Модель хищник-жертва. Вариант №59

Содержание

# Цель работы

Изучить жесткую модель хищник-жертва и построить эту модель.

# Теоретическое введение

* Модель Лотки—Вольтерры — модель взаимодействия двух видов типа «хищник — жертва», названная в честь её авторов, которые предложили модельные уравнения независимо друг от друга. Такие уравнения можно использовать для моделирования систем «хищник — жертва», «паразит — хозяин», конкуренции и других видов взаимодействия между двумя видами. [4]

Данная двувидовая модель основывается на следующих предположениях [4]:

1. Численность популяции жертв x и хищников y зависят только от времени (модель не учитывает пространственное распределение популяции на занимаемой территории)
2. В отсутствии взаимодействия численность видов изменяется по модели Мальтуса, при этом число жертв увеличивается, а число хищников падает
3. Естественная смертность жертвы и естественная рождаемость хищника считаются несущественными
4. Эффект насыщения численности обеих популяций не учитывается
5. Скорость роста численности жертв уменьшается пропорционально численности хищников

В этой модели – число жертв, - число хищников. Коэффициент описывает скорость естественного прироста числа жертв в отсутствие хищников, - естественное вымирание хищников, лишенных пищи в виде жертв. Вероятность взаимодействия жертвы и хищника считается пропорциональной как количеству жертв, так и числу самих хищников (). Каждый акт взаимодействия уменьшает популяцию жертв, но способствует увеличению популяции хищников (члены и в правой части уравнения).

Математический анализ этой (жёсткой) модели показывает, что имеется стационарное состояние, всякое же другое начальное состояние приводит к периодическому колебанию численности как жертв, так и хищников, так что по прошествии некоторого времени такая система вернётся в изначальное состояние.

Стационарное состояние системы (положение равновесия, не зависящее от времени решения) будет находиться в точке . Если начальные значения задать в стационарном состоянии , то в любой момент времени численность популяций изменяться не будет. При малом отклонении от положения равновесия численности как хищника, так и жертвы с течением времени не возвращаются к равновесным значениям, а совершают периодические колебания вокруг стационарной точки. Амплитуда колебаний и их период определяется начальными значениями численностей . Колебания совершаются в противофазе.

# Задачи

1. Построить график зависимости численности хищников от численности жертв
2. Построить график зависимости численности хищников и численности жертв от времени
3. Найти стационарное состояние системы

# Задание

Вариант 59:

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.

# Выполнение лабораторной работы

## Построение математической модели. Решение с помощью программ

### Julia

Код программы для нестационарного состояния:

using Plots  
using DifferentialEquations  
  
x0 = 6  
y0 = 21  
  
a = 0.48  
b = 0.52  
c = 0.053  
d = 0.048  
  
  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a\*u[1] + c \* u[1] \* u[2]  
 du[2] = b \* u[2] - d \* u[1] \* u[2]  
end  
  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
prob = ODEProblem(ode\_fn, v0, tspan)  
sol = solve(prob, dtmax=0.05)  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
T = [t for t in sol.t]  
  
plt = plot(  
 dpi=300,  
 legend=false)  
  
plot!(  
 plt,  
 X,  
 Y,  
 label="Зависимость численности хищников от численности жертв",  
 color=:blue)  
  
savefig(plt, "julia1-1.png")  
  
plt2 = plot(  
 dpi=300,  
 legend=true)  
  
plot!(  
 plt2,  
 T,  
 X,  
 label="Численность жертв",  
 color=:green)  
  
plot!(  
 plt2,  
 T,  
 Y,  
 label="Численность хищников",  
 color=:red)  
  
savefig(plt2, "julia1-2.png")

Код программы для стационарного состояния:

using Plots  
using DifferentialEquations  
  
a = 0.48  
b = 0.52  
c = 0.053  
d = 0.048  
  
x0 = c / d   
y0 = a / b  
  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a\*u[1] + c \* u[1] \* u[2]  
 du[2] = b \* u[2] - d \* u[1] \* u[2]  
end  
  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
prob = ODEProblem(ode\_fn, v0, tspan)  
sol = solve(prob, dtmax=0.05)  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
T = [t for t in sol.t]  
  
plt2 = plot(  
 dpi=300,  
 legend=true)  
  
plot!(  
 plt2,  
 T,  
 X,  
 label="Численность жертв",  
 color=:green)  
  
plot!(  
 plt2,  
 T,  
 Y,  
 label="Численность хищников",  
 color=:red)  
  
savefig(plt2, "julia2.png")

В стационарном состоянии решение вида будет представлять собой точку.

### Результаты работы кода на Julia

График численности хищников от численности жертв

График численности жертв и хищников от времени

Стационарное состояние

## OpenModelica

Код программы для нестационарного состояния:

model lab51  
 Real a = 0.48;  
 Real b = 0.52;  
 Real c = 0.053;  
 Real d = 0.048;  
 Real x;  
 Real y;  
  
initial equation  
 x = 6;  
 y = 21;  
equation  
 der(x) = -a\*x + c\*x\*y;  
 der(y) = b\*y - d\*x\*y;  
 annotation(  
 experiment(StartTime = 0, StopTime = 60, Tolerance = 1e-06, Interval = 0.05));  
end lab51;

Код программы для стационарного состояния:

model lab51  
 Real a = 0.48;  
 Real b = 0.52;  
 Real c = 0.053;  
 Real d = 0.048;  
 Real x;  
 Real y;  
initial equation  
 x = c/d;  
 y = a/b;  
equation  
 der(x) = -a\*x + c\*x\*y;  
 der(y) = b\*y - d\*x\*y;  
 annotation(  
 experiment(StartTime = 0, StopTime = 60, Tolerance = 1e-06, Interval = 0.05));  
end lab51;

В стационарном состоянии решение вида будет представлять собой точку.

### Результаты работы кода на OpenModelica

График численности хищников от численности жертв

График численности жертв и хищников от времени

Стационарное состояние

# Анализ полученных результатов. Сравнение языков.

В итоге проделанной работы мы построили график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв на языках Julia и OpenModelica. Построение модели хищник-жертва на языке openModelica занимает меньше строк, чем аналогичное построение на Julia.

# Вывод

В ходе выполнения лабораторной работы была изучена модель хищник-жертва и построена модель на языках Julia и Open Modelica.

# Список литературы. Библиография

[1] Документация по Julia: https://docs.julialang.org/en/v1/

[2] Документация по OpenModelica: https://openmodelica.org/

[3] Решение дифференциальных уравнений: https://www.wolframalpha.com/

[4] Модель Лотки—Вольтерры: https://math-it.petrsu.ru/users/semenova/MathECO/Lections/Lotka\_Volterra.pdf