Министерство науки и высшего образования Республики Казахстан

Евразийский Национальный университет им. Л.Н Гумилева Физико-технический факультет

Институт общей и теоретической физики

Моделирование звезды Algol (β Persei) и сравнение с реальными наблюдениями

Әлішер А. Қ.

Научный руководитель: Мырзакулов Р.

Астана, 2025

Введение

### **Актуальность темы:** Система Algol (β Персея) является одной из наиболее известных затменных двойных звёзд и служит классическим примером взаимодействия компонентов в тесной двойной системе. Благодаря своей относительной близости к Земле и яркости, Algol является удобным объектом для фотометрических и спектроскопических наблюдений. Изучение этой системы позволяет глубже понять процессы обмена массой между звёздами, эволюцию двойных систем, а также проверять и уточнять существующие модели звёздной структуры. Моделирование Algol и сравнение результатов с реальными наблюдениями способствует развитию методов численного моделирования и тестированию теоретических гипотез.

**Цель работы:** Целью данной работы является построение численной модели системы Algol (β Persei), с учетом её известных физических параметров, и сравнение полученных теоретических результатов с реальными астрономическими наблюдениями, в том числе с кривой блеска и орбитальными характеристиками.

1.Решение задачи двух тел.

2. Моделирование затмений.

3. Построить синтетическую кривую блеска, показывающую изменение яркости во времени с учётом затмений.

4. Сравнение и анализ результатов. Построить единый график, на котором будут отображены как модельные, так и реальные данные.

**Обьект исследование:** Algol (β Персея)

**Методология:** Сбор и анализ данных: изучение опубликованных научных наблюдений и параметров звёзд в системе Algol, включая массы, радиусы, орбитальные периоды и спектральные типы.

Моделирование: использование специализированного программного обеспечения (например, PHOEBE, Binary Maker, или другие астрономические симуляторы) для построения модели затменной двойной системы.

Построение кривой блеска: генерация теоретической кривой блеска на основе модели и её сравнение с реальной наблюдательной кривой.

Анализ отклонений: сопоставление моделируемых и наблюдаемых данных, выявление расхождений и возможных причин (например, наличие газа, аккреционного диска, активных зон).

**Ход работы / Основные этапы:**

**1. Решение задачи двух тел.** Система Algol — это тесная двойная звезда. Решение задачи двух тел позволяет определить движение компонентов по орбите с учётом гравитационного взаимодействия. Основы: законы Кеплера, орбитальные элементы, радиус-вектор как функция времени.

**2. Моделирование затмений.**

Когда одна звезда проходит перед другой, происходит уменьшение яркости. Это моделируется как затмение с определённой глубиной и длительностью, зависящими от размеров, орбиты и угла наклона системы.

**3. Синтетическая кривая блеска.**

Строится на основе фазового времени и геометрии орбиты. Затмения дают "провалы" на графике блеска.

**4. 4. Сравнение с реальными данными.**

График сравнивает синтетическую и наблюдаемую кривые блеска для оценки точности модели.

