

Отчёт по лабораторной работе 3

дисциплина: Математическое моделирование

Пейтель Андрей Андреевич, НПИбд-02-18

Содержание

Цель работы	1
Задание	1
Выполнение лабораторной работы	2
Выводы	7

Цель работы

Построить упрощенную модель боевых действий с помощью Python.

Задание

Вариант 7 Между страной X и страной Y идет война. Численности состава войск исчисляются от начала войны и являются временными функциями $x(t)$ и $y(t)$. В начальный момент времени страна X имеет армию численностью 24 000 человек, а в распоряжении страны Y армия численностью в 9 500 человек. Для упрощения модели считаем, что коэффициенты a, b, c, h постоянны. Также считаем $P(t)$ и $Q(t)$ непрерывными функциями.

Постройте графики изменения численности войск армии X и армии Y для следующих случаев:

1. Модель боевых действий между регулярными войсками

$$\frac{\partial x}{\partial t} = -0,3(t) - 0,87y(t) + |\sin(2t) + 1|$$

$$\frac{\partial y}{\partial t} = -0,5x(t) - 0,41y(t) + |\cos(3t) + 1|$$

2. Модель ведение боевых действий с участием регулярных войск и партизанских отрядов

$$\frac{\partial x}{\partial t} = -0,25x(t) - 0,64y(t) + |\sin(2t + 4)|$$

$$\frac{\partial y}{\partial t} = -0,2x(t)y(t) - 0,52y(t) + |\cos(t + 4)|$$

Выполнение лабораторной работы

1. Боевые действия между регулярными войсками

1.1. Изучил начальные условия. Коэффициент смертности, не связанный с боевыми действиями, у первой армии 0,3, а у второй – 0,41. Коэффициент эффективности первой и второй армии 0,5 и 0,87 соответственно. Функция, описывающая подход подкрепление первой армии, $P(t) = |\sin(2t) + 1|$, подкрепление второй армии описывается функцией $Q(t) = |\cos(3t) + 1|$. $x_0 = 24000$ – численность 1-ой армии, $y_0 = 9500$ – численность 2-ой армии.

1.2. Оформил начальные условия в код на Python:

```
x0 = 24000
y0 = 9500

a1 = 0.3
b1 = 0.87
c1 = 0.5
h1 = 0.41

def P1(t):
    p1 = np.fabs(np.sin (2t) +1)
    return p1
def Q1(t):
    q1 = np.fabs(np.cos(3t) +1)
    return q1
```

1.3. Для времени задал следующие условия: $t_0 = 0$ – начальный момент времени, $t_{max} = 1$ – предельный момент времени, $dt = 0,05$ – шаг изменения времени.

1.4. Добавил в программу условия, описывающие время:

```
t0 = 0
tmax = 1
dt = 0.05
t = np.arange(t0, tmax, dt)
```

1.5. Запрограммировал заданную систему дифференциальных уравнений, описывающих изменение численности армий:

```
def S1(f, t):
    s11 = -a1*f[0] - b1*f[1] + P1(t)
    s12 = -c1*f[0] - h1*f[1] + Q1(t)
    return s11, s12
```

1.6. Создал вектор начальной численности армий:

```
v = np.array([x0, y0])
```

1.7. Запрограммировал решение системы уравнений:

```
f1 = odeint(S1, v, t)
```

1.8. Описал построение графика изменения численности армий:

```
plt.plot(t, f1)
```

2. Боевые действия с участием регулярных войск и партизанских отрядов

2.1. Изучил начальные условия. Коэффициент смертности, не связанный с боевыми действиями, у первой армии 0,25, а у второй – 0,52. Коэффициент эффективности первой и второй армии 0,2 и 0,64 соответственно. Функция, описывающая подход подкрепление первой армии, $P(t) = |\sin(2t + 4)|$, подкрепление второй армии описывается функцией $Q(t) = |\cos(t + 4)|$. Изначальная численность армий такая же, как и в п. 1.1.

