Лабораторная работа №5

Модель хищник-жертва

Латыпова Диана. НФИбд-02-21

Содержание

# 1 Цель работы

* Разобраться в системе “хищник-жертва”.
* Реализовать модель “хищник-жертва”.

# 2 Задание

Вариант 46:

Для модели «хищник-жертва»:

Постройте график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: Найдите стационарное состояние системы.

# 3 Теоретическое введение

**Модель хищник-жертва (Модель Лотки — Вольтерры)**- одна из классических моделей в экологии, описывающая взаимодействие между популяциями двух видов: хищниками и их жертвами. Эта модель предполагает, что изменение численности каждого вида пропорционально численности другого вида и зависит от параметров рождаемости, смертности и взаимодействия между видами [1].

Пусть - численность популяции хищников в момент времени , - численность популяции жертв в момент времени .

Тогда изменение численности популяции хищников по времени определяется уравнением:

где - коэффициент рождаемости хищников, - коэффициент смертности хищников при взаимодействии с жертвами.

А изменение численности популяции жертв по времени определяется уравнением:

где - коэффициент рождаемости жертв, - коэффициент смертности жертв при взаимодействии с хищниками.

**Стационарное состояние.** В стационарном состоянии производные обоих видов по времени равны нулю [2]:

Это означает, что численности видов остаются постоянными, то есть не меняются со временем.

# 4 Выполнение лабораторной работы

Для начала реализуем нестационарное состояние.

Код на языке Julia (рис. 1) (рис. 2):

using Plots  
using DifferentialEquations  
  
# Начальные условия для численности жертв и хищников  
x0 = 13  
y0 = 27  
  
# Параметры модели хищник-жертва  
a = 0.25  
b = 0.05  
c = 0.6  
d = 0.061  
  
# Определение функции для системы дифференциальных уравнений  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a \* u[1] + b \* u[1] \* u[2] # Уравнение для изменения численности жертв  
 du[2] = c \* u[2] - d \* u[1] \* u[2] # Уравнение для изменения численности хищников  
end  
  
# Начальные условия и временной интервал  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
  
# Определение задачи ОДУ  
prob = ODEProblem(ode\_fn, v0, tspan)  
  
# Решение задачи  
sol = solve(prob, dtmax=0.05)  
X = [u[1] for u in sol.u] # Численность жертв  
Y = [u[2] for u in sol.u] # Численность хищников  
T = [t for t in sol.t] # Время  
  
# Построение графика фазового портрета  
plt = plot(  
 dpi=300,  
 legend=false)  
  
plot!(  
 plt,  
 X,  
 Y,  
 color=:blue)  
  
savefig(plt, "out/lab5\_1julia.png")  
  
# Построение графиков изменения численности жертв и хищников по времени  
plt2 = plot(  
 dpi=300,  
 legend=true)  
  
plot!(  
 plt2,  
 T,  
 X,  
 label="Численность жертв",  
 color=:red)  
  
plot!(  
 plt2,  
 T,  
 Y,  
 label="Численность хищников",  
 color=:purple)  
  
savefig(plt2, "out/lab5\_2julia.png")

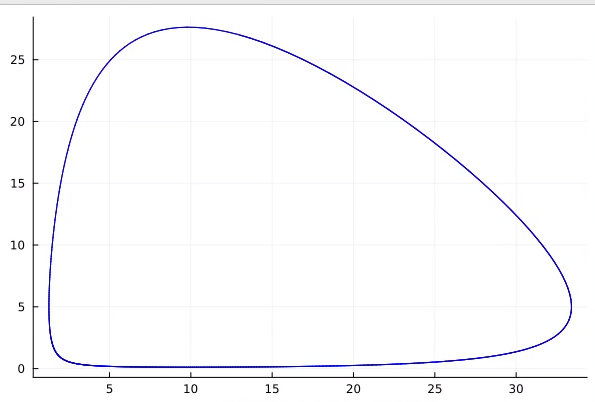


Рис. 1: Нестационарное состояние(1). Julia

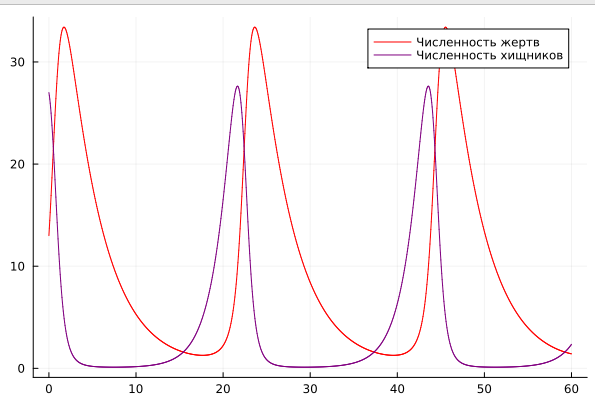


Рис. 2: Нестационарное состояние(2). Julia

Код для ПО OpneModelica (рис. 3) (рис. 4):

model lab5\_1  
Real x;  
Real y;  
initial equation  
x = 13;  
y = 27;  
equation  
der(x) = -0.25\*x + 0.05\*x\*y;  
der(y) = 0.6\*y - 0.061\*x\*y;  
end lab5\_1;

