Отчёт по лабораторной работе 3

дисциплина: Математическое моделирование

Никитаева Александра Семеновна, НПИбд-02-18

Содержание

List of Tables

List of Figures

1 Цель работы

Построить упрощенную модель боевых действий с помощью Python.

2 Задание

Вариант 18 Между страной и страной идет война. Численности состава войск исчисляются от начала войны и являются временными функциями $x(t)$ и $y(t)$. В начальный момент времени страна имеет армию численностью 105 000 человек, а в распоряжении страны армия численностью в 95 000 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 и армии для следующих случаев:

1. Модель боевых действий между регулярными войсками

$$\frac{\partial x}{\partial t} = -0,35x(t) - 0,45y(t) + 2sin(t)$$

$$\frac{\partial y}{\partial t} = -0,69x(t) - 0,61y(t) + cos(t) + 1$$

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

$$\frac{\partial x}{\partial t} = -0,35x(t) - 0,73y(t) + 2sin(2t)$$

$$\frac{\partial y}{\partial t} = -0,45x(t)y(t) - 0,41y(t) + cos(t) + 1$$

3 Выполнение лабораторной работы

1. Боевые действия между регулярными войсками

1.1. Изучила начальные условия. Коэффициент смертности, не связанный с боевыми действиями, у первой армии 0,35, а у второй – 0,45. Коэффициент эффективности первой и второй армии 0,69 и 0,61 соответственно. Функция, описывающая подход подкрепление первой армии, $P(t) = 2sin(t)$, подкрепление второй армии описывается функцией $Q(t) = cos(t) + 1$. $x_0 = 105000$ – численность 1-ой армии, $y_0 = 95000$ – численность 2-ой армии.

1.2. Оформила начальные условия в код на Python:

```
x0 = 105000
y0 = 95000

a1 = 0.35
b1 = 0.45
c1 = 0.69
h1 = 0.61

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

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

1.4. Добавила в программу условия, описывающие время:

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

1.5. Запрограммировала 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. Создала вектор начальной численности армий:

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

1.7. Запрограммировала решение системы уравнений:

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

1.8. Описала построение графика изменения численности армий:

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

2. Боевые действия с участием регулярных войск и партизанских отрядов

2.1. Изучила начальные условия. Коэффициент смертности, не связанный с боевыми действиями, у первой армии 0,35, а у второй – 0,73. Коэффициент эффективности первой и второй армии 0,45 и 0,41 соответственно. Функция, описывающая подход подкрепление первой армии, $P(t) = 2sin(2t)$, подкрепление второй армии описывается функцией $Q(t) = cos(t) + 1$. Изначальная численность армий такая же, как и в п. 1.1.

2.2. Дополнила начальные условия в коде на Python:

```
a2 = 0.35
b2 = 0.73
c2 = 0.45
h2 = 0.41


def P2(t):
    p2 = 2*np.sin(2*t)
    return p2
def Q2(t):
    q2 = np.cos(t)+1
    return q2
```

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

2.4. Запрограммировала 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 меняла.

2.6. Запрограммировала решение системы уравнений:

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

2.7. Описала построение графика изменения численности армий:

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

3. Сборка программы

3.1. Собрала код программы воедино:

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

x0 = 105000
y0 = 95000

a1 = 0.35
b1 = 0.45
c1 = 0.69
h1 = 0.61

a2 = 0.35
b2 = 0.73
c2 = 0.45
h2 = 0.41

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

def P1(t):
```

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

def P2(t):
    p2 = 2*np.sin(2*t)
    return p2
def Q2(t):
    q2 = np.cos(t)+1
    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.plot(t, f2)
plt.ylabel('Численность армии')
plt.xlabel('Время')
```

3.2. Получила графики изменения численностей армий (см. рис. 3.1 и 3.2):

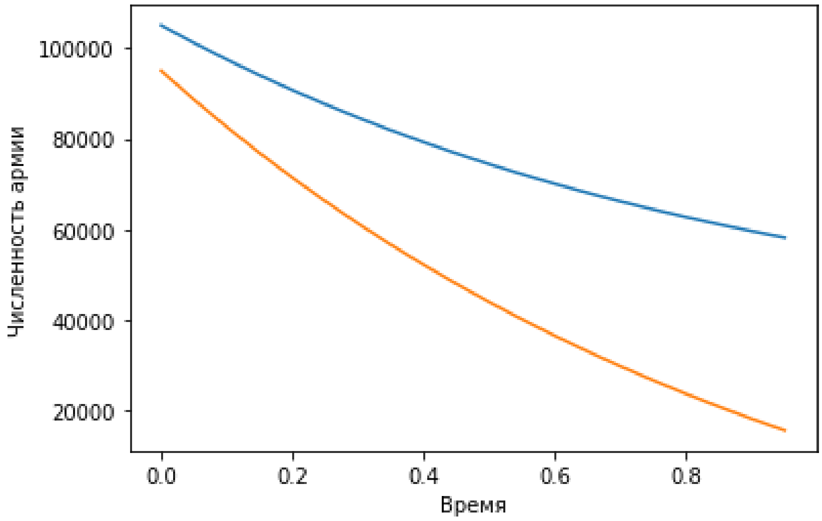

Figure 3.1: Боевые действия между регулярными войсками

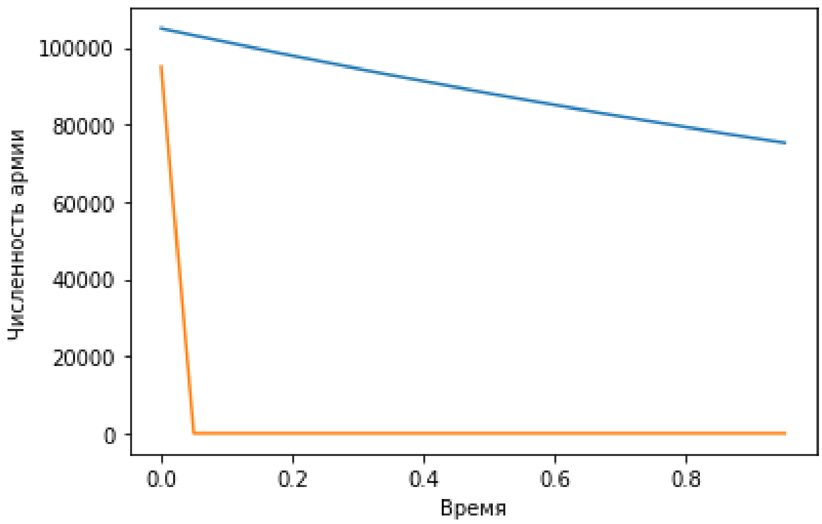

Figure 3.2: Боевые действия с участием регулярных войск и партизанских отрядов

4 Выводы

Построила 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 практический 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).