

# Analyse veränderlicher Sterne mittels TESS und Lightkurve

Dr. Peter Deeskow

**Abstract:** *The article delves into the application of the Lightkurve library for amateur astronomical observations, with a focus on data from the NASA TESS mission. It illustrates how Lightkurve, a Python-based tool, is instrumental in processing and analyzing the vast amounts of data generated by TESS, particularly in the study of variable stars. The article provides a detailed guide for amateur astronomers on how to utilize Lightkurve for tasks like light curve plotting, period determination, and photometry, making the complex TESS data accessible and usable for non-professional researchers. By empowering amateur astronomers with such advanced tools, the article highlights the potential for a broader community engagement in stellar astronomy and the significant contributions amateurs can make in this field, using data from cutting-edge space missions like TESS and KEPLER.*

Die TESS-Mission (Transiting Exoplanet Survey Satellite) [1,2] ist ein Weltraumteleskop der NASA, das 2018 gestartet wurde. Es hat die primäre Aufgabe, Exoplaneten zu entdecken, insbesondere solche, die erdähnlich sind und in der bewohnbaren Zone ihrer Sterne liegen. Dazu verwendet TESS 4 identische Kameras, die jeweils ein Feld von  $24^\circ \times 24^\circ$  abdecken, zusammen also  $24^\circ \times 96^\circ$ , die Größe der einzelnen Pixel beträgt etwa  $21''$ . Die Kameras werden für 27.5 Tage jeweils fest auf einen Himmelsausschnitt („Sektor“) ausgerichtet.

Für vordefinierte Targets werden in dieser Zeit mit hohen Kadenz kleine Bereiche um das Target gezielt ausgelesen. Dabei entstehen die Target Pixel Files (TPF), die diese Ausschnitte als Zeitserie zusammenfassen. Die Kadenz dafür betrug in der Primärmission 2018–2022 200 s, in der ersten (2020-2022) bzw. zweiten (ab 2022) Verlängerung 120 s und teilweise sogar 20 s. Eine „Pipeline“, also eine automatische Verarbeitung der Daten führt für die vordefinierten Targets auf der Basis der TPFs eine Photometrie durch und erstellt Lichtkurven. Das Gesamtbild (Full Frame Image, FFI) wurde in der Primärmission alle 30 min ausgelesen, in der ersten Verlängerung alle 10min und in der zweiten Verlängerung sogar alle 200 s [3,4,5,6]. Die Ergebnisse, TPFs und Lichtkurven für die vordefinierten Objekte und die FFIs, werden im MAST Archive (Mikulski Archive for Space Telescopes) veröffentlicht [7].

Die TESS-Mission deckt 85% des Himmels ab und ist ausgelegt auf die Beobachtung hellerer Sterne im Bereich Magnitude 9–15, in dem auch viele Amateurastronomen beobachten. Das TESS-Archiv kann daher eine wertvolle Quelle zusätzlicher Daten sein, um eigene Beobachtungen zu ergänzen oder vorzubereiten. Der Zugang zu diesen Daten wird durch die Python-Bibliothek „Lightkurve“ (mit „k“, kein Tipfehler) erleichtert [8]. Lightkurve erlaubt, mit wenigen Zeilen in Python umfassende Auswertungen zu erstellen. Das Arbeiten mit Lightkurve wird im Folgenden mit einigen Beispielen

vorge stellt. Der für die Beispiele verwendete Code wird vom Autor in Form von Jupyter-Notebooks im GitHub Repository TESS-BAV bereitgestellt [9]

Bei Verwendung der Anaconda Distribution [10] von Python ist die Installation einfach [8]. Lightkurve ist zwar nicht Bestandteil des base-Environments wird aber unterstützt und kann im Anaconda Prompt über

```
conda install -c conda-forge lightkurve
```

in das Enviroment geladen werden. Im Programmcode stehen dann nach

```
import lightkurve as lk
```

die Werkzeuge zur Verfügung.

