

Астрономическая кегля

М.А.Забелкин

Научный руководитель: А.С.Байгашов

Аннотация

В ходе данной работы на языке программирования *python* написана функция, изображающая аналемму солнца на заданную широту и местное среднее солнечное время. Для написания функции были использованы библиотеки *numpy* и *matplotlib*.

Введение

Аналемма – кривая, соединяющая ряд последовательных положений центральной звезды планетной системы (в нашем случае – Солнца) на небосводе одной из планет этой системы в одно и то же время суток в течение года.



Аналемма похожа на неправильную цифру восемь, причем в северном полушарии Земли верхняя часть этой восьмерки меньше, чем нижняя. В южном полушарии она, естественно, перевернута. Ее существование обусловлено тем, что ось вращения Земли наклонена к плоскости ее орбиты вокруг Солнца на $23^{\circ},45'$ и что эта орбита эллиптическая, а значит, Земля движется по ней не равномерно.

Постановка задачи

Для описания аналеммы необходимо рассчитать следующие параметры.

$$\eta = 7.53 \cos B + 1.5 \sin B - 9.87 \sin 2B$$
$$B = \frac{2\pi(x - 81)}{T}$$

Уравнение времени(η) на заданную дату (x - номер дня от 1 января). **Уравнение времени** — разница между средним солнечным и истинным солнечным временем.

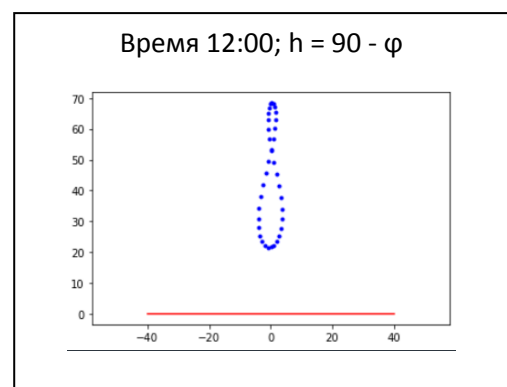
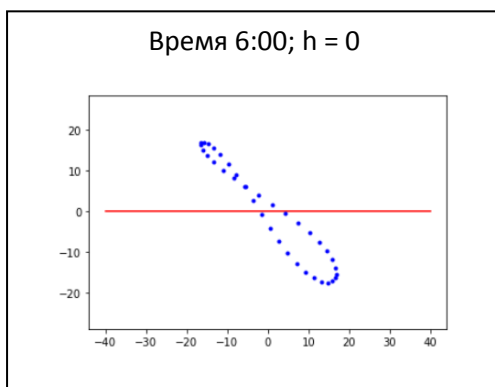
$$\delta = \sin^{-1} \left(\sin \varepsilon \cdot \sin \frac{2 \cdot \pi \cdot (x - 81)}{T} \right)$$

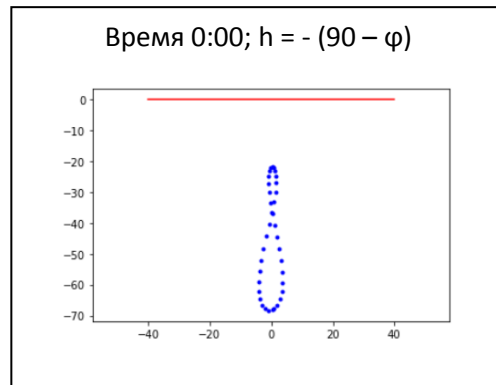
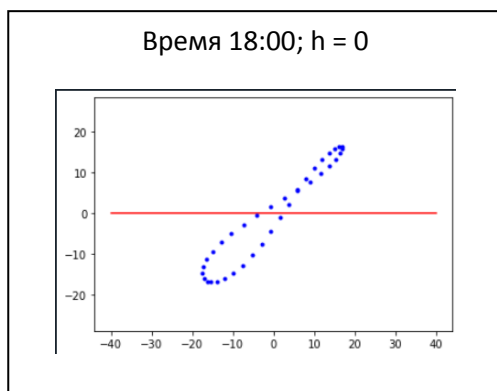
Формула склонения(δ) солнца на заданную дату (x - номер дня от 1 января).

