für WASP-12 durchgeführten Analysen.

Wie bereits in Kapitel 3.2.1 beschrieben, bezieht sich dieses Ergebnis auf zusätzliche Planeten auf koplanaren, nicht-exzentrischen Umlaufbahnen. Weist der hypothetische Planet eine andere Inklination auf als der bekannte Planet, so ist der zu erwartende Einfluss auf die Transitzeitvariation geringer, so dass die Grenzlinie der 1,5 min-Amplitude eine Art Mindestmasse angibt, die ein Körper besitzen muss, um die Transitzeitvariation hervorzurufen.

Aus Sicht der Stabilität des Systems beschreibt der Hill-Radius eines Körpers den Abstand, innerhalb welchem ein anderer Körper gravitativ an diesen gebunden ist. Für HAT-P-32b errechnet er sich zu

$$R_H = a \cdot \sqrt[3]{\frac{m_b}{3 \cdot m_S}} = 0,00216 \text{ AE} \quad (4.4)$$

mit den bereits in Tabelle 4.1 angegebenen Werten. In den Simulationen zeigte sich, dass innerhalb eines Bereiches von $4,5 R_H$ fast jede simulierte Planetenmasse einen sogenannten close encounter zur Folge hat. Dabei handelt es sich um eine gravitative Wechselwirkung bei nahen Vorbeiflügen der Körper untereinander, was zu einer Kollision der Körper, einer starken Änderung der Orbitalelemente und/oder dem Herausschleudern von einem der beteiligten Körper führt. Die Existenz eines Körpers in dieser Region kann damit ebenfalls ausgeschlossen werden. In Abbildung 4.6 ist gut erkennbar, dass in der Nähe der Resonanzen die Zahl der nahen Vorbeiflüge und damit der Bahninstabilitäten wie erwartet ansteigt, da hier die Interaktionen der beiden Planeten besonders stark sind.

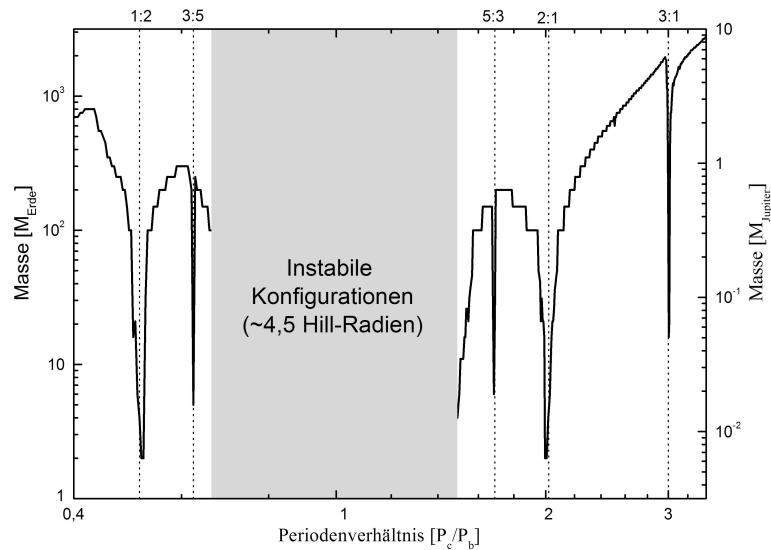


Abbildung 4.5: Die Mindestmasse eines zusätzlichen Körpers um HAT-P-32 die benötigt wird, um eine Amplitude von $\sim 1,5$ min zu erhalten (durchgezogene Linie). Dargestellt ist das Ergebnis in Abhängigkeit des Verhältnisses zwischen der Periode des zusätzlichen, hypothetischen Körpers P_c und der Periode des bekannten Planeten HAT-P-32b (P_b). Der grau schattierte Bereich markiert die dynamisch instabile Region um den bekannten Planeten ($\sim 4,5$ Hill-Radien).

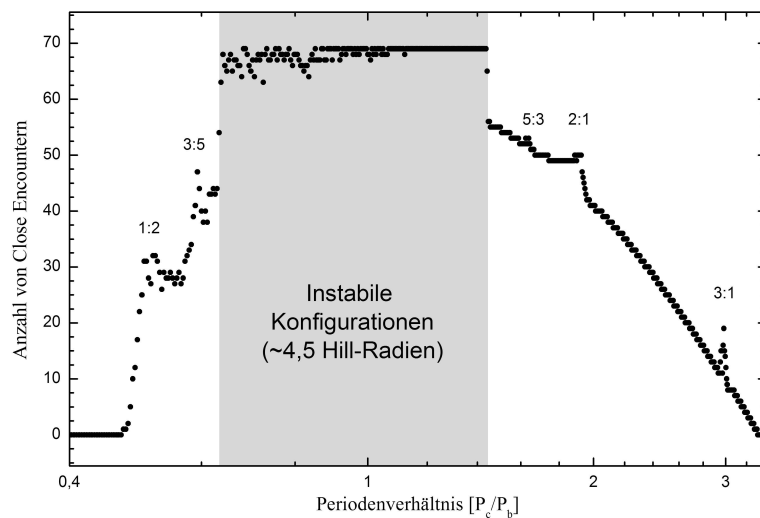


Abbildung 4.6: Die Anzahl an nahen Vorbeiflügen (*engl.* close encounters) in Abhängigkeit des Periodenverhältnisses. In dem grau markierten Bereich von $\pm 4,5 R_H$ führen fast alle simulierten Massen zu einer gravitativen Interaktion und damit zu Instabilität des Systems. Deutlich sichtbar ist auch der Anstieg nahe der Bahnresonanzen.

4.5.1 Einschränkungen aus der Radialgeschwindigkeitsschwankung

Neben der Möglichkeit, einen zusätzlichen Begleiter aus Sicht der Transitzeitvariation einzuschränken, kann auch die gemessene Radialgeschwindigkeitsschwankung herangezogen werden. Nach den Keplerschen Gesetzen ergibt sich die Umlaufperiode P eines Planeten nach

$$P^2 = \frac{4\pi^2 \cdot (a_S + a_P)^3}{G \cdot (m_S + m_P)} \quad (4.5)$$

aus der Summe der Abstände a des Sterns bzw. Planeten zum gemeinsamen Masseschwerpunkt und der Summe der Massen m der beiden Körper. Für die meisten Planeten können zwei Näherungen angenommen werden:

$a_P \gg a_S$ der gemeinsame Masseschwerpunkt befindet sich sehr nah am Stern, unter Umständen sogar innerhalb des Sterns, so dass der Abstand des Sterns von diesem wesentlich kleiner ist als der des Planeten und a_S vernachlässigt werden kann,

$m_P \ll m_S$ die Masse des Planeten ist wesentlich kleiner als die Masse des Sterns, so dass die Planetenmasse in den Rechnungen vernachlässigt werden kann.

Beide Annahmen führen vereinfacht zu:

$$\begin{aligned} P^2 &\approx \frac{4\pi^2}{G} \cdot \frac{a_P^3}{m_S} \\ a_P &= \left(\frac{G}{4\pi^2} \cdot P^2 \cdot m_S \right)^{1/3} \end{aligned} \quad (4.6)$$

Auf zirkularen Bahnen kann die Bahngeschwindigkeit des Planeten und des Sterns mit

$$v = \frac{2\pi \cdot a}{P} \quad (4.7)$$

berechnet werden. Unter Voraussetzung der Impulserhaltung gilt zudem

$$m_S \cdot a_S = m_P \cdot a_P. \quad (4.8)$$

Durch Einsetzen von Gleichung (4.6) in Gleichung (4.8) ergibt sich

$$a_S = \frac{m_P}{m_S} \cdot \left(\frac{G}{4\pi^2} \cdot P^2 \cdot m_S \right)^{1/3} \quad (4.9)$$

und schließlich für die Bahngeschwindigkeit des Sterns durch Einsetzen von (4.9) in (4.7)

$$\begin{aligned} v &= \frac{2\pi}{P} \cdot \frac{m_p}{m_s} \cdot \left(\frac{G}{4\pi^2} \cdot P^2 \cdot m_s \right)^{1/3} \\ &= (2\pi \cdot G)^{1/3} \cdot m_p \cdot P^{-1/3} \cdot m_s^{-2/3}. \end{aligned} \quad (4.10)$$

Diese Bahngeschwindigkeit spiegelt dabei auch die durch einen einzelnen Planeten verursachten Amplitude K der Radialgeschwindigkeitskurve unter der Annahme einer Inklination $i = 90^\circ$ wider. Werden die vorkommenden Größen in ihrer typischen Einheit ausgedrückt, d.h. Umlaufperiode in Tagen, Planetenmasse in Jupitermassen und Masse des Sterns in Sonnenmassen, so ergibt sich die Radialgeschwindigkeitsamplitude zu:

$$\begin{aligned} K &= \frac{(2\pi \cdot G)^{1/3} \cdot 1 M_{\text{Jup}}}{(1 \text{ d})^{1/3} \cdot (1 M_{\odot})^{2/3}} \cdot \frac{m_p}{[M_{\text{Jup}}]} \cdot \left(\frac{P}{[\text{d}]} \right)^{-1/3} \cdot \left(\frac{m_s}{[M_{\odot}]} \right)^{-2/3} \\ \frac{K}{[\text{m s}^{-1}]} &= 203,290 \cdot \frac{m_p}{[M_{\text{Jup}}]} \cdot \left(\frac{P}{[\text{d}]} \right)^{-1/3} \cdot \left(\frac{m_s}{[M_{\odot}]} \right)^{-2/3} \end{aligned} \quad (4.11)$$

Bei gegebener Sternmasse kann daher die Planetenmasse, die für eine gewisse Radialgeschwindigkeitsamplitude benötigt wird, in Abhängigkeit der Umlaufperiode bestimmt werden.

$$\begin{aligned} \frac{m_p}{[M_{\text{Jup}}]} &= \frac{K}{203,290} \cdot \left(\frac{m_s}{[M_{\odot}]} \right)^{2/3} \cdot \left(\frac{P}{[\text{d}]} \right)^{1/3} \\ \frac{m_p}{[M_{\text{Jup}}]} &= \frac{K \cdot P_b^{1/3}}{203,290} \cdot \left(\frac{m_s}{[M_{\odot}]} \right)^{2/3} \cdot \left(\frac{P}{P_b} \right)^{1/3} \end{aligned} \quad (4.12)$$

Wie bereits erwähnt, stellen Hartman u. a. (2011b) bereits in der Entdeckungsveröffentlichung von HAT-P-32b fest, dass der Stern HAT-P-32 nach Abzug der durch HAT-P-32b induzierten Radialgeschwindigkeit eine Schwankung der von $\Delta_{RV} = 78,7 \text{ m s}^{-1}$ aufweist.⁽⁹⁾ Knutson u. a. (2014) führten eine erneute Analyse der Radialgeschwindigkeit verschiedener Transitplanetensysteme durch und fanden unter anderem für HAT-P-32 eine Variation von $\Delta_{RV} = (64_{-9}^{+11}) \text{ m s}^{-1}$. Unter der Annahme, dass diese Variation von einem zusätzlichen Planeten HAT-P-32c stammt, der sich koplanar zu HAT-P-32b auf einer nicht-exzentrischen Bahn bewegt, kann mithilfe der genannten Herleitung die Masse eines potentiellen Störkörpers bei Vernachlässigung anderer Störeinflüsse in Abhängigkeit

⁽⁹⁾Der angegebene Wert wurde für einen nicht-exzentrischen Orbit von HAT-P-32b erhalten.

vom Verhältnis der Umlaufperioden zu

$$m_c \simeq 0,46367 \cdot (P_c/P_b)^{1/3} \quad (4.13)$$

bestimmt werden. Die damit berechenbare Planetenmasse gibt für jedes Periodenverhältnis diejenige Mindestmasse an, welche in einer Radialgeschwindigkeitsamplitude von $\Delta_{RV} = 64 \text{ m s}^{-1}$ resultiert. Größere Planetenmassen würden höhere Amplituden verursachen und können daher ausgeschlossen werden.

4.5.2 Einschränkungen aufgrund der Beobachtungsabdeckung

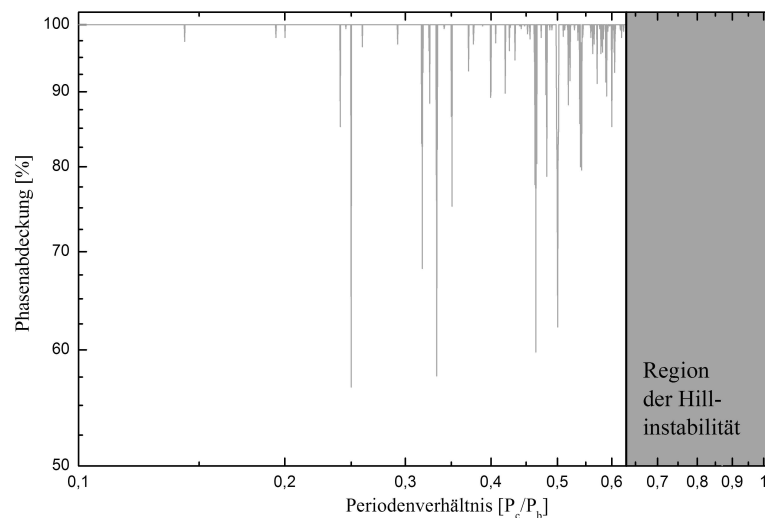


Abbildung 4.7: Die Abdeckung der Beobachtung eines kompletten Orbits eines zusätzlichen inneren Planeten. Der instabile Bereich von $\pm 4,5 R_H$ ist grau unterlegt. Für die meisten Periodenverhältnisse ist eine Abdeckung von mehr als 90% gegeben.

Aufgrund der Anzahl der Transitbeobachtungen kann auch die Frage untersucht werden, ob ein Transit eines zusätzlichen Planeten durch Zufall während der Beobachtung eines Transits von HAT-P-32b beobachtet wurde. Eine Betrachtung der Residuen der Transitlichtkurven (siehe Abbildung 4.2) zeigt, dass während der Transitbeobachtung keine deutliche Struktur eines weiteren Transitereignisses erkennbar ist. Abgesehen von einer mehr oder weniger gleichmäßigen Streuung während der Beobachtung ist keine systematische Abweichung sichtbar. Für eine mathematische Betrachtung der Beobachtungsabdeckung können die Residuen aller Lichtkurven mit Periodenverhältnissen von $P_c/P_b = [0,1 \dots 1]$ gefaltet und die Abdeckung der Phase bestimmt werden. Dazu wurde

ein von Manfred Kitzé zur Verfügung gestelltes Programm genutzt. Für äußere Planeten mit $P_c/P_b > 1$ sinkt die Abdeckung stark ab, so dass hier keine verwertbaren Abschätzungen möglich sind. Für die meisten Periodenverhältnisse im betrachteten Intervall zeigt sich eine Phasenabdeckung größer 90% (Abbildung 4.7). Innerhalb der Resonanzen sinkt die Phasenabdeckung erwartungsgemäß auf Werte $\gtrsim 55\%$ ab.

Wie in Tabelle 4.4 ersichtlich, weisen fast alle Transits eine Genauigkeit zwischen 0,9 mmag und 2,6 mmag auf. Lediglich drei Transitereignisse können nur mit einer photometrischen Genauigkeit von jeweils 3,3 mmag, 3,5 mmag und 4,5 mmag an das Modell angepasst werden. Davon ausgehend, dass alle transitartigen Strukturen innerhalb der Lichtkurven gesehen werden können, die eine Tiefe von mindestens $\Delta m = 3$ mmag aufweisen, kann mithilfe von Gleichung (1.2) unter der Annahme, dass der zusätzliche Planet selbst kein Licht emittiert, der maximale Radius des Planeten über

$$\begin{aligned} R_c &= R_S \cdot \sqrt{1 - 10^{-\Delta m/2,5}} \\ &= 1,219 \cdot 9,7203 \cdot R_{\gamma_+} \cdot \sqrt{1 - 10^{-3/2,5 \text{ mmag}}} \\ &\approx 0,62 R_{\gamma_+} \end{aligned} \tag{4.14}$$

mit $R_S = 1,219 R_{\odot}$ (Hartman u. a., 2011b) berechnet werden. Somit ist als grobe Näherung die Existenz eines inneren Planeten mit einem Radius $R_c \gtrsim 0,6 R_{\gamma_+}$ ausgeschlossen. Unter der Annahme einer ähnlich geringe Dichte wie die von HAT-P-32b ($\rho_b = 0,19 \text{ g cm}^{-3}$), ergäbe sich eine Mindestmasse von

$$m_c = 4/3\pi \cdot R_c^3 \cdot \rho_b = 0,035 M_{\gamma_+}. \tag{4.15}$$

Abhängig von der tatsächlich vorliegenden Dichte kann dieser Wert natürlich um bis zu einer Größenordnung variieren. So ergäbe sich zum Beispiel für die Dichte von Jupiter von $\rho_{\gamma_+} = 1,33 \text{ g cm}^{-3}$ von $m_c = 0,21 M_{\gamma_+}$. Der angegebene Wert stellt daher keine strikte Grenze dar, sondern soll vielmehr eine Abschätzung darstellen.

4.5.3 Zusammenfassung der Systemeinschränkungen

In Anbetracht dieser Einschränkungen ergibt sich für mögliche Begleiter der in Abbildung 4.8 dargestellte Parameterraum. Für die Beschränkung auf nicht-exzentrische, koplanare Bahnen können alle Konfigurationen unterhalb der durchgezogenen Linie aufgrund der Streuung im O–C Diagramm nicht ausgeschlossen werden. Besonders nahe der Resonanzen ergibt diese Einschränkung eine deutlich strengere Restriktion als die

Begrenzung durch die vorhandene Schwankung in der Radialgeschwindigkeit, welche als gepunktete Linie dargestellt ist und ebenfalls alle Konfigurationen oberhalb der Linie ausschließen lässt. Die hohe Abdeckung von Beobachtungsphasen $P_c/P_b < 1$ und das Fehlen von transitartigen Strukturen in den Residuen der Lichtkurven zeigt außerdem, Planeten oberhalb $M_c \gtrsim 11 M_\oplus$ unwahrscheinlich sind, wobei dies von der angenommenen Dichte eines hypothetischen Planeten abhängt. Die gestrichelte Markierung in Abbildung 4.8, die diese Grenze zeigt, stellt daher kein festes Limit dar.

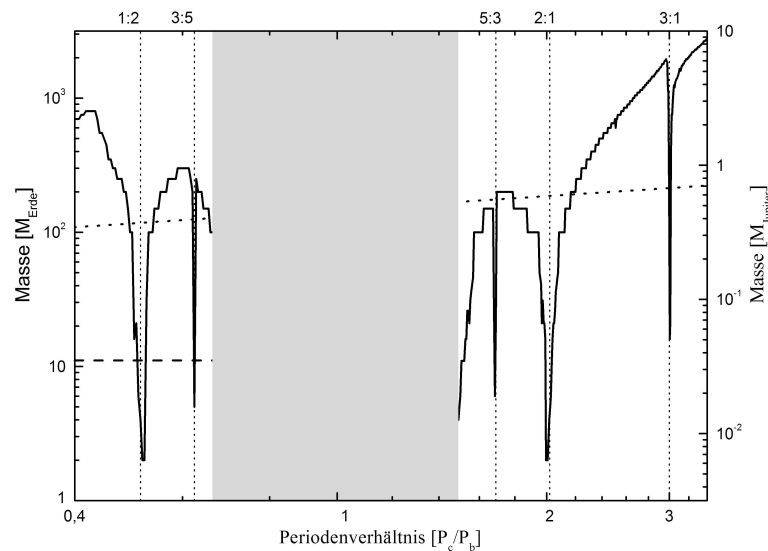


Abbildung 4.8: Die für HAT-P-32c verbleibenden Konfigurationen unter Einbeziehung der Ergebnisse aus der Simulation verschiedener Systemkonfigurationen (durchgezogene Linie), der Streuung innerhalb der Radialgeschwindigkeitsmessung (gepunktete Linie), sowie der Abschätzung der zufälligen Beobachtung eines Transitereignisses bei der Beobachtung von HAT-P-32b (gestrichelte Linie). Alle Systemkonfigurationen oberhalb der jeweils restriktivsten Kurve können für das HAT-P-32 System ausgeschlossen werden.

Bezüglich der durchgeführten Stabilitätsanalyse (siehe Kapitel 3.2.2) zeigte sich, dass alle der noch möglichen Begleiterszenarien über den simulierten Zeitraum von 600.000 Jahren stabil sind. Im Speziellen konnte festgestellt werden, dass die simulierten Systeme entweder in relativ kurzer Zeit instabil wurden, oder über die gesamte Simulationsdauer stabil waren. Abbildung 4.9 zeigt das Ergebnis der Analyse grafisch. Wie bereits erwähnt, sind die Resonanzen besonders interessant, da hier bereits kleine Massen zu starken Interaktionen führen. Entsprechend zeigt sich eine geringere Systemstabilität in deren Nähe.

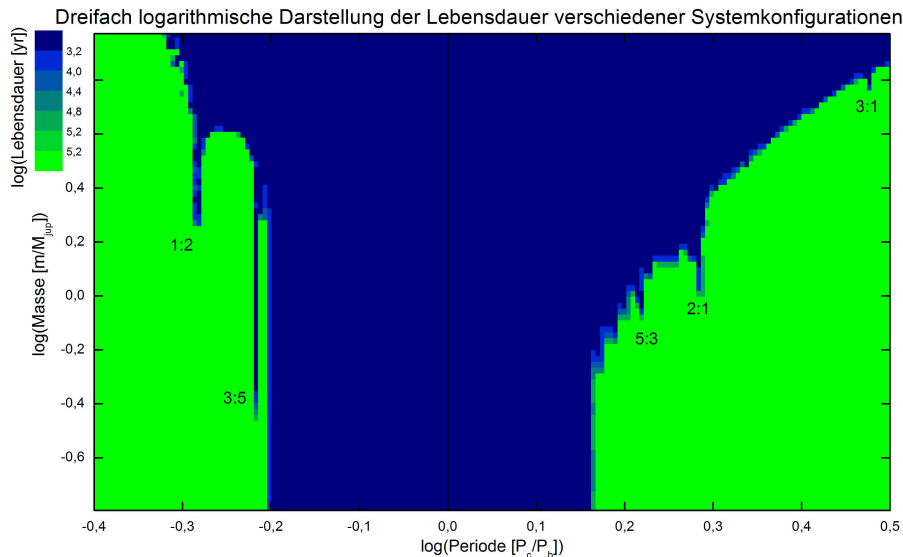


Abbildung 4.9: Ergebnis der Stabilitätsanalyse von HAT-P-32b. Besonders innerhalb der Resonanzen führen selbst geringe Massen zu großen Wechselwirkungen und damit Instabilitäten im System. Alle nach den vorherigen Einschränkungen verbleibenden Konfigurationen sind über den simulierten Zeitraum von 600.000 Jahren stabil.

4.6 Neubestimmung der Systemparameter

Neben dem genauen Zeitpunkt eines Transitereignisses wurden für jeden Transit auch die restlichen Parameter (Inklination i , Radienverhältnis R_b/R_s und normierter inverser stellarer Radius a/R_s) bestimmt. Von den 26 eigenen Beobachtungen konnten die vier Beobachtungen mit dem Tenerife 1,2m Teleskop nicht zur Bestimmung der Systemparameter genutzt werden.⁽¹⁰⁾

Zusätzlich zu den verbleibenden 22 Modellierungsergebnissen konnte das Ergebnis der Modellierung eines von Sada u. a. (2012) beobachteten Transitereignisses verwendet werden, sowie das ursprünglich von Hartman u. a. (2011b) publizierte Resultat. Gibson u. a. (2013) geben statt des Ergebnisses einer Einzelmodellierung der Transitparameter lediglich einen Wert an, der sich aus beiden beobachteten Transitereignissen ergibt. Dieses globale Ergebnis wurde dennoch verwendet. Als Zeitpunkt wurde Epoche 827 gewählt, welche sich zwischen den beiden Beobachtungsepochen 817 und 837 befindet. Die damit 25 Einzelergebnisse für die drei Parameter wurden jeweils über der Beobachtungsepoch

⁽¹⁰⁾Die Transitbeobachtungen mit dem Tenerife 1,2m Teleskop wurden zur Bestimmung des Transitzeitpunktes zur Verfügung gestellt. Die Auswertung der anderen Systemparametern, insbesondere des Radienverhältnisses wird gesondert mit Blick auf die Farbabhängigkeit von Bernt u. a. (in Vorbereitung) betrachtet. Erste Ergebnisse finden sich in Bernt u. a. (2013).

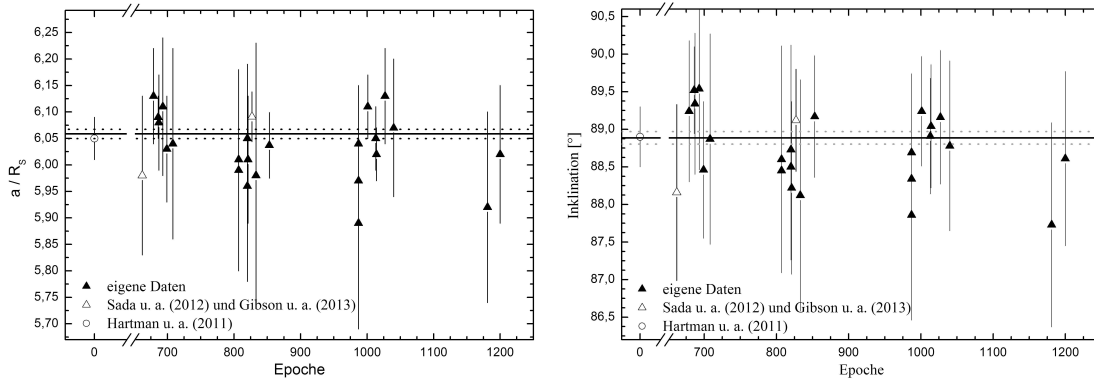


Abbildung 4.10: Das Ergebnis der Analyse des Verhältnisses von großer Halbachse und stellarem Radius a/R_S und der Inklination i . Beide neu bestimmten Parameter stimmen mit $a/R_S = (6,0590 \pm 0,0088)$ und $i = (88,89 \pm 0,09)^\circ$ mit dem Ergebnis früherer Studien überein.

dargestellt. Dafür wurde erneut das Programm ORIGIN verwendet. Für alle Systemparameter wurde zunächst angenommen, dass sie innerhalb des Beobachtungszeitraumes konstant sind. Daher konnte analog zur Vorgehensweise bei der Neubestimmung der Periode die in ORIGIN verfügbare Methode der linearen Regression mit einer festen Steigung von Null verwendet werden. Als Wichtung wurde auch hier eine instrumentelle Wichtung des Modellierungsfehlers genutzt.

Abbildung 4.10 zeigt grafisch die Ergebnisse der Neubestimmung des Verhältnisses von großer Halbachse und stellarem Radius, sowie der Inklination des Systems. Gut erkennbar ist, dass außer einer Streuung der erhaltenen Werte kein Trend in den Daten sichtbar ist. Die erhaltenen Ergebnisse stimmen mit $a/R_S = (6,0590 \pm 0,0088)$ und $i = (88,89 \pm 0,09)^\circ$ mit den bereits in der Literatur bestimmten Werten überein. Tabelle 4.6 listet vergleichend die drei Literaturwerte, sowie die in dieser Arbeit ermittelten Parameter auf. Aufgrund der Anzahl der Datenpunkte ist es dabei möglich, beide Parameter um bis zu eine Größenordnung genauer zu bestimmen als bisher. In Kapitel 3.1.3 wurde bereits erläutert, dass als Fehlerbalken der Einzeldatenpunkte jeweils der Größtfehler der Modelle verwendet wurde, um eine konservative Abschätzung zu erhalten. Die im Vergleich zu den Fehlern geringe Streuung in beiden Diagrammen spiegelt dieses Vorgehen wider. Die Anpassung des konstanten Modells an die Daten liefert mit einem $\chi_{\text{red}}^2 [a/R_S] = 0,32$ bzw. $\chi_{\text{red}}^2 [i] = 0,25$ auch den quantitativen Nachweis dafür. Die Durchführung eines χ^2 -Tests basiert auf der Grundlage, dass eine gute Modellanpassung das reduzierte χ^2 , d.h. ein auf die Anzahl der Datenpunkte abzüglich der Anzahl der Modellparameter normierter Wert minimiert wird und im besten Fall einen Wert von $\chi_{\text{red}}^2 \approx 1$

Tabelle 4.6: Ein Vergleich aller bisheriger Ergebnisse für die Systemparameter i , a/R_s und k mit den Ergebnissen dieser Studie. Das ermittelte Radienverhältnis gibt hierbei den bereits nach Gleichung (4.19) korrigierten Wert an.

	a/R_s	$k = R_p/R_s$	$i [^\circ]$
diese Analyse	$6,0590 \pm 0,0088$	$0,1512 \pm 0,0011$	$88,89 \pm 0,09$
Hartman u. a. (2011b)	$6,05^{+0,03}_{-0,04}$	$0,1508 \pm 0,0004$	$88,9 \pm 0,4$
Sada u. a. (2012)	$5,98^{+0,15}_{-0,10}$	$0,1531 \pm 0,0012$	$86,16^{+1,03}_{-1,17}$
Gibson u. a. (2013)	$6,091^{+0,047}_{-0,036}$	$0,1515 \pm 0,0012$	$89,12^{+0,68}_{-0,61}$

erreicht wird. Für den Fall, dass $\chi_{\text{red}}^2 < 1$ ist, kann dies zum einen als Indiz für eine Überinterpretation der Daten gewertet werden. Da die Modellanpassung mit einer konstanten Funktion und damit nur einem freien Parameter erfolgte, führt die Wahl zum Beispiel einer sinusförmigen Variation zu einer noch stärkeren Überinterpretation und ist daher abzulehnen. Zum anderen ist möglich, dass die Abschätzung der Einzelfehler zu konservativ ist. Betrachtet man die Größe der Einzelfehlerbalken im Vergleich zu den Ergebnissen der Modellierung zum Beispiel von Maciejewski u. a. (2011b), in welchem teilweise mit den gleichen Teleskopen ein ähnliches Objekt (WASP-12b) beobachtet wurde, so sind die hier erhaltenen Einzelfehler durchaus realistisch, so dass eine generelle Reskalierung nicht sinnvoll ist.