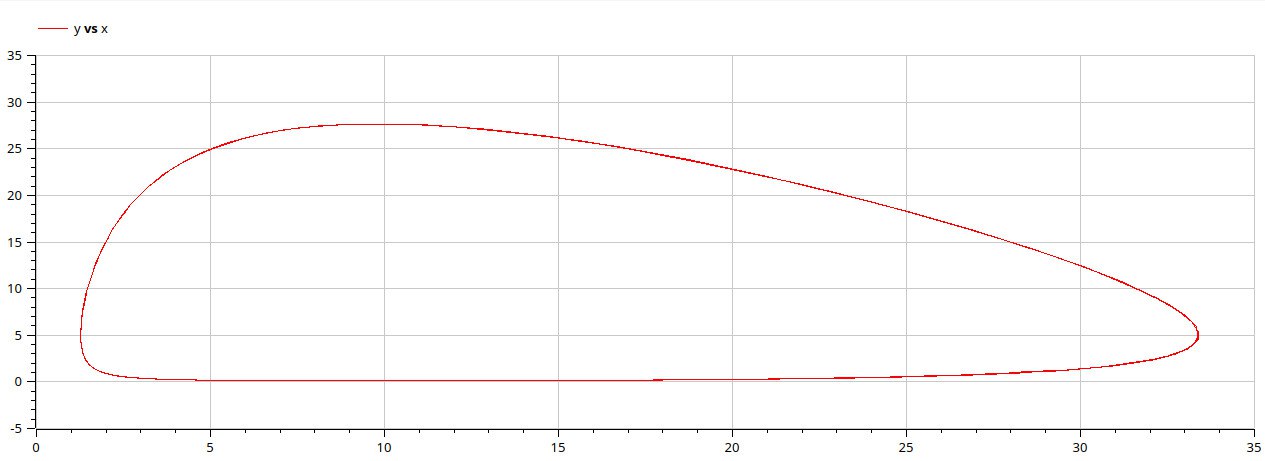


Рис. 3: Нестационарное состояние(1). OM

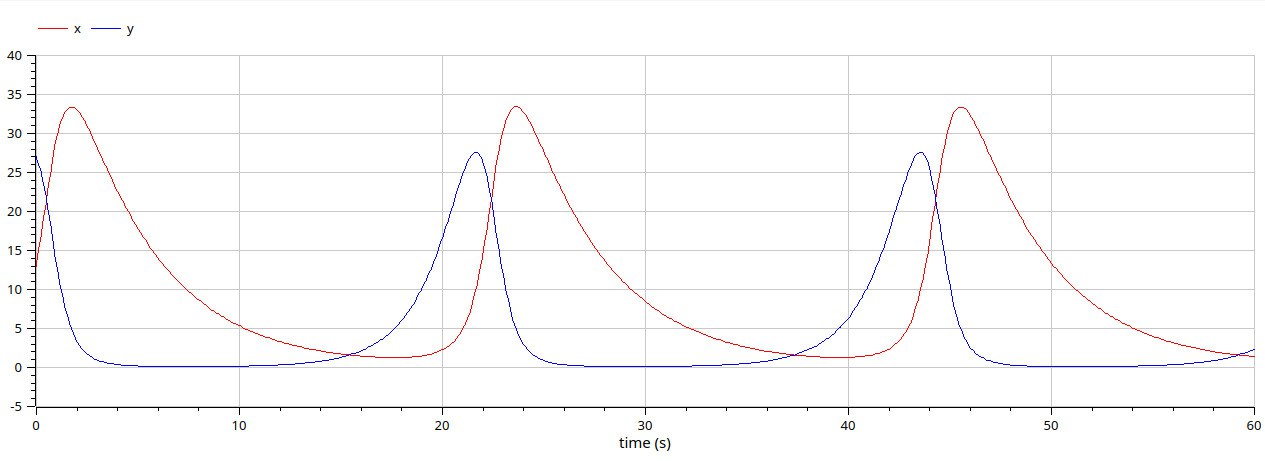


Рис. 4: Нестационарное состояние(2). OM

Реализуем стационарное состояние.

