Лабораторная работа #3

Модель боевых действий. Вариант 11

Баулин Егор Александрович, учебная группа: НКНбд-01-18

Содержание

# Цель работы

* Рассмотреть простейшую модель боевых действий – модель Ланчестера:
  + Просчитывать возможности подходов подкреплений к армиям;
  + Составлять системы дифференциальных уравнений изменения численностей армий;
  + Строить графики для моделей боевых действий.

# Задание

Между страной Х и страной У идет война. Численность состава войск исчисляется от начала войны, и являются временными функциями и . В начальный момент времени страна Х имеет армию численностью 120 000 человек, а в распоряжении страны У армия численностью в 90 000 человек. Для упрощения модели считаем, что коэффициенты постоянны. Также считаем и непрерывные функции.

Постройте графики изменения численности войск армии Х и армии У для следующих случаев:

Между регулярными войсками:

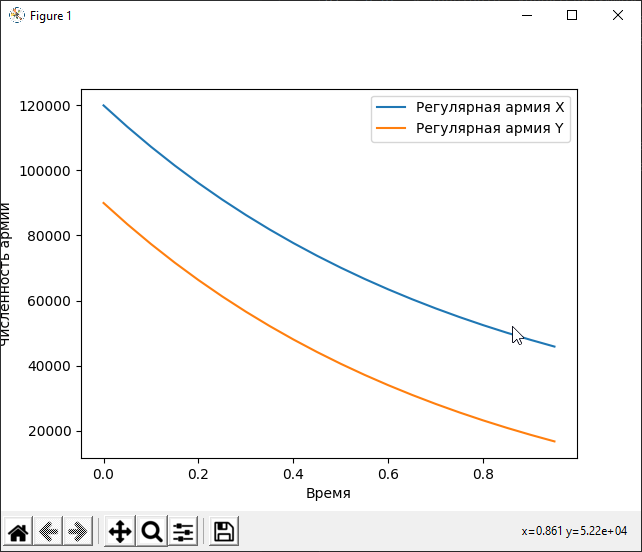
Между регулярными и партизанами:

# Выполнение лабораторной работы

Код на Python:

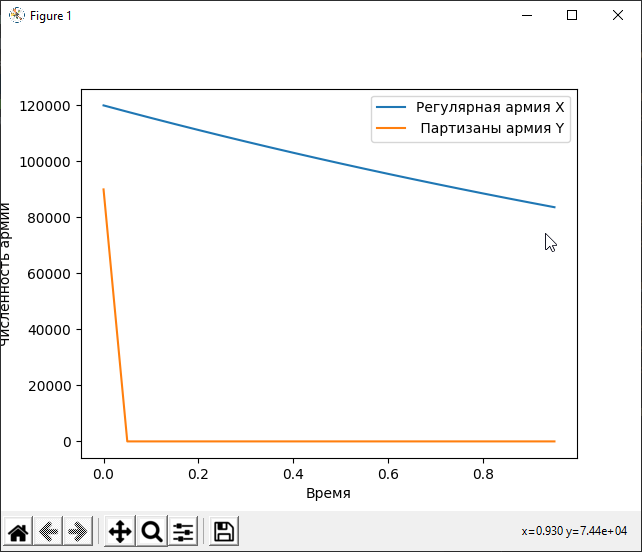
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt  
  
x0 = 120000 # Численность первой армии  
y0 = 90000 # Численность второй армии  
  
# Модель боевых действий между регулярными войсками  
a1 = 0.62 # Константа, характеризующая степень влияния различных факторов на потери  
b1 = 0.68 # Эффективность боевых действий армии Y  
c1 = 0.59 # Эффективность боевых действий армии X  
h1 = 0.71 # Константа, характеризующая степень влияния различных факторов на потери  
  
# Модель боевых действий между регулярными войсками и партизанами  
a2 = 0.38 # Константа, характеризующая степень влияния различных факторов на потери  
b2 = 0.68 # Эффективность боевых действий армии Y  
c2 = 0.21 # Эффективность боевых действий армии X  
h2 = 0.71 # Константа, характеризующая степень влияния различных факторов на потери  
  
# Показатель времени  
t0 = 0  
t\_max = 1  
dt = 0.05  
t = np.arange(t0, t\_max, dt)  
  
# Первый случай  
def P1(t):  
 p1 = np.sin(2\*t)  
 return p1  
  
  
def Q1(t):  
 q1 = np.cos(2\*t)  
 return q1  
  
# Второй случай  
def P2(t):  
 p2 = np.sin(2\*t)  
 return p2  
  
  
def Q2(t):  
 q2 = np.cos(2\*t)  
 return q2  
  
  
# Изменения численности  
# Первый случай  
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  
  
  
# Второй случай  
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]) # Вектор начальных условий  
  
# Решение  
f1 = odeint(S1, v, t)  
f2 = odeint(S2, v, t)  
  
# Построение графиков решений  
# Две регулярные армии  
plt.plot(t, f1)  
plt.ylabel('Численность армии')  
plt.xlabel('Время')  
plt.legend(['Регулярная армия X', 'Регулярная армия Y'])  
plt.show()  
  
# Регулярная армия и партизаны  
plt.plot(t, f2)  
plt.ylabel('Численность армии')  
plt.xlabel('Время')  
plt.legend(['Регулярная армия X', ' Партизаны армия Y'])  
plt.show()

График первого случая (рис. 1)



Две регулярные армии

График второго случай (рис. 2)



Регулярная армия и партизаны

# Выводы

* Рассмотрел простейшую модель боевых действий – модель Ланчестера:
  + Научился просчитывать возможности подходов подкреплений к армиям;
  + Научился оставлять системы дифференциальных уравнений изменения численностей армий;
  + Научился строить графики для моделей боевых действий.