Матвей Полубрюхов

m9175172036@gmail.com

Аннотация

[Заинтересуйте читателя с помощью аннотации (как правило, это краткое содержание документа).   
Если вы готовы добавить свой текст — просто щелкните здесь и введите его.]

Моделирование кривой блеска затменной звёздной системы

[Подзаголовок документа]

Оглавление

[Цели проекта 2](#_Toc108848102)

[Используемый «инструментарий» 2](#_Toc108848103)

[Хранение данных 2](#_Toc108848104)

[Алгоритм работы кода 2](#_Toc108848105)

[1. Вычисление истинной аномалии 2](#_Toc108848106)

[2. Расчёт расстояния в проекции на картинную плоскость 3](#_Toc108848107)

[3. Проверка дисков звёзд на наложение 3](#_Toc108848108)

[4. Подсчёт суммарной светимости/звёздной величины 4](#_Toc108848109)

[5. Добавление данных в массивы 4](#_Toc108848110)

[Примеры выполнения кода и анализ полученных данных 5](#_Toc108848111)

[Перспективы развития проекта (TODO) 6](#_Toc108848112)

[Заключение 6](#_Toc108848113)

[Список литературы и интернет-источников 6](#_Toc108848114)

# Цели проекта

* Изучить затменные звёздные системы, закономерность изменения звёздной величины, различия кривых блеска для разных типов звёздных систем
* Создать программу, выполняющую моделирование двойной звёздной системы
* Проанализировать полученные графические изображения изменения параметров во времени

# Используемый «инструментарий»

* Язык программирования Python 3.10. Выбор обусловлен лёгкостью и быстротой написания кода, понятностью синтаксиса, что я поставил приоритетней высокой скорости работы кода.
* Интегрированная среда разработки JetBrains PyCharm Professional EAP. На данный момент это одна из лучших по функционалу IDE для выбранного ЯП.
* Распределённая система контроля версий Git for Windows. Самая известная и простая в изучении.
* Подключаемые библиотеки (список см. [Dependencies](https://github.com/PM-95025/scientificProject/network/dependencies) и [requirements.txt](https://github.com/PM-95025/scientificProject/blob/master/requirements.txt)

# Хранение данных

Для хранения параметров звёзд используется файл ./data/data.ini. Внутри файла существует 4 раздела:

1. General – хранение общих данных о модели (таких, как )
2. Star 1 – хранение физических характеристик о 1ой звезде
3. Star 2 – хранение физических характеристик о 2ой звезде
4. System – хранение данных о звёздной системе (параметров орбиты)

Формат INI был выбран из-за простоты синтаксиса и удобочитаемости файлов этого формата.

# Алгоритм работы кода

1. Первоначальная настройка (инициализация)
2. Главный цикл (итерация по времени)
   1. Вычисление истинной аномалии
   2. Расчёт расстояния между центрами в видимой плоскости (обозначен за х на рис.2)
   3. Проверка дисков звёзд на наложение
   4. Подсчёт суммарной звёздной величины
   5. Добавление данных в массивы
3. Инициализация matplotlib
4. Построение графиков

### 1. Вычисление истинной аномалии

Производится путём численного решения уравнения Кеплера методом итераций:

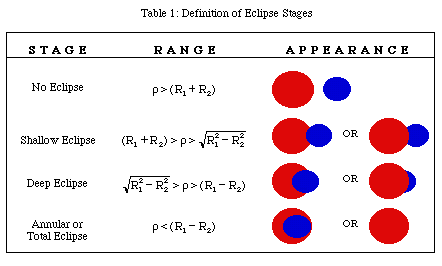
def elliptical\_orbit(t: float, T: float, e: float) -> float:  
 M: float = t / T \* 2\*pi  
 E: float = M  
   
 for n in range(20):  
 E = e \* sin(E) + M  
   
 phi: float = 2 \* degrees(atan(tan(E/2) / sqrt((1-e) / (1+e))))  
   
 if t/T > 0.5:  
 phi += 360  
   
 return phi

### 2. Расчёт расстояния в проекции на картинную плоскость

* def apparent\_distance(system, fi: float) -> float:if system.e == 0:  
   return system.a \* abs(sin(radians(fi)))  
     
   return abs((system.p \* sin(radians(system.periapsis\_argument + fi))) / (1 + system.e \* cos(radians(fi))))
* def inclination\_correction(system, fi: float) -> float:return system.a \* sin(radians(system.inclination)) \* cos(radians(system.ascending\_node\_longitude)) \* cos(radians(fi))
* fi = time2phi(t, system.period, system.e)  
  ap = apparent\_distance(system, fi)  
  x: float = sqrt(ap\*\*2 + inclination\_correction(system, fi)\*\*2)

