

Отчет по лабораторной работе 3

НФИбд-02-18

Оразклычев Довлет

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	16

List of Tables

List of Figures

2.1	Модели боевых действий	6
3.1	График модели первых боевых действий	10
3.2	График модели вторых боевых действий	10

1 Цель работы

Решить задачу о погоне

2 Задание

Между страной X и страной Y идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями $x(t)$ и $y(t)$. В начальный момент времени страна X имеет армию численностью 50 000 человек, а в распоряжении страны Y армия численностью в 69 000 человек. Для упрощения модели считаем, что коэффициенты a, b, c, h постоянны. Также считаем $P(t)$ и $Q(t)$ непрерывные функции.

Постройте графики изменения численности войск армии X и армии Y для следующих случаев: (рис. 2.1)

1. Модель боевых действий между регулярными войсками

$$\frac{dx}{dt} = -0,34x(t) - 0,72y(t) + \sin(t + 10)$$

$$\frac{dy}{dt} = -0,89x(t) - 0,43y(t) + \cos(t + 20)$$

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

$$\frac{dx}{dt} = -0,12x(t) - 0,51y(t) + \sin(20t)$$

$$\frac{dy}{dt} = -0,3x(t)y(t) - 0,61y(t) + \cos(13t)$$

Figure 2.1: Модели боевых действий

3 Выполнение лабораторной работы

Для начала мы импортируем библиотеки для построения кода и вводим наши переменные:

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

```
x0 = 50000
y0 = 69000
t0 = 0
```

```
a = 0.34
b = 0.72
c = 0.89
h = 0.43
```

```
tmax = 3
dt = 0.05
```

График будет таков: “Время” / “Численность армии”

Теперь мы создаем список значений t , которое мы будем использовать чтобы вычислять поточечно значения “Численность армии”:

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

Обратите внимание, что я также добавил элемент t_{\max} в конец списка. Дело в том, что функция `np.arange` заполняет от нуля до $t_{\max} - dt$, поэтому надо добавлять еще один элемент отдельно.

Теперь создаем непрерывные функции и систему уравнений:

```
def p(t):
    return (math.sin(t+10))
```

```
def q(t):
    return (math.cos(t+20))
```

```
def syst(f,t):
    x = f[0]
    y = f[1]
    dxdt = -a*x - b*y + p(t)
    dydt = -c*x - h*y + q(t)
    return (dxdt,dydt)
```

Создаем вектор значений наших данных и запускаем команду `odeint`, которая найдет значения “Численность армии” поточечно.

```
v0 = (x0, y0)
```

```
yf = odeint(syst, v0, t)
```

```
x = []
```



```
y = []
```

```
for i in range(len(yf)):
    x.append(yf[i][0])
    y.append(yf[i][1])
```

```
zero = []
for i in range (len(t)):
    zero = np.append(zero,0)
```

Создаем zero чтобы видеть полосу, где численность армии будет равняться нулю. Когда один из графиков пересечется с ним, то это будет значить, что кол-во людей в армии достигла нуля.

Теперь создаем график и выводим на экран.

Численность армии x будет “r”, т.е. красным. Численность армии y будет “g”, т.е. зеленым. Легенды ставим на правых верхний угол.

```
plt.figure(figsize = (20, 15))
plt.plot(t, zero, 'b')
plt.plot(t, x, 'r', label = 'x')
plt.plot(t, y, 'g', label = 'y')
plt.ylabel('Численность армии')
plt.xlabel('Время')
plt.title('Модель боевых действий №1')
plt.legend(loc = 'upper right')
```

```
plt.show()
```

И получаем: (рис. 3.1)

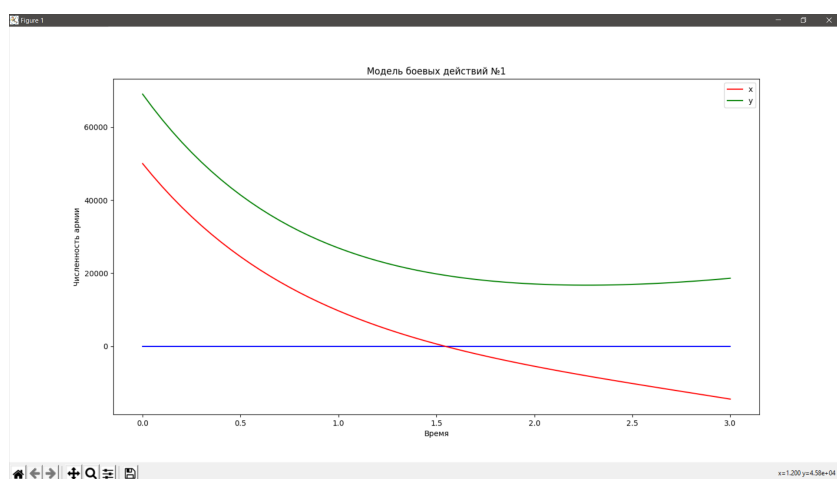


Figure 3.1: График модели первых боевых действий

(рис. 3.1)