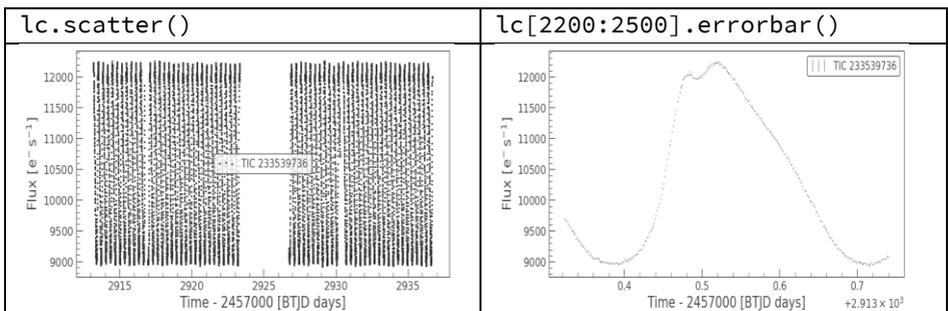
Betrachten wir DD Dra, der im BAV Rundbrief 3/2023 behandelt wurde [11]. Mit

```
search_result=lk.search_lightcurve(target="DD Dra",  
mission="TESS",cadence=120)
```

erhält man eine Liste von in diesem Fall 28 zur Verfügung stehenden Lichtkurven mit 2min Kadenz. Z.B. auf Position 24.

24 TESS Sector 59 2022 SPOC 120 233539736 0.0

Das ist die Lichtkurve, die durch Auswertung der in Sector 59 in 2022 aufgenommenen Daten mit 120 s Kadenz mit der SPOC Pipeline entstanden ist. SPOC steht für Science Processing Operations Center. Es gibt mehrere solche pipelines zur automatischen Photometrie. Es sind darin Algorithmen implementiert, die sich u.a. um die Festlegung einer optimalen Blende für die Photometrie und um die Korrektur von systematischen Fehlern aus der Bewegung des Satelliten kümmern [12].



**Abb. 1** TESS-Lichtkurve von DD Dra

Diese Lichtkurven kann man über den Index einfach herunterladen, hier die Pos. 24 auf der Liste

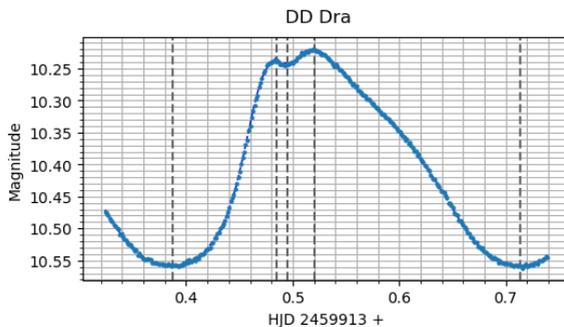
```
lc = search_result[24].download()
```

und ganz oder abschnittsweise, mit oder ohne Fehlerbalken plotten, vgl. Abb1.

Die heruntergeladene Lichtkurve enthält (inkl. der Datenlücke um BTJD=2925) insgesamt 18547 Einzelbeobachtungen, entsprechend etwa 80 aufeinanderfolgenden Maxima von DD Dra. Die Zeitangabe ist das baryzentrische JD, das sich vom heliozentrischen JD hier nur um Sekundenbruchteile unterscheidet (vgl. dazu die Berechnung der Differenz im Jupyter Notebook im Repository). Die Helligkeit wird als Fluss in der instrumentellen Einheit e/s angegeben. Für den Zusammenhang zwischen Fluss und Tess Magnitude Tmag findet sich die Relation [13].

$$Tmag = -2.5 * np.log10(Fluss) + 20.44$$

Damit kann aus den Daten eine Lichtkurve in der bei der BAV üblichen Form erstellt werden (vgl. der Python Code im Repository). Beim Vergleich der Magnituden mit BAV Daten ist zu berücksichtigen, dass die TESS-Beobachtungen im Cousins I-Band und nicht im V-Band erfolgen, dass sich also Farbeinflüsse ergeben können. Außerdem ist die Relation mit Unsicherheiten behaftet, die sich aus möglichen Fehlern bei der Korrektur von Einflüssen aus Streuung oder benachbarten hellen Sternen ergeben.