По этим 2 уравнениям мы можем построить аналемму солнца.

$$h = \sin \left((t - 6) \cdot \frac{2}{24} \cdot \pi \right) \cdot (90 - |\varphi|)$$

Формула расчёта высоты(h) центра аналеммы на заданной широте в заданное местное среднее солнечное время. Синус получен из следующих соображений:

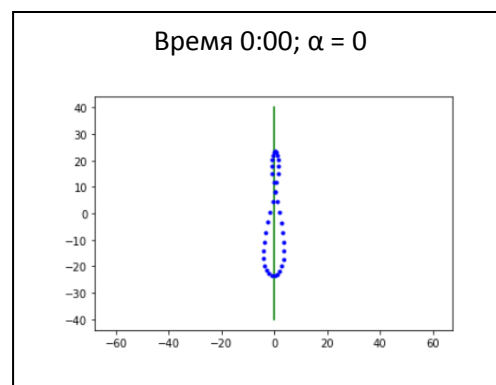
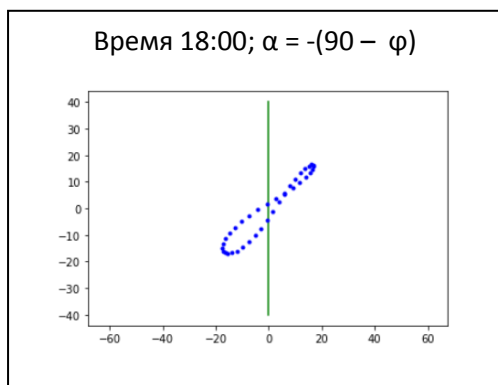
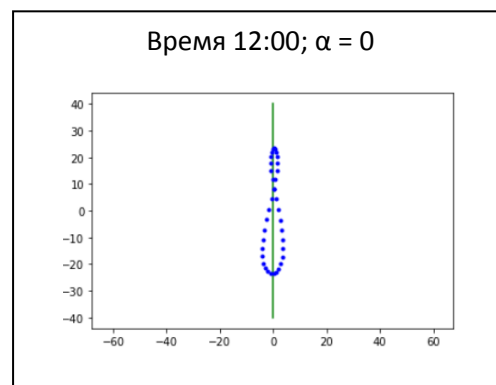
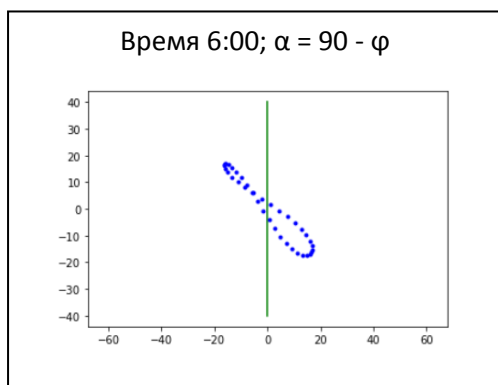




Высота h меняется от 0 до \max , от \max до 0, от 0 до $-\max$ и от $-\max$ до 0.

$$\alpha = \cos\left((t - 6) \cdot \frac{2}{24} \cdot \pi\right) \cdot (90 - |\varphi|)$$

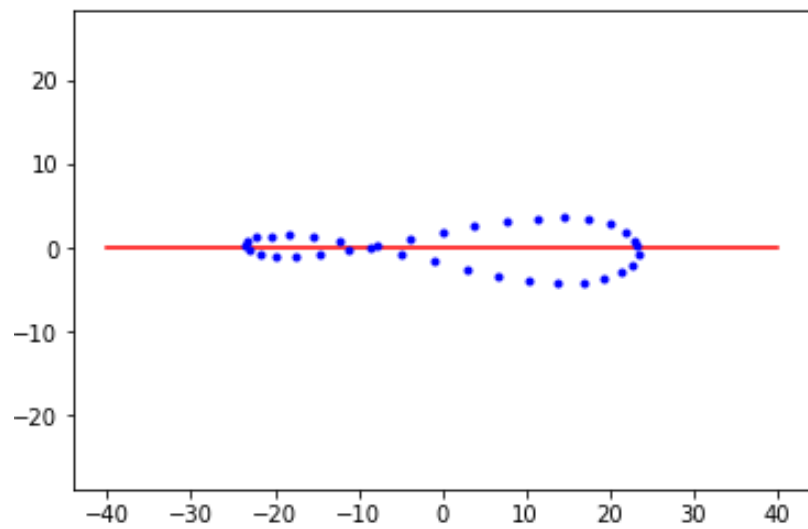
Формула расчёта угла наклона (α) аналеммы к прямой, перпендикулярной горизонту. Косинус получен из следующих соображений:



