Отчёт по лабораторной работе 2

дисциплина: Математическое моделирование

Василиса Михайловна Крючкова, НПИбд-02-18

Содержание

# Цель работы

Решить задачу о погоне, построить графики с помощью Python.

# Задание

**Вариант 41** На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 17,4 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 4,8 раза больше скорости браконьерской лодки.

1. Вывести дифференциальное уравнение, описывающее движение катера, с начальными условиями.
2. Построить траектории движения катера и лодки для двух случаев.
3. Определить точку пересечения катера и лодки.

# Выполнение лабораторной работы

**1. Вывод дифференциального уравнения**

1.1. Принимаем за , – место нахождения лодки браконьеров в момент обнаружения, км – место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки.

1.2. Введем полярные координаты. Считаем, что полюс – это точка обнаружения лодки браконьеров , а полярная ось проходит через точку нахождения катера береговой охраны. (см. рис. 1)

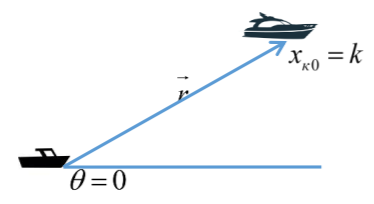


Figure 1: Положение катера и лодки в начальный момент времени

1.3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса , только в этом случае траектория катера пересечется с траекторией лодки. Поэтому для начала катер береговой охраны должен двигаться некоторое время прямолинейно, пока не окажется на том же расстоянии от полюса, что и лодка браконьеров. После этого катер береговой охраны должен двигаться вокруг полюса удаляясь от него с той же скоростью, что и лодка браконьеров.

1.4. Чтобы найти расстояние x (расстояние, после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение. Пусть через время катер и лодка окажутся на одном расстоянии x от полюса. За это время лодка пройдет , а катер (или , в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как $x \over v$ или $17,4 - x \over 4,8v$ (во втором случае $17,4 + x \over 4,8v$). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние можно найти из следующего(-их) уравнения(-й):

Тогда (км), а (км), задачу будем решать для двух случаев.

1.5. После того, как катер береговой охраны окажется на одном расстоянии от полюса, что и лодка, он должен сменить прямолинейную траекторию и начать двигаться вокруг полюса удаляясь от него со скоростью лодки . Для этого скорость катера раскладываем на две составляющие: – радиальная скорость и – тангенциальная скорость. (см. рис. 2)

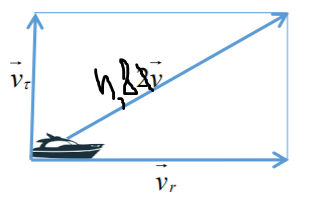


Figure 2: Разложение скорости катера на тангенциальную и радиальную составляющие

Радиальная скорость – это скорость, с которой катер удаляется от полюса, . Нам нужно, чтобы эта скорость была равна скорости лодки, поэтому полагаем .

Тангенциальная скорость – это линейная скорость вращения катера относительно полюса. Она равна произведению угловой скорости на радиус, .

Из рис. 2 по теореме Пифагора: , тогда получаем .

1.6. Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений:

Исключая из полученной системы производную по t, можно перейти к следующему уравнению:

Решив это уравнение, я получу траекторию движения катера в полярных координатах. Начальные условия:

**2. Построение траекторий движения катера и лодки**

2.1. Написала программу на Python:

import math  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt  
  
k = 17.4  
fi = 3\*math.pi/4  
  
#функция, описывающая движение катера береговой охраны  
def dr(r, tetha):   
 dr = 5\*r/math.sqrt(551)  
 return dr  
  
r01 = 5/29\*k #1 случай  
r02 = 5/19\*k #2 случай  
  
te = np.arange(0, 2\*math.pi, 0.01)  
  
r1 = odeint(dr, r01, te)  
r2 = odeint(dr, r02, te)  
  
#функция, описывающая движение лодки браконьеров  
def xt(t):   
 xt = math.tan(fi)\*t  
 return xt  
  
t = np.arange(0, 20, 1)  
  
#Перевод в полярные координаты  
tete = (np.tan(xt(t)/t))\*\*-1  
rr = np.sqrt(t\*t + xt(t)\*xt(t))  
  
#построение траектории движения катера в полярных координатах. 1 случай  
plt.polar(te, r1, 'g')  
#построение траектории движения лодки в полярных координатах  
plt.polar(tete, rr, 'b')   
  
#построение траектории движения катера в полярных координатах. 2 случай  
plt.polar(te, r2, 'g')  
#построение траектории движения лодки в полярных координатах  
plt.polar(tete, rr, 'b')

2.2. Получила следующие графики:(см. рис. 3 и 4)

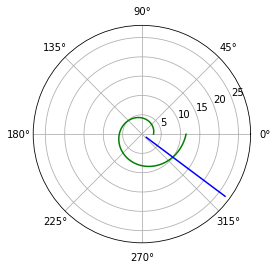


Figure 3: Траектории движения катера и лодки. 1 случай

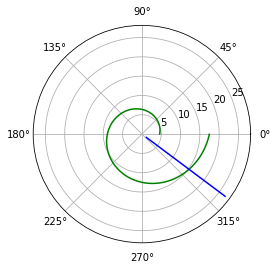


Figure 4: Траектории движения катера и лодки. 2 случай

**3. Точка пересечения**

3.1. Для определения точки пересечения я добавила в конце программы:

#для 1 случая  
idx = np.argwhere(np.diff(np.sign(rr - r1))).flatten()  
print (tete[-1])  
print (rr[idx[-1]])  
  
#для 2 случая  
idd = np.argwhere(np.diff(np.sign(rr - r2))).flatten()  
print (tete[-1])  
print (rr[idd[-1]])

3.2. В итоге я получила, что в 1 случае точка пересечения: , а во 2 случае: .

# Выводы

Решила задачу о погоне, построила графики с помощью Python.