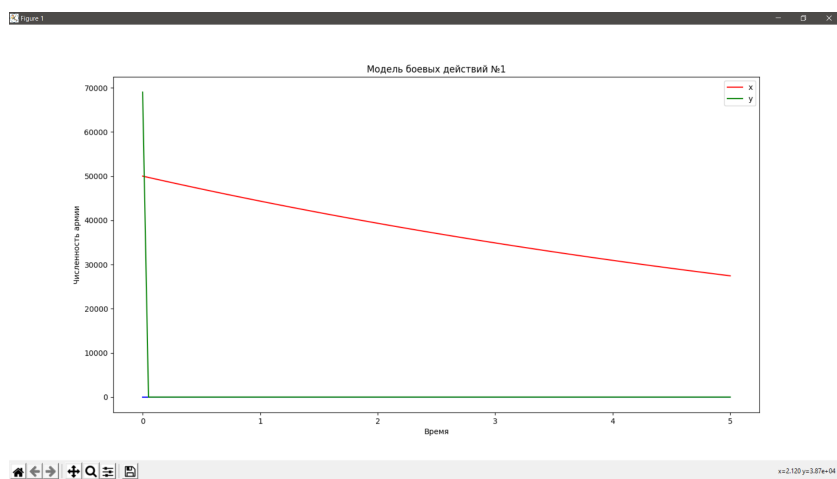


Figure 3.2: График модели вторых боевых действий

Код на Scilab для случая 1:

```
import numpy as np
import pandas as pd
import scipy as sp
import math
import matplotlib.pyplot as plt
```

```
from scipy.integrate import odeint
```

```
x0 = 50000
```

```
y0 = 69000
```

```
t0 = 0
```

```
a = 0.34
```

```
b = 0.72
```

```
c = 0.89
```

```
h = 0.43
```

```
tmax = 3
```

```
dt = 0.05
```

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

```
t = np.append(t, tmax)
```

```
def p(t):
```

```
    return (math.sin(t+10))
```

```
def q(t):
```

```
    return (math.cos(t+20))
```

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

```
    x = f[0]
```

```
    y = f[1]
```

```
    dxdt = -a*x - b*y + p(t)
```

```
    dydt = -c*x - h*y + q(t)
```

```
    return (dxdt,dydt)
```

```
v0 = (x0, y0)
```

```
yf = odeint(syst, v0, t)
```

```
x = []
```

```
y = []
```

```
for i in range(len(yf)):
```

```
    x.append(yf[i][0])
```

```
    y.append(yf[i][1])
```

```
zero = []
```

```
for i in range (len(t)):
```

```
    zero = np.append(zero,0)
```

```
plt.figure(figsize = (20, 15))
```

```
plt.plot(t, zero, 'b')
```

```
plt.plot(t, x, 'r', label = 'x')
```

```
plt.plot(t, y, 'g', label = 'y')
```

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

```
plt.xlabel('Время')
```

```
plt.title('Модель боевых действий №1')
```

```
plt.legend(loc = 'upper right')
```

```
plt.show()
```

Код на Scilab для случая 2:

```
import numpy as np
import pandas as pd
import scipy as sp
import math
import matplotlib.pyplot as plt

from scipy.integrate import odeint

x0 = 50000
y0 = 69000
t0 = 0

a = 0.12
b = 0.51
c = 0.30
h = 0.61

tmax = 5
dt = 0.05

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

def p(t):
    return (math.sin(t*20))

def q(t):
```

```

    return (math.cos(t*13))

def syst(f,t):
    x = f[0]
    y = f[1]
    dxdt = -a*x - b*y + p(t)
    dydt = -c*x*y - h*y + q(t)
    return (dxdt,dydt)

v0 = (x0, y0)

yf = odeint(syst, v0, t)

x = []
y = []

for i in range(len(yf)):

    x.append(yf[i][0])
    y.append(yf[i][1])


zero = []
for i in range (len(t)):
    zero = np.append(zero,0)

plt.figure(figsize = (20, 15))

```

```
plt.plot(t, zero, 'b')
plt.plot(t, x, 'r', label = 'x')
plt.plot(t, y, 'g', label = 'y')
print (y)
plt.ylabel('Численность армии')
plt.xlabel('Время')
plt.title('Модель боевых действий №1')
plt.legend(loc = 'upper right')

plt.show()
```

4 Выводы

Построили код на Python для решения и вывода на экран моделей боевых действий.