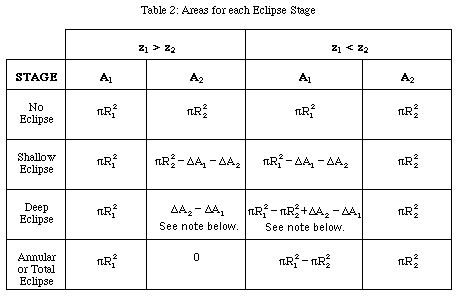
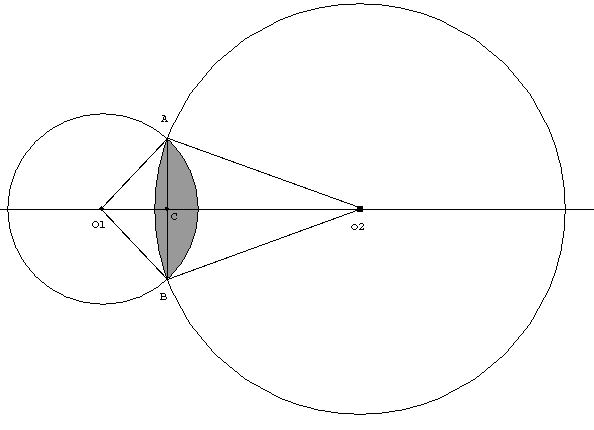
### 3. Проверка дисков звёзд на наложение

Производится с использование ранее вычисленного расстояния в проекции на картинную плоскость



В коде происходит проверка только на сценарий №1, т.к. остальные три идентичны в плане расчётов.

if star\_front.radius + star\_back.radius <= x:  
 return star\_system.L \* (1 + limb\_darkening\_edge) / 2



### 4. Подсчёт суммарной светимости/звёздной величины

Помимо факторов, обусловленных геометрией (эксцентриситет, наклон орбиты и т.д.), учитывается потемнение к краю. Данные для распределения яркости по диску звезды берутся из библиотеки PyLDTk (**P**ython **L**imb **D**arkening **T**ool**k**it)

# init ldtk

sc = LDPSetCreator(teff=(5500, 100), logg=(4.5, 0.2), z=(0.25, 0.05), filters=[sdss\_i], dataset='vis-lowres')  
limb\_darkening = sc.create\_profiles(nsamples=2000)  
limb\_darkening.resample\_linear\_z(300)  
  
limb\_darkening\_x = limb\_darkening.\_z  
limb\_darkening\_y = limb\_darkening.\_mean[0]  
  
limb\_darkening\_edge = min(limb\_darkening\_y)

# calculating

max\_radius: float = max(star\_front.radius, star\_back.radius)  
min\_radius: float = min(star\_front.radius, star\_back.radius)  
  
L\_front: float = star\_front.L \* (1 + limb\_darkening\_edge) / 2  
L\_back: float = star\_back.L \* (1 + limb\_darkening\_edge) / 2  
L\_corr\_square: float = interpolator((0, star\_back.square), (0, star\_back.L), calculate\_intersection(min\_radius, max\_radius, x))  
L\_corr\_limb\_darkening: float = limb\_darkening\_interpolator((max\_radius + min\_radius - x) / (4 \* star\_back.radius))  
  
L\_total: float = L\_front + L\_back - L\_corr\_square \* L\_corr\_limb\_darkening