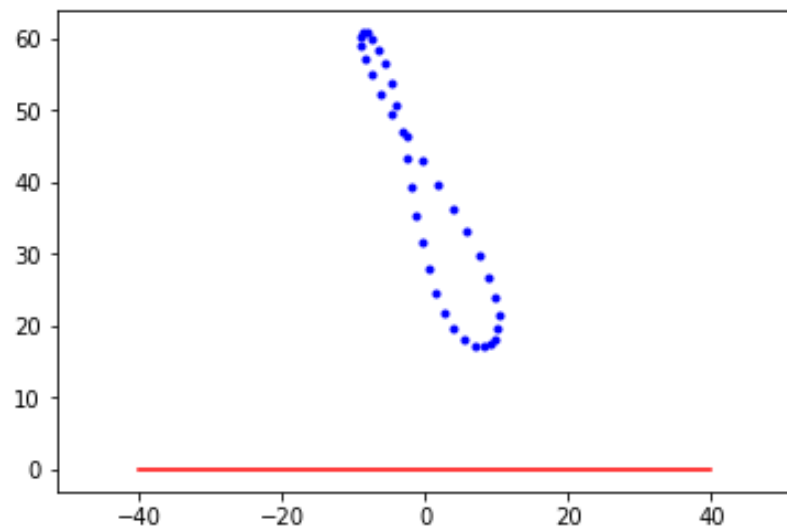
Угол α меняется от \max до 0, от 0 до $-\max$, от $-\max$ до 0 и от 0 до \max .

Результаты моделирования

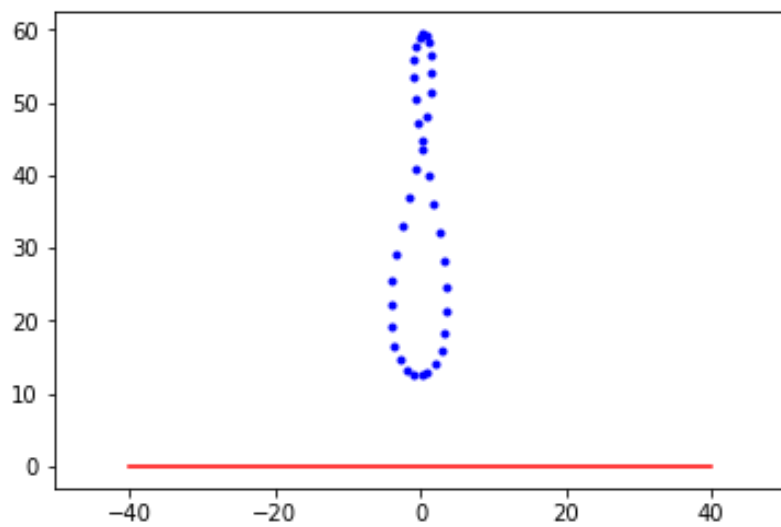
В результате моделирования получены следующие изображения аналеммы (это лишь малая часть всех изображений, которые можно получить). В полученных изображениях можно увидеть различия в наклоне аналеммы и её высоты над горизонтом, в зависимости от широты и местного времени. В южных широтах аналемма перевернута.