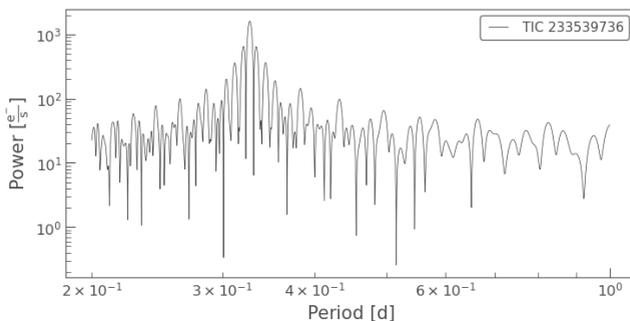
**Abb. 2** Umrechnung einer TESS-Lichtkurve in das BAV-Format

Hier ist also ein Max2 bei HJD = 2459913.5205. Das entspricht mit den im BAV Rundbrief 3/2023 [11] angegebenen Daten (Maximum bei 2460197.5389, Periode 0.326750) einem B-V von 0.073, also mehr als 1.5 h. Da die Unsicherheit der Maxima in jedem Fall sehr viel kleiner als diese Abweichung ist, stellt sich also die Frage nach der Periode.

Da ein TESS-Sektor ja 27 Tage und damit bis zu 80 Maxima von DD Dra erfasst, bietet sich die Möglichkeit, daraus eine Periode zu bestimmen. Lightkurve erlaubt dazu, mit

einem Befehl ein (Lomb-Scargle) Periodogramm zu erzeugen. Ein Lomb-Scargle-Periodogramm (Abb. 3) ist eine Methode zur Analyse von Zeitreihendaten, insbesondere zur Identifikation von periodischen Signalen in ungleichmäßig abgetasteten Datensätzen. Es wendet eine Variation der Fourier-Transformation an, um die Periodizität von Daten zu finden. Details zu diesem Verfahren sind in „Understanding the Lomb-Scargle Periodgram“ [14] nachzulesen.

```
pg = lc.to_periodogram(method="ls",minimum_period=0.2,  
    maximum_period=1,oversample_factor=100)  
pg.plot(view="period",scale="log")
```



**Abb. 3** Periodogramm für DD Dra aus TESS-Daten

Die Größe "Power" repräsentiert darin die Stärke oder Signifikanz eines periodischen Signals in den Daten. Ein höherer Power-Wert deutet darauf hin, dass das entsprechende Signal oder die Frequenz in den Daten stärker oder signifikanter ist, was auf eine periodische Komponente hinweist. Das Maximum ist bei

```
pg.period_at_max_power  
0.32682753d
```

Die Rückrechnung wie oben mit dieser Periode 0.32682753d ergibt dann ein B-V von 0.00528d bzw. 7.6 min, der Wert aus dem Periodogramm gibt das Verhalten also recht gut wieder.

Die von TESS bereitgestellten Lichtkurven werden aus „Target Pixel Files“ erzeugt. Die TPFs umfassen eine Reihe von kleinen (typisch 10 x 10 bis 20 x 20 Pixel großen) Ausschnitten aus den von TESS aufgenommen Gesamtbildern. Auf die TPFs von vordefinierten Targets kann über Lightkurve ebenfalls einfach zugegriffen werden, hier demonstriert am Beispiel AR Her:

```
tpf=lk.search_targetpixelfile('AR Her')
```

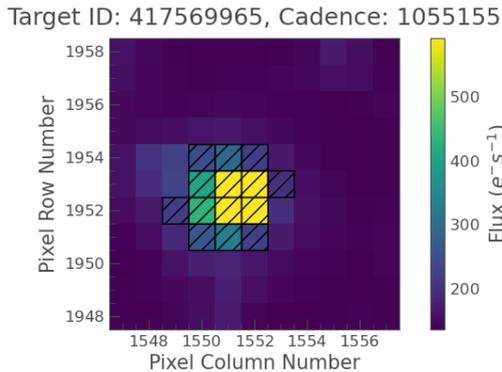
AR Her wurde im Sektor 51 mit 120 s Kadenz beobachtet, man erhält nur den einen Eintrag in der Liste der Suchergebnisse:

```
0 TESS Sector 51 2022 SPOC 120 417569965 0.0
```

