Лабораторная работа №3. Модель боевых действий

Радикорский Павел Михайлович

НФИбд-03-18

27.02.2021

Содержание

# Цели и задачи

**Цель:** Изучить виды модели боевых действий и реализовать программный код для её моделирования.

**Задачи:**

* изучить теорию о модели боевых действий
* реализовать программный код для 42 варианта

# Теоретическая справка

В теоретической части лабораторной работы рассмотрим все интерпретации модели боевых действий.

## Первая модель

где и - параметры, описывающие влияние побочных факторов на потери во время боевых действий, а и - параметры эффективности боевых действий со стороны армий и .

## Вторая модель

Во втором случае в борьбу добавляются партизанские отряды. Нерегулярные войска в отличии от постоянной армии менее уязвимы, так как действуют скрытно, в этом случае сопернику приходится действовать неизбирательно, по площадям, занимаемым партизанами. Поэтому считается, что тем потерь партизан, проводящих свои операции в разных местах на некоторой известной территории, пропорционален не только численности армейских соединений, но и численности самих партизан.

## Третья модель

## Подготовка к реализации

Коэффициенты и будут постоянными для реализации лабораторной работы, т.к. иной вариант сложнее смоделировать.

# Программная реализация

При выполнении задания лабораторной работы использовался Вариант 42.

## Начальные данные

import numpy as np  
from math import cos, sin  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt

x0 = 45000  
y0 = 50000  
t0 = 0   
  
tmax = 1  
dt = 0.05  
  
t = np.arange(t0,tmax,dt)  
  
v0 = np.array([x0, y0])

## Реализация моделей

### Модель №1

Зададим необходимые коэффициенты.

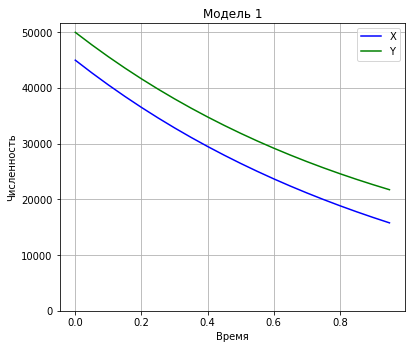
a = 0.29  
b = 0.67  
c = 0.6  
h = 0.38

Объявим функции и , функцию для СДУ.

def P(t):   
 return abs(sin(t) + 1)  
  
def Q(t):  
 return abs(cos(t) + 1)  
  
def derY1(y,t):  
 dy1 = -a\*y[0] - b\*y[1] + P(t)  
 dy2 = -c\*y[0] - h\*y[1] + Q(t)  
 return [dy1, dy2]

Решим через odeint систему дифференциальных уравнений с помощью функции (рис. 1).

y = odeint(derY1, v0, t)  
data1 = [y\_i[0] for y\_i in y]  
data2 = [y\_i[1] for y\_i in y]  
  
plt.plot(t, data1, 'b', label='X')   
plt.plot(t, data2, 'g', label='Y')  
plt.title('Модель 1')  
plt.xlabel('Время')  
plt.ylabel('Численность')  
plt.ylim(0, None)  
plt.legend()  
plt.grid(True)  
plt.margins(0.05)  
plt.subplots\_adjust(left=0, bottom=0, right=0.8, top=1)



Модель для регулярных войск

### Модель №2

Переопределим коэфициенты для второй модели ведения боевых действий.

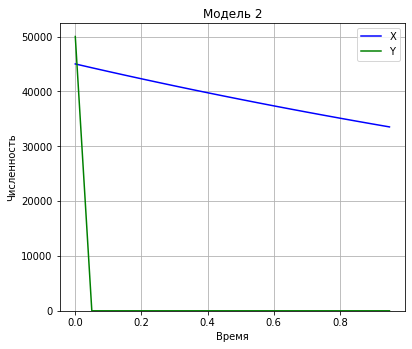
a = 0.31  
b = 0.67  
c = 0.42  
h = 0.53

Переопределим функции и , функцию решения СДУ.

def P(t):  
 return 2\*abs(sin(2\*t))  
  
def Q(t):  
 return abs(cos(t)+1)  
  
def derY2(y,t):  
 dy1 = -a\*y[0] - b\*y[1] + P(t)  
 dy2 = -c\*y[0]\*y[1] - h\*y[1] + Q(t)  
 return [dy1, dy2]

Решим обновленную СДУ (рис. 2).

y = odeint(derY2, v0, t)  
dataset\_1 = [y\_i[0] for y\_i in y]  
dataset\_2 = [y\_i[1] for y\_i in y]  
  
plt.plot(t, dataset\_1, 'b', label='X')   
plt.plot(t, dataset\_2, 'g', label='Y')  
plt.title('Модель 2')  
plt.xlabel('Время')  
plt.ylabel('Численность')  
plt.ylim(0, None)  
plt.legend()  
plt.grid(True)  
plt.margins(0.05)  
plt.subplots\_adjust(left=0, bottom=0, right=0.8, top=1)



Вторая модель

# Выводы

Были изучены модели боевых действий, а также была реализована практическая часть в виде реализации программного кода.

По построенным моделям можно судить, что при участии партизанских отрядов, армия Y понесет значительные потери, в отличие от первого случая, когда функции потерь обеих армий ведут себя приблизительно одинаково.