Die Genauigkeit der Bestimmung des Transitzeitpunktes und der Transiddauer, ist maßgeblich durch eine präzise Anpassung der Flanken bestimmt. Im Gegensatz dazu wird die Transittiefe, und damit das Radienverhältnis, sowohl von der photometrischen Streuung innerhalb des Transits, als auch während der Normallichtphase geprägt und ist daher wechselnden Beobachtungsbedingungen stärker ausgesetzt. Auch andere Artefakte in der Lichtkurve wirken sich stärker aus. Ein Beispiel hierfür ist die Beobachtung mit dem Trebur 1,2m Teleskop in Epoche 833, wo der in der Lichtkurve deutlich sichtbare Sprung die Modellanpassung erschwert. Für das Radienverhältnisses ergibt sich daher im Vergleich zu den beiden zuvor betrachteten Systemparametern ein anderes Bild. Zwar stimmt auch hier der neu bestimmte Wert $k = (0,15096 \pm 0,00041)$ mit den Literaturwerten überein, allerdings zeigt sich in Abbildung 4.11 eine wesentlich größere Streuung in den Ergebnissen.

Ein zusätzlicher Effekt, welcher die Bestimmung des Radienverhältnisses erschwert, ist die Kontamination mit sogenanntem Drittlcht. Befindet sich neben dem eigentlich beobachteten Stern ein weiterer Stern innerhalb der Apertur, so muss dieser in der Berechnung der Transittiefe berücksichtigt werden, da er sowohl in der Normallichtphase, als

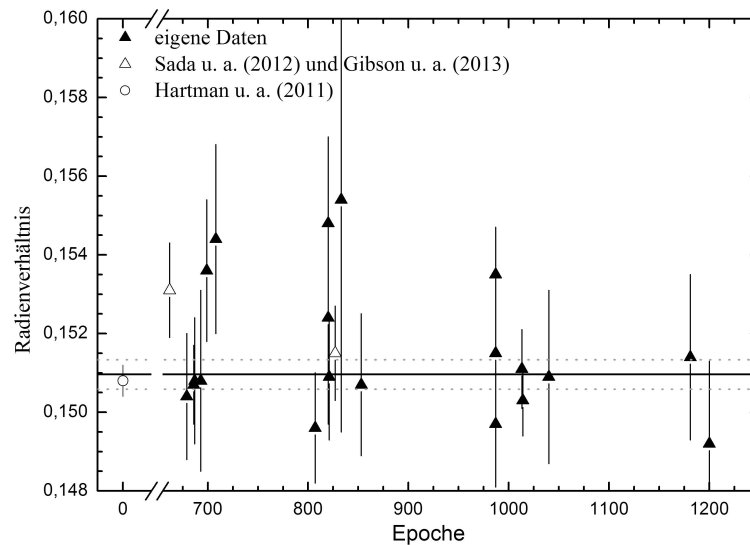


Abbildung 4.11: Die für die einzelnen Transitereignisse ermittelten Radienverhältnisse. Das Ergebnis der Anpassung einer Konstanten an die Datenpunkte ergibt $k = (0,15096 \pm 0,00041)$. Die Ergebnisse von Hartman u. a. (2011b) und – mit einer Abweichung von weniger als 2σ – auch die Ergebnisse von Sada u. a. (2012) sind mit diesem Resultat vereinbar.

auch während des Transits zur gemessenen Helligkeit des Sterns beiträgt. Für HAT-P-32 konnte ein solches Objekt unabhängig voneinander von Adams u. a. (2013) und Gibson u. a. (2013) gefunden und durch weitere Analysen von Zhao u. a. (2014) genauer charakterisiert werden. Letztere fanden heraus, dass es sich bei dem Objekt im Abstand von $d = (2,923 \pm 0,004)''$ von HAT-P-32 um einen frühen M-Stern handelt⁽¹⁾. Zusätzlich verweisen sie auf eine Veröffentlichung in Vorbereitung, in welcher mittels Eigenbewegungsanalyse nachgewiesen werden kann, dass es sich um ein gebundenes System handelt. Da sich ein so enger Begleiter stets innerhalb der zur Lichtkurvenerstellung genutzten Apertur befindet, muss dessen Einfluss auf die bestimmten Parameter kontrolliert werden. Im Besonderen trifft dies für die Bestimmung des Radienverhältnisses zu. Ohne einen zusätzlichen nahen Begleiter ergibt sich die Tiefe des Transitereignisses Δm aus dem Fluss des

⁽¹⁾Zhao u. a. (2014) konnten die Effektivtemperatur auf $T_{\text{eff}} = (3565 \pm 82)$ K (dies entspricht einem Spektraltyp von etwa M1,5) bestimmen, indem sie HAT-P-32 und den mutmaßlichen stellaren Begleiter mithilfe von adaptiver Optik aufgelöst beobachtet und die Flussverhältnisse zwischen beiden Objekten in verschiedenen Bändern bestimmt haben.

Sterns F_S und dem Fluss der vom Planeten verdeckten Sternfläche F_p mittels

$$\begin{aligned}\Delta m &= -2,5 \log \left(\frac{F_S - F_p}{F_S} \right) \\ &= -2,5 \log \left[1 - \left(\frac{R_p}{R_S} \right)^2 \right].\end{aligned}\quad (4.16)$$

Unter Berücksichtigung eines zusätzlichen Objektes mit dem Fluss F_B ändert sich dies zu

$$\begin{aligned}\Delta m &= -2,5 \log \left(\frac{F_S - F_p + F_B}{F_S + F_B} \right) \\ &= -2,5 \log \left[1 - \frac{F_p}{F_S + F_B} \right] \\ &= -2,5 \log \left[1 - \frac{F_p}{F_S} \frac{1}{1 + F_B/F_S} \right].\end{aligned}\quad (4.17)$$

Im Vergleich der Gleichungen (4.16) und (4.17) kann daher der Unterschied in den Radien zu

$$\left. \frac{R_p}{R_S} \right|_{\text{kor}} = \sqrt{1 + F_B/F_S} \cdot \left. \frac{R_p}{R_S} \right|_{\text{unkorr}} \quad (4.18)$$

bestimmt werden. Zhao u. a. (2014) geben für das Flussverhältnis im optischen Bereich einen Wert von $F_B/F_A < 0,012$ an.⁽¹²⁾ Da die den Werten zu Grunde liegenden Beobachtungen mit Filtern im SDSS-System (*engl.* Sloan Digital Sky Survey) gemacht wurden, im Speziellen g' , r' , i' und z' , sind die erhaltenen Werte für die in dieser Studie verwendeten Beobachtungen im Cousins- bzw. Besselsystem nicht exakt übertragbar. Die zentralen Wellenlängen von $\lambda_c^{r'} = 625,0$ nm (Fukugita u. a., 1996) und $\lambda_c^R = 647,0$ nm (Cousins, 1976) sind hinreichend nah beieinander, um die Flussverhältnisse vergleichen zu können. Mit $(F_B/F_A)_{r'} = 0,003 \pm 0,001$ ergibt sich daher ein Korrekturfaktor von

$$\sqrt{1 + F_B/F_S} = 1,0015 \pm 0,0050$$

und damit für das Radienverhältnis der leicht nach oben korrigierte Wert von

$$R_p/R_S = 0,1512 \pm 0,0011. \quad (4.19)$$

⁽¹²⁾Der Wert $F_B/F_A = 0,012$ entspricht dem SDSS- z' -Filter mit der zentralen Wellenlänge $\lambda = 913,4$ nm.

5 Weitere Ergebnisse

Neben HAT-P-32b wurden die gezeigten Analysen bezüglich möglicher Transitzeitvariationen, sowie die Bestimmung der Transit- und Orbitparameter auch für die weiteren vier Objekte HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b durchgeführt. Aufgrund der Anzahl der erhaltenen Beobachtungen ist eine detaillierte Betrachtung hypothetischer Begleiterszenarien hier allerdings nicht möglich. Die 16 Transitereignisse wurden dennoch analog der Vorgehensweise für HAT-P-32b ausgewertet und die Systemparameter neu bestimmt.

Dabei konnte zusätzlich auf verschiedene Literaturdaten zur Vergrößerung der Datengrundlage zurückgegriffen werden:

- HAT-P-18b: Esposito u. a. (2014) machten Radialgeschwindigkeitsmessungen und beobachteten einen Transit von HAT-P-18b. Sie veröffentlichten ihre Ergebnisse der Bestimmung der Systemparameter a/R_S , i und R_p/R_S .
- HAT-P-27b/WASP-40b: Neben den Daten der beiden Entdeckungspaper von Béky u. a. (2011) und Anderson u. a. (2011) haben auch Sada u. a. (2012) und Brown u. a. (2012) den Transitplaneten untersucht, so dass zusätzlich jeweils ein Wert für die Systemparameter a/R_S , i und R_p/R_S verfügbar waren. Zudem gibt Sada u. a. (2012) auch den Zeitpunkt des beobachteten Transits an.
- WASP-21b: Für WASP-21b konnten zusätzliche Transitmittelpunkte von Barros u. a. (2011b) und Ciceri u. a. (2013), sowie die Parameter a/R_S , i und R_p/R_S aus den beiden genannten Studien und Sada u. a. (2012) verwendet werden.

Bei der Betrachtung der erhaltenen Lichtkurven in Abbildung 5.1 und 5.2 fällt auf, dass die für die Auswertung berücksichtigten Daten stärker verrauscht sind, als die für HAT-P-32b verwendeten Daten. Dies liegt unter anderem an geringeren Transittiefen der bekannten Planeten und geringeren Helligkeiten der Muttersterne. Aufgrund der geringeren Anzahl an erfolgreichen Beobachtungen wurde hier bewusst auch auf Daten geringerer Qualität, im Speziellen mit einem $pnr < 5$, zurückgegriffen.

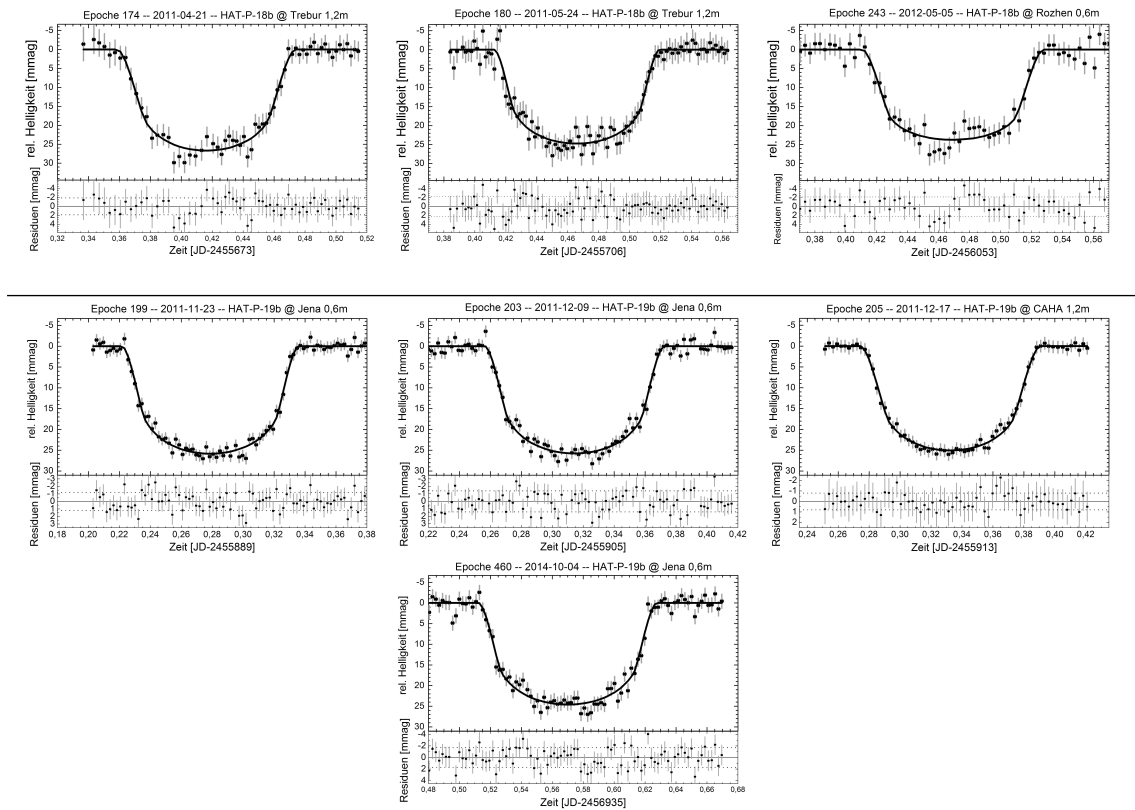


Abbildung 5.1: Die erhaltenen Lichtkurven von HAT-P-18b und HAT-P-19b.

Im Fall von HAT-P-18b zeigen alle erhaltenen Lichtkurven eine vergleichsweise starke Variabilität. Diese kann auf stellare Aktivität, wie zum Beispiel Sternflecken, hindeuten. Verdeckt ein Transitplanet bei seinem Vorbeiflug am Stern einen Fleck, d.h. im Vergleich zum restlichen Stern eine Region geringerer Helligkeit, so ist dies durch eine Änderung der Helligkeit in der Lichtkurve sichtbar. Eine weitere mögliche Ursache liegt in der Generierung der Lichtkurve. Während für die anderen betrachteten Objekte zwischen 20 und 80 Vergleichssterne zur Erstellung des künstlichen Vergleichssterne (siehe Kapitel 3.1.2) genutzt werden konnten, beruht die Analyse im Falle der drei erfolgreichen Transitbeobachtungen von HAT-P-18b, aufgrund des kleinen Gesichtsfeldes und der Helligkeit der Sterne im Gesichtsfeld, auf weniger als 10 Vergleichssterne. Aus der daraus resultierenden stärkeren Wichtung eines Einzelsterns kann eine höhere Schwankung der Lichtkurve resultieren. Wie zu erwarten, zeigen auch die Lichtkurven der Vergleichssterne eine größere Streuung, welche die Hypothese einer künstlichen Variabilität bestätigt. Eine Überprüfung durch ein Teleskop mit größerem Gesichtsfeld und damit mehr potentiellen Vergleichssterne konnte nicht realisiert werden.

Die restlichen Lichtkurven zeigen keine nennenswerten Artefakte, welche auf astro-

physikalische Effekte zurückführbar sind⁽¹³⁾. Dies zeigt sich auch deutlich in den erhaltenen Ergebnissen, welche in keinem Orbitparameter signifikante Variationen zeigen. Alle Daten lassen sich, abgesehen von der im Vergleich zu HAT-P-32b höheren Streuung, mit einem konstanten Wert erklären, welcher mit dem in der jeweiligen Literatur bekannten Wert vergleichbar ist (siehe Tabelle 5.1).

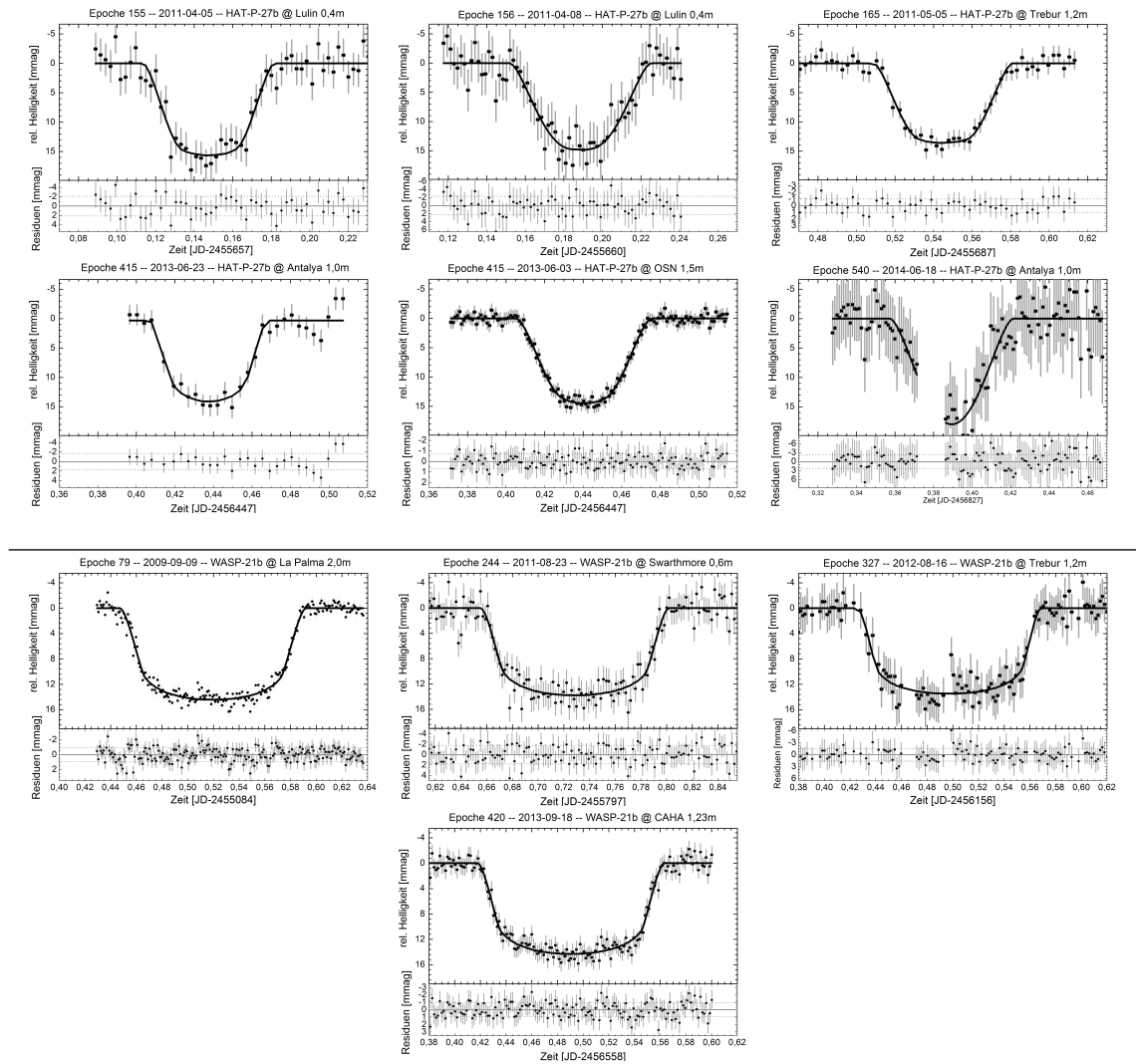


Abbildung 5.2: Die erhaltenen Lichtkurven von HAT-P-27b und WASP-21b.

Auffällig ist jedoch die Form der Transitereignisse von HAT-P-27b/WASP-40b. Bereits in den beiden Entdeckungsveröffentlichungen zeigen Béky u. a. (2011), dass die Transitform den erwarteten flachen Bereich (auch flacher Boden vom englischen Begriff “flat bottom”) zeigt, während Anderson u. a. (2011) einen eher runden Transit (runder Boden,

⁽¹³⁾Der Sprung in der Helligkeit der Beobachtung von WASP-21b mit dem Trebur 1,4m Teleskop am 16.8.2012 (Abbildung 5.2) wird von der Vignettierung des Teleskops durch die Kuppel hervorgerufen.

engl. “round bottom”) zeigt. Die im Rahmen dieser Arbeit erhaltenen Aufnahmen lassen diesbezüglich keine eindeutige Aussage zu, wenngleich der qualitativ hochwertigste Transit vom 3.6.2013, aufgenommen mit dem OSN 1,5m Teleskop, ebenfalls einen eher runden Boden besitzt. In den vorhandenen Literaturbeobachtungen von Sada u. a. (2012) zeigte sich vorrangig ein runder Transit, welcher den Autoren zufolge auf einen streifenförmigen Transit hindeutet. Dafür spricht auch die für Transitplaneten kleine Inklination von $i \approx 85^\circ$. Aus Sicht der hier gezeigten Transitbeobachtungen kann diese Hypothese nicht eindeutig bestätigt, wohl aber gestützt werden.

Wie schon bei den Beobachtungen von HAT-P-32b war es auch hier möglich, simultane Beobachtungen des gleichen Transits an verschiedenen Teleskopen durchzuführen (siehe Tabellen 2.2 und 2.3). Bei fast allen Simultanbeobachtungen ist dabei allerdings nur jeweils eine Beobachtung erfolgreich verlaufen. Die einzige nutzbare simultane Beobachtung eines Transits von HAT-P-27b am 3.6.2013 mit dem OSN 1,5m Teleskop und dem Antalya 1,0m Teleskop stellte sich dabei im Rahmen der Analyse als besonders wichtig heraus. Eine Betrachtung der Unterschiede zwischen beiden Modellanpassungen in Tabelle 5.2 zeigt auf, dass die Ergebnisse innerhalb des größeren der beiden Fehlerbalken übereinstimmen. Eine Ausnahme bildet der ermittelte Transitzeitpunkt. Hier weichen die Ergebnisse um 4,36 min von einander ab (bei individuellen Fehlerbalken der Einzelmodelle von 2,4 min, bzw. 0,4 min). Trägt man beide Werte in das entsprechende O–C Diagramm ein (Abbildung 5.3), ist ersichtlich, dass einer der beiden Werte mit der Erwartung übereinstimmt, während der andere stark abweicht. Bei der Suche nach der Ursache für diese starke Diskrepanz wurden zunächst die Lichtkurven genauer untersucht. Mit 1,2 mmag bzw. 2,6 mmag weisen beide Lichtkurven keine auffällig große Streuung auf. Auch in den jeweiligen Beobachtungsmitschriften wurden keine Vorkommnisse während der Beobachtungen aufgeführt und die Wetterbedingungen als gut eingestuft. Bei Rücksprachen mit lokal ansässige Kollegen konnte allerdings festgestellt werden, dass die Zeitsynchronisation an einem der beiden Teleskope mittels GPS in regelmäßigen Abständen durchgeführt wird. Jeweils kurz vor einer neuen Synchronisation ist der Zeitversatz allerdings sehr groß, die Synchronisationsabstände folglich zu lang. Die Größe des Zeitversatzes ist dabei mit der festgestellten Verschiebung des Transitmittelpunktes konsistent. Mithilfe unserer simultanen Transitbeobachtung konnte dieses Problem gefunden und anschließend durch eine Verkürzung des Synchronisationsintervalls behoben werden.

Dieses Beispiel verdeutlicht einerseits noch einmal den Nutzen simultaner Beobachtungen, zum anderen zeigt es auf, wie wichtig eine Synchronisation der Teleskopzeit mit der koordinierten Weltzeit ist. Ohne eine korrekte Zeiterfassung kann mit nur einer Beob-

Tabelle 5.1: Ein Vergleich zwischen den Ergebnissen dieser Analyse mit den vorhandenen Literaturdaten. Alle Null-Epochen T_0 wurden in die Zeitbasis BJD_{TDB} konvertiert. Eine Bestimmung der Nullepoche wurde von Brown u. a. (2012) nicht durchgeführt.

	T_0 [d]	P [d]	a/R_S	$k = R_p/R_S$	i [°]
HAT-P-18b					
diese Analyse	2 454 715,022 54 ± 0,000 39	5,508 029 1 ± 0,000 004 2	17,09 ± 0,71	0,1362 ± 0,001 1	88,79 ± 0,21
Hartman u. a. (2011a)	2 454 715,022 51 ± 0,000 20	5,508 023 ± 0,000 006	16,04 ± 0,75	0,1365 ± 0,001 5	88,3 ± 0,3
Esposito u. a. (2014)	2 455 706,7 ± 0,7	5,507 978 ± 0,000 043	16,76 ± 0,82	0,136 ± 0,011	88,79 ± 0,25
HAT-P-19b					
diese Analyse	2 455 091,535 00 ± 0,000 15	4,008 784 2 ± 0,000 000 7	12,36 ± 0,09	0,1378 ± 0,001 4	88,51 ± 0,22
Hartman u. a. (2011a)	2 455 091,534 94 ± 0,000 34	4,008 778 ± 0,000 006	12,24 ± 0,67	0,1418 ± 0,002 0	88,2 ± 0,4
HAT-P-27b					
diese Analyse	2 455 186,019 91 ± 0,000 44	3,039 580 3 ± 0,000 001 5	10,01 ± 0,13	0,1192 ± 0,001 5	85,08 ± 0,07
Béky u. a. (2011)	2 455 186,019 55 ± 0,000 54	3,039 486 ± 0,000 012	9,65 ^{+0,54} _{-0,40}	0,1186 ± 0,003 1	84,7 ^{+0,7} _{-0,4}
Anderson u. a. (2011)	2 455 368,394 76 ± 0,000 18	3,039 572 1 ± 0,000 007 8	9,88 ± 0,39	0,1250 ± 0,001 5	84,98 ^{+0,20} _{-0,14}
Sada u. a. (2012)	2 455 186,198 22 ± 0,000 32	3,039 582 4 ± 0,000 003 5	9,11 ^{+0,71} _{-1,01}	0,1344 ^{+0,017 4} _{-0,038 9}	84,23 ± 0,88
Brown u. a. (2012)	–	3,039 577 ± 0,000 006	9,80 ^{+0,38} _{-0,29}	0,120 ^{+0,009} _{-0,007}	85,0 ± 0,2
WASP-21b					
diese Analyse	2 454 743,042 17 ± 0,000 65	4,322 512 6 ± 0,000 002 2	9,62 ± 0,17	0,1030 ± 0,000 8	87,12 ± 0,24
Bouchy u. a. (2010)	2 454 743,042 6 ± 0,002 2	4,322 482 ^{+0,000 024} _{-0,000 019}	10,56 ± 0,49	0,1040 ^{+0,001 7} _{-0,001 8}	88,75 ^{+0,70} _{-0,84}
Barros u. a. (2011b)	2 455 084,520 48 ± 0,000 20	4,322 506 0 ± 0,000 003 1	9,68 ^{+0,30} _{-0,19}	0,107 1 ^{+0,000 9} _{-0,000 8}	87,34 ± 0,29
Ciceri u. a. (2013)	2 454 743,040 54 ± 0,000 71	4,322 518 6 ± 0,000 003 0	9,46 ± 0,27	0,1055 ± 0,002 3	86,97 ± 0,33
Southworth (2012)	2 455 084,520 40 ± 0,000 16	4,322 506 0 ± 0,000 003 1	9,35 ± 0,34	0,1095 ± 0,001 3	86,77 ± 0,45

achtung je Transit ein echtes nicht von einem künstlich induzierten Signal unterschieden werden. Eine rückwirkende Korrektur ist in den meisten Fällen, auch im Fall der hier geschilderten Beobachtungen, nicht möglich.⁽¹⁴⁾

Trägt man die erhaltenen Transitzeitpunkte in die entsprechenden O–C Diagramme (Abbildung 5.3) ein, kann in allen Fällen eine Neubestimmung der Periode die meisten Datenpunkte erklären. Neben dem bereits erwähnten abweichenden Datenpunkt aufgrund von Synchronisationsfehlern, sowie einer generellen stärkeren Streuung kann allerdings keine signifikante Variabilität festgestellt werden. Innerhalb ihrer Fehlerbalken sind fast alle Datenpunkte mit einer konstanten Periode erklärbar.

Tabelle 5.2: Vergleich der Beobachtung des Transitereignisses des extrasolaren Planeten HAT-P-27b am 3.6.2013 mit den Teleskopen OSN 1,5m und Antalya 1,0m.

Teleskop	$T_{\text{mid}} [\text{d}] - 2\,456\,447$	a/R_s	$k = R_p/R_s$	$i [^\circ]$
Antalya 1,0m	$0,442\,68 \pm 0,001\,66$	$10,64 \pm 1,30$	$0,118\,4 \pm 0,008\,1$	$85,51 \pm 0,94$
OSN 1,5m	$0,445\,71 \pm 0,000\,30$	$10,18 \pm 0,29$	$0,122\,4 \pm 0,003\,7$	$85,23 \pm 0,21$
Unterschiede	0,003 03	0,46	0,0040	0,28

Auch bezüglich der ermittelten Parameter i , a/R_s und k können die in der Literatur angegebenen Werte bestätigt werden (Abbildungen 5.4, 5.5 und 5.6). Im Fall von WASP-21b ist das ermittelte Verhältnis von Halbachse zu stellarem Radius a/R_s geringer als der ursprünglich von Bouchy u. a. (2010) gefundene Wert. Dies deckt sich mit den Ergebnissen späterer Studien von Barros u. a. (2011b), Southworth (2012) und Ciceri u. a. (2013), mit welchen das hier erhaltene Resultat übereinstimmt.

⁽¹⁴⁾Alle Transitereignisse des betreffenden Teleskops wurden anschließend noch einmal auf mögliche Synchronisationsfehler hin untersucht.

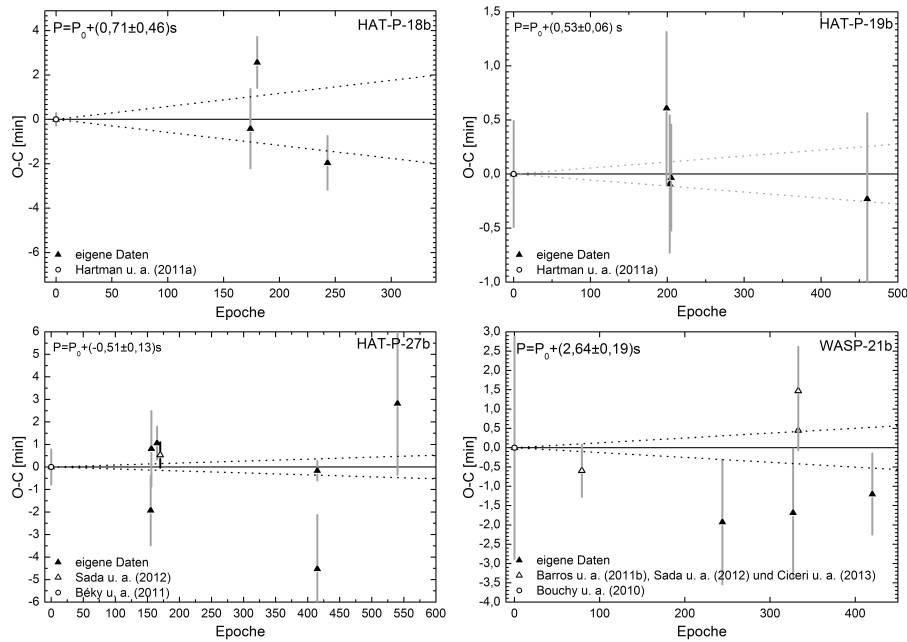


Abbildung 5.3: Die O-C Diagramme für HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b. In allen Fällen zeigt sich, dass eine geringfügige Neubestimmung der Periode die gemessenen Transitzeitpunkte gut erklären kann. Die große Abweichung eines Datenpunktes im O-C Diagramm von HAT-P-27b ist auf eine fehlerhafte Zeitsynchronisation zurückzuführen.