Код на языке Julia (рис. 5):

using Plots  
using DifferentialEquations  
  
# Определение параметров модели  
a = 0.25  
b = 0.05  
c = 0.6  
d = 0.061  
  
# Вычисление начальных условий из уравнений равновесия  
x0 = c / d  
y0 = a / b  
  
# Определение функции для системы дифференциальных уравнений  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a\*u[1] + b \* u[1] \* u[2] # Уравнение для изменения численности жертв  
 du[2] = c \* u[2] - d \* u[1] \* u[2] # Уравнение для изменения численности хищников  
end  
  
# Начальные условия и временной интервал  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
  
# Определение задачи ОДУ  
prob = ODEProblem(ode\_fn, v0, tspan)  
  
# Решение задачи  
sol = solve(prob, dtmax=0.05)  
  
# Извлечение результатов  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
T = [t for t in sol.t]  
  
# Построение графиков  
plt2 = plot(  
 dpi=300,  
 legend=true)  
  
plot!(  
 plt2,  
 T,  
 X,  
 label="Prey's count", # Численность жертв  
 color=:red)  
  
plot!(  
 plt2,  
 T,  
 Y,  
 label="Predator's count", # Численность хищников  
 color=:purple)  
  
savefig(plt2, "out/lab5\_3julia.png") # Сохранение графика в файл

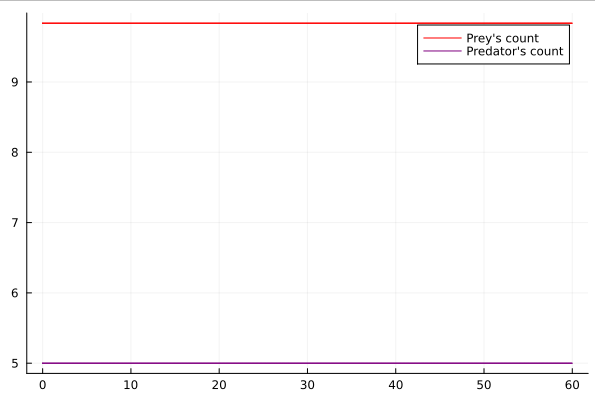


Рис. 5: Стационарное состояние. Julia

Код для ПО OpneModelica (рис. 6):

model lab5\_2  
Real x;  
Real y;  
initial equation  
x = 0.6 / 0.061;  
y = 0.25 / 0.05;  
equation  
der(x) = -0.25\*x + 0.05\*x\*y;  
der(y) = 0.6\*y - 0.061\*x\*y;  
end lab5\_2;

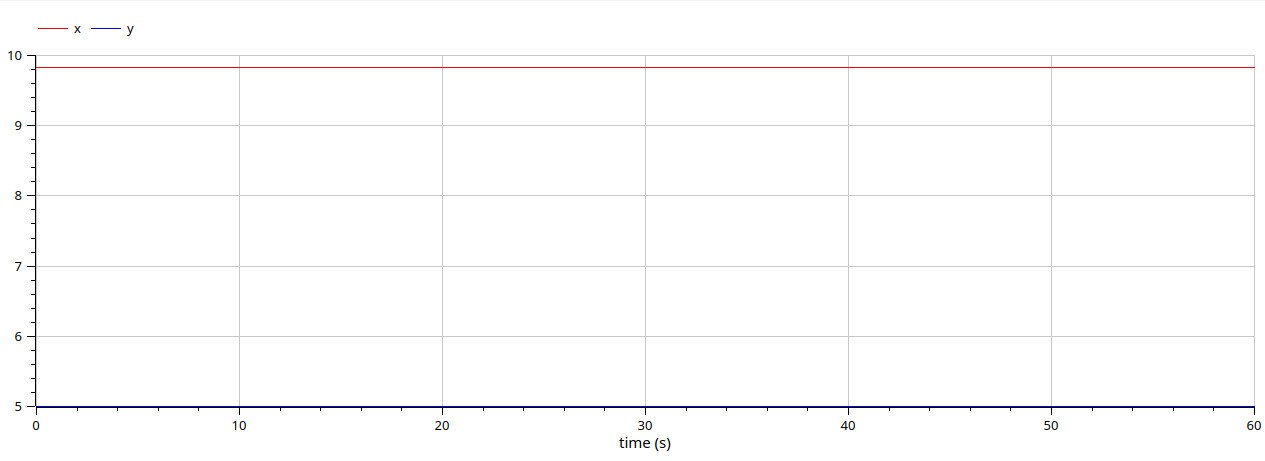


Рис. 6: Стационарное состояние. OM

# 5 Выводы

Я разобралась в системе “хищник-жертва”. А также реализовала модель “хищник-жертва” на языке программирования julia и на ПО OpenModelica. Нарисовала графики для нестационарного и стационарного состояния.

# Список литературы

1. Система "хищник-жертва" [Электронный ресурс]. Wikimedia Foundation, Inc., 2023. URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%C2%AB%D1%85%D0%B8%D1%89%D0%BD%D0%B8%D0%BA_%E2%80%94_%D0%B6%D0%B5%D1%80%D1%82%D0%B2%D0%B0%C2%BB>.

2. Стационарное состояние [Электронный ресурс]. Copyright © 2024 OpenModelica., 2017. URL: <https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5>.