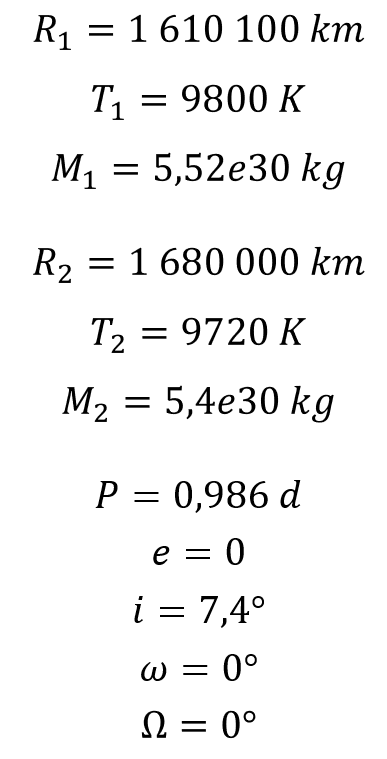
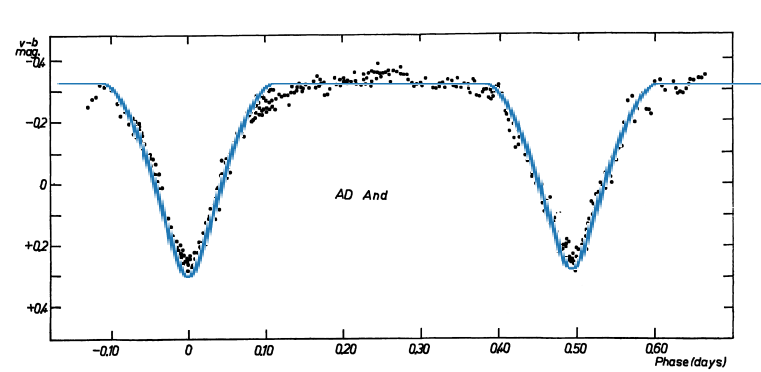
### 5. Добавление данных в массивы

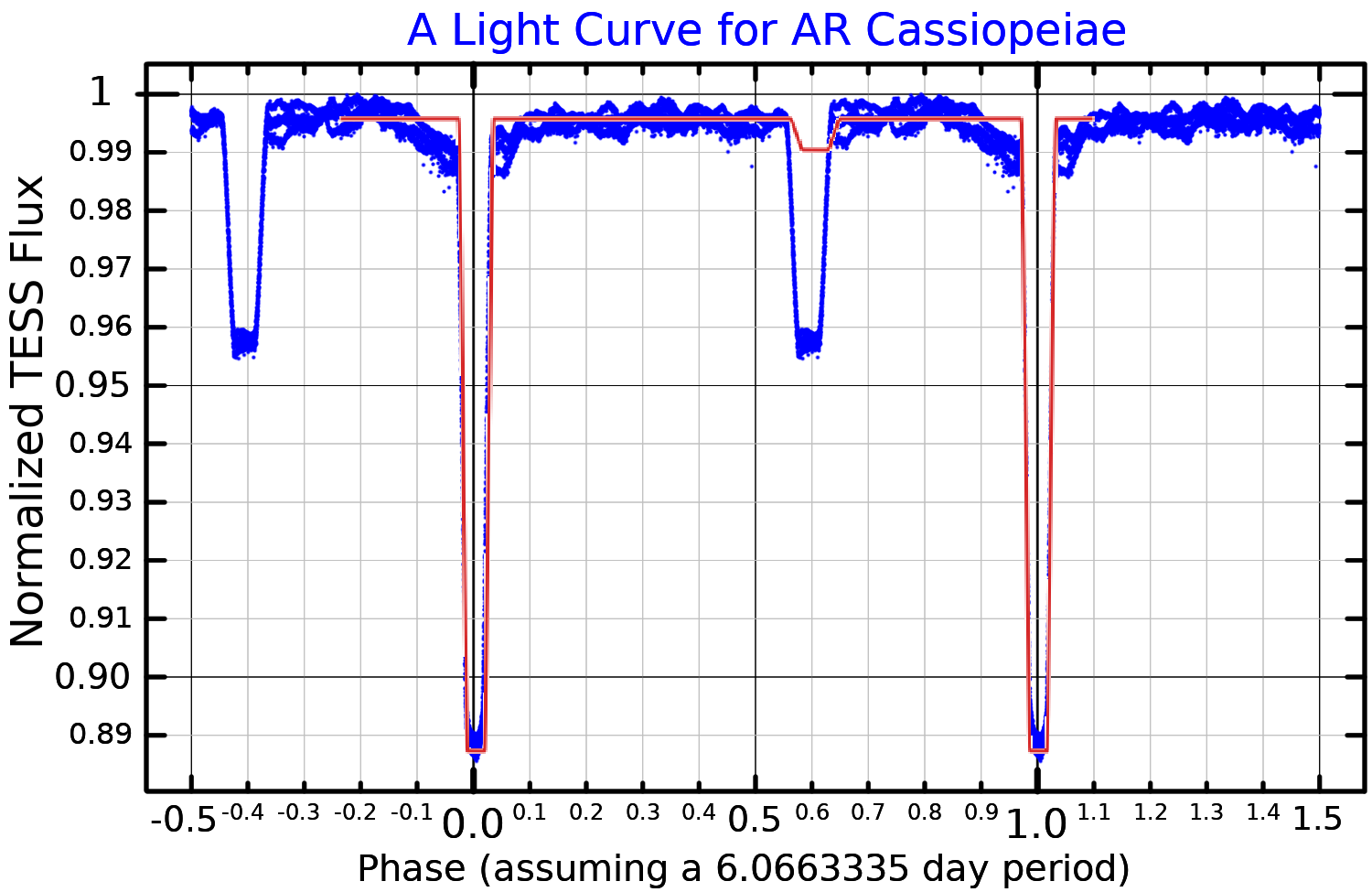
Списки принадлежат а типу данных builtins.list[float]