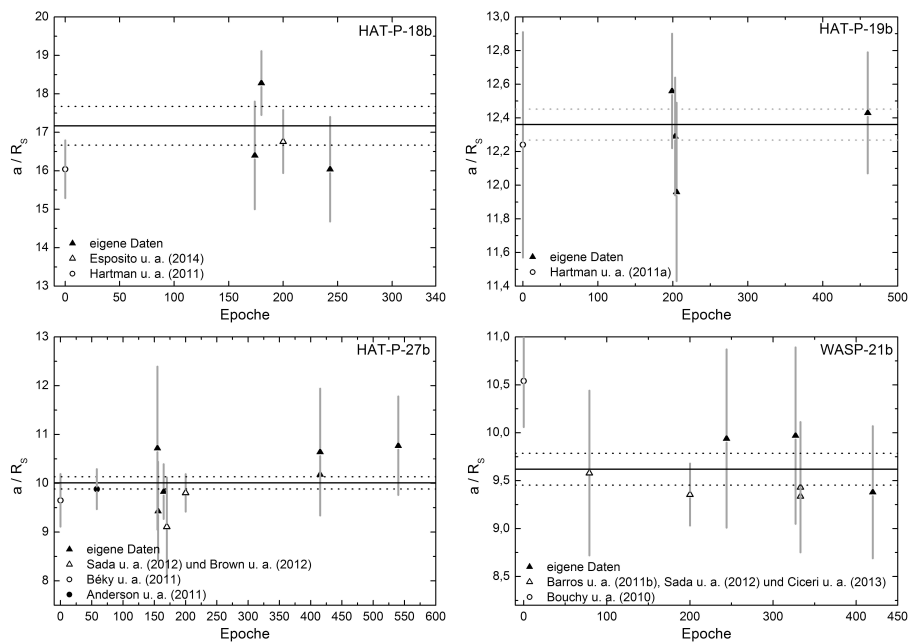


Abbildung 5.4: Die Verhältnisse a/R_S für HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b zeigen keine Variabilität.

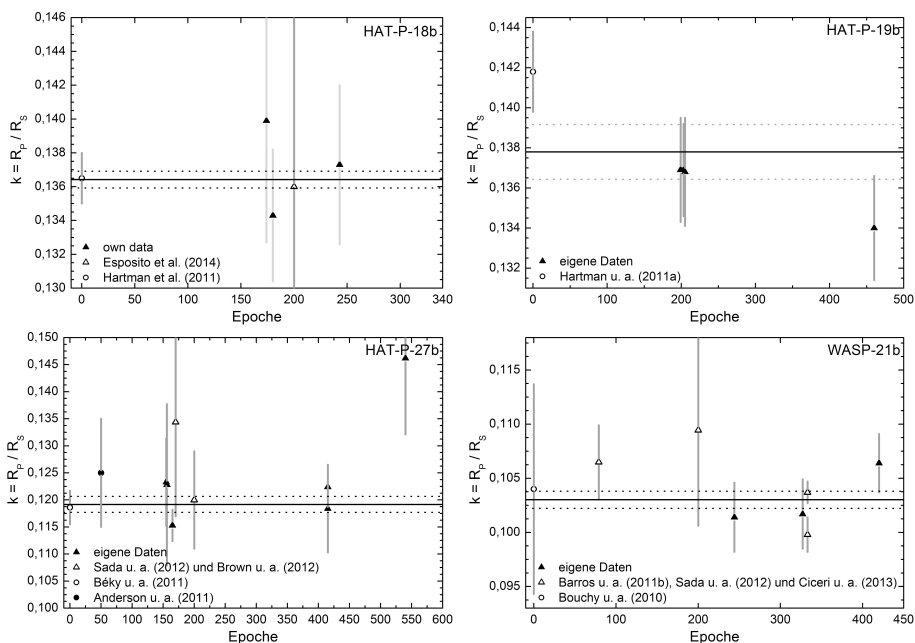


Abbildung 5.5: Das Radienverhältnis als Maß für die Transittiefe für HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b.

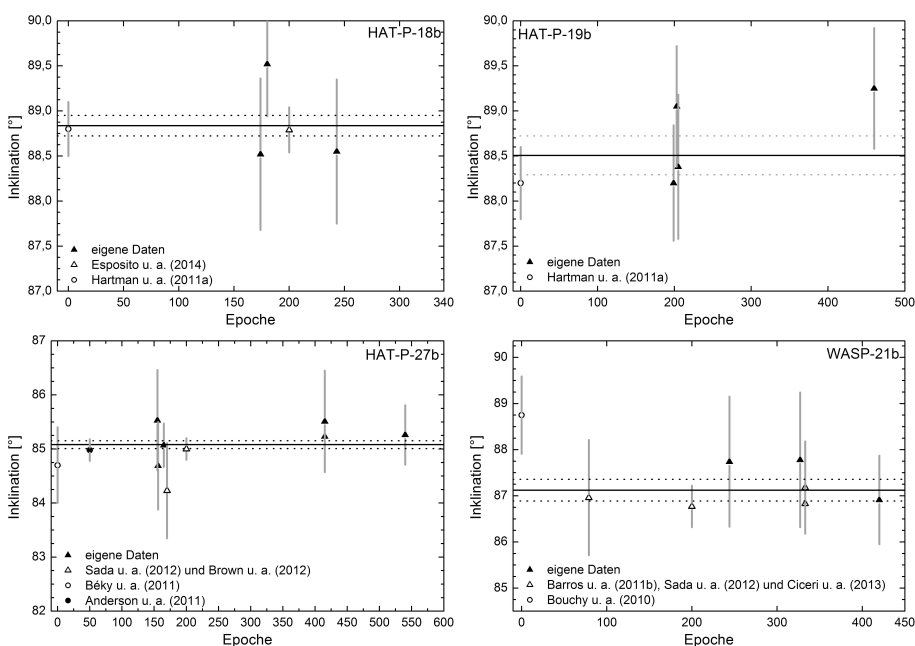


Abbildung 5.6: Das grafische Ergebnis für die Analyse der Inklination für HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b.

6 Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurden bekannte Transitplaneten untersucht, um durch eine präzise Messung des Transitzeitpunktes Indikatoren für die Existenz zusätzlicher Objekte im System zu erhalten. Dazu fiel die Auswahl auf sieben bekannte Transitplaneten, die aufgrund ihrer Eigenschaften zusätzliche Objekte vermuten ließen.

Unter Zuhilfenahme des YETI-Netzwerkes konnten über einen Zeitraum von vier Jahren 93 Transitbeobachtungen mit einer Gesamtbelichtungszeit von 291 Stunden an 17 verschiedenen Beobachtungsstandorten weltweit durchgeführt werden. Aufgrund verschiedener Einflüsse, maßgeblich wechselnder Wetterbedingungen, verblieben dabei nur 41 Transitbeobachtungen zur weiteren Analyse. Die restlichen Beobachtungen mussten entweder bereits während des Transitereignisses abgebrochen, oder nachträglich aufgrund zu hoher photometrischer Fehler aussortiert werden. Zuzüglich zu den eigenen Daten wurden auch publizierte Resultate anderer Forschungsgruppen verwendet.

Mit einer Zahl von 26 konnten dabei für den extrasolaren Planeten HAT-P-32b die meisten Beobachtungsdaten ausgewertet werden. Neben der Analyse des Transitmittelpunktes und der damit einhergehenden Neubestimmung der Umlaufperiode umfasste dies die Bestimmung der Inklination i , des Verhältnisses aus großer Halbachse und stellarem Radius a/R_S , sowie des Radienverhältnisses R_p/R_S . In keinem der Parameter wurde dabei eine signifikante Änderung im Vergleich zu den in der Literatur verfügbaren Werten gefunden.

Das erstellte O–C Diagramm für das HAT-P-32 System zeigt neben einer gewissen Streuung weder einen Trend, noch eine periodische Variabilität. Aufgrund der Vielzahl der Daten war es dennoch möglich, zu untersuchen, welche Objekte Effekte hervorrufen würden, die innerhalb der Streuung nicht entdeckt werden können. Dazu wurde anhand der gemessenen Radialgeschwindigkeitsschwankungen eine Abschätzung über einen zusätzlichen Begleiter getroffen. Zusätzlich wurde die Frage gestellt, ob ein Objekt zufällig einen Transit während einer Transitbeobachtung von HAT-P-32b durchgeführt hat. Zuletzt wurde eine Simulation von insgesamt 127.385 verschiedenen Systemkonfigurationen durchgeführt, um die zu erwartenden TTV-Amplituden zu ermitteln. Diese ergeben wiederum eine Einschränkung möglicher Begleiter im HAT-P-32 System, welche in Ab-

bildung 4.8 zusammengefasst sind. Besonders innerhalb der Resonanzen können dabei Planeten mit einer Masse von wenigen Erdmassen ausgeschlossen werden.

Nach den Ergebnisse anderer Studien zu HAT-P-32b, insbesondere den Resultaten von Knutson u. a. (2014), ist davon auszugehen, dass sich ein weiteres, noch unentdecktes Objekt im System befindet. Aufgrund der Radialgeschwindigkeitsanalyse ist dies ein massereiches Objekt auf einer äußeren Bahn, welches nach der aktuellen Beobachtungslage nicht durch Transitzeitanalysen bestätigt oder widerlegt werden kann. Die Auswirkungen eines solchen Objektes auf Transitzeitvariationen von HAT-P-32b sind im Rahmen aktueller bodengebundener Beobachtungen nicht signifikant, so dass eine Bestätigung eines solchen Objektes durch eine direkte Transitbeobachtung im Falle einer Inklination nahe 90° oder genauere Radialgeschwindigkeitsmessungen notwendig ist. Die durch den weiten Orbit bedingte kurze Transitdauer macht eine direkte Transitbeobachtungen mit bodengebundenen Teleskopen mit Durchmessern $\gtrsim 1$ m, wie sie im YETI-Netzwerk vorhanden sind, schwierig. Ein Langzeitprojekt zur präzisen Messung der Radialgeschwindigkeit stellt daher die bessere Vorgehensweise dar.

Die Untersuchung der Parameter Inklination, Radienverhältnis und inverser normierter Sternradius brachte ebenfalls keine signifikante Variation zum Vorschein. Aufgrund der Vielzahl der durchgeführten Transitbeobachtungen konnten die genannten Parameter, sowie die Orbitparameter Periode und Zeitpunkt der Nullepoche allerdings genauer als bisher bestimmt werden. Die neu gefundenen Werte stimmen dabei innerhalb ihrer Messfehler mit den Literaturdaten überein.

Für die Transitplaneten HAT-P-18b, HAT-P-19b, HAT-P-27b/WASP-40b und WASP-21b war es möglich, die Systemparameter neu zu bestimmen. Die Anzahl und die Zeitspanne der Beobachtungen ist dabei ausreichend, um vor allem die Periode genauer zu bestimmen als dies bisher möglich war. Um aufgrund der Streuung im O-C Diagramm bestimmte Systemkonfigurationen auszuschließen ist die Datenlage unzureichend. Wie auch schon für HAT-P-32b konnten die Parameter Inklination, Radienverhältnis und inverser normierter Sternradius bestimmt und mit den Literaturwerten verglichen werden. Dabei wurde sowohl innerhalb dieses Beobachtungsprojektes, als auch im Vergleich zu den Literaturdaten keine Variabilität festgestellt.

Für zwei Objekte, HAT-P-15b und WASP-38b, war es trotz der Anzahl der weltweit beteiligten Teleskope nicht möglich, Transitbeobachtungen durchzuführen. Dies liegt einerseits an der langen Transitdauer, die eine Beobachtung nur in langen Winternächten erlaubt, andererseits an der jeweiligen Umlaufperiode.

Als besonders nützlich erwiesen sich simultane Beobachtungen von Transitereignissen

mit verschiedenen Teleskopen, da diese systematische Fehler aufzeigen können, die bei Einzelbeobachtungen meist unentdeckt bleiben.

6.1 Implikationen für die Suche nach TTV-Signalen

Durch die kontinuierliche und simultane Beobachtung tausender Planetenkandidaten mit den Weltraumteleskopen *CoRoT* und *Kepler*, sowie den Ergebnissen von Einzelstudien bodengebundener Beobachtungen, ist es heute möglich, einen genaueren Blick auf die Suche nach TTV-Signalen zu werfen. Bei der Analyse von 64 Heißen Jupitern, also Planeten mit einer Masse von $\sim 1 M_{\text{J}}$ auf einem sehr engen Orbit, die mit dem Weltraumteleskop *Kepler* gefunden wurden, fanden Steffen u. a. (2012) keine TTV-Signale und schlussfolgerten, "Heiße Jupiter sind einsam". Im Gegenzug stellten sie fest, dass bei den ebenfalls untersuchten Stichproben von Warmen Jupitern und Heißen Neptunen bereits zusätzliche Begleiter im System bekannt sind und TTV-Analysen solcher Objekte vielversprechender wären. Eine spätere Studie von Szabó u. a. (2013) konnte auf Daten von 159 Heißen Jupitern zurückgreifen und fand bei 3 Objekten Variationen, welche durch keine anderen Effekte erklärt werden können, als dynamisch induziert zu sein.

Wenn Heiße Jupiter daher Begleiter haben, so gleicht ihre Suche der Suche nach der Nadel im Heuhaufen. Die Objekte dieser Studie, sowie der Studien von Raetz u. a. (2014) und Maciejewski u. a. (2011a,b, 2013b), sind ausgewählte Objekte, bei denen ein Indiz für das Vorhandensein eines zusätzlichen Begleiters gegeben ist. Dabei konnte nur im Fall von WASP-12b ein periodisches Signal gefunden werden, welches auf einen weiteren Planeten im System zurückgeführt werden kann (Maciejewski u. a., 2013a). Die Interpretation des gefundenen Signals ist allerdings schwierig und bedarf weiterer Nachbeobachtungen. So konnte zum Beispiel von Essen u. a. (2013) für den Transitplaneten Qatar-1b zeigen, dass eine mögliche Erklärung der gefundenen Signale die Existenz eines planetaren Begleiters im System ist. Aufgrund all dieser Ergebnisse ist es im Nachhinein daher nicht ungewöhnlich, dass für HAT-P-32b keine Transitzeitvariationen gefunden wurden. Zwar lassen die bisherigen Einschränkungen noch immer einige mögliche Begleiterszenarien zu, in Anbetracht der Resultate der *Kepler*-Mission ist aber nicht zu erwarten, dass im System noch weitere planetare Begleiter existieren. Das Ergebnis dieser Studie kann daher als zusätzlicher Beweis dafür gesehen werden, dass Systeme mit Heißen Jupitern in der Regel keine weiteren Planeten auf sternnahen Umlaufbahnen besitzen. Dass im Allgemeinen mithilfe der Analyse von Transitzeitvariationen zusätzliche Begleiter bei bekannten Transitplaneten gefunden werden können, zeigen die bereits erwähnten Bei-

spiele von Kepler-9b (Ballard u. a., 2011) und Kepler-6b (Nesvorný u. a., 2012). Ähnlich dazu konnte für das Kepler-88-System durch Nesvorný u. a. (2013) eine c-Komponente vorausgesagt und später durch Barros u. a. (2014) werden.

Aus der Perspektive eines möglichen Entstehungsszenarios war es durchaus möglich, dass weitere Planeten bei der Migration eines Heißen Jupiters gebildet werden (Mardling, 2007). Auch die Existenz von Planeten kleiner Masse innerhalb einer Orbitresonanz wäre dabei nicht ausgeschlossen. Ogihara u. a. (2014) stellten sogar fest, dass resonante Orbits stabile Konfigurationen sein können, wenngleich aufgrund der Interaktionen ein innerer Planet unwahrscheinlich ist. Dies machte die Suche nach erdähnlichen Planeten mittels TTV-Analysen lohnend, da innerhalb der Resonanzen große Signale zu erwarten sind. Ein Blick auf die von Goldreich & Schlichting (2014) zusammengetragene Statistik von Multiplanetensystemen unter den Planetenkandidaten des Kepler-Sateliten zeigt allerdings auf, dass sich keines der analysierten Systeme in (einige Systeme jedoch nahe) einer Bahnresonanz befinden.

Die meisten Resultate in der Literatur stammen aus Analysen kontinuierlicher Daten des *Kepler*-Sateliten und haben eine hohe photometrische Qualität. Eine weitere Suche nach Transitzeitvariationen mittels bodengebundener Teleskope ist dennoch sinnvoll, wenn:

- eine Verlagerung des Beobachtungsschwerpunktes von Heißen Jupitern auf Warme Jupiter und Heiße Neptune erfolgt, was allerdings eine Verringerung der Transittiefe sowie eine Vergrößerung der Umlaufperiode mit sich bringt,
- mithilfe eines Teleskopnetzwerkes wie dem YETI-Netzwerk auf Teleskope auf verschiedenen Längengraden zugegriffen werden kann, um eine möglichst hohe Anzahl aufeinanderfolgender Transitereignisse beobachten zu können und
- die Qualität der Daten aufgrund des Spiegeldurchmessers, der Wetterbedingungen während der Beobachtung, der Nachführung des Teleskops sowie der Zeitsynchronisation ausreicht, um den Transitzeitpunkt auf $\Delta T_{\text{mid}} \lesssim 30$ Sekunden genau bestimmen zu können.

Literaturverzeichnis

- Adams, E. R.; Dupree, A. K.; Kulesa, C. & McCarthy, D., 2013. *Adaptive Optics Images. II. 12 Kepler Objects of Interest and 15 Confirmed Transiting Planets*. *Astron. J.*, 146, 9.
- Agol, E.; Steffen, J.; Sari, R. & Clarkson, W., 2005. *On detecting terrestrial planets with timing of giant planet transits*. *MNRAS*, 359, 567–579.
- Anderson, D. R.; Barros, S. C. C.; Boisse, I. u. a., 2011. *WASP-40b: Independent Discovery of the 0.6 M Transiting Exoplanet HAT-P-27b*. *Publications of the ASP*, 123, 555–560.
- Anderson, D. R.; Gillon, M.; Hellier, C. u. a., 2008. *WASP-5b: a dense, very hot Jupiter transiting a 12th-mag Southern-hemisphere star*. *MNRAS*, 387, L4–L7.
- Bakos, G.; Noyes, R. W.; Kovács, G. u. a., 2004. *Wide-Field Millimagnitude Photometry with the HAT: A Tool for Extrasolar Planet Detection*. *Publications of the ASP*, 116, 266–277.
- Ballard, S.; Fabrycky, D.; Fressin, F. u. a., 2011. *The Kepler-19 System: A Transiting 2.2 R_{\oplus} Planet and a Second Planet Detected via Transit Timing Variations*. *Astrophys. J.*, 743, 200.
- Barros, S. C. C.; Díaz, R. F.; Santerne, A. u. a., 2014. *SOPHIE velocimetry of Kepler transit candidates. X. KOI-142 c: first radial velocity confirmation of a non-transiting exoplanet discovered by transit timing*. *Astron. Astrophys.*, 561, L1.
- Barros, S. C. C.; Faedi, F.; Collier Cameron, A. u. a., 2011a. *WASP-38b: a transiting exoplanet in an eccentric, 6.87d period orbit*. *Astron. Astrophys.*, 525, A54.
- Barros, S. C. C.; Pollacco, D. L.; Gibson, N. P. u. a., 2011b. *A lower mass for the exoplanet WASP-21b*. *MNRAS*, 416, 2593–2599.

- Béky, B.; Bakos, G. Á.; Hartman, J. u. a., 2011. *HAT-P-27b: A Hot Jupiter Transiting a G Star on a 3 Day Orbit*. *Astrophys. J.*, 734, 109.
- Bernt, I.; Müller, M.; Strassmeier, K. G. & Granzer, T., 2013. *Detection and initial characterisation of an exoplanet atmosphere with small aperture telescopes*. European Planetary Science Congress 2013, held 8-13 September in London, UK. Online at: "http://meetings.copernicus.org/epsc2013", id.EPSC2013-14, 8, EPSC2013.
- Bodenheimer, P.; Lin, D. N. C. & Mardling, R. A., 2001. *On the Tidal Inflation of Short-Period Extrasolar Planets*. *Astrophys. J.*, 548, 466–472.
- Borucki, W. J.; Koch, D. G.; Basri, G. u. a., 2011. *Characteristics of Kepler Planetary Candidates Based on the First Data Set*. *Astrophys. J.*, 728, 117.
- Bouchy, F.; Hebb, L.; Skillen, I. u. a., 2010. *WASP-21b: a hot-Saturn exoplanet transiting a thick disc star*. *Astron. Astrophys.*, 519, A98.
- Broeg, C.; Fernández, M. & Neuhäuser, R., 2005. *A new algorithm for differential photometry: computing an optimum artificial comparison star*. *Astronomische Nachrichten*, 326, 134–142.
- Brown, D. J. A.; Collier Cameron, A.; Díaz, R. F. u. a., 2012. *Analysis of Spin-Orbit Alignment in the WASP-32, WASP-38, and HAT-P-27/WASP-40 Systems*. *Astrophys. J.*, 760, 139.
- Brown, T. M.; Charbonneau, D.; Gilliland, R. L. u. a., 2001. *Hubble Space Telescope Time-Series Photometry of the Transiting Planet of HD 209458*. *Astrophys. J.*, 552, 699–709.
- Butler, R. P.; Marcy, G. W.; Williams, E. u. a., 1997. *Three New "51 Pegasi-Type" Planets*. *Astrophys. J. Letters*, 474, L115–L118.
- Carter, J. A. & Winn, J. N., 2009. *Parameter Estimation from Time-series Data with Correlated Errors: A Wavelet-based Method and its Application to Transit Light Curves*. *Astrophys. J.*, 704, 51–67.
- Chambers, J. E., 1999. *A hybrid symplectic integrator that permits close encounters between massive bodies*. *MNRAS*, 304, 793–799.

- Chambers, J. E. & Migliorini, F., 1997. *Mercury - A New Software Package for Orbital Integrations*. In *AAS/Division for Planetary Sciences Meeting Abstracts #29*, Band 29 von *Bulletin of the American Astronomical Society*, 1024.
- Christian, D. J.; Gibson, N. P.; Simpson, E. K. u. a., 2009. *WASP-10b: a $3M_J$, gas-giant planet transiting a late-type K star*. *MNRAS*, 392, 1585–1590.
- Ciceri, S.; Mancini, L.; Southworth, J. u. a., 2013. *Simultaneous follow-up of planetary transits: revised physical properties for the planetary systems HAT-P-16 and WASP-21*. *Astron. Astrophys.*, 557, A30.
- Claret, A., 2000. *A new non-linear limb-darkening law for LTE stellar atmosphere models. Calculations for $-5.0 \leq \log[M/H] \leq +1$, $2000 \text{ K} \leq T_{\text{eff}} \leq 50000 \text{ K}$ at several surface gravities*. *Astron. Astrophys.*, 363, 1081–1190.
- Claret, A. & Bloemen, S., 2011. *Gravity and limb-darkening coefficients for the Kepler, CoRoT, Spitzer, uvby, UBVRIJHK, and Sloan photometric systems*. *Astron. Astrophys.*, 529, A75.
- Claret, A. & Hauschildt, P. H., 2003. *The limb-darkening for spherically symmetric Next-Gen model atmospheres: A-G main-sequence and sub-giant stars*. *Astron. Astrophys.*, 412, 241–248.
- Cousins, A. W. J., 1976. *VRI standards in the E regions*. *Memoirs of the Royal Astronomical Society*, 81, 25.
- Davis, L., 1987. *Specifications for the Aperture Photometry Package*.
- , 1994. *A Reference Guide to the IRAF/DAOPHOT Package*.
- Diaz-Cordoves, J. & Gimenez, A., 1992. *A new nonlinear approximation to the limb-darkening of hot stars*. *Astron. Astrophys.*, 259, 227–231.
- Eastman, J.; Gaudi, B. S. & Agol, E., 2013. *EXOFAST: A Fast Exoplanetary Fitting Suite in IDL*. *Publications of the ASP*, 125, 83–112.
- Eastman, J.; Siverd, R. & Gaudi, B. S., 2010. *Achieving Better Than 1 Minute Accuracy in the Heliocentric and Barycentric Julian Dates*. *Publications of the ASP*, 122, 935–946.
- Errmann, R.; Torres, G.; Schmidt, T. O. B. u. a., 2014. *Investigation of a transiting planet candidate in Trumpler 37: An astrophysical false positive eclipsing spectroscopic binary star*. *Astronomische Nachrichten*, 335, 345.

- Esposito, M.; Covino, E.; Mancini, L. u. a., 2014. *The GAPS Programme with HARPS-N at TNG. III: The retrograde orbit of HAT-P-18b*. *Astron. Astrophys.*, 564, L13.
- Etzel, P. B., 1981. *A Simple Synthesis Method for Solving the Elements of Well-Detached Eclipsing Systems*. In E. B. Carling & Z. Kopal, Verleger, *Photometric and Spectroscopic Binary Systems*, 111.
- Fischer, D. A.; Marcy, G. W.; Butler, R. P. u. a., 2008. *Five Planets Orbiting 55 Cancri*. *Astrophys. J.*, 675, 790–801.
- Ford, E. B. & Holman, M. J., 2007. *Using Transit Timing Observations to Search for Trojans of Transiting Extrasolar Planets*. *Astrophys. J. Letters*, 664, L51–L54.
- Fukugita, M.; Ichikawa, T.; Gunn, J. E. u. a., 1996. *The Sloan Digital Sky Survey Photometric System*. *Astron. J.*, 111, 1748.
- Fulton, B. J.; Shporer, A.; Winn, J. N. u. a., 2011. *Long-term Transit Timing Monitoring and Refined Light Curve Parameters of HAT-P-13b*. *Astron. J.*, 142, 84.
- Gazak, J. Z.; Johnson, J. A.; Tonry, J. u. a., 2012. *Transit Analysis Package: An IDL Graphical User Interface for Exoplanet Transit Photometry*. *Advances in Astronomy*, 2012, 697967.
- Gibson, N. P.; Aigrain, S.; Barstow, J. K. u. a., 2013. *The optical transmission spectrum of the hot Jupiter HAT-P-32b: clouds explain the absence of broad spectral features?* *MNRAS*, 436, 2974–2988.
- Goldreich, P. & Schlichting, H. E., 2014. *Overstable Librations can Account for the Paucity of Mean Motion Resonances among Exoplanet Pairs*. *Astron. J.*, 147, 32.
- Gorshanov, D. L.; Shakht, N. A. & Kisselev, A. A., 2006. *Observations of the binary star 61 Cyg on the 26 inch refractor at the Pulkovo observatory*. *Astrophysics*, 49, 386–396.
- Hartman, J. D.; Bakos, G. Á.; Sato, B. u. a., 2011a. *HAT-P-18b and HAT-P-19b: Two Low-density Saturn-mass Planets Transiting Metal-rich K Stars*. *Astrophys. J.*, 726, 52.
- Hartman, J. D.; Bakos, G. A.; Torres, G. u. a., 2011b. *HAT-P-32b and HAT-P-33b: Two Highly Inflated Hot Jupiters Transiting High-jitter Stars*. *Astrophys. J.*, 742, 59.
- Haswell, C. A., 2010. *Transiting Exoplanets*.

- Holman, M. J. & Murray, N. W., 2005. *The Use of Transit Timing to Detect Terrestrial-Mass Extrasolar Planets*. *Science*, 307, 1288–1291.
- Howell, S. B., 2006. *Handbook of CCD Astronomy*.
- Irwin, A. W. & Fukushima, T., 1999. *A numerical time ephemeris of the Earth*. *Astron. Astrophys.*, 348, 642–652.
- Kipping, D. M., 2009. *Transit timing effects due to an exomoon*. *MNRAS*, 392, 181–189.
- Kipping, D. M.; Hartman, J.; Buchhave, L. A. u. a., 2013. *The Hunt for Exomoons with Kepler (HEK). II. Analysis of Seven Viable Satellite-hosting Planet Candidates*. *Astrophys. J.*, 770, 101.
- Kipping, D. M.; Huang, C. X.; Nesvorný, D. u. a., 2014. *The Possible Moon of Kepler-90g is a False Positive*. *ArXiv e-prints*.
- Klinglesmith, D. A. & Sobieski, S., 1970. *Nonlinear Limb Darkening for Early-Type Stars*. *Astron. J.*, 75, 175.
- Knutson, H. A.; Fulton, B. J.; Montet, B. T. u. a., 2014. *Friends of Hot Jupiters. I. A Radial Velocity Search for Massive, Long-period Companions to Close-in Gas Giant Planets*. *Astrophys. J.*, 785, 126.
- Koch, D. G.; Borucki, W. J.; Basri, G. u. a., 2010. *Kepler Mission Design, Realized Photometric Performance, and Early Science*. *Astrophys. J. Letters*, 713, L79.
- Kopal, Z., 1950. *Detailed effects of limb darkening upon light and velocity curves of close binary systems*. *Harvard College Observatory Circular*, 454, 1–12.
- Kovács, G.; Bakos, G. Á.; Hartman, J. D. u. a., 2010. *HAT-P-15b: A 10.9 Day Extrasolar Planet Transiting a Solar-type Star*. *Astrophys. J.*, 724, 866–877.
- Maciejewski, G.; Dimitrov, D.; Neuhäuser, R. u. a., 2011a. *Transit timing variation and activity in the WASP-10 planetary system*. *MNRAS*, 411, 1204–1212.
- Maciejewski, G.; Dimitrov, D.; Seeliger, M. u. a., 2013a. *Multi-site campaign for transit timing variations of WASP-12 b: possible detection of a long-period signal of planetary origin*. *Astron. Astrophys.*, 551, A108.
- Maciejewski, G.; Errmann, R.; Raetz, S. u. a., 2011b. *High-precision photometry of WASP-12 b transits*. *Astron. Astrophys.*, 528, A65.