2.2. Дополнил начальные условия в коде на Python:

```
a2 = 0.25
```

```
b2 = 0.64
```

```
c2 = 0.2
```

```
h2 = 0.52
```

```
def P2(t):
```

```
    p2 = np.fabs(np.sin(2t+4))
```

```
    return p2
```

```
def Q2(t):
```

```
    q2 = np.fabs(np.cos(t+4))
```

```
    return q2
```

2.3. Условия для времени оставил такие же, как и в п. 1.3, соответственно, не дублировала их в программе.

2.4. Запрограммировал заданную систему дифференциальных уравнений, описывающих изменение численности армий:

```
def S2(f, t):
```

```
    s21 = -a2*f[0] - b2*f[1] + P2(t)
```

```
    s22 = -c2*f[0]*f[1] - h2*f[1] + Q2(t)
```

```
    return s21, s22
```

2.5. Т. к. начальная численность армий не изменилась, вектор начальных условий тоже не менял.

2.6. Запрограммировал решение системы уравнений:

```
f2 = odeint(S2, v, t)
```

2.7. Описал построение графика изменения численности армий:

```
plt.plot(t, f2)
```

3. Сборка программы

3.1. Собрал код программы воедино и получила следующий код:

```
import math
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
```

```
x0 = 24000
```

```
y0 = 9500
```

```
a1 = 0.3
```

```
b1 = 0.87
```

```
c1 = 0.5
```

```
h1 = 0.41
```

```
a2 = 0.25
```

```
b2 = 0.64
```

```
c2 = 0.2
```

```
h2 = 0.52
```

```
t0 = 0
```

```
tmax = 1
```

```
dt = 0.05
```

```
t = np.arange(t0, tmax, dt)
```

```
def P1(t):
```

```
    p1 = np.fabs(np.sin(2t)+1)
```

```
    return p1
```

```
def Q1(t):
```

```
    q1 = np.fabs(np.cos(3t)+1)
```

```
    return q1
```

```
def P2(t):
```

```
    p2 = np.fabs(np.sin(2t+4))
```

```
    return p2
```

```
def Q2(t):
```

```
    q2 = np.fabs(np.cos(t+4))
```

```
    return q2
```

```
def S1(f, t):
```

```
    s11 = -a1*f[0] - b1*f[1] + P1(t)
```

```
    s12 = -c1*f[0] - h1*f[1] + Q1(t)
```

```
    return s11, s12
```

```
def S2(f, t):
```

```
    s21 = -a2*f[0] - b2*f[1] + P2(t)
```

```
    s22 = -c2*f[0]*f[1] - h2*f[1] + Q2(t)
```

```
    return s21, s22
```

```
v = np.array([x0, y0])
```

```
f1 = odeint(S1, v, t)
```

```
f2 = odeint(S2, v, t)
```

```
plt.plot(t, f1)
```

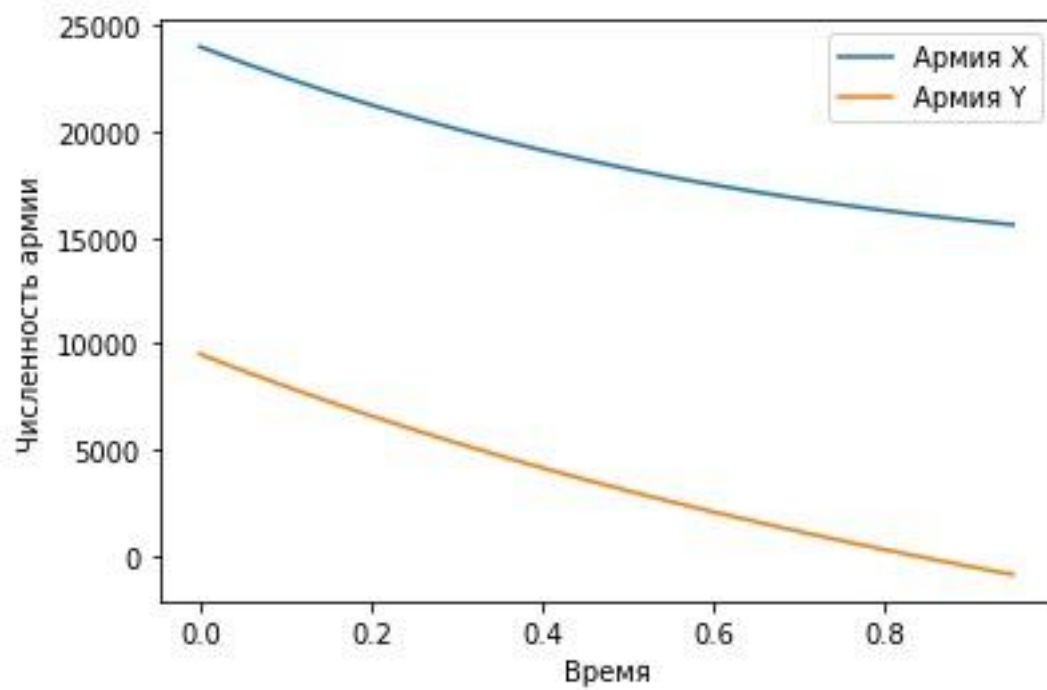
```
plt.ylabel('Численность армии')
```

```
plt.xlabel('Время')
```