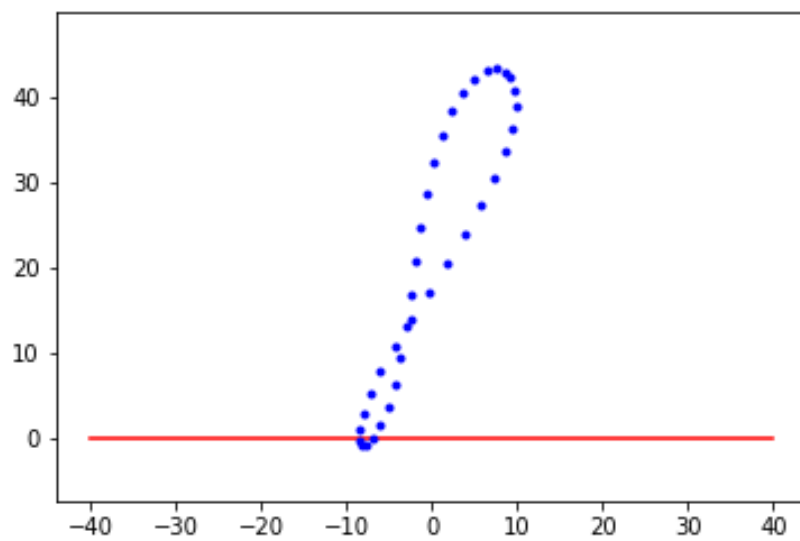
Широта 0 | Время 6:00



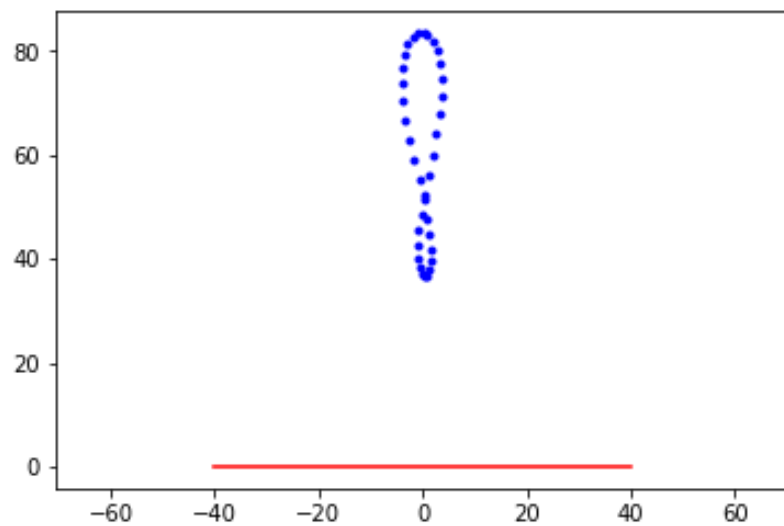
Широта 45 с.ш. | Время 10:00



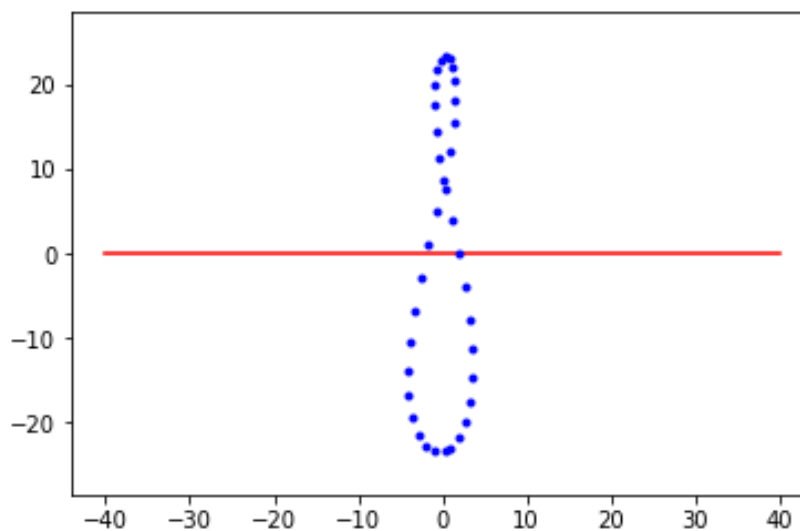
Широта 54 с.ш. | Время 12:00



Широта 60 ю.ш. | Время 9:00



Широта 30 ю.ш. | Время 12:00



Широта 90 с.ш. | Время 3:00

Закключение и перспективы

В ходе работы написана функция, изображающая аналемму на заданную широту и местное среднее солнечное время. Данная функция может быть использована для составления задач по астрономии; фотографами, желающими сделать фотографии аналеммы, для примерного определения композиции. В перспективе можно написать к данной программе приложение на мобильные устройства.

Листинг кода:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def analemma(phi = 45, time_h = 6, time_m = 0):
5     #Функция, рисующая аналемму на заданную широту и время.(phi<0 - южные широты)
6     #Красная линия - горизонт
7     E = 23.44
8     dates = np.arange(1,365,10)
9     B = 2 * np.pi * (dates - 81)/365.2564
10
11     x0 = 7.53 * np.cos(B) + 1.5 * np.sin(B) - 9.87 * np.sin(2 * B)
12     x0 = x0*360/60/24
13     y0 = np.arcsin(np.sin(E*np.pi/180)*np.sin(2*np.pi*(dates-80)/365.2422)) * 180 / np.pi
14     t = time_h + time_m/60
15     alpha = np.cos((t-6) * 2 / 24 * np.pi )*(90-abs(phi))
16     h = np.sin((t-6)*2/24*np.pi)*(90-abs(phi))
17     x = x0 * np.cos(-alpha*np.pi/180) + y0 * np.sin(-alpha*np.pi/180)
18     if phi >=0:
19         y = y0 * np.cos(-alpha*np.pi/180) - x0 * np.sin(-alpha*np.pi/180) + h
20     else:
21         y = -(y0 * np.cos(-alpha*np.pi/180) - x0 * np.sin(-alpha*np.pi/180)) + h
22
23     plt.axis('equal')
24     plt.plot([-40,40],[0,0], '-', color = 'r')
25     plt.plot(x,y,'.',color = 'b')
26
27 analemma(45,18)
28
29
30
31
32
```