- Maciejewski, G.; Niedzielski, A.; Wolszczan, A. u. a., 2013b. *Constraints on a Second Planet in the WASP-3 System*. *Astron. J.*, 146, 147.
- Mandel, K. & Agol, E., 2002. *Analytic Light Curves for Planetary Transit Searches*. *Astrophys. J. Letters*, 580, L171–L175.
- Mardling, R. A., 2007. *Long-term tidal evolution of short-period planets with companions*. *MNRAS*, 382, 1768–1790.
- Mayor, M. & Queloz, D., 1995. *A Jupiter-mass companion to a solar-type star*. *Nature*, 378, 355–359.
- Miralda-Escudé, J., 2002. *Orbital Perturbations of Transiting Planets: A Possible Method to Measure Stellar Quadrupoles and to Detect Earth-Mass Planets*. *Astrophys. J.*, 564, 1019–1023.
- Montenbruck, O., 1984. *Grundlagen der Ephemeridenrechnung*.
- Morbidelli, A.; Levison, H. F.; Tsiganis, K. & Gomes, R., 2005. *Chaotic capture of Jupiter's Trojan asteroids in the early Solar System*. *Nature*, 435, 462–465.
- Mugrauer, M. & Berthold, T., 2010. *STK: A new CCD camera at the University Observatory Jena*. *Astronomische Nachrichten*, 331, 449.
- Nelson, B. & Davis, W. D., 1972. *Eclipsing-Binary Solutions by Sequential Optimization of the Parameters*. *Astrophys. J.*, 174, 617.
- Nesvorný, D.; Kipping, D.; Terrell, D. u. a., 2013. *KOI-142, The King of Transit Variations, is a Pair of Planets near the 2:1 Resonance*. *Astrophys. J.*, 777, 3.
- Nesvorný, D.; Kipping, D. M.; Buchhave, L. A. u. a., 2012. *The Detection and Characterization of a Nontransiting Planet by Transit Timing Variations*. *Science*, 336, 1133–.
- Nesvorný, D. & Morbidelli, A., 2008. *Mass and Orbit Determination from Transit Timing Variations of Exoplanets*. *Astrophys. J.*, 688, 636–646.
- Neuhäuser, R.; Errmann, R.; Berndt, A. u. a., 2011. *The Young Exoplanet Transit Initiative (YETI)*. *Astronomische Nachrichten*, 332, 547.
- Ogihara, M.; Kobayashi, H. & Inutsuka, S.-i., 2014. *N-body Simulations of Terrestrial Planet Formation under the Influence of a Hot Jupiter*. *Astrophys. J.*, 787, 172.

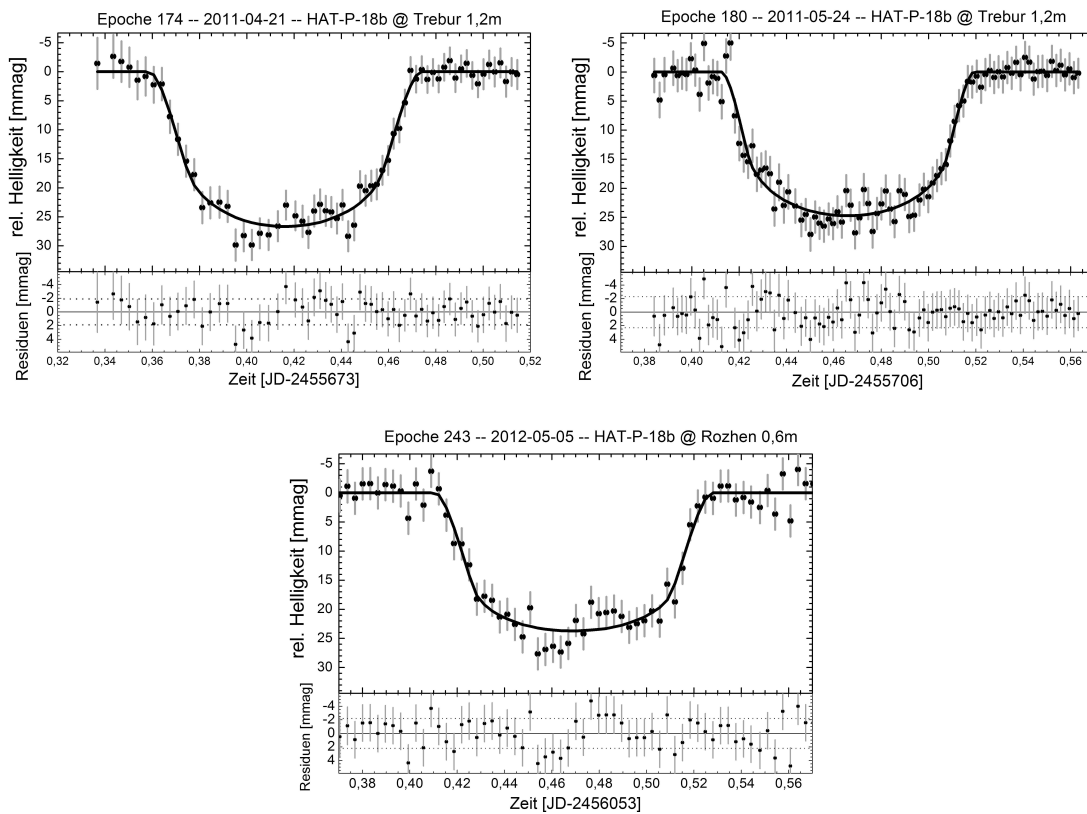
- Perryman, M. A. C., 2000. *Extra-solar planets*. Reports on Progress in Physics, 63, 1209–1272.
- Popper, D. M. & Etzel, P. B., 1981. *Photometric orbits of seven detached eclipsing binaries*. Astron. J., 86, 102–120.
- Raetz, S.; Maciejewski, G.; Ginski, C. u. a., 2014. *Transit timing of TrES-2: a combined analysis of ground- and space-based photometry*. MNRAS, 444, 1351–1368.
- Rømer, O., 1676. *Démonstration touchant le mouvement de la lumière trouvé par M.Rømer de l'Académie royale des sciences*. Le Journal des Sçavans, Paris, 1676, 233–236.
- Sada, P. V.; Deming, D.; Jennings, D. E. u. a., 2012. *Extrasolar Planet Transits Observed at Kitt Peak National Observatory*. Publications of the ASP, 124, 212–229.
- Sasselov, D. D., 2003. *The New Transiting Planet OGLE-TR-56b: Orbit and Atmosphere*. Astrophys. J., 596, 1327–1331.
- Schwarzschild, K., 1906. *On the equilibrium of the Sun's atmosphere*. Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen. Math.-phys. Klasse, 195, p. 41-53, 195, 41–53.
- Seager, S. & Mallén-Ornelas, G., 2003. *A Unique Solution of Planet and Star Parameters from an Extrasolar Planet Transit Light Curve*. Astrophys. J., 585, 1038–1055.
- Seeliger, M.; Dimitrov, D.; Kjurkchieva, D. u. a., 2014. *Transit timing analysis in the HAT-P-32 system*. MNRAS, 441, 304–315.
- Shapiro, I. I., 1964. *Fourth Test of General Relativity*. Physical Review Letters, 13, 789–791.
- Sing, D. K.; Désert, J.-M.; Lecavelier Des Etangs, A. u. a., 2009. *Transit spectrophotometry of the exoplanet HD 189733b. I. Searching for water but finding haze with HST NICMOS*. Astron. Astrophys., 505, 891–899.
- Sitarz, E., 1983. *Kulturen am Rande der Bibel*. Katholisches Bibelwerk, Stuttgart.
- Southworth, J., 2008. *Homogeneous studies of transiting extrasolar planets - I. Light-curve analyses*. MNRAS, 386, 1644–1666.

- , 2011. *Homogeneous studies of transiting extrasolar planets - IV. Thirty systems with space-based light curves*. MNRAS, 417, 2166–2196.
- , 2012. *Homogeneous studies of transiting extrasolar planets - V. New results for 38 planets*. MNRAS, 426, 1291–1323.
- Southworth, J.; Hinse, T. C.; Jørgensen, U. G. u. a., 2009. *High-precision photometry by telescope defocusing - I. The transiting planetary system WASP-5*. MNRAS, 396, 1023–1031.
- Southworth, J.; Maxted, P. F. L. & Smalley, B., 2004. *Eclipsing binaries in open clusters - II. V453 Cyg in NGC 6871*. MNRAS, 351, 1277–1289.
- Steffen, J. H.; Fabrycky, D. C.; Agol, E. u. a., 2013. *Transit timing observations from Kepler - VII. Confirmation of 27 planets in 13 multiplanet systems via transit timing variations and orbital stability*. MNRAS, 428, 1077–1087.
- Steffen, J. H.; Gaudi, B. S.; Ford, E. B. u. a., 2007. *Detecting and Characterizing Planetary Systems with Transit Timing*. ArXiv e-prints.
- Steffen, J. H.; Ragozzine, D.; Fabrycky, D. C. u. a., 2012. *Kepler constraints on planets near hot Jupiters*. Proceedings of the National Academy of Science, 109, 7982–7987.
- Stetson, P. B., 1987. *DAOPHOT - A computer program for crowded-field stellar photometry*. Publications of the ASP, 99, 191–222.
- Stoer, J. & Bulrish, R., 1980. *Introduction to Numerical Analysis*.
- Strand, K. A., 1944. *The astrometric study of unseen companions in double stars*. Astron. J., 51, 12–13.
- Struve, O., 1952. *Proposal for a project of high-precision stellar radial velocity work*. The Observatory, 72, 199–200.
- Szabó, R.; Szabó, G. M.; Dály, G. u. a., 2013. *Multiple planets or exomoons in Kepler hot Jupiter systems with transit timing variations?* Astron. Astrophys., 553, A17.
- Tody, D., 1993. *IRAF in the Nineties*. In R. J. Hanisch, R. J. V. Brissenden, & J. Barnes, Verleger, *Astronomical Data Analysis Software and Systems II*, Band 52 von *Astronomical Society of the Pacific Conference Series*, 173.

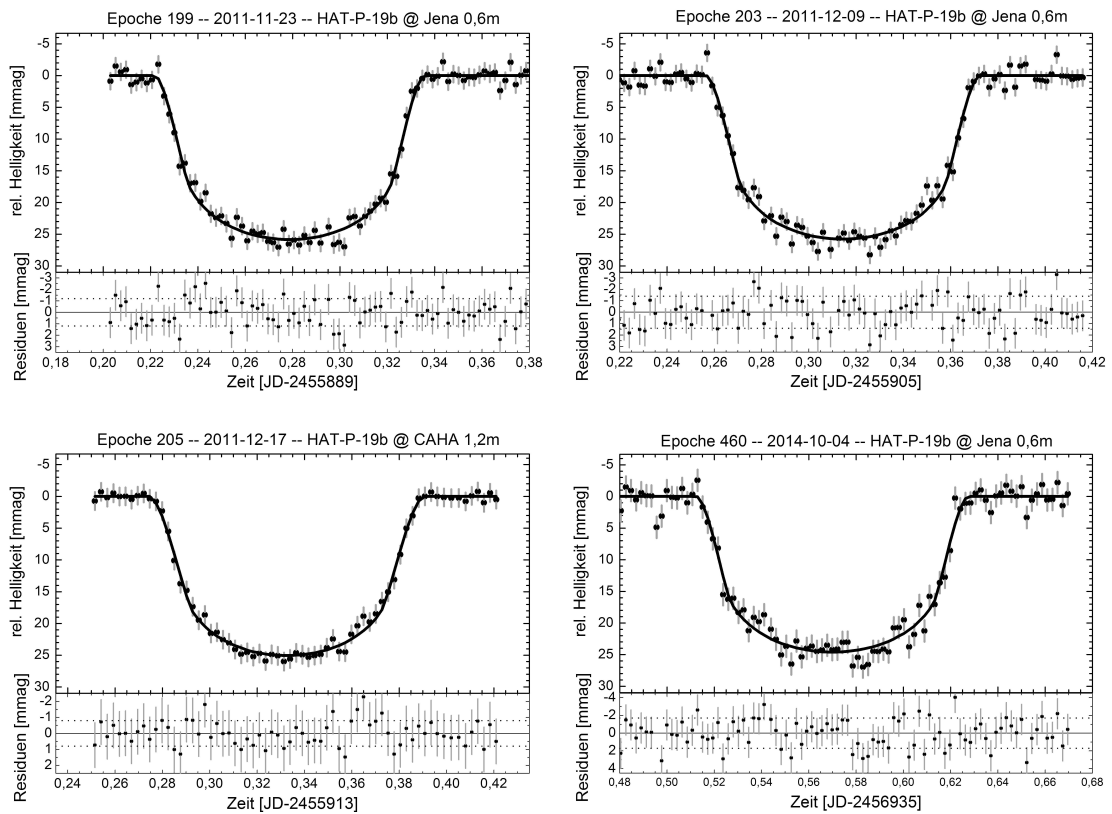
- von Essen, C.; Schröter, S.; Agol, E. & Schmitt, J. H. M. M., 2013. *Qatar-1: indications for possible transit timing variations*. *Astron. Astrophys.*, 555, A92.
- Winn, J. N., 2010. *Transits and Occultations*. ArXiv e-prints.
- Wolszczan, A. & Frail, D. A., 1992. *A planetary system around the millisecond pulsar PSR1257 + 12*. *Nature*, 355, 145–147.
- Young, A. T.; Genet, R. M.; Boyd, L. J. u. a., 1991. *Precise automatic differential stellar photometry*. *Publications of the ASP*, 103, 221–242.
- Zahn, J.-P., 1977. *Tidal friction in close binary stars*. *Astron. Astrophys.*, 57, 383–394.
- Zhao, M.; O'Rourke, J. G.; Wright, J. T. u. a., 2014. *Characterization of the Atmosphere of the Hot Jupiter HAT-P-32Ab and the M-dwarf Companion HAT-P-32B*. *Astrophys. J.*, 796, 115.

A Transitlichtkurven

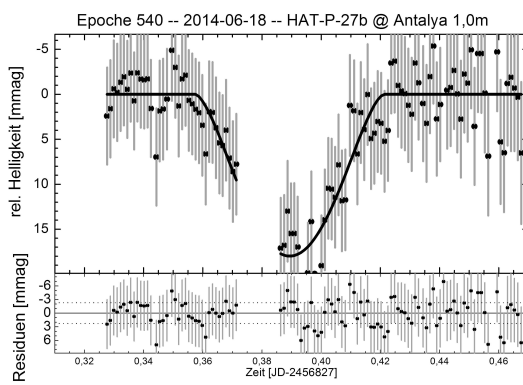
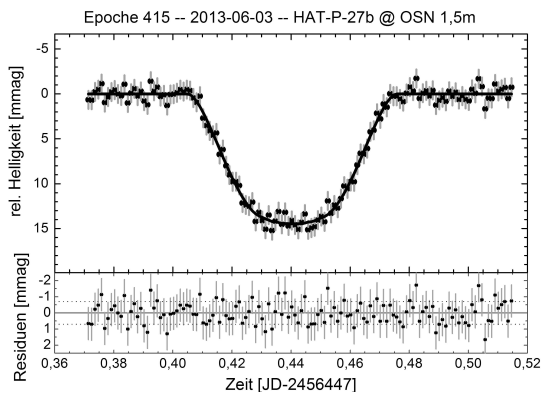
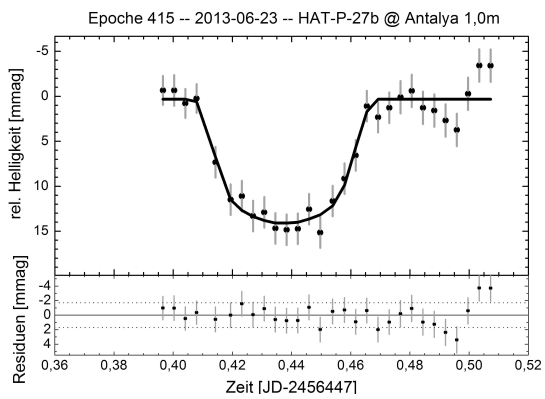
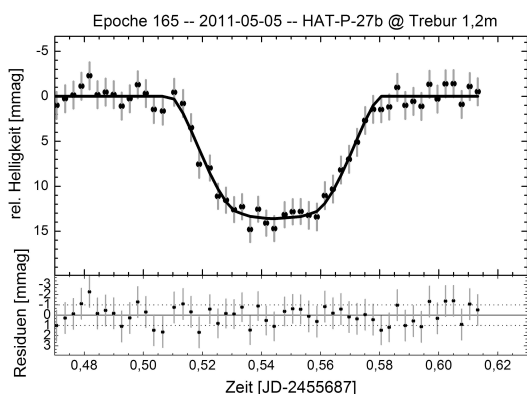
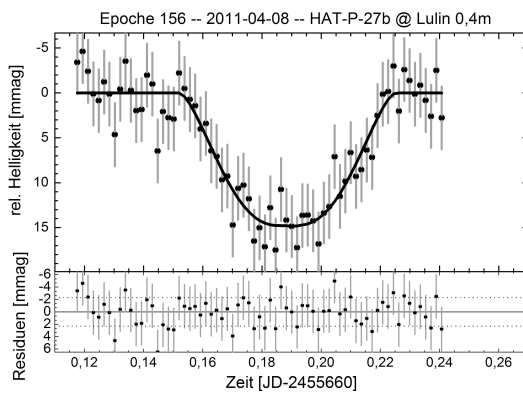
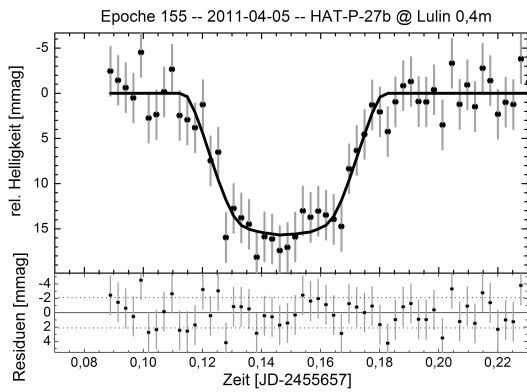
A.1 Transitlichtkurven von HAT-P-18b



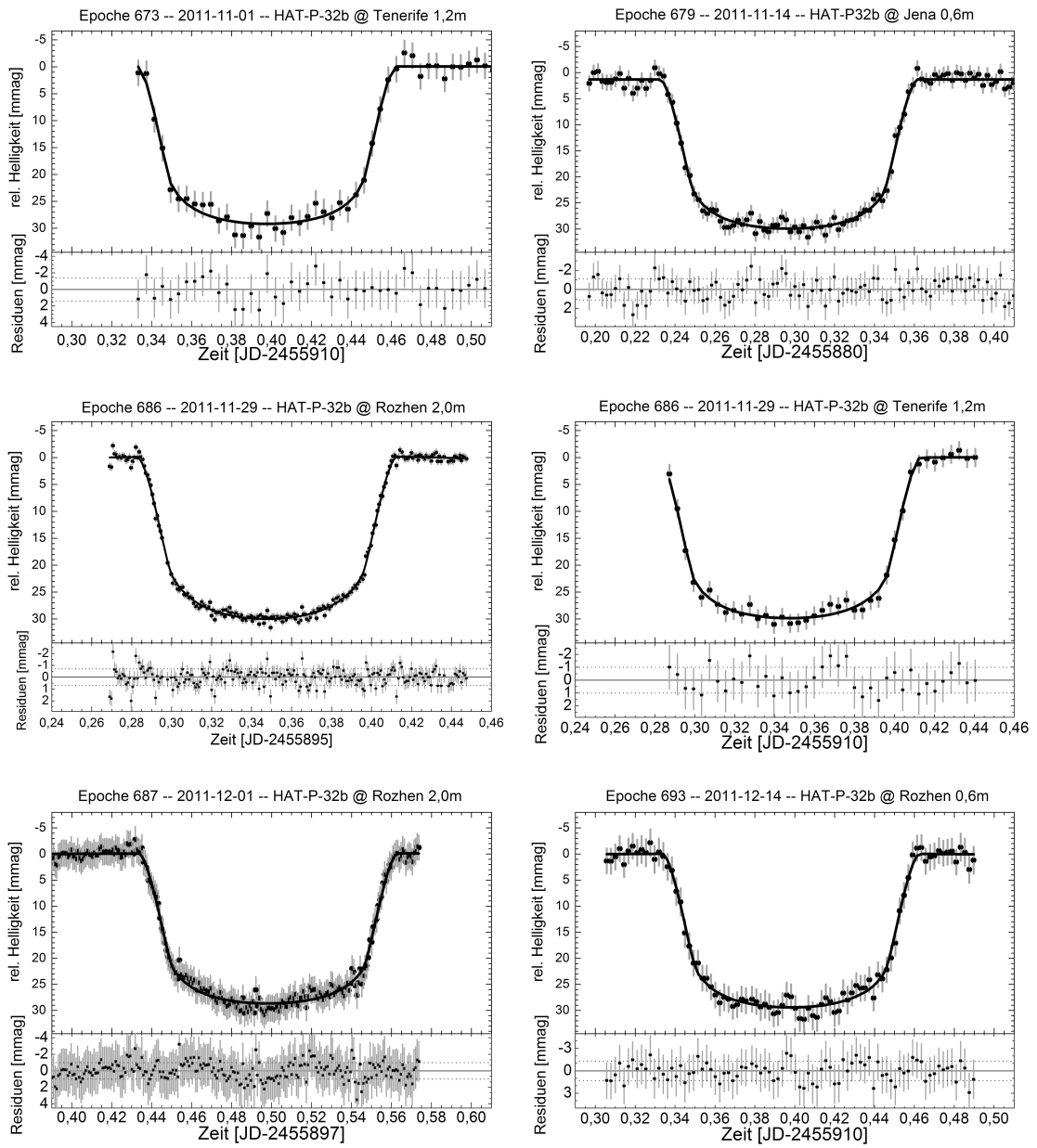
A.2 Transitlichtkurven von HAT-P-19b



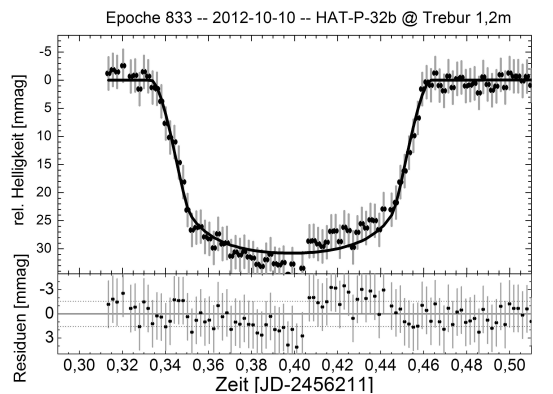
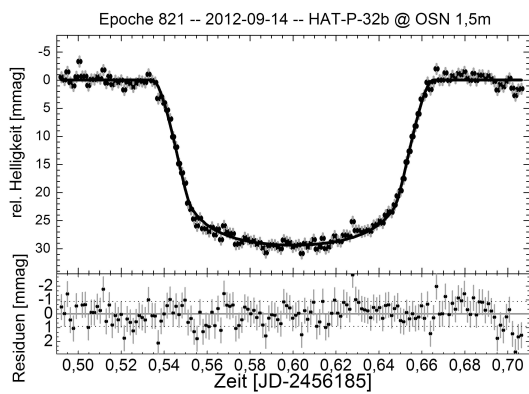
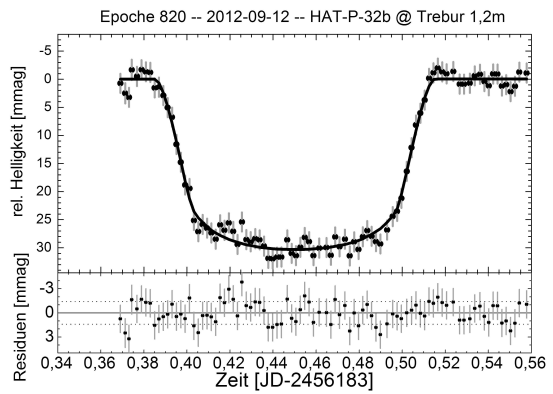
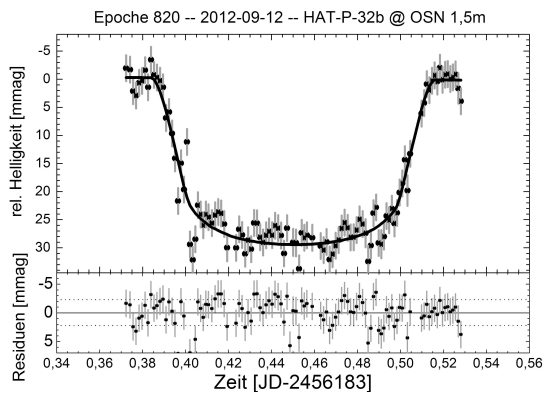
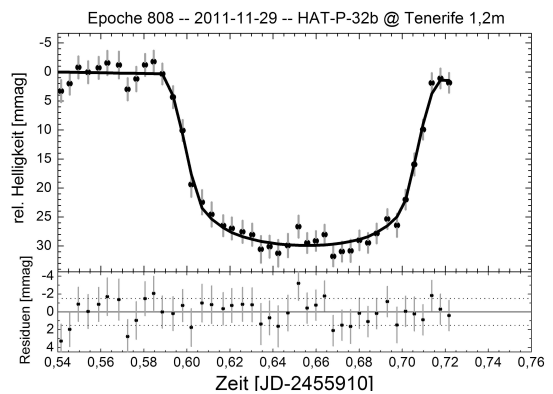
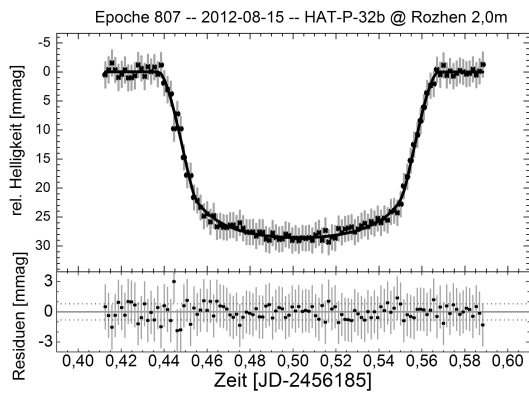
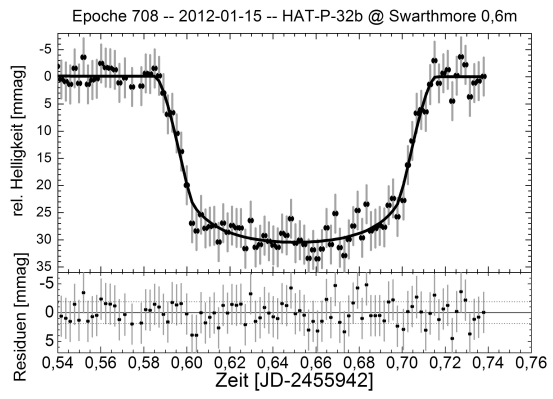
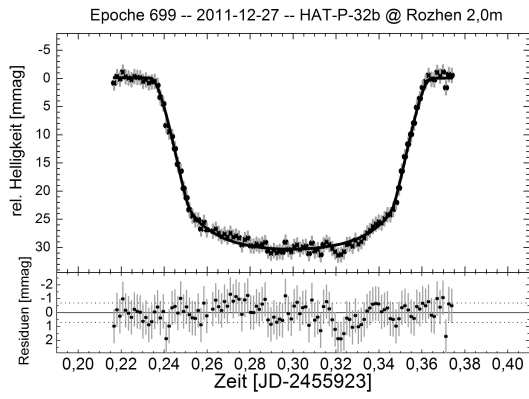
A.3 Transitlichtkurven von HAT-P-27b/WASP-40b