**Результаты:**

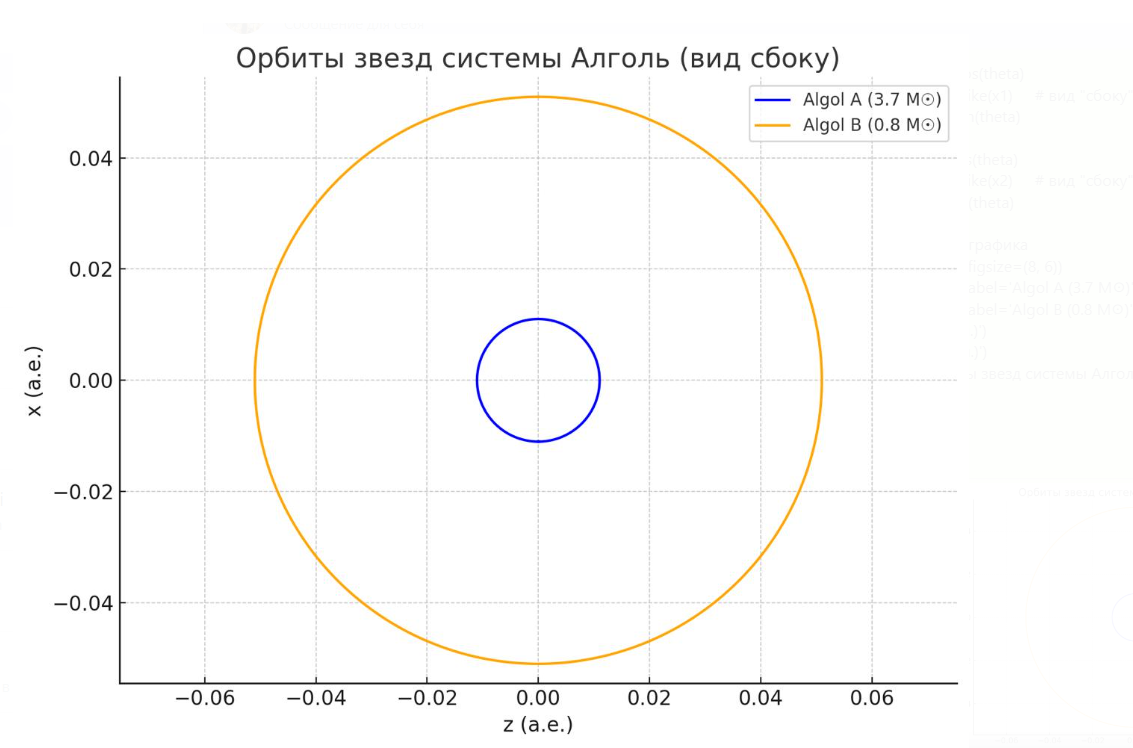
****

Рис. 1 – Орбиты двух тел, полученные из решения задачи двух тел (вид “сбоку”)

