Лабораторная работа №2

Модель боевых действий

Давтян Артур Арменович

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 12 000 человек, а в распоряжении страны У армия численностью в 15 000 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 math  
import numpy as np  
from scipy.integrate import odeint  
import matplotlib.pyplot as plt  
  
x0 = 12000 #численность армии Х  
y0 = 15000 #численность армии Y  
  
# Между регулярными:  
a1 = 0.34 #константа, характеризующая степень влияния различных факторов на потери  
b1 = 0.75 #эффективность боевых действий армии у  
c1 = 0.65 #эффективность боевых действий армии х  
h1 = 0.45 #константа, характеризующая степень влияния различных факторов на потери  
  
# Между регулярными и партизанами:  
a2 = 0.24 #константа, характеризующая степень влияния различных факторов на потери  
b2 = 0.64 #эффективность боевых действий армии у  
c2 = 0.31 #эффективность боевых действий армии х  
h2 = 0.38 #константа, характеризующая степень влияния различных факторов на потери  
  
# время  
t0 = 0  
tmax = 1  
dt = 0.05  
t = np.arange(t0, tmax, dt)  
  
# Первый случай  
def P1(t):  
 p1 = np.sin(3\*t)  
 return p1  
def Q1(t):  
 q1 = np.cos(4\*t)  
 return q1  
  
# Второй случай  
def P2(t):  
 p2 = abs(np.cos(2\*t))  
 return p2  
def Q2(t):  
 q2 = abs(np.sin(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  
  
# Два решения  
f1 = odeint(S1, v, t)  
f2 = odeint(S2, v, t)  
  
# Первый случай (две регулярные армии)  
plt.plot(t, f1)  
plt.ylabel('Численность армии')  
plt.xlabel('Время')  
plt.legend(['Армия X (рег)', 'Армия Y (рег)'])  
  
# Второй случай (регулярная армия и партизаны)  
plt.plot(t, f2)  
plt.ylabel('Численность армии')  
plt.xlabel('Время')  
plt.legend(['Армия X (рег)', 'Армия Y (парт)'])

График первого случая (рис. 1)

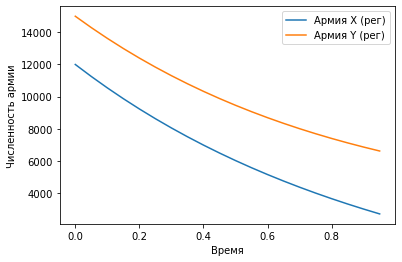


Figure 1: Две регулярные армии

График второго случай (рис. 2)

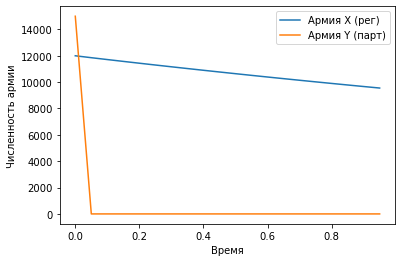


Figure 2: Регулярная армия и партизаны

# 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.