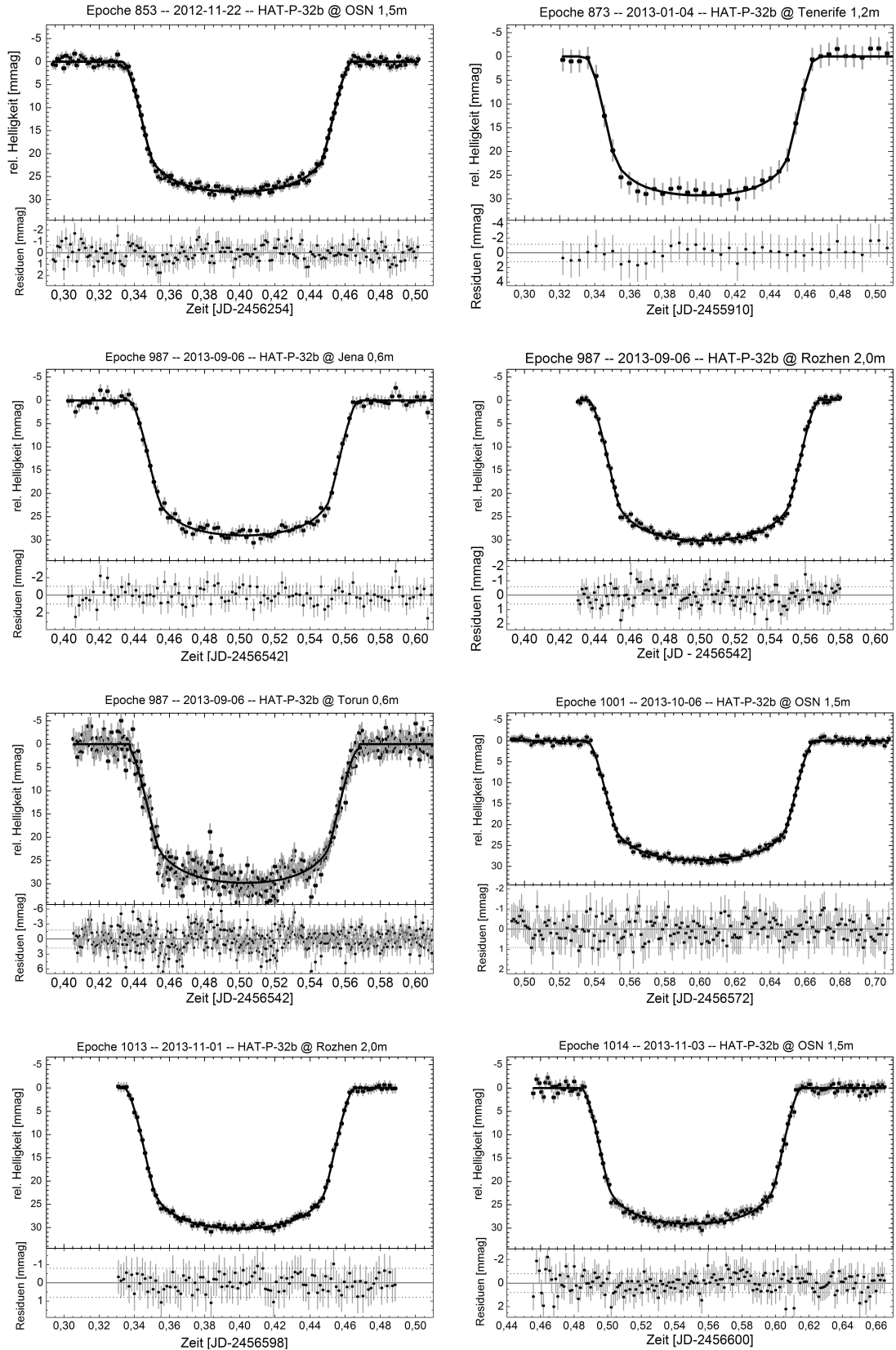


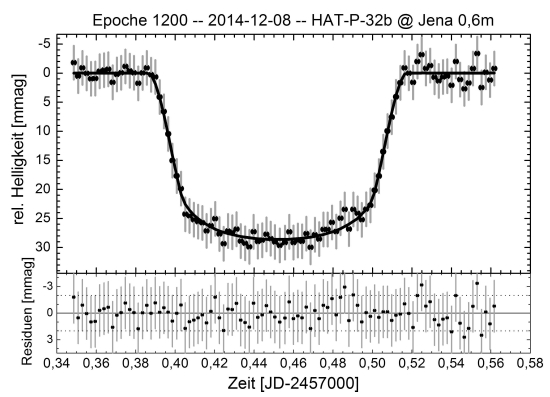
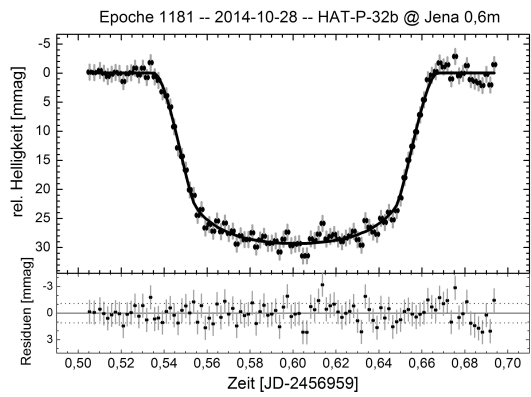
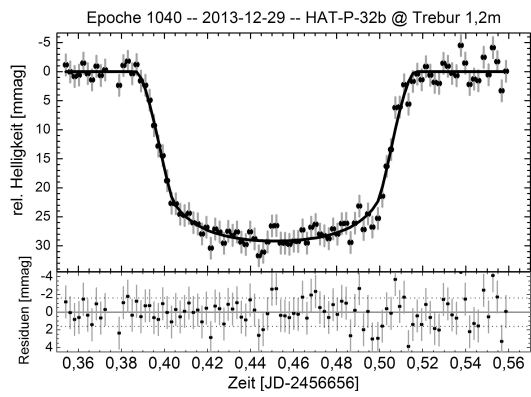
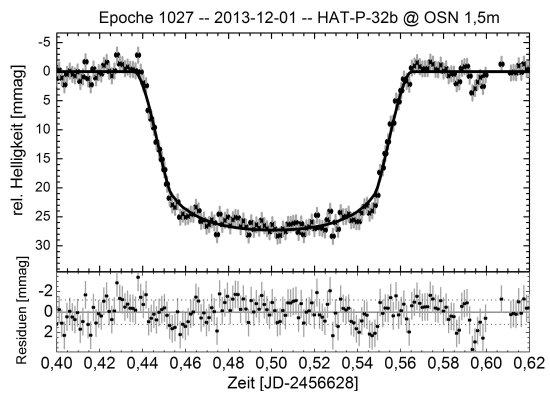
A.4 Transitlichtkurven von HAT-P-32b



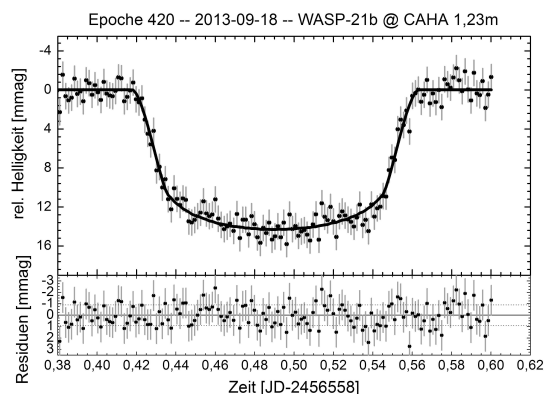
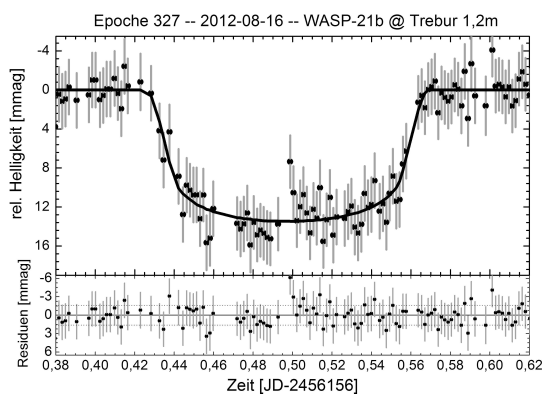
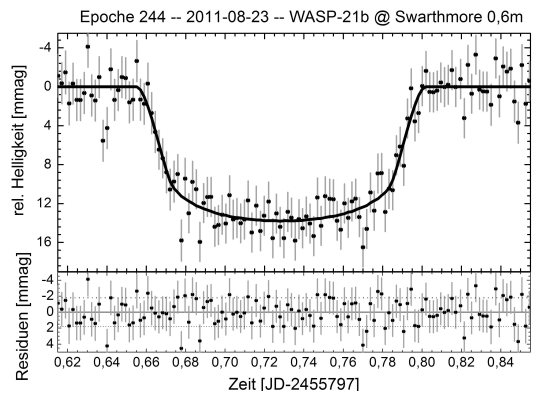
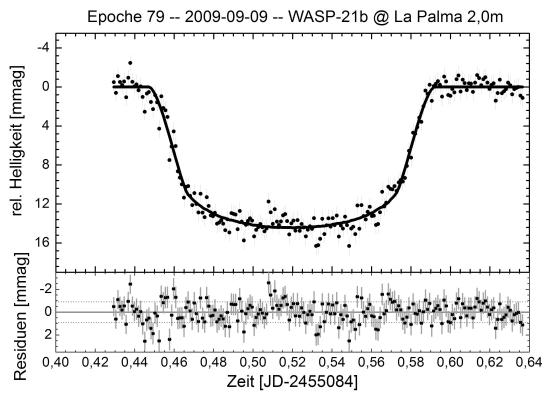
ANHANG A. TRANSITLICHTKURVEN







A.5 Transitlichtkurven von WASP-21b



B Quellcodes

B.1 Programme zur Berechnung von Transitzeitpunkten

Quellcode B.1: Hilfsprogramm zur Definition von Klassen, Importanweisungen und Funktionen. Die Berechnungen beruhen zum Teil auf Gleichungen von Montenbruck (1984)

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
## my_astro.py

importerrors=False
try:
    import math
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import os
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import sys
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import copy
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import string
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import numpy
except ImportError as e:
    print "Fehler:_{0}..Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
importerrors=True
try:
    import pytz
except ImportError as e:
```

```

    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
try:
    import datetime
except ImportError as e:
    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
try:
    import time
except ImportError as e:
    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
try:
    import ephem
except ImportError as e:
    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
try:
    from pexpect import *
except ImportError as e:
    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
try:
    import re
except ImportError as e:
    print "Fehler:_{0}._Bitte_nachinstallieren_oder_Programm_auf_narsil_ausfuehren".format
        (e.args)
    import errors=True
if import errors:
    exit(1)

#####
# Klassen-Definitionen
#####
class Target(object):
    def __init__(self, abbreviation, name, ra, dec, t0, period, duration):
        self.Abbreviation=abbreviation
        self.Name=name
        self.RA=ra
        self.DEC=dec
        self.T0=float(t0)
        self.Period=float(period)
        self.Duration=float(duration)
    def set_Target(self, abbreviation, name, ra, dec, t0, period, duration):
        self.Abbreviation=abbreviation
        self.Name=name
        self.RA=ra
        self.DEC=dec
        self.T0=float(t0)
        self.Period=float(period)
        self.Duration=float(duration)
    def get_Name(self):
        return self.Name
    def get_RA(self):
        return self.RA
    def get_DEC(self):
        return self.DEC
    def get_T0(self):
        return self.T0
    def get_Period(self):

```

```

        return self.Period
    def get_Duration(self):
        return self.Duration
class Transit:
    def __init__(self, name, begin, end, observatory, moon,fullmoondistance, el_start,
        el_stop, sun_start, sun_stop, flag):
        self.name = name
        self.begin = begin
        self.end = end
        self.observatory = observatory
        self.moon = moon
        self.fullmoondistance = fullmoondistance
        self.el_start = el_start
        self.el_stop = el_stop
        self.sun_start = sun_start
        self.sun_stop = sun_stop
        self.beginINT = int(begin*1000)
        self.flag = flag
    def __repr__(self):
        return repr((self.name, self.begin, self.end, self.observatory, self.moon, self.
            fullmoondistance, self.el_start, self.el_stop, self.sun_start, self.sun_stop,
            self.flag))

class Telescope(object):
    def __init__(self, abbreviation, name, latitude, Elongitude, elevation, pressure=1000,
        numberobservatories=0):
        self.__Latitude=latitude
        self.__Elongitude=Elongitude
        self.__Name=name
        self.__Abbreviation=abbreviation
        self.__Elevation=int(elevation)
        self.__Pressure=int(pressure)
        self.__NumberObservatories=numberobservatories
    def set_Telescope(self,latitude, Elongitude, name, abbreviation, elevation, pressure
        =1000):
        self.__Latitude=latitude
        self.__Elongitude=Elongitude
        self.__Name=name
        self.__Abbreviation=abbreviation
        self.__Elevation=int(elevation)
        self.__Pressure=int(pressure)
    def Increase_NumberObs(self):
        self.__NumberObservatories+=1
    def get_NumberObs(self):
        return self.__NumberObservatories
    def get_Latitude(self):
        return self.__Latitude
    def get_Elongitude(self):
        return self.__Elongitude
    def get_Name(self):
        return self.__Name
    def get_Abbreviation(self):
        return self.__Abbreviation
    def get_Elevation(self):
        return self.__Elevation
    def get_Pressure(self):
        return self.__Pressure

#####
# Abrunden
#####
def div_ceil(a, b):
    return ((a+(b-1))/b)

```

```

def test_schaltjahr(jahr):
    try:
        datetime.date(jahr,2,29)
    except ValueError:
        return False
    return True

#####
# Addition von Daten
#####
def date_plus_hour(date, hour):
    jd = gregdate_to_juldate(date['date'], date['time'])['jd']
    hour_jd = hour/24.

    return copy.deepcopy(juldate_to_gregdate(jd+hour_jd))
#####

#####
# Umrechnung vom Sexigesimal ins Grad-System
#####
def sex_to_deg(ra, dec):

    ra_deg = 15*(ra[0] + ra[1]/60.0 + ra[2]/3600.0)

    if (dec[0] != 0):
        if (dec[0] < 0):
            dec_deg = dec[0] - dec[1]/60.0 - dec[2]/3600.0
        else:
            dec_deg = dec[0] + dec[1]/60.0 + dec[2]/3600.0

    elif (dec[0] == 0) and (dec[1] != 0):
        if (dec[1] < 0):
            dec_deg = dec[1]/60.0 - dec[2]/3600.0
        else:
            dec_deg = dec[1]/60.0 + dec[2]/3600.0

    else:
        dec_deg = dec[0] + dec[1]/60.0 + dec[2]/3600.0

    return copy.deepcopy({'ra_deg':ra_deg, 'dec_deg':dec_deg})
#####

#####
# Umrechnung vom Gregorianischen ins Julianische Datum
#
# (siehe Buch "Grundlagen der Ephemeridenrechnung" von
# Oliver Montenbruck)
#####
def gregdate_to_juldate(Date, UT_Time):

    if Date[1] < 3:
        year = Date[2] - 1
        month = Date[1] + 12
    else:
        year = Date[2]
        month = Date[1]
    day = Date[0]

    JD = 1720996.5 + math.floor(365.25*year) + math.floor(30.6001*(month+1)) + math.floor(
        year/400.) - math.floor(year/100.) + day + (UT_Time[0] + UT_Time[1]/60. + UT_Time
        [2]/3600.)/24.

    MJD = JD - 2400000.5

```

```

    return copy.deepcopy({'jd':JD, 'mjd':MJD})
#####

#####
# Umrechnung vom Julianischen ins Gregorianische Datum
#
# (siehe Buch "Grundlagen der Ephemeridenrechnung" von
# Oliver Montenbruck)
#####
def juldate_to_gregdate(juldate, flag='jd'):

    if flag == 'mjd':
        JD = juldate + 2400000.5
    else:
        JD = juldate

    a = math.floor(JD+0.5)

    if a < 2299161:
        c = a+1524
    else:
        b = math.floor((a-1867216.25)/36524.25)
        c = a+b-math.floor(b/4.)+1525

    d = math.floor((c-122.1)/365.25)
    e = math.floor(365.25*d)
    f = math.floor((c-e)/30.6001)

    day = math.modf(c-e-math.floor(30.6001*f)+(JD+0.5-a))
    month = f-1-12*math.floor(f/14.)
    year = d-4715-math.floor((7+month)/10.)

    hour = math.modf(24.*day[0])
    minute = math.modf(60.*hour[0])
    sec = 60.*minute[0]

    if month > 12:
        year = year + 1
        month = month - 12

    return copy.deepcopy({'date':[int(day[1]), int(month), int(year)], 'time':[int(hour
        [1]), int(minute[1]), sec]})
#####

#####
# Berechnung von string in h,m,sec
#####
def string_to_time(time_string):
    time = string.split(time_string, ':')
    h = string.atoi(time[0])
    minute = string.atoi(time[1])
    sec = string.atof(time[2])

    return [copy.deepcopy(h), copy.deepcopy(minute), copy.deepcopy(sec)]
#####

#####
# Berechnung von string in day,month,year
#####
def string_to_date(Date_string):
    trenn = ['-','/']

```

```
if len(Date_string) == 6:
    year = string.atoi(Date_string[:2])
    month = string.atoi(Date_string[2:4])
    day = string.atoi(Date_string[4:])
else:
    for i in trenn:
        if string.find(Date_string, i) != -1:
            #print Date_string
            datum = string.split(Date_string, i)
            #print datum
            day = string.atoi(datum[2])
            month = string.atoi(datum[1])
            year = string.atoi(datum[0])

if (day > 31) and (year < 32):
    hilfs_tag = copy.deepcopy(day)
    day = copy.deepcopy(year)
    year = hilfs_tag

del hilfs_tag

if year < 1900:
    year += 1900

return [copy.deepcopy(day), copy.deepcopy(month), copy.deepcopy(year)]
#####
```


Quellcode B.2: Programm zur Berechnung von Transitzeitpunkten

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
### Name : predictions.py
### Version: 0.2 (alpha)
### Zweck : Transit-Berechnungen fuer Yeti-netzwerk
### Autor : Martin Seeliger
### Date : 2011/03/10
###
### Bemerkungen: Basiert auf Grundlage des airmass_plot von Tristan Roell
### Basiert auf pyephem (Algorithmen von Xephem)
### und auf Algorithmen aus dem Buch
### 'Grundlagen der Ephemeridenrechnung' von
### Oliver Montenbruck
###
#####
### The Program predictions.py is called by invoking the command
###
### >>> python predictions.py
###
### from the console. It is dedicated to compute all transit events at a certain
telescope
### or list of telescopes provided within further files.
###
### below you can find import declarations that should NOT be changed
### all

###
###
#####
## import declarations
#####
###
###

try:
    import my_astro
    from my_astro import *
except ImportError as e:
    print "Fehler beim Import von: {0}. Die Datei my_astro.py muss sich im Programmordner
        befinden!".format(e.args)
    exit(1)

#####
#####
###
###
### specific input data start:
#####
###
###

###
#####
## Namen von Input- und Output Dateien
#####
###
transit_source_file = "input_transit.pylst"
telescope_source_file = "input_telescope.pylst"

```

```

tex_datei_name      = "Treur2015"

####
#####
## Start und Ende der Berechnungen
##
## Fuer Kalenderuebersicht muss ein ganzes Jahr (1.1. bis 31.12.) gewaehlt werden
#####
####
chosen_start_date   = '01-01-2015'      # Startdatum der Berechnung in DD-MM-YYYY
chosen_end_date     = '31-12-2015'     # Enddatum der Berechnung in DD-MM-YYYY

####
#####
## Laenge der out-of-transit phase vor und nach dem Transit die mitbeobachtet wird
## in Stunden
#####
####

len_of_OOT_phase    = 1.00              # Laenge der out-of-transit-phase in stunden
                    # default Transits: 1.00
                    # default DWARF: 0.50

####
#####
## Programm Flags zur berechnung von:
## two_flag        : Tiefe targets waehrend Normallichtphase zulassen
## three_flag      : hohe sonne waehrend Normallichtphase zulassen
## twothree_flag   : beides zulassen
## partial_flag    : partielle transits zulassen (minimum beobachtung eines halben transits
)
#####
####
#two_flag          = True               # Transits finden, die waehrend Normallicht
                    manchmal tiefer sind als min_target_elev
two_flag           = False              # Transits finden, die auch waehrend Normallicht
                    nie tiefer sind als min_target_elev

#three_flag        = True               # Transits finden, bei denen die Sonne waehrend
                    Normallicht manchmal hoeher ist als max_sun_elev
three_flag         = False              # Transits finden, bei denen die Sonne auch
                    waehrend Normallicht nie hoeher ist als max_sun_elev

#twothree_flag     = True               # Transits finden, bei denen beide o.g. Punkte
                    zutreffen
twothree_flag      = False              # Transits finden, bei denen beide o.g. Punkte
                    nicht zutreffen

#partial_flag      = True               # Auf partielle Transits finden
partial_flag       = False              # Auf partielle Transits nicht finden

####
#####
## Kalenderuebersicht aktivieren / deaktivieren
#####
####

printcalendar      = True              # True / False # Soll ein Kalender gedruckt werden (nur fÄijr
                    ein observatorium empfehlenswert
#printcalendar     = False              # True / False # Soll ein Kalender gedruckt werden (nur
                    fÄijr ein observatorium empfehlenswert
timezonetime       = 1                  # Anzahl der stunden die noch abgezogen werden mÄijssen fÄijr
                    lokale zeit (Dtl: 1, asien: 6, USA: -6)
#Monate=["Januar","Februar","MÄdrz","April","Mai","Juni","Juli","August","September",
"Oktober","November","Dezember"]
Monate=["January","February","March","April","May","June","July","August","September",
"October","November","December"]

```

```

####
#####
## Ausgabestatistiken anpassen
#####
####

    # Anzahl der Sterne bzw. Observatorien nebeneinander in der Statistik
cStarCount=4          # default: 5; Dwarf: 3-4
cObsCount=4          # default: 4
    # Anzahl der Transits pro Seite
cTransitsPerPage=45  # default: 45
    # Anzahl der Transitobjekte pro Spalte in der bersicht
cTransitsPerColumn=14 # default: 10
    # Anzahl der Teleskope pro Spalte in der bersicht
cTelecopesPerColumn=6 # default: 5

####
#####
## Benutzerdefinierte Mindesthoehen
#####
####

U_min_target_elev    = False          # Benutzerdefinierte Minimale Targethoehe
    waehlen (default: 30)? True/False
min_target_elev     = 30              # Benutzerdefinierte minimale Targethoehe [grad]
####
U_max_sun_elev      = False          # Benutzerdefinierte maximale Sonnenhoehe
    waehrend Normalllicht waehlen (default: -15)? True/False
    # astronomische nacht: -18 ; nautische Nacht: -12; buergerliche Nacht:
    -6
max_sun_elev        = -15            # Benutzerdefinierte maximale Sonnenhoehe [grad]
####
U_max_sun_elev_transit = False      # Benutzerdefinierte maximale Sonnenhoehe
    waehrend Transit waehlen (default: -18)? True/False
max_sun_elev_transit = -18          # Benutzerdefinierte maximale Sonnenhoehe waehrend
    Transit [grad]
####
U_moon_distance     = False          # Benutzerdefinierte minimale Mondddistanz
    waehlen (default: 50)? True/False
moon_distance       = 50

#####
#####
#####
## specific input data end
#####
#####
#####
#####

#####
#####
## loading of telescope and target file lists
#####
#####

target_list=[]
tele_list=[]
try:

```

```

telescope_file=open(telescope_source_file,'r')
for line in telescope_file:
    temp_telescope_item=re.split("\t+",line)
    if temp_telescope_item[0]=="Teleskop>>":
        temp_telescope = Telescope(temp_telescope_item[1],temp_telescope_item[2],
            temp_telescope_item[3],temp_telescope_item[4],temp_telescope_item[5],
            temp_telescope_item[6])
        tele_list.append(temp_telescope)
telescope_file.close()
except:
    print "Error_on_importing_Telescope_Source_File"
    exit(1)
try:
target_file=open(transit_source_file,'r')
for line2 in target_file:
    temp_target_item=re.split("\t+",line2)
    if temp_target_item[0]=="Target>>":
        temp_target = Target(temp_target_item[1],temp_target_item[2],temp_target_item[3],
            temp_target_item[4],temp_target_item[5],temp_target_item[6],temp_target_item
            [7])
        target_list.append(temp_target)
target_file.close()
except:
    print "Error_on_importing_Object_Source_File"
    exit(1)
if len(tele_list)==0:
    print "No_valid_observatory_entry_found!_Program_terminated"
    exit(1)
if len(target_list)==0:
    print "No_valid_target_entry_found!_Program_terminated"
    exit(1)
#####
#####
#####
## configuration of calender printing tool
#####
#####
#####

printjahr=chosen_end_date[6:]
if printcalendar:
    # if not (chosen_start_date[0:5]=="01-01" and chosen_end_date[0:5]=="31-12"):
    # print "Kalender soll gedruckt werden, aber kein ganzes Jahr ausgewdhlt! Diese
    # FunktionalitÃd't steht nur fÃd'jr Jahresplots zur VerfÃd'gung"
    # time.sleep(3)
    # printcalendar=False
    if not ((int(chosen_end_date[6:])-int(chosen_start_date[6:])==0)):
        print "Bitte_nur_ein_Jahr_zum_Drucken_wÃd'hlen,_da_sonst_die_Ãd'bersichtlichkeit_
        verloren_geht."
        time.sleep(3)
    printcalendar=False
if ((len(tele_list)>1) and printcalendar):
    print "Bitte_nur_ein_Teleskop_wÃd'hlen,_da_sonst_die_Ãd'bersichtlichkeit_verloren_geht."
    time.sleep(3)
    printcalendar=False
if ((len(target_list)>5) and printcalendar):
    print "Bitte_weniger_Targets_wÃd'hlen,_da_sonst_die_Ãd'bersichtlichkeit_verloren_geht."
    time.sleep(3)

#####
#####
#####
## set standard values if not given
#####
#####
#####

```

```
#####

if not(tex_datei_name[-4:]==" .tex"):
    tex_datei_name+=".tex"
if not U_min_target_elev:
    min_target_elev = 30
if not U_max_sun_elev:
    max_sun_elev = -15
if not U_max_sun_elev_transit:
    max_sun_elev_transit = -18
if not U_moon_distance:
    moon_distance = 50

targets_not_found = []
transits = []
target_list_clearname = []
tele_list_clearname = []
TransitsPerSter = {}
numberobservatories = {}
tex_observatorylist = '\\begin{tabular}{'+l*(my_astro.div_ceil(len(tele_list),
    cTelescopesPerColumn))+'}\n'
tex_starlist = '\\begin{tabular}{'+l*(my_astro.div_ceil(len(target_list),
    cTransitsPerColumn))+'}\n'
cyclenumber = 0
telescope_number = 0
len_of_00T_phase_days = len_of_00T_phase*0.0416667 # laenge der out-of-transit-phase
    in tagen

for tele_choice in range(len(tele_list)):
    cyclenumber += 1
    telescope_number += 1
    telescope = ephem.Observer()
    telescope.lat = (tele_list[tele_choice]).get_Latitude()
    telescope.long = (tele_list[tele_choice]).get_ELongitude()
    telescope.elevation = (tele_list[tele_choice]).get_Elevation()
    telescope.name = (tele_list[tele_choice]).get_Name()
    telescope.temp = telescope_number
    telescope.pressure = (tele_list[tele_choice]).get_Pressure()
    numberobservatories[telescope.name]=0

    if telescope.name not in tele_list_clearname:
        tele_list_clearname.append(telescope.name)

    print '----->_Berechne_Transits_in_' + telescope.name

    for target_choice in range(len(target_list)):
        target = {}
        target['name'] = (target_list[target_choice]).get_Name()
        target['ra_sex'] = (target_list[target_choice]).get_RA()
        target['dec_sex'] = (target_list[target_choice]).get_DEC()
        target['T_0'] = (target_list[target_choice]).get_T0()
        target['P'] = (target_list[target_choice]).get_Period()
        target['d'] = (target_list[target_choice]).get_Duration()
        targetfile = ephem.readdb(target['name']+',f|S|??,'+target['ra_sex'] +'|0.0,'
            +target['dec_sex']+'|0.0,0.0,2000,0')
        if cyclenumber==1:
            TransitsPerSter[target['name']]=0
        print target['name']
        if not (target['name'] == 'No_Target') and (target['name'] not in
            target_list_clearname):
            target_list_clearname.append(target['name'])

#####
## erzeuge Sonne, Mond und Planeten-Objekte
#####
```

```

sun = ephem.Sun()
moon = ephem.Moon()
#####
startdate_UT = my_astro.string_to_date(chosen_start_date)
enddate_UT = my_astro.string_to_date(chosen_end_date)
UT_Time = my_astro.string_to_time('00:00:00')
startdate_JD = my_astro.gregdate_to_juldate(startdate_UT,UT_Time)['jd']
enddate_JD = my_astro.gregdate_to_juldate(enddate_UT,UT_Time)['jd']
p_start = math.floor((startdate_JD-target['T_0'])/target['P'])
p_end =math.floor((enddate_JD-target['T_0'])/target['P'])
if cyclenumber==len(tele_list):
    first = my_astro.juldate_to_gregdate(target['T_0']+target['P']*int(p_start))
    firstUT = datetime.datetime(first['date'][2], first['date'][1], first['date'][0],
        first['time'][0],first['time'][1],int(first['time'][2]))
    last = my_astro.juldate_to_gregdate(target['T_0']+target['P']*int(p_end+1))
    lastUT = datetime.datetime(last['date'][2], last['date'][1], last['date'][0],
        last['time'][0],last['time'][1],int(last['time'][2]))
for i in range(int(p_start),int(p_end+1)):
    T_mid = target['T_0']+target['P']*i
    T_mid_UT = my_astro.juldate_to_gregdate(T_mid)
    T_start=T_mid-0.04167*target['d']/2
    T_start_UT=my_astro.juldate_to_gregdate(T_start)
    T_begin=T_start-len_of_00T_phase_days
    T_begin_UT=my_astro.juldate_to_gregdate(T_begin)
    T_stop=T_mid+0.04167*target['d']/2
    T_stop_UT=my_astro.juldate_to_gregdate(T_stop)
    T_end=T_stop+len_of_00T_phase_days
    T_end_UT=my_astro.juldate_to_gregdate(T_end)

    date = datetime.datetime(T_mid_UT['date'][2], T_mid_UT['date'][1], T_mid_UT['date']
        ] [0],T_mid_UT['time'][0],T_mid_UT['time'][1],int(T_mid_UT['time']
        ] [2]))
    telescope.date=date
    sun.compute(telescope)
    sun_height_mid = math.degrees(float(sun.alt))
    targetfile.compute(telescope)
    target_height_mid = math.degrees(float(targetfile.alt))

    date = datetime.datetime(T_begin_UT['date'][2], T_begin_UT['date'][1], T_begin_UT[
        'date'] [0],T_begin_UT['time'][0],T_begin_UT['time'][1],int(T_begin_UT['time']
        ] [2]))
    telescope.date=date
    sun.compute(telescope)
    sun_height_begin = math.degrees(float(sun.alt))
    targetfile.compute(telescope)
    target_height_begin = math.degrees(float(targetfile.alt))

    date = datetime.datetime(T_start_UT['date'][2], T_start_UT['date'][1], T_start_UT[
        'date'] [0],T_start_UT['time'][0],T_start_UT['time'][1],int(T_start_UT['time']
        ] [2]))
    telescope.date=date
    sun.compute(telescope)
    sun_height_start = math.degrees(float(sun.alt))
    targetfile.compute(telescope)
    target_height_start = math.degrees(float(targetfile.alt))

    date = datetime.datetime(T_stop_UT['date'][2], T_stop_UT['date'][1], T_stop_UT['
        date'] [0],T_stop_UT['time'][0],T_stop_UT['time'][1],int(T_stop_UT['time']
        ] [2]))
    telescope.date=date
    sun.compute(telescope)
    sun_height_stop = math.degrees(float(sun.alt))
    targetfile.compute(telescope)
    target_height_stop = math.degrees(float(targetfile.alt))

    date = datetime.datetime(T_end_UT['date'][2], T_end_UT['date'][1], T_end_UT['date']
        ] [0],T_end_UT['time'][0],T_end_UT['time'][1],int(T_end_UT['time']
        ] [2]))