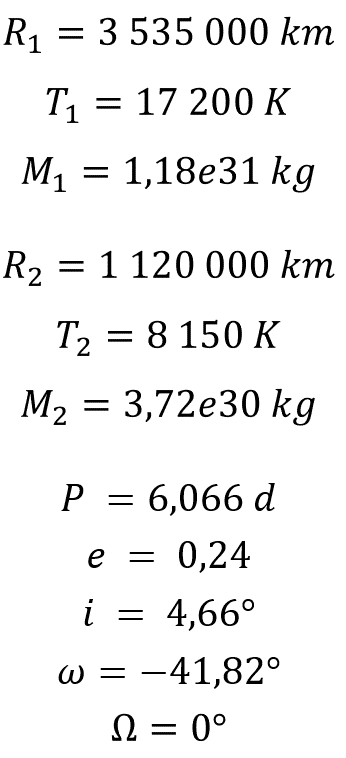
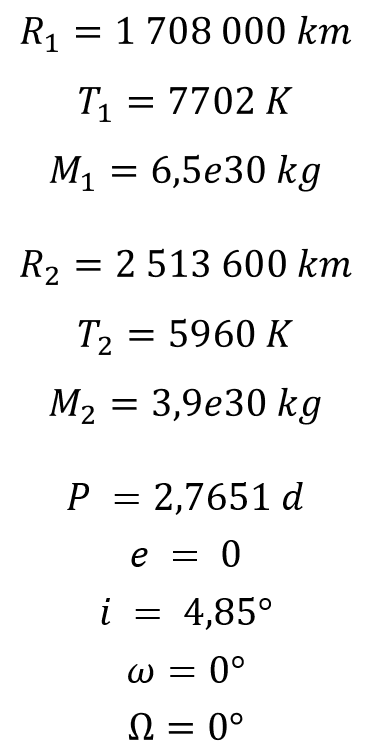
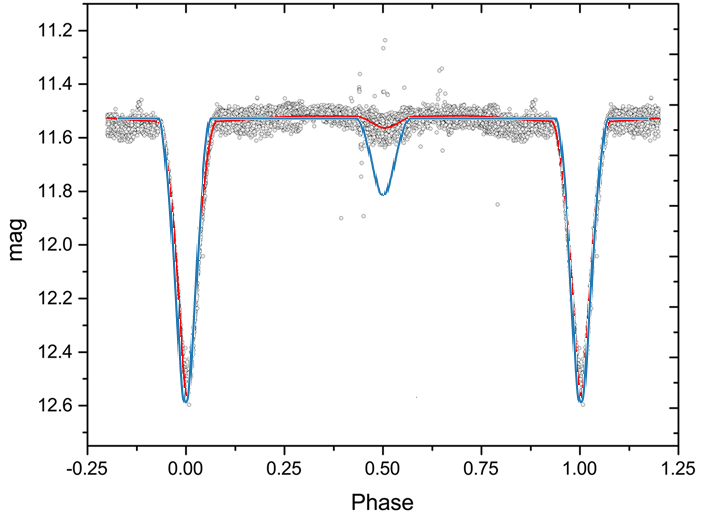
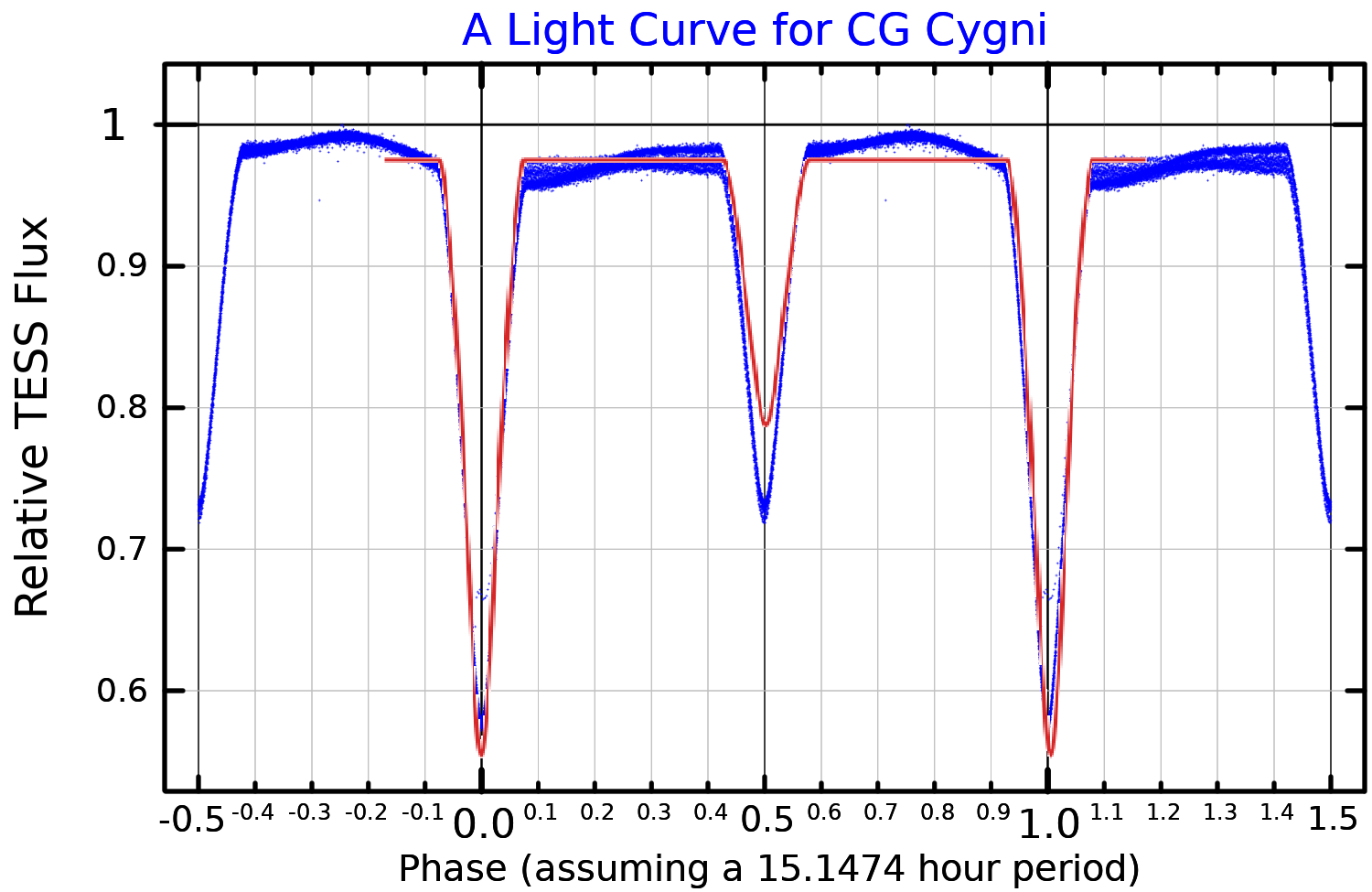
x\_axis\_data.append(t / system.period)  
mags.append(result[0])  
mags\_without\_correction.append(result[1])  
distances\_visual.append(x)  
distances.append(system.r(fi))  
fis.append(fi)