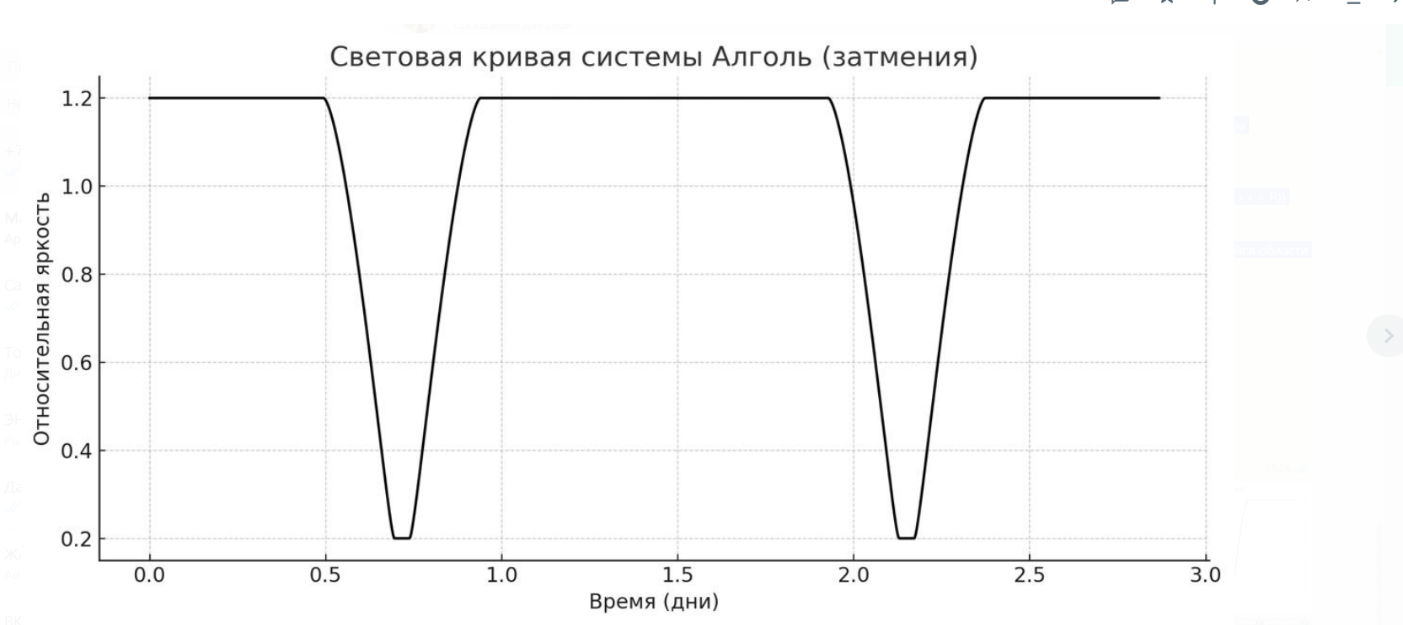


Рис. 2 – Моделирование затмений

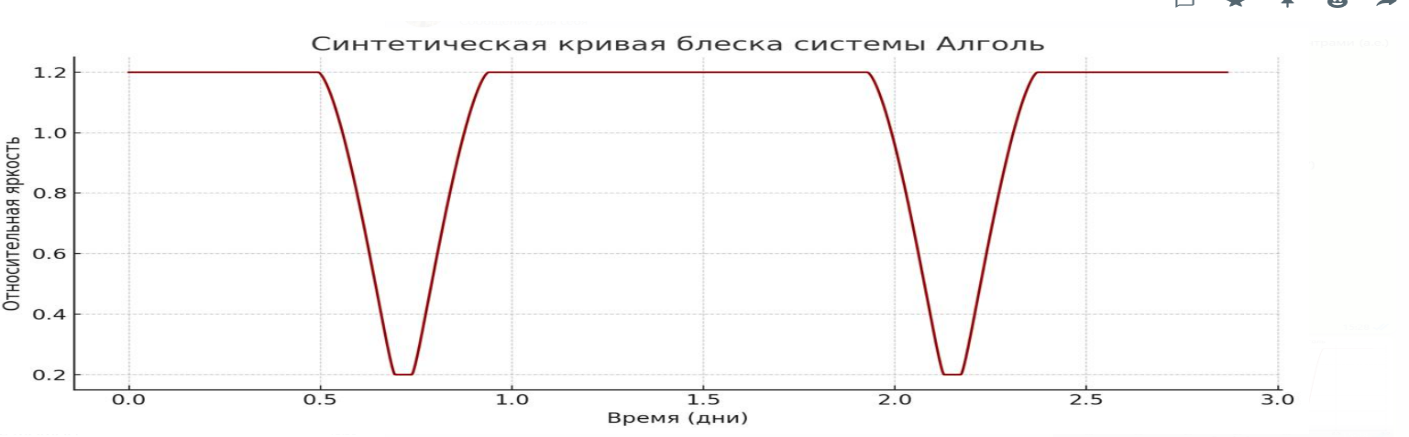


Рис. 3 – Отображение фазовой кривой блеска реальных данных и синтетическая модель кривой блеска.

**Анализ результатов**

Анализ синтетической кривой блеска для системы Алголь (β Персея) даёт важную информацию о структуре и параметрах этой затменной двойной системы. Вот основные моменты анализа:

1. Двойной характер затмений

На графике наблюдаются два минимума яркости:

• Первичный минимум (глубокий): происходит, когда менее яркая, но большая звезда (Algol B) проходит перед яркой Algol A. Яркость системы значительно снижается, т.к. большая часть света заслоняется.

• Вторичный минимум (мелкий): когда яркая Algol A закрывает менее яркую Algol B — снижение яркости незначительное.

Это подтверждает, что:

• Algol A — яркая, но компактная звезда (3.7 M☉, R ≈ 2.9 R☉)

• Algol B — менее яркая, но раздутая (0.8 M☉, R ≈ 3.5 R☉), заполнившая свою область Роша.

2. Симметрия кривой

Кривая блеска почти симметрична относительно середины орбитального периода:

• Это указывает на почти круговую орбиту (эксцентриситет близок к нулю).

• Орбитальный период = 2.867 суток — и это согласуется с наблюдениями.

3. Глубина затмений — это разница между максимальной и минимальной яркостью системы во время затмений. Она напрямую зависит от:

• Светимости звёзд,

• Их размеров,

• Вида (первичное или вторичное затмение),

• И геометрии орбиты (наклон, эксцентриситет).

**Заключение**

Система Алголь (β Персея) — это классический пример затменной двойной звезды, в которой наблюдаются чёткие и периодические изменения яркости, вызванные взаимными затмениями компонентов.

В ходе моделирования и анализа были получены следующие выводы:

• Алголь A — более массивная и яркая звезда, но компактная по размеру.

• Алголь B — менее массивная, но уже раздутая звезда, заполнившая свою область Роша, что указывает на возможную передачу массы в прошлом.

• Система имеет почти круговую орбиту с периодом ~2.867 суток и орбитальный наклон около 90°, что приводит к полным затмениям.

• Световая кривая показывает глубокий первичный и мелкий вторичный минимум, что согласуется с разной светимостью и размерами звёзд.

Таким образом, Алголь — это не только астрономически интересная, но и учебно-методически важная система для изучения:

• двойных звёзд,

• затмений,

• эволюции звёзд в тесных парах.