Den TPF kann man herunterladen und sich einzelne Bilder aus der Serie anzeigen lassen (Abb. 4), hier das erste (mit Index 0), Abb. 4

```
tpf = tpf[0].download()
```

```
tpf[0].plot(aperture_mask=tpf[0].pipeline_mask, mask_color='k')
```



**Abb. 4** TargetPixelFile von AR Her mit Maske für die Photometrie

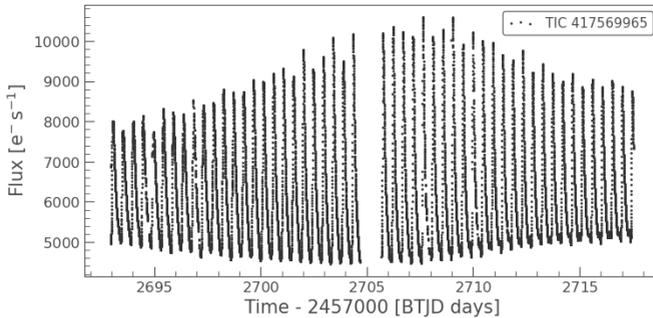
(Mit `tpf.animate()` kann die Serie von Einzelbildern auch als kleiner Film dargestellt werden, vgl. den Code im Repository.) Die Daten, die man auf diese Weise erhält, sind nicht die Rohdaten der Tess-Kamera. Sie haben bereits eine automatische Verarbeitung erfahren, die z.B. systematische Fehler aus Streulichteinflüssen beseitigt. Dazu im nächsten Beispiel mehr.

Die Photometrie wird auf den TPFs durch Aufsummieren der Flüsse derjenigen Pixel durchgeführt, die in einer Maske, der „Targetaperture“ liegen. Diese kann automatisch generiert oder auch manuell erstellt werden. In Abb. 4 ist als schraffierte Pixel die von der Verarbeitungspipeline automatisch generierte Maske eingeblendet über die Parameter `aperture_mask=tpf[0].pipeline_mask`, `mask_color='k'`. Man kann die Maske, die für die Photometrie verwendet wird, umdefinieren, aber dafür ist hier kein Anlass zu erkennen. Wir können also mit dem Befehl `to_lightkurve()` direkt mit der Standardmaske eine Lichtkurve aus dem TPF erzeugen und plotten:

```
lc = tpf.to_lightkurve(aperture_mask=tpf.pipeline_mask)
```

```
lc.scatter();
```

Man erkennt in der Darstellung Abb. 5 den Blazhko-Effekt:

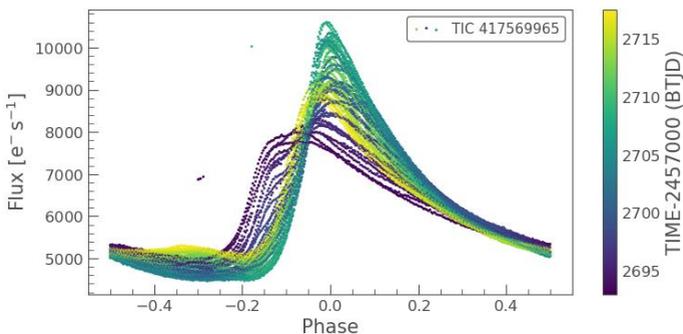


**Abb. 5** TESS-Lichtkurve für AR Her

Der lässt dich mit den Werkzeugen von Lightkurve noch deutlicher darstellen. Das Periodogramm ergibt die Grundfrequenz der Oszillation zu  $pg.period\_at\_max\_power=0.46923402d$ , mit der dann über den Befehl `fold()` die Lichtkurve gefaltet werden kann:

```
pg =  
lc.to_periodogram(method="ls",minimum_period=0.2,maximum_period=  
1,oversample_factor=100)  
lcf=lc.fold(period=pg.period_at_max_power,epoch_phase=0.7,norma  
lize_phase=True)  
lcf.scatter(c=lcf.time_original.value);
```

Das ergibt dann eine deutliche Darstellung des Blazhko-Effektes über die etwa 50 Zyklen, die im Sektor erfaßt wurden, Abb. 6.