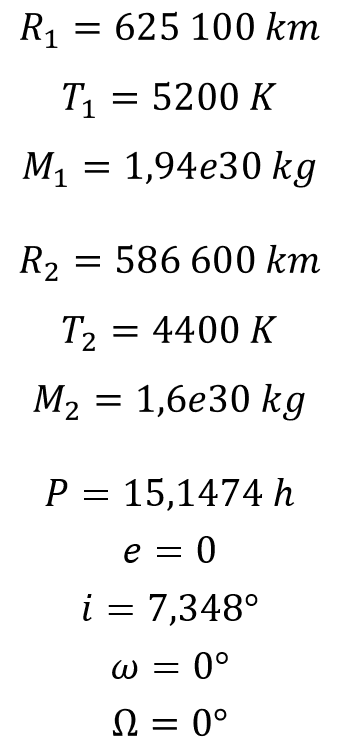
# Примеры выполнения кода и анализ полученных данных

Для возможности мгновенной оценки результатов в качестве примеров были использованы реально существующие звёздные системы.

1. AD Андромеды



1. AR Кассиопеи
2. TT Андромеды
3. CG Лебедя



# Перспективы развития проекта (TODO)

1. Добавить поддержку тесных звёздных систем (), точнее – звёзд, деформированных силой гравитации со стороны другого компонента;
2. Уточнить модель, введя ещё один эффект – ‘’эффект фазы’ или частичное отражение одной звезды на другой

# Заключение

* В ходе проектной работы создана программа, выполняющая поставленную задачу – моделирование двойной звёздной системы;
* Проведены анализ полученных кривых блеска, сравнение результата работы кода с наблюдательными данными. Так как данные почти совпадают, можно сделать вывод о корректности работы программы.

# Список литературы и интернет-источников

1. <https://forum.sources.ru/index.php?showtopic=9381> - тема с форума Sources.Ru (задача о пересечении двух кругов)
2. [https://ru.wikipedia.org/wiki/Уравнение\_Кеплера#Приближённые\_методы](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%9A%D0%B5%D0%BF%D0%BB%D0%B5%D1%80%D0%B0#%D0%9F%D1%80%D0%B8%D0%B1%D0%BB%D0%B8%D0%B6%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B)
3. Официальная документация к библиотеке matplotlib:
   1. <https://matplotlib.org/stable/gallery/lines_bars_and_markers/fill_between_demo.html>
   2. <https://matplotlib.org/stable/gallery/shapes_and_collections/hatch_style_reference.html>
   3. <https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html>
4. <https://github.com/hpparvi/ldtk/blob/master/notebooks/01_Example_basics.ipynb> - jupiter ноутбук с инструкцией по получению данных о распределении яркости по диску звезды
5. Обсуждения на форуме StackOverflow (настройка matplotlib):
   1. <https://stackoverflow.com/questions/28132936/axes-invert-axis-does-not-work-with-sharey-true-for-matplotlib-subplots>
   2. <https://stackoverflow.com/questions/56894740/matplotlib-why-does-setting-grid-with-pyplot-setp-causes-error>
   3. <https://stackoverflow.com/questions/19626530/python-xticks-in-subplots>
   4. <https://stackoverflow.com/questions/5993206/is-it-possible-to-have-multiple-pyplot-windows-or-am-i-limited-to-subplots>
6. Обсуждения на форуме OverCoder (настройка matplotlib):
   1. [https://overcoder.net/q/1195838/как-мне-заполнить-область-только-штриховкой-без-фона-в-matplotlib-20](https://overcoder.net/q/1195838/%D0%BA%D0%B0%D0%BA-%D0%BC%D0%BD%D0%B5-%D0%B7%D0%B0%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D1%8C-%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C-%D1%82%D0%BE%D0%BB%D1%8C%D0%BA%D0%BE-%D1%88%D1%82%D1%80%D0%B8%D1%85%D0%BE%D0%B2%D0%BA%D0%BE%D0%B9-%D0%B1%D0%B5%D0%B7-%D1%84%D0%BE%D0%BD%D0%B0-%D0%B2-matplotlib-20)
7. <https://www.geeksforgeeks.org/how-to-implement-linear-interpolation-in-python/amp/> - учебный материал с портала GeeksForGeeks "How to implement linear interpolation in Python?"
8. Информация о двойных звёздах (в т.ч. их параметры)
   1. <https://en.wikipedia.org/wiki/AD_Andromedae>
   2. <https://en.wikipedia.org/wiki/CG_Cygni>
   3. <https://en.wikipedia.org/wiki/AR_Cassiopeiae>
   4. <https://www.scirp.org/journal/paperinformation.aspx?paperid=77049> (TT And)
   5. <http://www.physics.sfasu.edu/astro/ebstar/ebstar.html> (битая ссылка, архивная версия на [WayBack Machine](https://web.archive.org/web/20220501021646/http:/www.physics.sfasu.edu/astro/ebstar/ebstar.html) - статья из электронной библиотеки SFASU "ECLIPSING BINARY STARS. A Simple Model for Computing Light Curves"
   6. <http://crydee.sai.msu.ru/ak4/Chapt_11_3_154.htm> - статья из электронной библиотеки ГАИШ "§ 154. Общие характеристики двойных систем"

QR-код со ссылкой на GitHub-репозиторий моего проекта