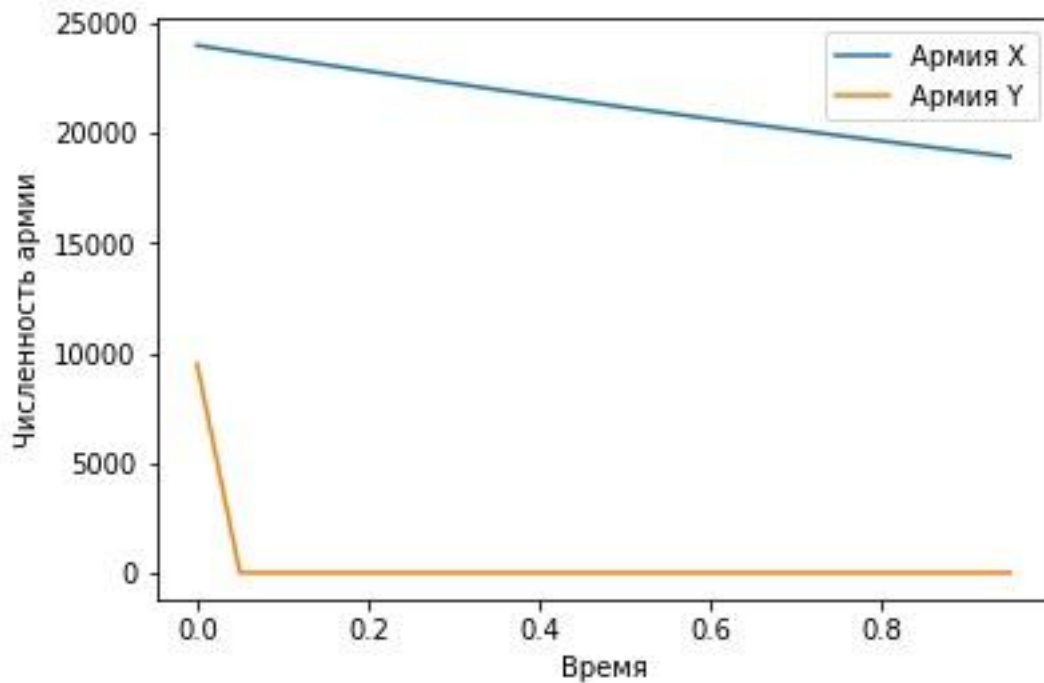
```
plt.legend(['Армия X', 'Армия Y'])
```

```
plt.plot(t, f2)
```

```
plt.ylabel('Численность армии')  
plt.xlabel('Время')  
plt.legend(['Армия X', 'Армия Y'])
```



Боевые действия между регулярными войсками



Боевые действия с участием регулярных войск и партизанских отрядов

Выводы

Построил упрощенную модель боевых действий с помощью Python.

В боевых действиях между регулярными войсками победит армия X, причем ей на это потребуется довольно много времени (видим по графику, что численность армии Y будет на исходе практически в предельный момент времени).

В боевых действиях с участием регулярных войск и партизанских отрядов также победит армия X, но уже намного быстрее, чем в 1-ом случае (видим по графику, что армия Y потеряла всех бойцов практически сразу после начала войны).