**Abb. 6** Gefaltete TESS-Lichtkurve für AR Her

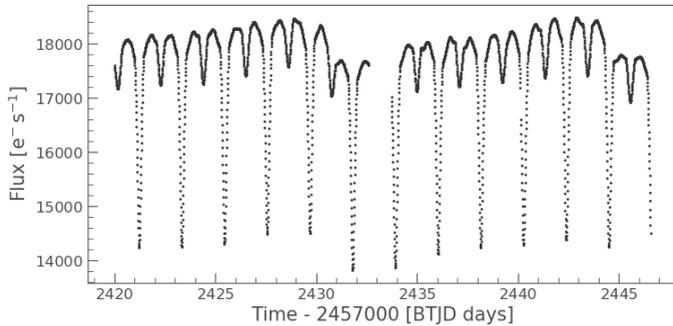
In der Darstellung wurde über `c=lcf.time_original.value` der jedem Punkt zuzuordnende Beobachtungszeitpunkt verwendet, um die Markierungen im Bild entsprechend des Zeitverlaufs von Dunkelblau (am Anfang des Sektors) über Türkis zu Gelb (am Ende) einzufärben.

Die beiden bis hier gezeigten Beispiele - DD Dra und AR Her - gehören zu den Zielen in der Beobachtungsliste von TESS. Für diese werden die `TargetPixelFiles` und die daraus anzuleitenden Lichtkurven automatisch erstellt. Auch wenn diese Liste umfangreich ist (ca. 200.000+ Einträge) so enthält sie doch u.U. nicht alle Sterne, die für BAV Beobachter interessant sein können. Es besteht dann die Möglichkeit, aus den ebenfalls verfügbaren Vollformatbildern (Full Frame Images, FFIs) die TPFs als Zeitreihe kleiner Ausschnitte selbst zu erzeugen und dann wie oben Lichtkurven und Auswertungen zu erstellen.

Das soll im letzten Beispiel am Bedeckungsveränderlichen V0463 Cyg demonstriert werden. Der Befehl `search_tess_cut` prüft, in welchen Sektoren die FFI das gewünschte Objekt enthalten ist. Über den `download(cutoutsize=xx)` Befehl können wir für eines der Suchergebnisse, hier die Listenposition 1, einen `TargetPixelfile` der Größe `cutoutsize` erzeugen. Der so erzeugte TPF enthält -anders als die von einer Pipeline erzeugten TPFs im letzten Beispiel- noch keine vordefinierte Maske für die Photometrie, die müssen wir erstellen als `target_mask`. Damit kann dann mit wenigen Zeilen eine Roh-Lichtkurve generiert werden:

```
search_result = lk.search_tesscut('V0463 Cyg')
tpf = search_result[1].download(cutout_size=20)
target_mask=np.empty((20,20))
target_mask[:,:]=False
target_mask[9:12,9:12]=True
lc=tpf.to_lightcurve(aperture_mask=target_mask)
lc.scatter();
```

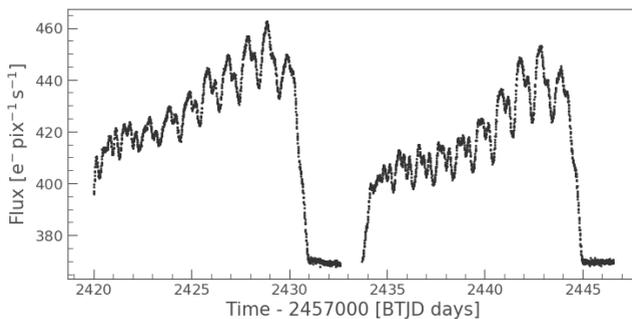
Das Ergebnis in Abb. 7 sieht deutlich anders aus, als für einen Bedeckungsveränderlichen erwartet. Hier wird ein systematischer Fehler in den TESS FFI Daten deutlich, nämlich das mit der Position von Erde und Mond relativ zu Tess variierende Streulicht, das in den 27 Tage umfassenden Sektoren eine 27/2 Periodizität aufweist. In den von den Pipelines erzeugten Lichtkurven ist das herausgerechnet, hier müssen wir uns darum kümmern.