```

```

telescope.date=date
sun.compute(telescope)
sun_height_end = math.degrees(float(sun.alt))
targetfile.compute(telescope)
target_height_end = math.degrees(float(targetfile.alt))

moon.compute(telescope)
fullmoondistance=int(moon.phase)
moondist=math.degrees(ephem.separation(moon, targetfile))
local_flag = "indef"
if ((sun_height_begin<max_sun_elev_transit)&(sun_height_end<max_sun_elev_transit)
    &(moondist>moon_distance) & (target_height_begin>min_target_elev)& (
    target_height_end>min_target_elev)):
    local_flag = "1" #no restrictions
elif two_flag & ((sun_height_begin<max_sun_elev_transit)&(sun_height_end<
    max_sun_elev_transit)&(moondist>moon_distance) & (target_height_start>
    min_target_elev) & (target_height_stop>min_target_elev)):
    local_flag = "2" #target too low at normal light
elif three_flag & ((sun_height_start<max_sun_elev_transit) & (sun_height_stop<
    max_sun_elev_transit) & (sun_height_begin<max_sun_elev) & (sun_height_begin<
    max_sun_elev) & (moondist>moon_distance) & (target_height_begin>
    min_target_elev) & (target_height_end>min_target_elev)):
    local_flag = "3" #sun too high at normal light
elif twothree_flag & ((sun_height_start<max_sun_elev_transit) & (sun_height_stop<
    max_sun_elev_transit) & (sun_height_begin<max_sun_elev) & (sun_height_begin<
    max_sun_elev) & (moondist>moon_distance) & (target_height_start>
    min_target_elev) & (target_height_stop>min_target_elev)):
    local_flag = "2,3" #target too low at normal light and sun to high at normal
    light
elif partial_flag & ((moondist>moon_distance) & ((sun_height_mid<max_sun_elev)&(
    target_height_mid>max_sun_elev)) & ((sun_height_begin<min_target_elev)&(
    target_height_begin>min_target_elev))):
    local_flag = "P/B" #partial beginning
elif partial_flag & ((moondist>moon_distance) & ((sun_height_mid<max_sun_elev)&(
    target_height_mid>min_target_elev)) & ((sun_height_end<max_sun_elev)&(
    target_height_end>min_target_elev))):
    local_flag = "P/E" #partial end
if (local_flag != "indef") & (T_begin > startdate_JD) & (T_end < enddate_JD):
    Trans=Transit(target['name'], T_begin, T_end, telescope.name, moondist,
        fullmoondistance, target_height_start, target_height_stop, sun_height_start,
        sun_height_stop, local_flag)
    transits.append(Trans)
tex_kalender=''
if printcalendar:
    KalenderVektor=[]

    for i in range(31):
        KalenderVektor.append(['']*12)

for j in (range(cTelecopesPerColumn)):
    for i in range(my_astro.div_ceil(len(tele_list_clearname),cTelecopesPerColumn)):
        aktuellesTeleskop=i*cTelecopesPerColumn+j
        if (aktuellesTeleskop<(len(tele_list_clearname))):
            if i < (my_astro.div_ceil(len(tele_list_clearname),cTelecopesPerColumn)-1):
                tex_observatorylist=tex_observatorylist+tele_list_clearname[aktuellesTeleskop]+'
                &'
            else:
                tex_observatorylist=tex_observatorylist+tele_list_clearname[aktuellesTeleskop]+'
                '\\\\n'
        else:
            tex_observatorylist=tex_observatorylist+'\\\\n'

tex_observatorylist=tex_observatorylist+'\\end{tabular}'

for j in (range(cTransitsPerColumn)):
    for i in range(my_astro.div_ceil(len(target_list_clearname),cTransitsPerColumn)):

```

```

aktuellerTransit=i*cTransitsPerColumn+j
if (aktuellerTransit<(len(target_list_clearname))):
    if i < (my_astro.div_ceil(len(target_list_clearname),cTransitsPerColumn)-1):
        tex_starlist=tex_starlist+target_list_clearname[aktuellerTransit]+'&'
    else:
        tex_starlist=tex_starlist+target_list_clearname[aktuellerTransit]+'\\\\\\n'
    else:
        tex_starlist=tex_starlist+'\\\\\\n'
tex_starlist=tex_starlist+'\\end{tabular}'

texfile_raw=''

if len(transits)==0:
    print "no_transits_found"
elif len(transits)==1:
    print "1_transit_found"
if (len(transits)>0):
    texfile=open(tex_dateiname, 'w')
    texfile.write('\\documentclass[a4paper, 10pt, oneside]{article}\\n\\n\\usepackage[utf8x]{
    inputenc}\\n\\usepackage{lscape}\\n\\usepackage[usenames, dvipsnames]{color}\\n\\n\\
    usepackage{graphicx, a4wide, array}\\n\\n\\pagestyle{empty}\\n\\n\\parindent_0pt\\n\\n\\
    usepackage{threeparttable}\\n\\n\\usepackage{booktabs}\\n\\n\\usepackage[margin=1.5cm]{
    geometry}\\n\\n\\date{'+str(datetime.date.today())+'}\\n\\n\\pdfinfo{%n/Title_(Transit
    _predictions)_n_/Author_(C)_n_/Creator_(C)_n_/Producer_(C)_n_/Subject_(C)_n_/
    Keywords_(C)_n}\\n\\n\\begin{document}\\n\\n')

    transits=sorted(transits, key=lambda transits: transits.beginINT)

    numbertransits=0
    totalnumber=len(transits)
    for i in range(0, (len(transits)), cTransitsPerPage):
        if len(tele_list)==1:
            texfile_raw=texfile_raw + ('\\newpage\\n\\emph{start/end}_of_monitoring_(incl.'+"
            %1.2f"%len_of_00T_phase+'h_normal_light)_in_UT;_\\emph{moon:}_moondistance_[$
            ^\\circ$]_to_target;_\\emph{sun/target:}_altitudes_of_sun/target_at_transit-
            start/-end;_\\textit{Flags:}_1_--_no_restrictions, _target_at_high_elevations, _
            transit_within_astronomical_night')
            if two_flag or twothree_flag:
                texfile_raw=texfile_raw + (';_2_--_target_below_'+"%1.2f"%min_target_elev+'
                degree_in_normal_light_phase')
            if three_flag or twothree_flag:
                texfile_raw=texfile_raw + (';_3_--_sun_above_'+"%1.2f"%max_sun_elev_transit+'
                degree_in_normal_light_phase')
            if partial_flag:
                texfile_raw=texfile_raw + (';_P/E_or_P/B_--_only_half_of_transit_visible_
                including_beginning_(B)_or_end_(E)_n')
            texfile_raw=texfile_raw + ('\\begin{table}[!ht]\\n\\centering\\n\\begin{tabular}{
            lllr@{/}rlllc}\\n\\toprule\\nname_&_start_(UT)_&_end_(UT)_&_\\multicolumn{2}{l}{
            moon}&_&_sun_&_target_&_Flag\\\\\\n\\midrule\\n')
        else:
            texfile_raw=texfile_raw + ('\\newpage\\n\\emph{start/end}_of_monitoring_(incl.'+"
            %1.2f"%len_of_00T_phase+'h_normal_light)_in_UT;_\\emph{moon:}_moondistance_[$
            ^\\circ$]_to_target;_\\emph{sun/target:}_altitudes_of_sun/target_at_transit-
            start/-end;_\\textit{Flags:}_1_--_no_restrictions, _target_at_high_elevations, _
            transit_within_astronomical_night')
            if two_flag or twothree_flag:
                texfile_raw=texfile_raw + (';_2_--_target_below_'+"%1.2f"%min_target_elev+'
                degree_in_normal_light_phase')
            if three_flag or twothree_flag:
                texfile_raw=texfile_raw + (';_3_--_sun_above_'+"%1.2f"%max_sun_elev_transit+'
                degree_in_normal_light_phase')
            if partial_flag:
                texfile_raw=texfile_raw + (';_P/E_or_P/B_--_only_half_of_transit_visible_
                including_beginning_(B)_or_end_(E)_n')
            texfile_raw=texfile_raw + ('\\begin{table}[!ht]\\n\\centering\\n\\begin{tabular}{

```



```

llllr@{/}rlllc}\n\toprule\name_\&_start_(UT)\&_end_(UT)\&_Observatory_\&_
multicolumn{2}{1}{moon}\&_sun_\&_target_\&_Flag\\\n\midrule\n')
for j in range (i,i+cTransitsPerPage):
    if j < len(transits):
        tname = str(transits[j].name)
        tstartUT=my_astro.juldate_to_gregdate(transits[j].begin)
        if printcalendar:
            calendertime=my_astro.juldate_to_gregdate(transits[j].begin-0.5-timezonetime
                /24)
        if tstartUT['date'][0]<10:
            tstartUT['date'][0]='0'+str(tstartUT['date'][0])
        if tstartUT['date'][1]<10:
            tstartUT['date'][1]='0'+str(tstartUT['date'][1])
        tstartstr =str(tstartUT['date'][0])+'.'+str(tstartUT['date'][1])+'.'+str(
            tstartUT['date'][2])+ '/' +str(datetime.time(tstartUT['time'][0],tstartUT['
            time'][1],int(tstartUT['time'][2])))
        tendUT=my_astro.juldate_to_gregdate(transits[j].end)
        if tendUT['date'][0]<10:
            tendUT['date'][0]='0'+str(tendUT['date'][0])
        if tendUT['date'][1]<10:
            tendUT['date'][1]='0'+str(tendUT['date'][1])
        tendstr =str(tendUT['date'][0])+'.'+str(tendUT['date'][1])+'.'+str(tendUT['date'
            ][2])+ '/' +str(datetime.time(tendUT['time'][0],tendUT['time'][1],int(tendUT['
            time'][2])))
        tObservatory = str(transits[j].observatory)
        tmoon =str(int(transits[j].moon))
        tmoonphase=str(transits[j].fullmoondistance)
        tsun=str(int(transits[j].sun_start))+'.'+str(int(transits[j].sun_stop))
        ttarget=str(int(transits[j].el_start))+'.'+str(int(transits[j].el_stop))
        tflag=str(transits[j].flag)
        if len(tele_list)==1:
            texfile_raw=texfile_raw + (tname + '_&' + tstartstr + '_&' + tendstr + '_&'
                + tmoon + '$^\\circ$_&' + tmoonphase+ '\\%$_&' + tsun + '_&' + ttarget
                + '_&' + tflag)
            if printcalendar and (int(calendertime['date'][2])==int(chosen_end_date[6:])):
                temp_day=int(calendertime['date'][0])-1
                temp_month=int(calendertime['date'][1])-1
                if KalenderVektor[temp_day][temp_month]=='':
                    KalenderVektor[temp_day][temp_month]=tname
                else:
                    KalenderVektor[temp_day][temp_month]+=", "+tname
        else:
            texfile_raw=texfile_raw + (tname + '_&' + tstartstr + '_&' + tendstr + '_&'
                + tObservatory + '_&' + tmoon + '$^\\circ$_&' + tmoonphase+ '\\%$_&' +
                tsun + '_&' + ttarget + '_&' + tflag)
        numberobservatories[tObservatory]=numberobservatories[tObservatory]+1
    if ((j<(len(transits)-1))&(j!=i+cTransitsPerPage-1)):
        if ((transits[j].name!=transits[j+1].name) or (transits[j].begin!=transits[j
            +1].begin)):
            texfile_raw=texfile_raw + ('\\vspace*{0.1cm}\\\n')
            numbertransits=numbertransits+1
            TransitsPerSter[transits[j].name]=TransitsPerSter[transits[j].name]+1
        else:
            texfile_raw=texfile_raw + ('\\\n')
    elif ((j<(len(transits)-1))):
        if ((transits[j].name!=transits[j+1].name) or (transits[j].begin!=transits[j
            +1].begin)):
            texfile_raw=texfile_raw + ('\\\n')
            numbertransits=numbertransits+1
            TransitsPerSter[transits[j].name]=TransitsPerSter[transits[j].name]+1
        else:
            texfile_raw=texfile_raw + ('\\\n')
    else:
        texfile_raw=texfile_raw + ('\\\n')
        TransitsPerSter[transits[j].name]=TransitsPerSter[transits[j].name]+1
        numbertransits=numbertransits+1

```

```

    texfile_raw=texfile_raw + ('\\bottomrule\\end{tabular}\\end{table}\\n')
print 'total_number_of_transits:' + str(totalnumber)
print 'number_of_different_transits:' + str(numbertransits)
texfile.write('\\parbox[c][2cm][c]{\\textwidth}{\\begin{center}\\n\\textbf{\\huge_
Transit_predictions}\\n\\end{center}\\n}\\n\\parbox[c][5cm][c]{\\textwidth}{\\n\\
begin{center}\\n{\\em_for_the_stars}\\n\\vspace*{0.5cm}\\n{\\small_n' +
tex_starlist + '\\n}\\n\\end{center}\\n}\\n\\parbox[c][9cm][c]{\\textwidth}{\\begin
{center}\\n{\\em_at_the_telescopes}\\n\\vspace*{0.5cm}\\n}\\n' + tex_observatorylist+
\\n\\end{center}\\n}\\vspace*{0.5cm}\\n\\begin{center}\\n_covering_the_time_+
chosen_start_date + '_till_' + chosen_end_date+ '\\n\\end{center}\\n\\n\\vspace*{0.5cm
}\\n\\n')
texfile.write('\\emph{statistics:_number_of_visible_transits_(different_/total):_'+
str(numbertransits)+ '/' + str(totalnumber)+'\\n\\n')
texfile.write('\\vspace*{0.5cm}\\n\\n\\emph{single_star_statistics:}')
texfile.write('\\n\\begin{table}[!h]\\n\\centering\\n\\begin{tabular}{*'+str(cStarCount
-1)+'{ll}|ll}\\n\\hline\\n')
count=0
for k, v in TransitsPerSter.iteritems():
    count=count+1
    texfile.write(k + ':' + str(v))
    if count==cStarCount:
        texfile.write('\\\\\\n')
        count=0
    if count>0:
        texfile.write('&')
texfile.write('\\\\\\n\\end{tabular}\\n\\end{table}\\n')

texfile.write('\\begin{table}[!h]\\n\\centering\\n\\begin{tabular}{*'+str(cObsCount-1)+'
{ll}|ll}\\n\\hline\\n')
count=0
for h, j in numberobservatories.iteritems():
    count=count+1
    texfile.write(h + ':' + str(j))
    if count==cObsCount:
        texfile.write('\\\\\\n')
        count=0
    if count>0:
        texfile.write('&')
texfile.write('\\\\\\n\\end{tabular}\\n\\end{table}\\n\\n')

texfile.write(texfile_raw)
if (printcalendar and (len(tele_list)==1)):
    texfile.write('\\n\\newpage\\n\\definecolor{dark-gray}{gray}{0.50}\\begin{
landscape}\\n\\begin{center}\\nCalendar_overview_for_' + telescope.name+'_' +
chosen_end_date[6:] + '\\n\\end{center}\\n\\renewcommand{\\arraystretch}{2}\\n\\
begin{table}[!ht]\\n\\centering\\n\\tiny\\n\\begin{tabular}{*{12}{|p{0.2cm}|p{1.05
cm}|}}\\n\\hline\\n')
    texfile.write('\\multicolumn{2}{|c|}{\\footnotesize{'+Monate[0]+'}}')
    for i in range(1,11):
        texfile.write('\\multicolumn{2}{c|}{\\footnotesize{'+Monate[i]+'}}&')
    texfile.write('\\multicolumn{2}{c|}{\\footnotesize{'+Monate[11]+'}}\\\\\\n\\hline')
    for tag in range(31):
        for monat in range(11):
            kalendertag=""
            try:
                if (datetime.datetime(int(chosen_end_date[6:]),int(monat+1), int(tag+1)).
                    weekday()>4):
                    kalendertag="\\textbf{\\textcolor{dark-gray}{'+str(tag+1)+'}}'"
                else:
                    kalendertag=str(tag+1)
            except:
                kalendertag=str(tag+1)
        if monat in [0,2,4,6,7,9]:
            texfile.write("\\footnotesize{'+kalendertag+'}&"+KalenderVektor[tag][monat]+'&
")
        elif monat in [3,5,8,10]:

```

```

    if tag==30:
        texfile.write("\\multicolumn{2}{c|}{}&")
    else:
        texfile.write("\\footnotesize{"+kalendertag+"}&"+KalenderVektor[tag][monat]+
"&")
elif monat == 1:
    if tag >=29:
        texfile.write("\\multicolumn{2}{c|}{}&")
    elif tag ==28:
        if not (my_astro.test_schaltjahr(int(tstartUT['date'][2]))):
            texfile.write("\\multicolumn{2}{c|}{}&")
        else:
            texfile.write("\\footnotesize{"+kalendertag+"}&"+KalenderVektor[tag][monat
]+"&")
    else:
        texfile.write("\\footnotesize{"+kalendertag+"}&"+KalenderVektor[tag][monat]+
"&")
    texfile.write("\\footnotesize{"+kalendertag+"}&"+KalenderVektor[tag][11]+"\\\\\\
hline\\n")
    texfile.write('\\n\\end{tabular}\\n\\end{table}\\n\\end{landscape}')
texfile.write('\\end{document}')
del transits
texfile.close()
print 'erstelle_pdf-datei:_' + tex_datei_name[:-3] + 'pdf'
run ('pdflatex_' + tex_datei_name)
run ('rm_' + tex_datei_name[:-3] + 'aux')
run ('rm_' + tex_datei_name[:-3] + 'log')
os.system('chmod_f_g+w_' + tex_datei_name[:-3] + '*')
if len(targets_not_found)>0:
    print 'Targets_nicht_gefunden:_' + str(targets_not_found)
os.system("cp_predictions.py_/astro/p5sema2/Programs/python/TTV/Predictions/Version2/
program_safes/predictions_" + time.strftime("%y-%m-%d_%H-%M-%S", time.localtime()) + ".
py")
os.system("bzip2_/astro/p5sema2/Programs/python/TTV/Predictions/Version2/program_safes/
predictions*.py")
print 'Habe_fertig'

```

B.2 Programme zur N-Körper Simulation

Quellcode B.3: Hilfsprogramm zur Definition von Klassen und Funktionen

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : OCcalculations.py
## Version: 0.1 (alpha)
## Zweck : O-C Calculations of Transiting planets
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import time
import math
import random
import thread
from routines import *

#Parameters to be used for param.in
param={}
param[1] ="BS"          # algorithm (MVS, BS, BS2, RADAU, HYBRID)
param[2] ="2451000.500" # start time in days
param[3] ="2451000.500" # stop time (days)
tmpparam =10           # outout interval (minutes)
param[4] =str(round(0.0006944*tmpparam,3)) # output interval (days) as needes by
program
param[5] =param[4]      # timestep (days)
param[6] ="1.d-12"     # accuracy parameter

param[7] ="yes"        # stop integration after a close encounter
param[8] ="no"         # allow collisions to occur
param[9] ="no"         # include collisional fragmentation
param[10]="days"      # express time in days or years
param[11]="no"         # express time relative to integration start time
param[12]="high"       # output precision (high, medium, low)
param[13]="no"         # include relativity in integration
param[14]="no"         # include user-defindes force

param[15]="100"        # ejection distance [AU]
param[16]="0.00567"    # radius of central body [AU]
param[17]="1.16"       # central mass [solar masses]
param[18]="0"          # central J2
param[19]="0"          # central J4
param[20]="0"          # central J6
param[21]="3"          # Hybrid integrator changeover [Hill radii]
param[22]="1000"       # number of timesteps between data dumps
param[23]="100"        # number of timesteps between periodic effects

#Parameters to be used for big.in
big={}
big[1] ="Ast"          # style (Cartesian, Asteroidal, Cometary)
big[2] ="2451000.5"    # epoch [days]

```

ANHANG B. QUELLCODES

```

#first known planet
tmpbig = "HatP32b"          # 1st Planet name
big[3] = (tmpbig.replace("_","")).ljust(8,"_") # 1st Planet name as needed by program
big[4] = "8.17001226e-04"   # 1st Planet mass in solar masses
big[5] = "4.d0"           # 1st Planet maximum close encounter distance in Hill radii
big[6] = "0.19"           # 1st Planet density in g/cm^3

big[7] = "0.0343"         # 1st planet semi major axis [AU]
big[8] = "0.000"         # 1st planet eccentricity
big[9] = "88.9"          # 1st planet inclination [degree]
big[10] = "000.0"        # 1st planet argument of pericentre [degrees]
big[11] = "0"            # 1st planet longitude of ascending node [degrees]
big[12] = "0"            # 1st planet mean anomaly [degrees]
big[13] = "0"            # 1st planet 1st spin angular momentum
big[14] = "0"            # 1st planet 2nd spin angular momentum
big[15] = "0"            # 1st planet 3rd spin angular momentum

#second unknown planet

tmpbig = "HatP32c"        # 2nd Planet name
big[16] = (tmpbig.replace("_","")).ljust(8,"_") # 2nd Planet name as needed by program
big[17] = "1.84870277E-03" # 2nd Planet mass in solar masses
big[18] = "4.d0"          # 2nd Planet maximum close encounter distance in Hill radii
big[19] = "1.00"          # 2nd Planet density in g/cm^3

big[20] = "0.0964"        # 2nd planet semi major axis [AU]
big[21] = "0.000"         # 2nd planet eccentricity
big[22] = "88.9"          # 2nd planet inclination [degree]
big[23] = "000.0"        # 2nd planet argument of pericentre [degrees]
big[24] = "0"            # 2nd planet longitude of ascending node [degrees]
big[25] = "0"            # 2nd planet mean anomaly [degrees]
big[26] = "0"            # 2nd planet 1st spin angular momentum
big[27] = "0"            # 2nd planet 2nd spin angular momentum
big[28] = "0"            # 2nd planet 3rd spin angular momentum

fileparam={}
fileparam[1]="big.in"      # big.in-filename
fileparam[2]="small.in"   # small.in filename
fileparam[3]="param.in"   # param.in filename
fileparam[4]="xv.out"      # xv.out filename
fileparam[5]="ce.out"      # ce.out filename
fileparam[6]="info.out"    # info.out filename
fileparam[7]=(fileparam[1])[:-2]+"dmp" # big.dmp filename
fileparam[8]=(fileparam[2])[:-2]+"dmp" # big.dmp filename
fileparam[9]=(fileparam[3])[:-2]+"dmp" # big.dmp filename
fileparam[10]="restart.dmp" # restart.dmp filename

#Parameterfile for element.in
elementparam={}
elementparam[1]=fileparam[4] # input file name
elementparam[2]="Cen"        # type of elements (central body, barycentric, Jacobi)
elementparam[3]=param[4]     # minimum interval between outputs [days]
elementparam[4]="days"      # express time in days or years
elementparam[5]="yes"        # express time relative to integration start time
elementparam[6]="x8.4_a8.4" # output format
elementparam[7]=big[3]       # body name

numberepochs=5000 #100 #number of epochs to calculate
additionalepochs=10
param[3] =str(2451000.5+365.25*(numberepochs+additionalepochs)*math.sqrt(float(big[7])*
float(big[7])*float(big[7])*(float(param[17])+float(big[4]))))[0:11]
# stop time (days)

massstep=[2,6,1] #[1,3000,5] # massrange and step in M_earth (5 -> 5M_earth) start
at 1M_earth, end at 3000Mearth=1Mjup, step 5Mearth
austep=[2,3] #[20,100] # au step in 0,0001AU units (20 -> 0.002AU; minimum: 2) and

```

```

        maximum Au in number of AU of known planet (should be ~factor 10)
        # inner au step is half this value
pfad=os.getcwd()+"/"
if len(sys.argv)<3:
    print "mit_(Mercury6_/_ptmet)_und_Anzahl_der_Kerne_starten"
    exit(1)

numbercores=int(sys.argv[2])
print "Anzahl_kerne:_" +str(numbercores)
missing_files=""
if sys.argv[1]=="Mercury6":
    try:
        file=open('mercury6','r')
        file.close()
    except:
        missing_files=missing_files+"mercury6,_"
    try:
        file=open('element6','r')
        file.close()
    except:
        missing_files=missing_files+"element6,_"
    try:
        file=open('message.in','r')
        file.close()
    except:
        missing_files=missing_files+"message.in,_"
    if missing_files=="":
        missing_files="all_required_files_exist"
#    print missing_files
    MainMercuryProgram(elementparam, fileparam, big, param, numbercores, massstep, austep, pfad
        , numberepochs+additionalepochs)
else:
    missing_files="Missing_files:_" +missing_files[:-2] +"_-_Program_terminated"
    print missing_files
    exit(1)
elif sys.argv[1]=="ptmet":
    try:
        file=open('ptmet','r')
        file.close()
    except:
        missing_files=missing_files+"ptmet,_"
    try:
        file=open('asc2bin','r')
        file.close()
    except:
        missing_files=missing_files+"asc2bin,_"
    try:
        file=open('coef_01.br4','r')
        file.close()
    except:
        missing_files=missing_files+"coef_01.br4,_"
    try:
        file=open('coef_01.dat','r')
        file.close()
    except:
        missing_files=missing_files+"coef_01.dat,_"

    if missing_files=="":
        missing_files="all_required_files_exist"
#    print missing_files
    else:
        missing_files="Missing_files:_" +missing_files[:-2] +"_-_Program_terminated"
        print missing_files
        exit(1)

else:

```

```
    print "mit_(Mercury6_/_lptmet)_und_Anzahl_der_Kerne_starten"  
    exit(1)  
#Mercury6 startbefehl  
MainMercuryProgram(elementparam, fileparam, big, param, numbercores, massstep, astep, pfad,  
    numberepochs+additionalepochs)
```

Quellcode B.4: Programm zur Berechnung der theoretischen TTV Signale

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

import os
import sys
import time
import math
import random
import threading
import Queue
from scipy import stats

queueLock=threading.Lock()
stateLock=threading.Lock()
filesLock=threading.Lock()
elementLock=threading.Lock()
OCLock=threading.Lock()
printScreenLock=threading.Lock()
workQueue=Queue.Queue(10)
exitFlag=0
threadstatus=[]
prog_info=[]

def WriteParamIn(param, paramfilename):
    paramfile=open(paramfilename,"w")
    paramfile.write("0+_06_Integration_parameters_(WARNING:_Do_not_delete_this_line!)\n")
    paramfile.write(")_Lines_beginning_with_(')'_are_ignored.\n")
    paramfile.write(")-----\n")
    paramfile.write(")_Important_integration_parameters:\n")
    paramfile.write(")-----\n")
    paramfile.write("_algorithm_(MVS,_BS,_BS2,_RADAU,_HYBRID_etc)_____="+param[1] +"\n")
    paramfile.write("_start_time_(days)_____="+param[2] +"\n")
    paramfile.write("_stop_time_(days)_____="+param[3] +"\n")
    paramfile.write("_output_interval_(days)_____="+param[4] +"\n")
    paramfile.write("_time_steps_(days)_____="+param[5] +"\n")
    paramfile.write("_accuracy_parameter_____="+param[6] +"\n")
    paramfile.write(")-----\n")
    paramfile.write(")_Important_options:\n")
    paramfile.write(")-----\n")
    paramfile.write("_stop_integration_after_a_close_encounter_____="+param[7] +"\n")
    paramfile.write("_allow_collisions_to_occur_____="+param[8] +"\n")
    paramfile.write("_include_collisional_fragmentation_____="+param[9] +"\n")
    paramfile.write("_express_time_in_days_or_years_____="+param[10]+"\n")
    paramfile.write("_express_time_relative_to_integration_start_time_____="+param[11]+"\n")
    paramfile.write("_output_precision_____="+param[12]+"\n")
    paramfile.write("<<not_used_at_present>\n")
    paramfile.write("_include_relativity_in_integration_____="+param[13]+"\n")
    paramfile.write("_include_user-defined_force_____="+param[14]+"\n")
    paramfile.write(")-----\n")
    paramfile.write(")_These_parameters_do_not_need_to_be_adjusted_often:\n")
    paramfile.write(")-----\n")
    paramfile.write("_ejection_distance_[AU]_____="+param[15]+"\n")
    paramfile.write("_radius_of_central_body_[AU]_____="+param[16]+"\n")
    paramfile.write("_central_mass_[solar]_____="+param[17]+"\n")
    paramfile.write("_central_J2_____="+param[18]+"\n")
    paramfile.write("_central_J4_____="+param[19]+"\n")

```



```

paramfile.write("_central_J6_=====" + param[20] + "\n")
paramfile.write("<<not_used_at_present>\n")
paramfile.write("<<not_used_at_present>\n")
paramfile.write("_Hybrid_integrator_changeover_[Hill_radii]_=====" + param[21] + "\n")
paramfile.write("_number_of_timesteps_between_data_dumps_=====" + param[22] + "\n")
paramfile.write("_number_of_timesteps_between_periodic_effects_=====" + param[23] + "\n")
paramfile.close()
return 0

def WriteBigIn(big, bigfilename):
    bigfile=open(bigfilename, "w")
    bigfile.write("0+_06_Big-body_initial_data_((WARNING:_Do_not_delete_this_line!))\n")
    bigfile.write("_Lines_beginning_with_')_are_ignored.\n")
    bigfile.write(")-----\n")
    bigfile.write("_style_(Cartesian, _Asteroidal, _Cometary)_=_"+big[1]+"")
    bigfile.write("_epoch_(in_days)_=====" + big[2] + "\n")
    bigfile.write(")-----\n")
    bigfile.write(big[3] + "_m=" + big[4] + "_r=" + big[5] + "_d=" + big[6] + "\n")
    bigfile.write(big[7] + "_"+big[8] + "_"+big[9] + "_"+big[10] + "_"+big[11] + "_"+big[12] + "_"+big
    [13] + "_"+big[14] + "_"+big[15] + "\n")
    bigfile.write(big[16] + "_m=" + big[17] + "_r=" + big[18] + "_d=" + big[19] + "\n")
    bigfile.write(big[20] + "_"+big[21] + "_"+big[22] + "_"+big[23] + "_"+big[24] + "_"+big[25] + "_"+
    big[26] + "_"+big[27] + "_"+big[28] + "\n")
    bigfile.close()
    return 0

def WriteFilesIn (fileparam, filefilename):
    filefile=open(filefilename, "w")
    filefile.write("_"+fileparam[1]+"\n")
    filefile.write("_"+fileparam[2]+"\n")
    filefile.write("_"+fileparam[3]+"\n")
    filefile.write("_"+fileparam[4]+"\n")
    filefile.write("_"+fileparam[5]+"\n")
    filefile.write("_"+fileparam[6]+"\n")
    filefile.write("_"+fileparam[7]+"\n")
    filefile.write("_"+fileparam[8]+"\n")
    filefile.write("_"+fileparam[9]+"\n")
    filefile.write("_"+fileparam[10]+"\n")
    return 0

def WriteSmallIn (smallfilename):
    try:
        file=open(smallfilename, 'r')
        file.close()
    except:
        smallfile=open(smallfilename, 'w')
        smallfile=open(smallfilename, "w")
        smallfile.write("0+_06_Small-body_initial_data_((WARNING:_Do_not_delete_this_line
        !))\n")
        smallfile.write("_Lines_beginning_with_')_are_ignored.\n")
        smallfile.write(")-----\n")
        smallfile.write("_style_(Cartesian, _Asteroidal, _Cometary)_=_Ast\n")
        smallfile.write(")-----\n")
        smallfile.close()
    return 0

def WriteElementIn (elementparam, elementfilename):
    elementfile=open(elementfilename, "w")
    elementfile.write("0+_06_element_((WARNING:_Do_not_delete_this_line!))\n")
    elementfile.write("_Lines_beginning_with_')_are_ignored.\n")
    elementfile.write(")-----\n")
    elementfile.write("_number_of_input_files_=_1\n")