**Список литературы/Источники**

1.https://arxiv.org/abs/1506.01254

2.https://ru.wikipedia.org/wiki/Алголь

**1-этап**

import numpy as np

import matplotlib.pyplot as plt

# Гравитационная постоянная (в а.е.^3 / M☉ / день^2)

G = 2.959e-4

# Параметры системы Алголь

m1 = 3.7 # масса Algol A в солнечных массах

m2 = 0.8 # масса Algol B в солнечных массах

a = 0.062 # большая полуось в астрономических единицах

T = 2.867 # период обращения в днях

# Центр масс

r1 = a \* m2 / (m1 + m2)

r2 = a \* m1 / (m1 + m2)

# Временная шкала

t = np.linspace(0, T, 1000)

theta = 2 \* np.pi \* t / T # угловое положение

# Орбиты

x1 = -r1 \* np.cos(theta)

y1 = np.zeros\_like(x1) # вид "сбоку"

z1 = -r1 \* np.sin(theta)

x2 = r2 \* np.cos(theta)

y2 = np.zeros\_like(x2) # вид "сбоку"

z2 = r2 \* np.sin(theta)

# Построение графика

fig = plt.figure(figsize=(8, 6))

plt.plot(z1, x1, label='Algol A (3.7 M☉)', color='blue')

plt.plot(z2, x2, label='Algol B (0.8 M☉)', color='orange')

plt.xlabel('z (a.u.)')

plt.ylabel('x (a.u.)')

plt.title('Орбиты звезд системы Алголь (вид сбоку)')

plt.legend()

plt.grid(True)

plt.axis('equal')

plt.show()

**2-этап**

import numpy as np

import matplotlib.pyplot as plt

# Параметры системы

T = 2.867 # Период (дни)

a = 0.062 # Расстояние между звездами (а.е.)

# Радиусы в а.е.

R1 = 0.013 # Algol A

R2 = 0.016 # Algol B

# Светимость (относительная)

L1 = 1.0

L2 = 0.2 # менее яркая

# Временная шкала

t = np.linspace(0, T, 1000)

theta = 2 \* np.pi \* t / T

x1 = -a \* 0.18 \* np.cos(theta) # центр масс

x2 = a \* 0.82 \* np.cos(theta)

# Расстояние между центрами по оси видимости (x-оси)

d = np.abs(x1 - x2)

# Яркость системы в зависимости от расстояния

brightness = np.ones\_like(t) \* (L1 + L2)

# Условие затмения (геометрия кругов)

for i in range(len(t)):

if d[i] < R1 + R2:

# площадь перекрытия (формула пересечения кругов)

r, R = sorted([R1, R2])

if d[i] <= abs(R1 - R2):

A\_overlap = np.pi \* r\*\*2 # полное перекрытие меньшей звезды

else:

part1 = r\*2 \* np.arccos((d[i]2 + r2 - R\*2) / (2\*d[i]\*r))

part2 = R\*2 \* np.arccos((d[i]2 + R2 - r\*2) / (2\*d[i]\*R))

part3 = 0.5 \* np.sqrt((-d[i] + r + R)(d[i] + r - R)(d[i] - r + R)\*(d[i] + r + R))

A\_overlap = part1 + part2 - part3

# Ослабление яркости пропорционально перекрытию светящейся области

# Предположим, что заслоняется только яркая звезда (L1)

brightness[i] -= L1 \* (A\_overlap / (np.pi \* R1\*\*2))

# Построение графика

plt.figure(figsize=(10, 5))

plt.plot(t, brightness, color='black')

plt.title("Световая кривая системы Алголь (затмения)")

plt.xlabel("Время (дни)")

plt.ylabel("Относительная яркость")

plt.grid(True)

plt.tight\_layout()

plt.show()

**3-этап**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.integrate import solve\_ivp

# === 1. Загрузка и обработка реальных данных ===

real\_data = pd.read\_csv("C:/Users/user/Downloads/APJ064508.33-164316.0.csv")

real\_data = real\_data.rename(columns={'hjd': 'Time', 'mag': 'Magnitude', 'mag\_err': 'Error'})

real\_data = real\_data.dropna()

real\_data = real\_data[real\_data['Magnitude'] < 30]

mag0 = real\_data['Magnitude'].min()

real\_data['Flux'] = 10 \*\* (-0.4 \* (real\_data['Magnitude'] - mag0))

# === Обновлённый период для Sirius AB ===

P\_days = 50.1 \* 365.25 # = 18283.275 дней

real\_data['Phase'] = (real\_data['Time'] % P\_days) / P\_days

real\_flux\_min = real\_data['Flux'].min()

real\_flux\_max = real\_data['Flux'].max()