**Abb. 7** Mit tesscut erzeugte Rohlichtkurve

Lightkurve bietet Werkzeuge, solche Korrekturen vornehmen. Der einfachste Weg ist, die Funktion `estimate_background` zu verwenden. `estimate_background` sucht über einen Grenzwert bezogen auf den Median der Helligverteilung im TPF alle Pixel, die nicht einem Stern zuzuordnen sind und leitet daraus eine Lichtkurve für die Hintergrundhelligkeit ab, Abb. 8.

```
lcb=tpf.estimate_background()
lcb.scatter();
```

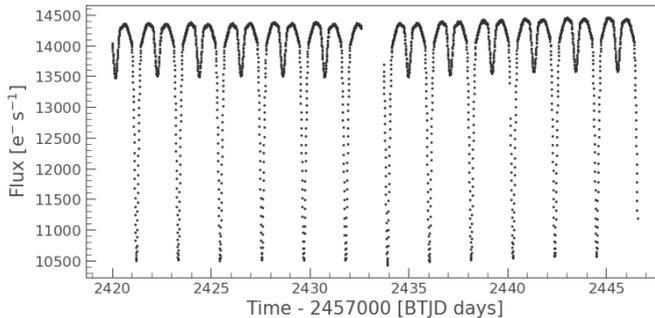


**Abb. 8** Lichtkurve der Hintergrundhelligkeit im TPF für V0463 Cyg

Diese Hintergrundhelligkeit kann dann von der Rohlichtkurve abgezogen werden. Dabei ist beachten, dass die flux Werte in den Lichtkurven Einheiten tragen. Die Lichtkurve des Sterns wird in Elektronen/s, die des Hintergrunds in Elektronen/pixel/s angegeben.

```
n_target_pixels = target_mask.sum()
n_target_pixels
```

```
lcc=lc - lcb.flux*ap.units.pix*n_target_pixels
lcc.scatter();
```



**Abb. 9** Korrigierte Lichtkurve für V0463 Cyg

Das einfache Verfahren führt hier offenbar schon zu gutem Ergebnis, Abb. 9. Bei Bedarf für komplexere Korrekturmöglichkeiten sei auf die Tutorials in der Onlinedokumentation [8] verwiesen.

Die vorstehenden Beispiele machen deutlich, dass mit der Python Bibliothek Lightkurve mit geringem Aufwand der Datenbestand der TESS Mission (und im Übrigen genauso der KEPLER Mission) ausgewertet werden kann, um Beobachtungsvorhaben bei der BAV vorzubereiten oder zu ergänzen.

- [1] <https://tess.mit.edu/>
- [2] <https://heasarc.gsfc.nasa.gov/docs/tess/>
- [3] <https://heasarc.gsfc.nasa.gov/docs/tess/data-products.html>
- [4] <https://heasarc.gsfc.nasa.gov/docs/tess/primary.html>
- [5] <https://heasarc.gsfc.nasa.gov/docs/tess/extended.html>
- [6] <https://heasarc.gsfc.nasa.gov/docs/tess/second-extended.html>
- [7] <https://archive.stsci.edu/>
- [8] <https://docs.lightkurve.org/>
- [9] <https://github.com/pdeeskow/TESS-BAV>
- [10] <https://www.anaconda.com/>
- [11] D. Bannuscher, DD Dra wieder eingefangen, BAV Rundbrief 3/2023, 72. Jahrgang
- [12] <https://heasarc.gsfc.nasa.gov/docs/tess/pipeline.html>
- [13] <https://tess.mit.edu/public/tesstransients/pages/readme.html>
- [14] VanderPlas, J. Understanding the Lomb-Scargle Periodogram, ApJS 236.1:16 (2018) <https://ui.adsabs.harvard.edu/abs/2018ApJS..236...16V>

Kontakt: [pdeeskow@googlemail.com](mailto:pdeeskow@googlemail.com)