```

```

elementfile.write("")
-----\n")
elementfile.write("\List the input files, one per line\n")
elementfile.write("\"+elementparam[1]+\n")
elementfile.write("")
-----\n")
elementfile.write("\type of elements (central body, barycentric, Jacobi)="+
elementparam[2]+\n")
elementfile.write("\minimum interval between outputs (days)===== "+
elementparam[3]+\n")
elementfile.write("\express time in days or years===== "+
elementparam[4]+\n")
elementfile.write("\express time relative to integration start time===== "+
elementparam[5]+\n")
elementfile.write("")
-----\n")
elementfile.write("\Output formats (e.g. a8.4=> semi-major axis with 8 digits & 4
dec. places\n")
elementfile.write("\"+elementparam[6]+\n")
elementfile.write("")
-----\n")
elementfile.write("\Which bodies do you want (one per line\n")
elementfile.write("\n")
elementfile.write(elementparam[7])
elementfile.close()

class MercuryThread (threading.Thread):
    def __init__(self, ID, name, queue, elementparam, fileparam, big, param, pfa, massstep, astep)
    :
        threading.Thread.__init__(self)
        self.ID=ID
        self.name = name
        self.queue=queue
        self.elementparam=elementparam
        self.fileparam=fileparam
        self.big=big
        self.param=param
        self.pfa=pfa
        self.massstep=massstep
        self.astep=astep
    def run(self):
        run_Mercury(self.ID, self.name, self.queue, self.elementparam, self.fileparam, self.big,
            self.param, self.pfa, self.massstep, self.astep)

class PTmetThread (threading.Thread):
    def __init__(self, ID, name, queue, elementparam, fileparam, big, param, pfa, massstep, astep,
        numberepochs):
        threading.Thread.__init__(self)
        self.ID=ID
        self.name = name
        self.queue=queue
        self.elementparam=elementparam
        self.fileparam=fileparam
        self.big=big
        self.param=param
        self.pfa=pfa
        self.massstep=massstep
        self.astep=astep
        self.numberepochs=numberepochs
    def run(self):
        print "Starting_" + self.name
        run_PTmet(self.name, self.queue, self.elementparam, self.fileparam, self.big, self.param,
            self.pfa, self.massstep, self.astep, self.numberepochs)

def run_Mercury(threadnummer, name, queue, elementparam, fileparam, bigparam, paramparam, pfa,

```

```

    massstep, austep):

while not exitFlag:
    setthreadstatus(threadnummer, "waiting")
    setthreadpercentage(threadnummer, "--")
    PrintStatus(threadstatus, prog_info)
    time.sleep(10)
    queueLock.acquire()
    if (not workQueue.empty()) and (not exitFlag):
        element=queue.get()
        setcurrentelement(threadnummer, element)
        queueLock.release()
        filesLock.acquire()
        time.sleep(2)
        bigparam[17]=str(massstep[int((element.split('_'))[0])])
        bigparam[20]=str(austep[int((element.split('_'))[1])])
        bitparamlname=bigparam[3]
        bigparam[3]=name
        fileparam[1]="big_"+element+".in" # big.in-filename
        fileparam[2]="small_"+element+".in" # small.in filename
        fileparam[3]="param_"+element+".in" # param.in filename
        fileparam[4]="xv_"+element+".out" # xv.out filename
        fileparam[5]="ce_"+element+".out" # ce.out filename
        fileparam[6]="info_"+element+".out" # info.out filename
        fileparam[7]=(fileparam[1])[:-2]+"dmp" # big.dmp filename
        fileparam[8]=(fileparam[2])[:-2]+"dmp" # small.dmp filename
        fileparam[9]=(fileparam[3])[:-2]+"dmp" # param.dmp filename
        fileparam[10]="restart_"+element+".dmp" # restart.dmp filename
        elementparam[1]=fileparam[4]
        WriteSmallIn(fileparam[2])
        WriteParamIn(paramparam, fileparam[3])
        WriteBigIn(bigparam, fileparam[1])
        WriteFilesIn(fileparam, "files.in")
        setthreadstatus(threadnummer, "Mercury6")
        PrintStatus(threadstatus, prog_info)
        filesLock.release()
        os.system("nice_./mercury6")
        infofile=open("info_"+element+".out", 'r')
        infofile_data=infofile.readlines()
        infofile.close()
        encounter_found=False
        for line in range(len(infofile_data)):
            if (infofile_data[line]).startswith("_WARNING:_Stopping_integration_due_to_an_
                encounter"):
                prog_info[5].append(str(element))
                prog_info[5][0]+=1
                prog_info[5][1]=str(element)
                PrintStatus(threadstatus, prog_info)
                encounter_found=True
                break
        if not encounter_found:
            filesLock.acquire()
            time.sleep(2)
            elementparam[7]=name
            bigparam[17]=str(massstep[int((element.split('_'))[0])])
            bigparam[20]=str(austep[int((element.split('_'))[1])])
            bitparamlname=bigparam[3]
            bigparam[3]=name
            fileparam[1]="big_"+element+".in" # big.in-filename
            fileparam[2]="small_"+element+".in" # small.in filename
            fileparam[3]="param_"+element+".in" # param.in filename
            fileparam[4]="xv_"+element+".out" # xv.out filename
            fileparam[5]="ce_"+element+".out" # ce.out filename
            fileparam[6]="info_"+element+".out" # info.out filename
            fileparam[7]=(fileparam[1])[:-2]+"dmp" # big.dmp filename
            fileparam[8]=(fileparam[2])[:-2]+"dmp" # small.dmp filename

```

```

fileparam[9]=(fileparam[3])[:-2]+"dmp"      # param.dmp filename
fileparam[10]="restart_"+element+".dmp"     # restart.dmp filename
elementparam[1]=fileparam[4]
WriteSmallIn(fileparam[2])
WriteParamIn(paramparam, fileparam[3])
WriteBigIn(bigparam, fileparam[1])
WriteFilesIn(fileparam, "files.in")
WriteElementIn(elementparam, "element.in")
filesLock.release()
setthreadstatus(threadnummer, "Element6")
setthreadpercentage(threadnummer, "--")
PrintStatus(threadstatus, prog_info)
os.system("./element6")
setthreadstatus(threadnummer, "TTV_calc")
setthreadpercentage(threadnummer, "--")
PrintStatus(threadstatus, prog_info)
time.sleep(2)
orbitfile=open(name+".aei", "r")
i=1
timeelements=[]
orbitelements=[]
for line in orbitfile:
    if i<4:
        i=i+1
    else:
        tmp=(line[:-1]).split()
        timeelements.append(tmp[0])
        orbitelements.append(tmp[1])
orbitfile.close()
os.system("mv_"+name+".aei_orbit_"+element+".aei")
transittimes=[]
for i in range(len(timeelements)-1):
    if (((orbitelements[i]).startswith("-"))&(not (orbitelements[i+1]).startswith(
        "-"))):
        if (((orbitelements[i]).startswith("-0.0000"))&((orbitelements[i]).
            startswith("0.0000"))):
            try:
                y1= float(orbitelements[i-1])
                y2= float(orbitelements[i+2])
                x1= float(timeelements[i-1])
                x2= float(timeelements[i+2])
                transittimes.append(str(x2-y2*(x2-x1)/(y2-y1)))
            except:
                pass
        else:
            y1= float(orbitelements[i])
            y2= float(orbitelements[i+1])
            x1= float(timeelements[i])
            x2= float(timeelements[i+1])
            transittimes.append(float(x2-y2*(x2-x1)/(y2-y1)))
xvalues=[]
for i in range(len(transittimes)):
    xvalues.append(i+1)
if len(xvalues)==0:
    print "WARNING: xvalues_hat_laenge_0"
try:
    slope, intercept, r_value, p_value, std_err = stats.linregress(xvalues,
        transittimes)
    for i in range(len(transittimes)):
        transittimes[i]=(transittimes[i]-intercept-((i+1)*slope))
    OCLock.acquire()
    file=open("transittimes_"+(element.split('_')[0])+".lst", "a")
    file.write((element.split('_')[1]+", "+str(transittimes)[1:-1]+"\n"))
    file.close()
    OCLock.release()
except:

```

```

        print "linegress_nicht_moeglich_(xvalues_laenge_0?)"
        OCLock.acquire()
        file=open("transittimes_"+(element.split('_'))[0]+".lst","a")
        file.write(str((element.split("_"))[1])+", "+str(transittimes)[1:-1]+"\n")
        file.close()
        OCLock.release()
    else:
        OCLock.acquire()
        file=open("transittimes_"+(element.split('_'))[0]+".lst","a")
        file.write(str((element.split("_"))[1])+", "+str(CE)+"\n")
        file.close()
        OCLock.release()
    stateLock.acquire()
    file=open("lastprogressstate.lst","r")
    states=[]
    for line in file:
        states=line.split()
    file.close()
    states.remove(element)
    file=open("lastprogressstate.lst","w")
    for i in range(len(states)):
        if i < (len(states)-1):
            file.write(states[i]+", ")
        else:
            file.write(states[i])
    file.close()
    stateLock.release()
    os.system("rm -r_*_"+element+"*")
else:
    queueLock.release()
    setthreadstatus(threadnummer, "dead")
    PrintStatus(threadstatus, prog_info)

def setthreadpercentage(nummer, percentage):
    if percentage=="--":
        threadstatus[nummer-1][6]="--"
    else:
        if threadstatus[nummer-1][2]=="x":
            threadstatus[nummer-1][6]=str(int(percentage)*100/196809643)+"%"
        else:
            threadstatus[nummer-1][6]="--"

def setcurrentelement(nummer, elementname):
    threadstatus[nummer-1][7]=elementname

def setthreadstatus(nummer, status):
    if status=="waiting":
        threadstatus[nummer-1][1]="x"
    else:
        threadstatus[nummer-1][1]=""
    if status=="Mercury6":
        threadstatus[nummer-1][2]="x"
    else:
        threadstatus[nummer-1][2]=""
    if status=="Element6":
        threadstatus[nummer-1][3]="x"
    else:
        threadstatus[nummer-1][3]=""
    if status=="TTV_calc":
        threadstatus[nummer-1][4]="x"
    else:
        threadstatus[nummer-1][4]=""
    if status=="dead":
        threadstatus[nummer-1][5]="x"

```

```

else:
    threadstatus[nummer-1][5]=""

def PrintStatus(statusarray, proginfo):
    printScreenLock.acquire()
    os.system("clear")
    zeile_threads="Thread_#:\t"
    zeile_waiting="waiting:\t"
    zeile_mercury="Mercury6:\t"
    zeile_percentage="finished:\t"
    zeile_element="Element6:\t"
    zeile_TTV="TTV_calc:\t"
    zeile_dead="Process_dead:\t"
    zeile_elemente="lastprogressed:_"+str(proginfo[2])
    zeile_Anzahl="Massenelemente/AU-Elemente:_"+str(proginfo[0])+"_(gesamt_"+str(proginfo
        [1])+"_Elemente_zu_je_"+str(proginfo[4])+"_Epochen)"
    if proginfo[5][0]==0:
        zeile_encounter="keine"
    else:
        zeile_encounter=str(proginfo[5][0])+"_-last:_"+str(proginfo[5][1])
    for l in range(len(statusarray)):
        zeile_threads+=str(statusarray[l][0])+"\t"
        zeile_waiting+=str(statusarray[l][1])+"\t"
        zeile_mercury+=str(statusarray[l][2])+"\t"
        zeile_element+=str(statusarray[l][3])+"\t"
        zeile_percentage+=str(statusarray[l][6])+"\t"
        zeile_TTV+=str(statusarray[l][4])+"\t"
        zeile_dead+=str(statusarray[l][5])+"\t"
    print "Status:\n-----\n" + zeile_Anzahl + "\n"
        -----\n" + zeile_threads + "\n" + zeile_waiting
        + "\n" + zeile_mercury + "\n"+zeile_percentage+ "\n" + zeile_element + "\n" +
        zeile_TTV + "\n-----\n" + zeile_dead + "\n" +
        zeile_elemente + "\n-----\nadditional_info:_"+
        str(proginfo[3]) + "\nclose_encounter:_"+ zeile_encounter+ "\nSpeicherbedarf:_"+
    os.system("du-sh")
    os.system("sensors")
    printScreenLock.release()

def MainMercuryProgram(elementparam, fileparam, big, param, numbercores, massstep, austep, pfd
    , numberepochs):
    global exitFlag
    global threadstatus
    global prog_info
    for i in range(numbercores):
        threadstatus.append([i, "", "", "", "", "x", "", ""])
    massrange=range(massstep[0], massstep[1], massstep[2])    #massrange in erdmassen 5 ->
        alle 5M_erd ein punkt
    for l in range(len(massrange)):
        massrange[l]=float(massrange[l]*3)/1000000    #massragen in sonnenmassen
    aurange=range((3*int(float(param[16])*10000)), (int(float(big[20])*10000)-5*austep[0]),
        int((austep[0])/2))
    aurange=aurange+(range(int(float(big[20])*10000+5*austep[0]), austep[1]*int(float(big
        [20])*10000), austep[0]))    # aurange 20 -> alle 0,002AU ein wert
    print aurange
    prog_info.append(str(len(massrange))+"/"+str(len(aurange)))
    prog_info.append(str(len(massrange)*len(aurange)))
    prog_info.append("")
    prog_info.append("")
    prog_info.append(str(numberepochs))
    prog_info.append([0,0])

    PrintStatus(threadstatus, prog_info)
    kombinationen=[]
    for l in range(len(aurange)):
        aurange[l]=float(aurange[l])/10000

```

```

try:
    kombfile=open("kombinationen.lst", "r")
    print "Kombinationsfile existiert"
    kombfile.close()
    kombfile=open("kombinationen.lst", "a")
    try:
        lastprogressfile=open("lastprogressstate.lst", "r")
        print "last-progress-file gefunden"
        for line in lastprogressfile:
            print "folgende Elemente werden zurueckgeschrieben:" +line
            kombfile.write("_"+line)
        lastprogressfile.close()
        os.system("rm lastprogressstate.lst")
    except:
        print "kein last-progress-file gefunden"
    kombfile.close()
except:
    for i in range(len(massrange)-1, -1, -1):
        kombinationen.append(str(i)+"_a")
    kombfile=open("kombinationen.lst", "w")
    print "kein Kombinationsfile gefunden - neues wird angelegt"
    for i in range(len(kombinationen)):
        if i < (len(kombinationen)-1):
            kombfile.write(kombinationen[i]+"_")
        else:
            kombfile.write(kombinationen[i])
    kombfile.close()
    kombfile=open("kombinationen.lst", "r")
    time.sleep(10)
    kombinationen=""
    for line in kombfile:
        kombinationen=line.split()
    threadlist=[]
    for i in range(numbercores):
        threadlist.append("Thread-"+str(i+1))
    threads=[]
    threadID=1
    for tname in threadlist:
        thread = MercuryThread(threadID, tname, workQueue, elementparam, fileparam, big, param,
            pfad, massrange, aurange)
        thread.start()
        threads.append(thread)
        threadID+=1
    while not exitFlag:
        try:
            file=open("exitFlag", "r")
            prog_info[3]= "exitFlag_found_-_waiting_for_threads_to_finish"
            PrintStatus(threadstatus, prog_info)
            file.close()
            exitFlag=1
        except:
            queueLock.acquire()
            if (workQueue.empty() & (len(kombinationen) > 0)):
                file=open("kombinationen.lst", "r")
                for line in file:
                    kombinationen=line.split()
                file.close()
                neuesElement=(kombinationen.pop())
                if neuesElement.split("_")[1]=="a":
                    for i in range(len(aurange)-1, -1, -1):
                        kombinationen.append(neuesElement.split("_")[0]+"_"+str(i))
                neuesElement=str(kombinationen.pop())
    # print "Main Thread new Element: "+neuesElement
    file=open("kombinationen.lst", "w")
    for i in range(len(kombinationen)):
        file.write(kombinationen[i]+"_")

```

```

file.close()
workQueue.put(neuesElement)
queueLock.release()
stateLock.acquire()
try:
    file=open("lastprogressstate.lst","r")
    for line in file:
        prog_info[2]=str(line)
        PrintStatus()
        #print "Mainthread read lastprogressstate: "+str(line)
    file.close()
except:
    pass
    #print "Mainthread lastprogressstate empty"
file=open("lastprogressstate.lst","a")
file.write("_"+neuesElement)
file.close()
file=open("lastprogressstate.lst","r")
try:
    for line in file:
        prog_info[2]=str(line)
        PrintStatus()
except:
    pass
    #print "Mainthread lastprogressstate still empty"
file.close()
stateLock.release()
else:
    queueLock.release()
if not (workQueue.empty()):
    time.sleep(4)
#time.sleep(2)
for i in range(numbercores):
    try:
        setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[7]+".out")))
    except:
        setthreadpercentage(i+1, "--")
PrintStatus(threadstatus, prog_info)
if (workQueue.empty())&(len(kombinationen)==0):
    exitFlag=1
prog_info[3]= "Alle_Kombinationen_abgearbeitet_--_warte_auf_threads"
PrintStatus(threadstatus, prog_info)
while threading.activeCount()>1:
    prog_info[3]= "Alle_Kombinationen_abgearbeitet_-_waiting_for_" + str(threading.
        activeCount()-1)+"_thread"+"(s" if (threading.activeCount()-1)>1 else "")+
        "_to_finish"
    for i in range(numbercores):
        try:
            setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[7]+".
                out")))
        except:
            setthreadpercentage(i+1, "--")
PrintStatus(threadstatus, prog_info)
time.sleep(30)
prog_info[3]= "FÄijhre_TTV-Signale_zusammen..."
PrintStatus(threadstatus, prog_info)
for line in os.popen("ls_1_transit*"):
    print "FÄijhre_Transitsignale_in_"+line[:-1]+"_zusammen"
    transits=[]
    file=open(line[:-1], "r")
    m=0
    for line2 in file:
        transits.append(0)
        transits[m]=(line2[:-1]).split(",")
        m=m+1
    file.close()

```



```

file=open("newsorted"+line[: -1], "w")
file.write("Massenelement_"+str(massrange[int(line[13: -5])])+ "_M_sun_\n")
file.write("distance\t")
for i in range(numberepochs):
    file.write(str(i)+"\t")
for i in range(len(transits)-1):
    file.write(str(aurange[int(transits[i][0])])+ "_AU\t")
    for j in range(numberepochs):
        try:
            file.write(str(transits[i+1][j+1])+ "\t")
        except:
            transits[i+1].append(transits[i+1][j])
            file.write(str(transits[i+1][j+1])+ "\t")
    file.write("\n")
file.close()
while threading.activeCount() > 1:
    prog_info[3]= "exitFlag_found_-_waiting_for_" + str(threading.activeCount()-1)+ "_
    thread"+"s" if (threading.activeCount()-1) > 1 else ""+ "_to_finish"
    for i in range(numbercores):
        try:
            setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[7]+ ".out")))
        except:
            setthreadpercentage(i+1, "--")
    PrintStatus(threadstatus, prog_info)
    time.sleep(30)
os.system("rm *.tmp_element.*_files.in")
try:
    file=open("exitFlag", "r")
    file.close()
    os.system("rm_exitFlag")
except:
    pass
prog_info[3]= "Alles_beendet_-_beende_Programm"
PrintStatus(threadstatus, prog_info)
return 0

```

Quellcode B.5: Programm zur Extraktion der Positionsdaten aus den MERCURY6-Daten

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : Extract_from_aie.py
## Version: 0.1 (alpha)
## Zweck : extract transit times from an *.aie file containing t and x
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import time
import math
import random

orbitfile=open("HatP32b.aie","r")
i=1
timeelements=[]
orbitelements=[]
for line in orbitfile:
    if i<4:
        i=i+1
    else:
        tmp=(line[:-1]).split()
        timeelements.append(tmp[0])
        orbitelements.append(tmp[1])
orbitfile.close()
transittimes=[]
for i in range(len(timeelements)-1):
    if (((orbitelements[i]).startswith("-"))&(not (orbitelements[i+1]).startswith("-"))):
        if (((orbitelements[i]).startswith("-0.0000"))&((orbitelements[i]).startswith("0.0000"))):
            try:
                y1= float(orbitelements[i-1])
                y2= float(orbitelements[i+2])
                x1= float(timeelements[i-1])
                x2= float(timeelements[i+2])
                transittimes.append(str(x2-y2*(x2-x1)/(y2-y1)))
            except:
                pass
        else:
            y1= float(orbitelements[i])
            y2= float(orbitelements[i+1])
            x1= float(timeelements[i])
            x2= float(timeelements[i+1])
            transittimes.append(float(x2-y2*(x2-x1)/(y2-y1)))
transittimefile=open("HatP32b.ttt","w")
for line in range(len(transittimes)-1):
    transittimefile.write(str(transittimes[line])+"\n")
transittimefile.close()

```

Quellcode B.6: Hilfsprogramm zur Sortierung der Daten

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import os
import sys
import time

for line in os.popen("ls -l newsortedtransit*"):
    print "Sortiere Zeilen in "+line[:-1]
    transits=[]
    file=open(line[:-1],"r")
    m=0
    for line2 in file:
        transits.append(0)
        transits[m]=line2[:-1]
        m=m+1
    file.close()
    for j in range(5):
        for i in range(len(transits)-2):
            if i > 1:
                if float((transits[i].split())[0]) > float((transits[i+1].split())[0]):
                    temp=transits[i]
                    transits[i]=transits[i+1]
                    transits[i+1]=temp
    file=open("secondnewsorted"+line[:-1],"w")
    for i in range(len(transits)-1):
        file.write(transits[i)+"\n")
    file.close()
```

Quellcode B.7: Programm zur Darstellung der Mindestmasse in Abhängigkeit des Periodenverhältnisses

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : OCcalculations.py
## Version: 0.1 (alpha)
## Zweck : O-C Calculations of Transiting planets
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import time
import numpy

number_mass_elements=73 #59
werte=["--"]

minimumtime=1.5 #time in minutes
minimumtime=minimumtime/(24*60) #time in days
max_signal=0
aktuelle_AU=""
for i in range(number_mass_elements):
    if i < 10:
        akt="0"+str(i)
    else:
        akt=str(i)
    datei=open("secondnewsortednewsortedtransittimes_"+akt+".lst",'r')
    print "secondnewsortednewsortedtransittimes_"+akt+".lst"
    dateizeile=[]
    zeile=0
    for line in datei.readlines():
        if zeile==0:
            werte[zeile]+="\t"+(line.split())[1]
        if zeile >1:
            dateizeile=(line.split("\t"))[:-1]
            aktuelle_AU=dateizeile[0]
            try:
                wertezeile=[abs(float(dateizeile[x+1])) for x in range(len(dateizeile)-1)]
                max_signal=numpy.std(wertezeile)
            except:
                max_signal=0
            if i==0:
                werte.append(aktuelle_AU)
                werte[zeile-1]+="\t"+str(max_signal)
            zeile+=1
    datei.close()
neue_datei=open("max_signale.lst",'w')
for i in range(len(werte)):
    neue_datei.write(werte[i)+"\n")
neue_datei.close()

neue_datei=open("max_signale.lst",'r')
daten=[]
enddaten=["distance_[AU]_\t_minimum_mass_[Msun]"]
zeile1=[]
dateizeile=[]
zeile=0

```

```

for line in neue_datei.readlines():
    if zeile==0:
        zeile1=((line[:-1]).split("\t"))[1:]
        zeile=1
    else:
        dateizeile=(line.split("\t"))[:-1]
        daten=[abs(float(dateizeile[x+1])) for x in range(len(dateizeile)-1)]
        index=0
        found=0
        for item in range(len(daten)):
            if float(daten[item])>float(minimumtime):
                found=1
                break
            index+=1
        if found==1:
            enddaten.append(str(line.split("\t")[0]).split()[0]+\t+str(zeile1[index]))
        else:
            enddaten.append(str(line.split("\t")[0]).split()[0]+\t+str(0.000001502))
neue_datei.close()
end_datei=open("end_datei_1_5min.lst","w")
for i in range(len(enddaten)):
    end_datei.write(str(enddaten[i])+"\n")
end_datei.close()

```

Quellcode B.8: Programm zur Darstellung der Anzahl von close encounters

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : OCcalculations.py
## Version: 0.1 (alpha)
## Zweck : O-C Calculations of Transiting planets
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import time
import numpy

number_mass_elements=69
au_werte=['']
CE_werte=['']

max_signal=0
aktuelle_AU=""
for i in range(number_mass_elements):
    if i < 10:
        akt="0"+str(i)
    else:
        akt=str(i)
    datei=open("secondnewsortednewsortedtransittimes_"+akt+".lst",'r')
    print "secondnewsortednewsortedtransittimes_"+akt+".lst"
    dateizeile=[]
    zeile=0
    for line in datei.readlines():
        if zeile >1:
            dateizeile=(line.split("\t"))[:-1]
            aktuelle_AU=dateizeile[0]
            if i==0:
                au_werte.append(aktuelle_AU)
                CE_werte.append(0)
            if dateizeile[1]=="CE":
                CE_werte[zeile-2]+=1
        zeile+=1
    datei.close()
neue_datei=open("CE_Anzahl.lst",'w')
for i in range(len(au_werte)):
    neue_datei.write(str(au_werte[i])+"\t"+str(CE_werte[i])+"\n")
neue_datei.close()

```

B.3 Programme zur Bestimmung der Stabilität

Quellcode B.9: Hilfsprogramm zur Definition von Klassen und Funktionen

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : OCcalculations.py
## Version: 0.1 (alpha)
## Zweck : O-C Calculations of Transiting planets
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import time
import math
import random
import thread
from routines import *

#Parameters to be used for param.in
param={}
param[1] ="BS"          # algorithm (MVS, BS, BS2, RADAU, HYBRID)
param[2] ="2451000.500"  # start time in days
param[3] ="4002451000.0" # stop time (days)
tmpparam =1000000       # outout interval (minutes)
param[4] =str(round(0.0006944*tmpparam,3)) # output interval (days) as needed by
    program
param[5] ="10"          # timestep (days)
param[6] ="1.d-12"     # accuracy parameter

param[7] ="yes"        # stop integration after a close encounter
param[8] ="yes"        # allow collisions to occur
param[9] ="no"         # include collisional fragmentation
param[10]="days"      # express time in days or years
param[11]="no"         # express time relative to integration start time
param[12]="high"       # output precision (high, medium, low)
param[13]="no"         # include relativity in integration
param[14]="no"         # include user-defindes force

param[15]="100"        # ejection distance [AU]
param[16]="0.00502"    # radius of central body [AU]
param[17]="1.01"       # central mass [solar masses]
param[18]="0"          # central J2
param[19]="0"          # central J4
param[20]="0"          # central J6
param[21]="3"          # Hybrid integrator changeover [Hill radii]
param[22]="10000000"   # number of timesteps between data dumps
param[23]="10000"      # number of timesteps between periodic effects

#Parameters to be used for big.in
big={}
big[1] ="Ast"         # style (Cartesian, Asteroidal, Cometary)
big[2] ="2451000.5"   # epoch [days]

```

```

#first known planet
tmpbig ="HatP32b"          # 1st Planet name
big[3] =(tmpbig.replace("_","").ljust(8,"_")) # 1st Planet name as needed by program
big[4] ="8.93951341E-04"   # 1st Planet mass in solar masses
big[5] ="4.d0"            # 1st Planet maximum close encounter distance in Hill radii
big[6] ="0.19"           # 1st Planet density in g/cm^3

big[7] ="0.0343"         # 1st planet semi major axis [AU]
big[8] ="0.000"         # 1st planet eccentricity
big[9] ="88.9"          # 1st planet inclination [degree]
big[10] ="000.0"        # 1st planet argument of pericentre [degrees]
big[11] ="0"            # 1st planet longitude of ascending node [degrees]
big[12] ="0"            # 1st planet mean anomaly [degrees]
big[13] ="0"            # 1st planet 1st spin angular momentum
big[14] ="0"            # 1st planet 2nd spin angular momentum
big[15] ="0"            # 1st planet 3rd spin angular momentum

#second unknown planet

tmpbig ="HatP32c"          # 2nd Planet name
big[16] =(tmpbig.replace("_","").ljust(8,"_")) # 2nd Planet name as needed by program
big[17] ="1.84870277E-03"  # 2nd Planet mass in solar masses
big[18] ="4.d0"           # 2nd Planet maximum close encounter distance in Hill radii
big[19] ="1.96"          # 2nd Planet density in g/cm^3

big[20] ="0.0000"        # 2nd planet semi major axis [AU]
big[21] ="0.000"        # 2nd planet eccentricity
big[22] ="88.9"         # 2nd planet inclination [degree]
big[23] ="000.0"        # 2nd planet argument of pericentre [degrees]
big[24] ="0"            # 2nd planet longitude of ascending node [degrees]
big[25] ="0"            # 2nd planet mean anomaly [degrees]
big[26] ="0"            # 2nd planet 1st spin angular momentum
big[27] ="0"            # 2nd planet 2nd spin angular momentum
big[28] ="0"            # 2nd planet 3rd spin angular momentum

fileparam={ }
fileparam[1] ="big.in"   # big.in-filename
fileparam[2] ="small.in" # small.in filename
fileparam[3] ="param.in" # param.in filename
fileparam[4] ="xv.out"   # xv.out filename
fileparam[5] ="ce.out"   # ce.out filename
fileparam[6] ="info.out" # info.out filename
fileparam[7] =(fileparam[1])[:-2] + "dmp" # big.dmp filename
fileparam[8] =(fileparam[2])[:-2] + "dmp" # big.dmp filename
fileparam[9] =(fileparam[3])[:-2] + "dmp" # big.dmp filename
fileparam[10] ="restart.dmp" # restart.dmp filename