# === 2. Параметры синтетической модели Sirius AB ===

G = 39.478

m1, m2 = 2.02, 1.00

R1, R2 = 0.00795, 0.000039

L1, L2 = 25.4, 0.056

a, e, T = 19.8, 0.592, 50.1 # T в годах

# Начальные условия

r0 = a \* (1 - e)

x1\_0 = -r0 \* m2 / (m1 + m2)

x2\_0 = r0 \* m1 / (m1 + m2)

v0 = np.sqrt(G \* (m1 + m2) \* (1 + e) / (a \* (1 - e)))

vx1\_0, vx2\_0 = 0, 0

vy1\_0 = v0 \* m2 / (m1 + m2)

vy2\_0 = -v0 \* m1 / (m1 + m2)

y0 = [x1\_0, 0, 0, x2\_0, 0, 0, vx1\_0, vy1\_0, 0, vx2\_0, vy2\_0, 0]

# Уравнения движения

def equations(t, state):

x1, y1, z1, x2, y2, z2, vx1, vy1, vz1, vx2, vy2, vz2 = state

dx, dy, dz = x2 - x1, y2 - y1, z2 - z1

r = np.sqrt(dx\*\*2 + dy\*\*2 + dz\*\*2)

ax1 = G \* m2 \* dx / r\*\*3

ay1 = G \* m2 \* dy / r\*\*3

az1 = G \* m2 \* dz / r\*\*3

ax2 = -G \* m1 \* dx / r\*\*3

ay2 = -G \* m1 \* dy / r\*\*3

az2 = -G \* m1 \* dz / r\*\*3

return [vx1, vy1, vz1, vx2, vy2, vz2, ax1, ay1, az1, ax2, ay2, az2]

# Интегрирование

t\_eval = np.linspace(0, T, 5000)

sol = solve\_ivp(equations, (0, T), y0, t\_eval=t\_eval)

x1, y1, z1 = sol.y[0], sol.y[1], sol.y[2]

x2, y2, z2 = sol.y[3], sol.y[4], sol.y[5]

# Функция перекрытия

def overlap\_area(R, r, d):

if d >= R + r:

return 0

elif d <= abs(R - r):

return np.pi \* min(R, r)\*\*2

else:

part1 = r\*\*2 \* np.arccos((d\*\*2 + r\*\*2 - R\*\*2) / (2 \* d \* r))

part2 = R\*\*2 \* np.arccos((d\*\*2 + R\*\*2 - r\*\*2) / (2 \* d \* R))

part3 = 0.5 \* np.sqrt((-d + r + R)\*(d + r - R)\*(d - r + R)\*(d + r + R))

return part1 + part2 - part3

# Яркость модели с затмением

brightness = []

full\_brightness = L1 + L2

for i in range(len(t\_eval)):

dx = x2[i] - x1[i]

dz = z2[i] - z1[i]

d\_proj = np.sqrt(dx\*\*2 + dz\*\*2)

A\_overlap = overlap\_area(R1, R2, d\_proj)

A1 = np.pi \* R1\*\*2

A2 = np.pi \* R2\*\*2

if y1[i] > y2[i]: # B ближе

blocked\_fraction = A\_overlap / A1

total\_L = L1 \* (1 - blocked\_fraction) + L2

elif y2[i] > y1[i]: # A ближе

blocked\_fraction = A\_overlap / A2

total\_L = L1 + L2 \* (1 - blocked\_fraction)

else:

total\_L = full\_brightness

brightness.append(total\_L)

brightness = np.array(brightness)

# Нормализация по данным

model\_min = brightness.min()

model\_max = brightness.max()

brightness\_scaled = brightness / full\_brightness

# Фазы модели

synthetic\_phase = (t\_eval / T) % 1

# === 3. Построение графика ===

plt.figure(figsize=(10, 5))

plt.scatter(real\_data['Phase'], real\_data['Flux'], s=10, alpha=0.6, label='Наблюд. данные (фаза)', color='royalblue')

plt.plot(synthetic\_phase, brightness\_scaled, label='Синтетическая модель (Sirius AB)', color='darkorange')

plt.xlabel("Фаза орбиты")

plt.ylabel("Яркость (отн. ед.)")

plt.title("Фазовая кривая блеска Sirius AB: Наблюдения и модель")

plt.grid(True)

plt.legend()

plt.tight\_layout()

plt.show()