#Parameterfile for element.in
elementparam={ }
elementparam[1] =fileparam[4] # input file name
elementparam[2] ="Cen"        # type of elements (central body, barycentric, Jacobi)
elementparam[3] =param[4]     # minimum interval between outputs [days]
elementparam[4] ="days"      # express time in days or years
elementparam[5] ="yes"        # express time relative to integration start time
elementparam[6] ="x8.4_a8.4" # output format
elementparam[7] =big[3]       # body name

numberepochs=100000000.0 #100 #number of epochs to calculate
additionalepochs=0
param[3] =str(2451000.5+365.25*(numberepochs+additionalepochs)*math.sqrt(float(big[7])*
float(big[7])*float(big[7])*(float(param[17])+float(big[4]))))
# stop time (days)

massstep=[50,3000,50] # [1,3000,5] # massrange and step in M_earth (5 -> 5M_earth) start
at 1M_earth, end at 3000Mearth=1Mjup, step 5Mearth
austep=[4,3] # [20,100] # au step in 0,0001AU units (20 -> 0.002AU; minimum: 2) and

```



```

        maximum Au in number of AU of known planet (should be ~factor 10)
        # inner au step is half this value
pfad=os.getcwd()+"/"
if len(sys.argv)<3:
    print "mit_>_Mercury6_<_und_Anzahl_der_Kerne_starten"
    exit(1)

numbercores=int(sys.argv[2])
print "Anzahl_kerne:_"+str(numbercores)
missing_files=""
if sys.argv[1]=="Mercury6":
    try:
        file=open('mercury6','r')
        file.close()
    except:
        missing_files=missing_files+"mercury6,_"
    try:
        file=open('message.in','r')
        file.close()
    except:
        missing_files=missing_files+"message.in,_"
if missing_files=="":
    missing_files="all_required_files_exist"
    time.sleep(10)
else:
    missing_files="Missing_files:_"+missing_files[:-2] + "_-Program_terminated"
    print missing_files
    exit(1)
else:
    print "mit_(Mercury6_/_ptmet)_und_Anzahl_der_Kerne_starten"
    exit(1)

##Mercury6 startbefehl
MainMercuryProgram(elementparam,fileparam,big,param,numbercores,massstep,austep,pfad,
    numberepochs+additionalepochs)

```

Quellcode B.10: Programm zur Berechnung der Stabilität

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

import os
import sys
import time
import math
import random
import threading
import Queue

queueLock=threading.Lock()
stateLock=threading.Lock()
filesLock=threading.Lock()
elementLock=threading.Lock()
lifetimeLock=threading.Lock()
printScreenLock=threading.Lock()
workQueue=Queue.Queue(10)
exitFlag=0
threadstatus=[]
prog_info=[]

def WriteParamIn(param, paramfilename):
    paramfile=open(paramfilename, "w")
    paramfile.write("")0+_06_Integration_parameters(WARNING: Do not delete this line!)\n
    ")
    paramfile.write(")_Lines_beginning_with(')'_are_ignored.\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_Important_integration_parameters:\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_algorithm_(MVS,_BS,_BS2,_RADAU,_HYBRID_etc)=====")+param[1] +"\n")
    paramfile.write(")_start_time_(days)=====")+param[2] +"\n")
    paramfile.write(")_stop_time_(days)=====")+param[3] +"\n")
    paramfile.write(")_output_interval_(days)=====")+param[4] +"\n")
    paramfile.write(")_time_steps_(days)=====")+param[5] +"\n")
    paramfile.write(")_accuracy_parameter=====")+param[6] +"\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_Important_options:\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_stop_integration_after_a_close_encounter=====")+param[7] +"\n")
    paramfile.write(")_allow_collisions_to_occur=====")+param[8] +"\n")
    paramfile.write(")_include_collisional_fragmentation=====")+param[9] +"\n")
    paramfile.write(")_express_time_in_days_or_years=====")+param[10]+"\n")
    paramfile.write(")_express_time_relative_to_integration_start_time=====")+param[11]+"\n")
    paramfile.write(")_output_precision=====")+param[12]+"\n")
    paramfile.write(")_<_not_used_at_present_>\n")
    paramfile.write(")_include_relativity_in_integration=====")+param[13]+"\n")
    paramfile.write(")_include_user-defined_force=====")+param[14]+"\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_These_parameters_do_not_need_to_be_adjusted_often:\n")
    paramfile.write("")
    -----\n")
    paramfile.write(")_ejection_distance_[AU]=====")+param[15]+"\n")
    paramfile.write(")_radius_of_central_body_[AU]=====")+param[16]+"\n")
    paramfile.write(")_central_mass_[solar]=====")+param[17]+"\n")
    paramfile.write(")_central_J2=====")+param[18]+"\n")
    paramfile.write(")_central_J4=====")+param[19]+"\n")
    paramfile.write(")_central_J6=====")+param[20]+"\n")
    paramfile.write(")_<_not_used_at_present_>\n")
    paramfile.write(")_<_not_used_at_present_>\n")

```

```

paramfile.write("_Hybrid_integrator_changeover_[Hill_radii]_=====" + param[21] + "\n")
paramfile.write("_number_of_timesteps_between_data_dumps_=====" + param[22] + "\n")
paramfile.write("_number_of_timesteps_between_periodic_effects_=====" + param[23] + "\n")
paramfile.close()

def WriteBigIn(big, bigfilename):
    bigfile=open(bigfilename,"w")
    bigfile.write("0+_06_Big-body_initial_data_((WARNING:_Do_not_delete_this_line!!)\n")
    bigfile.write(")_Lines_beginning_with_')'_are_ignored.\n")
    bigfile.write(")-----\n")
    bigfile.write("_style_(Cartesian,_Asteroidal,_Cometary)_=" + big[1] + "\n")
    bigfile.write("_epoch_(in_days)_=====" + big[2] + "\n")
    bigfile.write(")-----\n")
    bigfile.write(big[3] + "_m=" + big[4] + "_r=" + big[5] + "_d=" + big[6] + "\n")
    bigfile.write(big[7] + "_" + big[8] + "_" + big[9] + "_" + big[10] + "_" + big[11] + "_" + big[12] + "_" + big
    [13] + "_" + big[14] + "_" + big[15] + "\n")
    bigfile.write(big[16] + "_m=" + big[17] + "_r=" + big[18] + "_d=" + big[19] + "\n")
    bigfile.write(big[20] + "_" + big[21] + "_" + big[22] + "_" + big[23] + "_" + big[24] + "_" + big[25] + "_" +
    big[26] + "_" + big[27] + "_" + big[28] + "\n")
    bigfile.close()

def WriteFilesIn (fileparam, filefilename):
    filefile=open(filefilename,"w")
    filefile.write("_" + fileparam[1] + "\n")
    filefile.write("_" + fileparam[2] + "\n")
    filefile.write("_" + fileparam[3] + "\n")
    filefile.write("_" + fileparam[4] + "\n")
    filefile.write("_" + fileparam[5] + "\n")
    filefile.write("_" + fileparam[6] + "\n")
    filefile.write("_" + fileparam[7] + "\n")
    filefile.write("_" + fileparam[8] + "\n")
    filefile.write("_" + fileparam[9] + "\n")
    filefile.write("_" + fileparam[10] + "\n")
    filefile.close()

def WriteSmallIn (smallfilename):
    smallfile=open(smallfilename,"w")
    smallfile.write("0+_06_Small-body_initial_data_((WARNING:_Do_not_delete_this_line!!)\n")
    smallfile.write(")_Lines_beginning_with_')'_are_ignored.\n")
    smallfile.write(")-----\n")
    smallfile.write("_style_(Cartesian,_Asteroidal,_Cometary)_=" + Ast + "\n")
    smallfile.write(")-----\n")
    smallfile.close()

def WriteElementIn (elementparam, elementfilename):
    elementfile=open(elementfilename,"w")
    elementfile.write("0+_06_element_((WARNING:_Do_not_delete_this_line!!)\n")
    elementfile.write(")_Lines_beginning_with_')'_are_ignored.\n")
    elementfile.write(")-----\n")
    elementfile.write("_number_of_input_files_=" + 1 + "\n")
    elementfile.write(")-----\n")
    elementfile.write(")_List_the_input_files,_one_per_Line\n")
    elementfile.write("_" + elementparam[1] + "\n")
    elementfile.write(")-----\n")
    elementfile.write("_type_of_elements_(central_body,_barycentric,_Jacobi)_=" +
    elementparam[2] + "\n")
    elementfile.write("_minimum_interval_between_outputs_(days)_=====" +
    elementparam[3] + "\n")

```

```

elementfile.write("_express_time_in_days_or_years_=====" +
    elementparam[4] + "\n")
elementfile.write("_express_time_relative_to_integration_start_time_=====" +
    elementparam[5] + "\n")
elementfile.write("
-----\n")
elementfile.write("_Output_formats_(e.g._a8.4_=>_semi-major_axis_with_8_digits_&_4_
dec_places\n")
elementfile.write("_"+elementparam[6]+"\n")
elementfile.write("
-----\n")
elementfile.write("_Which_bodies_do_you_want_(one_per_line\n")
elementfile.write("\n")
elementfile.write(elementparam[7])
elementfile.close()

class MercuryThread (threading.Thread):
    def __init__(self, ID, name, queue, elementparam, fileparam, big, param, pfa, massstep, astep)
        :
        threading.Thread.__init__(self)
        self.ID=ID
        self.name = name
        self.queue=queue
        self.elementparam=elementparam
        self.fileparam=fileparam
        self.big=big
        self.param=param
        self.pfa=pfa
        self.massstep=massstep
        self.aste=aste
    def run(self):
        run_Mercury(self.ID, self.name, self.queue, self.elementparam, self.fileparam, self.big,
            self.param, self.pfa, self.massstep, self.aste)

def setthreadpercentage(number, percentage):
    if percentage=="--":
        threadstatus[number-1][5]="--"
        threadstatus[number-1][7]="--"
    else:
        if threadstatus[number-1][2]=="x":
            threadstatus[number-1][5]=str(round(float(percentage)*100/36954013,1))+"%"
            threadstatus[number-1][7]=str(round(((1.0-float(percentage)/36954013)*11.7158),1))+"
            h"
        else:
            threadstatus[number-1][5]="--"
            threadstatus[number-1][7]="--"

def setcurrentelement(number, elementname):
    threadstatus[number-1][6]=elementname

def setthreadstatus(number, status):
    if status=="waiting":
        threadstatus[number-1][1]="x"
    else:
        threadstatus[number-1][1]=""
    if status=="Mercury6":
        threadstatus[number-1][2]="x"
    else:
        threadstatus[number-1][2]=""
    if status=="encounter":
        threadstatus[number-1][3]="x"
    else:
        threadstatus[number-1][3]=""
    if status=="dead":
        threadstatus[number-1][4]="x"

```

```

else:
    threadstatus[nummer-1][4]=""

def PrintStatus(statusarray, proginfo):
    printScreenLock.acquire()
    os.system("clear")
    zeile_threads="Thread_#:\t"
    zeile_waiting="waiting:\t"
    zeile_mercury="Mercury6:\t"
    zeile_percentage="finished:\t"
    zeile_timefinished="remaining:\t"
    zeile_encounter="Calculate:\t"
    zeile_dead="Process_dead:\t"
    zeile_elemente="lastprogressed:_"+str(proginfo[2])
    zeile_Anzahl="Massenelemente/AU-Elemente:_"+str(proginfo[0])+"_(gesamt_"+str(proginfo
        [1])+"_Elemente_zu_je_"+str(proginfo[4])+"_Jahre)"
    for l in range(len(statusarray)):
        zeile_threads+=str(statusarray[l][0])+"\t"
        zeile_waiting+=str(statusarray[l][1])+"\t"
        zeile_mercury+=str(statusarray[l][2])+"\t"
        zeile_percentage+=str(statusarray[l][5])+"\t"
        zeile_timefinished+=str(statusarray[l][7])+"\t"
        zeile_encounter+=str(statusarray[l][3])+"\t"
        zeile_dead+=str(statusarray[l][4])+"\t"
    print "Status:\n-----\n" + zeile_Anzahl + "\n
        -----\n" + zeile_threads + "\n" + zeile_waiting
        + "\n" + zeile_mercury + "\n"+zeile_percentage+"\n" +zeile_timefinished+"\n" +
        zeile_encounter + "\n-----\n" + zeile_dead + "\n
        " + zeile_elemente + "\n-----\nadditional_info:_
        " + str(proginfo[3]) + "\nSpeicherbedarf:_
    os.system("du_-sh")
    printScreenLock.release()

def run_Mercury(threadnummer,name,queue,elementparam,fileparam,bigparam,paramparam,pfad,
    massstep,austep):
    while not exitFlag:
        setthreadstatus(threadnummer,"waiting")
        PrintStatus(threadstatus,prog_info)
        queueLock.acquire()
        if not workQueue.empty():
            element=queue.get()
            queueLock.release()
            setthreadstatus(threadnummer,"Mercury6")
            setcurrentelement(threadnummer,element)
            PrintStatus(threadstatus,prog_info)
            filesLock.acquire()
            time.sleep(5)
            try:
                tempfile=open("files.in",'r')
                logfile=open("logfile.info",'a')
                logfile.write(element+"_"+str(threadnummer)+"_--_vor_aenderung\n")
                for line in tempfile:
                    logfile.write(line)
                tempfile.close()
                logfile.close()
            except:
                pass
            bigparam[17]=str(massstep[int((element.split('_'))[0])])
            bigparam[20]=str(austep[int((element.split('_'))[1])])
            bitparamlname=bigparam[3]
            bigparam[3]=name
            fileparam[1]="big_"+element+".in" # big.in-filename
            fileparam[2]="small_"+element+".in" # small.in filename
            fileparam[3]="param_"+element+".in" # param.in filename
            fileparam[4]="xv_"+element+".out" # xv.out filename

```

```

fileparam[5]="ce_"+element+".out"           # ce.out filename
fileparam[6]="info_"+element+".out"        # info.out filename
fileparam[7]=(fileparam[1])[:-2]+"dmp"    # big.dmp filename
fileparam[8]=(fileparam[2])[:-2]+"dmp"    # small.dmp filename
fileparam[9]=(fileparam[3])[:-2]+"dmp"    # param.dmp filename
fileparam[10]="restart_"+element+".dmp"    # restart.dmp filename
elementparam[1]=fileparam[4]
WriteSmallIn(fileparam[2])
WriteParamIn(paramparam, fileparam[3])
WriteBigIn(bigparam, fileparam[1])
WriteFilesIn(fileparam, "files.in")
file=open("namen_xv.info", "a")
file.write(element+"_"+str(threadnummer)+"_"+str("xv_"+element+".out\n"))
file.close()
try:
    tempfile=open("files.in", 'r')
    logfile=open("logfile.info", 'a')
    logfile.write(element+"_"+str(threadnummer)+"_nach_aenderung\n")
    for line in tempfile:
        logfile.write(line)
    tempfile.close()
    logfile.close()
except:
    pass
filesLock.release()
os.system("nice./mercury6")
infofile=open("info_"+element+".out", 'r')
infofile_data=infofile.readlines()
infofile.close()
setthreadstatus(threadnummer, "encounter")
PrintStatus(threadstatus, prog_info)
local_lifetime=str(massstep[int((element.split('_'))[0])]+"_t"+str(austep[int((
    element.split('_'))[1])])+"\t")
encounter_found=False
for line in range(len(infofile_data)):
    if (infofile_data[line]).startswith("_WARNING:_Stopping_integration_due_to_an_
        encounter"):
        local_lifetime+=str(max(0, round((((int((((infofile_data[line+1].split())[1]).
            split(".")[0])))-float(paramparam[2]))/365.25), 0))+"\n")
        encounter_found=True
        break
if not encounter_found:
    local_lifetime+=str(round(((float(paramparam[3])-float(paramparam[2]))/365.25)
        ,0))+"\n")
lifetimeLock.acquire()
file=open("lifetimes.lst", "a")
file.write(local_lifetime)
file.close()
lifetimeLock.release()
stateLock.acquire()
file=open("lastprogressstate.lst", "r")
states=[]
for line in file:
    states=line.split()
file.close()
states.remove(element)
file=open("lastprogressstate.lst", "w")
for i in range(len(states)):
    if i < (len(states)-1):
        file.write(states[i]+"_")
    else:
        file.write(states[i])
file.close()
stateLock.release()
os.system("rm_big_"+element+".*_ce_"+element+".*_t_param_"+element+".*_restart_"+
    element+".*_small_"+element+".*_xv_"+element+".*")

```

```

        os.system("mv_info_"+element+".out_infofiles/")
    else:
        queueLock.release()
        time.sleep(10)
setthreadstatus(threadnummer,"dead")
setthreadpercentage(threadnummer,"--")
PrintStatus(threadstatus,prog_info)

def MainMercuryProgram(elementparam, fileparam, big, param, numbercores, massstep, austepp, pfaad
, numberepochs):
    global exitFlag
    global threadstatus
    global prog_info
    for i in range(numbercores):
        threadstatus.append([i, "", "", "", "x", "--", "", "--"])
    massrange=range(massstep[0], massstep[1], massstep[2])    #massrange in erdmassen 5 ->
        alle 5M_erd ein punkt
    for l in range(len(massrange)):
        massrange[l]=float(massrange[l]*3)/1000000    #massragen in sonnenmassen
    aurange=range((3*int(float(param[16])*100000)), (int(float(big[7])*100000)-5*austepp[0]),
        int((austepp[0])/2))
    aurange=aurange+(range(int(float(big[7])*100000+5*austepp[0]), austepp[1]*int(float(big
        [7])*100000), austepp[0]))    # aurange 20 -> alle 0,002AU ein wert
    prog_info.append(str(len(massrange))+"/"+str(len(aurange)))
    prog_info.append(str(len(massrange)*len(aurange)))
    prog_info.append("")
    prog_info.append("")
    prog_info.append(str(round(((float(param[3])-float(param[2]))/365.25), 0))+ "yr")
    if (len(aurange)==0) or (len(massrange)==0):
        prog_info[3]= "massrange_oder_aurange_ist_0"
        PrintStatus(threadstatus, prog_info)
        exit(1)
    PrintStatus(threadstatus, prog_info)
    kombinationen=[]
    for l in range(len(aurange)):
        aurange[l]=float(aurange[l])/100000
    try:
        file=open("lifetimes.lst", "r")
        file.close()
    except:
        file=open("lifetimes.lst", "w")
        file.write("Mass[Msun]\tdistance[AU]\tlifetime[yr]\n")
        file.close()
    try:
        kombfile=open("kombinationen.lst", "r")
        print "Kombinationsfile_existiert"
        kombfile.close()
        kombfile=open("kombinationen.lst", "a")
        try:
            lastprogressfile=open("lastprogressstate.lst", "r")
            print "last-progress-file_gefunden"
            for line in lastprogressfile:
                print "folgende_Elemente_werden_zurueckgeschrieben:_" +line
                kombfile.write("_"+line)
            lastprogressfile.close()
            os.system("rm_lastprogressstate.lst")
        except:
            print "kein_last-progress_file_gefunden"
        kombfile.close()
    except:
        for i in range(len(massrange)-1, -1, -1):
            kombinationen.append(str(i)+"_a")
        kombfile=open("kombinationen.lst", "w")
        print "kein_Kombinationsfile_gefunden_-_neues_wird_angelegt"

```

```

for i in range(len(kombinationen)):
    if i < (len(kombinationen)-1):
        kombfile.write(kombinationen[i]+"_")
    else:
        kombfile.write(kombinationen[i])
kombfile.close()
kombfile=open("kombinationen.lst","r")
kombinationen=""
for line in kombfile:
    kombinationen=line.split()
threadlist=[]
for i in range(numbercores):
    threadlist.append("Thread-"+str(i+1))
threads=[]
threadID=1
for tname in threadlist:
    thread = MercuryThread(threadID, tname, workQueue, elementparam, fileparam, big, param,
        pfad, massrange, aorange)
    thread.start()
    threads.append(thread)
    threadID+=1
while not exitFlag:
    try:
        file=open("exitFlag","r")
        print "exitFlag_found_-_Waiting_for_Threads_to_finish"
        file.close()
        exitFlag=1
    except:
        queueLock.acquire()
        if (workQueue.empty()&(len(kombinationen)>0)):
            file=open("kombinationen.lst","r")
            for line in file:
                kombinationen=line.split()
            file.close()
            neuesElement=(kombinationen.pop())
            if neuesElement.split("_")[1]=="a":
                for i in range(len(aorange)-1,-1,-1):
                    kombinationen.append(neuesElement.split("_")[0]+"_"+str(i))
            neuesElement=str(kombinationen.pop())
            file=open("kombinationen.lst","w")
            for i in range(len(kombinationen)):
                file.write(kombinationen[i]+"_")
            file.close()
            workQueue.put(neuesElement)
            queueLock.release()
            stateLock.acquire()
            try:
                file=open("lastprogressstate.lst","r")
                for line in file:
                    prog_info[2]=str(line)
                    PrintStatus()
                file.close()
            except:
                pass
            file=open("lastprogressstate.lst","a")
            file.write("_"+neuesElement)
            file.close()
            file=open("lastprogressstate.lst","r")
            try:
                for line in file:
                    prog_info[2]=str(line)
                    PrintStatus()
            except:
                pass
            file.close()
            stateLock.release()

```



```

else:
    queueLock.release()
    time.sleep(10)
    for i in range(numbercores):
        try:
            setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[6]+".out")))
        except:
            setthreadpercentage(i+1, "--")
    PrintStatus(threadstatus, prog_info)
    if (workQueue.empty()) & (len(kombinationen) == 0):
        exitFlag = 1
        prog_info[3] = "Alle_Kombinationen_abgearbeitet_--_warte_auf_threads"
        prog_info[3] = "exitFlag_found_--_waiting_for_" + str(threads.activeCount()) + "
            threads_to_finish"
        for i in range(numbercores):
            try:
                setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[6]+".out
                    ")))
            except:
                setthreadpercentage(i+1, "--")
    PrintStatus(threadstatus, prog_info)
    while threading.activeCount() > 1:
        prog_info[3] = "exitFlag_found_--_waiting_for_" + str(threading.activeCount() - 1) + "
            threads_to_finish"
        for i in range(numbercores):
            try:
                setthreadpercentage(i+1, str(os.path.getsize("xv_"+(threadstatus[i])[6]+".out")))
            except:
                setthreadpercentage(i+1, "--")
    PrintStatus(threadstatus, prog_info)
    time.sleep(30)

os.system("rm *.tmp_element.*_files.in")
try:
    file = open("exitFlag", "r")
    file.close()
    os.system("rm exitFlag")
except:
    pass
prog_info[3] = "Alles_beendet_--_beende_Programm"
PrintStatus(threadstatus, prog_info)
return 0

```

Quellcode B.11: Programm zur Sortierung der Stabilitätsergebnisse

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

#####
## Name : OCcalculations.py
## Version: 0.1 (alpha)
## Zweck : O-C Calculations of Transiting planets
## Autor : Martin Seeliger
## Date : 2011/11/04
##
## Bemerkungen: Basiert auf Mercury6 inkl. Element6
##
#####

import os
import sys
import math
import numpy
from matplotlib.patches import Patch
from pylab import *

def sort_vectors(xvector, yvector, zvector):
    found=False
    for i in range(len(xvector)-1):
        for j in range(len(xvector)-1-i):
            if xvector[j]>xvector[j+1]:
                tmpx=xvector[j]
                xvector[j]=xvector[j+1]
                xvector[j+1]=tmpx
                tmpy=yvector[j]
                yvector[j]=yvector[j+1]
                yvector[j+1]=tmpy
                tmpz=zvector[j]
                zvector[j]=zvector[j+1]
                zvector[j+1]=tmpz
                found=True
            if (yvector[j]>yvector[j+1]) and (xvector[j]==xvector[j+1]):
                tmpx=xvector[j]
                xvector[j]=xvector[j+1]
                xvector[j+1]=tmpx
                tmpy=yvector[j]
                yvector[j]=yvector[j+1]
                yvector[j+1]=tmpy
                tmpz=zvector[j]
                zvector[j]=zvector[j+1]
                zvector[j+1]=tmpz
                found=True
    return xvector, yvector, zvector, found

def all_indices(value, qlist):
    indices = []
    idx = -1
    while True:
        try:
            idx = qlist.index(value, idx+1)
            indices.append(idx)
        except ValueError:
            break
    return indices

log=False
Minimum=3

x=[]

```

```

differentx=[]
x_label=""
y=[]
differenty=[]
y_label=""
z=[]
z_label=""
file=open("mass_sorted_lifetimes_comb.lst",'r')
i=0

for line in file:
    tmp=(line[:-1]).split()
    if i==0:
        y_label="Mass_[M$_{jup}$]"
        x_label="P$_a$/P$_b$"
        z_label=tmp[2]
    else:
        tmp[0]=str(float(tmp[0]))
        y.append(float(tmp[0]))
        x.append(float(tmp[1]))
        if log:
            if float(tmp[2])>Minimum:
                z.append(log10(float(tmp[2])))
            else:
                z.append(Minimum)
        else:
            z.append(float(tmp[2]))
        if i==1:
            if not differentx.count(float(tmp[1])):
                differentx.append(float(tmp[1]))
            if not differenty.count(float(tmp[0])):
                differenty.append(float(tmp[0]))
        i=1
sort_gefunden=True
sort_nummer=1
while sort_gefunden:#2*len(differentx):
    print "sorting_number:_" +str(sort_nummer)
    x,y,z,sort_gefunden=sort_vectors(x,y,z)
    sort_nummer+=1

file.close()
new_file=open("au_sorted_lifetimes_comb.lst",'w')
new_file.write("Mass_[Msun]\tPa/Pb\tlog(lifetime)\n")
for i in range(len(y)):
    new_file.write(str(y[i])+"\t"+str(x[i])+"\t"+str(z[i])+"\n")
new_file.close()

```


Danksagung

Ich möchte an dieser Stelle die Gelegenheit nutzen, mich zu bedanken. Personen und Gründe gibt es genug, auch wenn hier nur einige wenige genannt werden können.

Mein Dank geht zunächst an Prof. Dr. Ralph Neuhäuser, welcher mir durch die Annahme als Doktorand die letzten vier Jahre überhaupt ermöglichte. Außerdem danke ich Dr. Gracjan Maciejewski dafür, dass er mich für das Thema der Transitzeitvariationen so sehr begeistern konnte, dass ich es während meiner Promotion bearbeiten wollte.

Beiden Personen sowie auch Dr. Christian Ginski gebührt ebenfalls der Dank für die Möglichkeit, an den Teleskopen des Deutsch-Spanischen Zentrums für Astronomie am Calar Alto Beobachtungen durchzuführen und so ein wenig internationale Beobachtungsluft zu schnuppern.

Für die nationalen und internationalen Beobachtungen, die für mein Projekt stets unkompliziert durchgeführt werden konnten, danke ich allen Kollegen, vor allem allen Mitgliedern des YETI-Netzwerkes und allen Beobachtern in Jena.

Die Arbeit in der Wissenschaft kann des öfteren sehr zäh sein und teilweise auch ziemlich frustrierend. In diesen Fällen sind Kollegen, die einem helfend zur Seite stehen oder einmal mit zum Kaffee holen an die frische Luft kommen, unentbehrlich. Das gesamte Team des Astrophysikalischen Instituts ist mir dabei sehr ans Herz gewachsen.

Einen besonderen Dank möchte ich dabei meinen Kollegen János Schmidt, Manfred Kitze, Ronny Errmann, Claudia Marka, Jonas Greif, Alexandra Berndt, Anna Pannicke, Dr. Tobias O. B. Schmidt und Tamara Zehe aussprechen, die stets gute Ideen und Ratschläge hatten. Außerdem auch Dr. Torsten Löhne, Nicole Pawellek, Caroline Reinert und Dr. Christian Vitense, welche die Teepausen am Institut erst lohnenswert machten.

Für die Wartungen des Computersystems danke ich Jürgen Weiprecht, für die gute Instandhaltung des Teleskops in Großschwabhausen Dr. Markus Mugrauer und Dr. Frank Gießler.

Zu guter Letzt gilt mein größter Dank meinen Eltern, meiner Schwester Christin sowie meinem Partner Chris, die mir in den letzten Jahren sehr viel Rückhalt gegeben haben.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig, ohne unzulässige Hilfe Dritter und ohne die Benutzung anderer als der angegebenen Hilfsmittel und Literatur angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Nachstehend aufgeführte Personen haben mir in der jeweils beschriebenen Weise unentgeltlich geholfen:

- Beobachtungen an der Beobachtungsstation Großschwabhausen der Universitäts-Sternwarte Jena werden von allen Mitarbeitern und Studenten des Astrophysikalischen Institutes durchgeführt. Im Speziellen waren die folgenden Personen an Transit-Beobachtungen beteiligt: Christian Adam, Sven Buder, Annika Dathe, Baha Dincel, Ronny Errmann, Christian Ginski, Jonas Greif, Markus M. Hohle, Manfred Kitze, Claudia Marka, Markus Mugrauer, Anna Pannicke, Stefanie Rätz, Susanne Richter, János G. Schmidt, Tobias O.B. Schmidt Kim Werner und Verena Wolf.
- Die Beobachtungen an anderen nationalen und internationalen Teleskopen wurden von jeweils lokal ansässigen Astronomen durchgeführt, namentlich Berk Aydın Tansel Ak, Selcuk Bilir, Jano Budaj, Victor Casanoca, Wen Ping Chen, Rolf Chini, David Cohen, Dinko Dimitrov, Matilde Fernandez, Cezary Gałan, Ersin Göğüş, Tolga Güver, Eda Güzel, Martin Haas, Moritz Hackstein, Stephan Hellmich, Eric Jensen, Diana Kjurkchieva, Emil Kundra, Gracjan Maciejewski, Matthias Mallonn, Larry Marschall, Stefano Mottola, Johannes M. Ohlert, Aykut Özdönmez, Theo Pri-bulla, Damian Puchalski, Gozde Saral, Martin Vaňko, Zhen-Yu Wu und Xu Zhou.
- Das Programm zur Erstellung der Lichtkurven wurde von Ronny Errmann geschrieben.
- Das Programm zur Phasenfaltung und Berechnung der Phasenabdeckung wurde von Manfred Kitze verfasst.

Weitere Personen waren nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die geltende Promotionsordnung der Physikalisch-Astronomischen Fakultät ist mir bekannt.

Ich versichere ehrenwörtlich, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Jena, den 23. Juni 2015

Martin Seeliger