Отчёт по лабораторной работе №5 Математическое моделирование

Модель хищник-жертва. Вариант №8

Выполнил: Маляров Семён Сергеевич,
НПИбд-01-21, 1032209505

Содержание

1 Цель работы

Изучить жесткую модель хищник-жертва и построить эту модель.

2 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

$$\begin{cases} \frac{dx}{dt} = \left(-ax(t) + by(t)x(t)\right) \\ \frac{dy}{dt} = \left(cy(t) - dy(t)x(t)\right) \end{cases}$$

В этой модели x – число жертв, y - число хищников. Коэффициент a описывает скорость естественного прироста числа жертв в отсутствие хищников, с - естественное вымирание хищников, лишенных пищи в виде жертв. Вероятность взаимодействия жертвы и хищника считается пропорциональной как количеству жертв, так и числу самих хищников (xy). Каждый акт взаимодействия уменьшает популяцию жертв, но способствует увеличению популяции хищников (члены $-bxy$ и dxy 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 $x_0 = \frac{c}{d}, y_0 = \frac{a}{b}$. Если начальные значения задать в стационарном состоянии $x(0) = x_0, y(0) = y_0$, 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 $x(0), y(0)$. Колебания совершаются в противофазе.

3 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

4 Задание

Вариант 59:

Для модели «хищник-жертва»:

$$\begin{cases} \frac{dx}{dt} = -0.19x(t) + 0.048y(t)x(t) \\ \frac{dy}{dt} = 0.39y(t) - 0.036y(t)x(t) \end{cases}$$

Постройте график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: $x_0 = 13, y_0 = 18$ Найдите стационарное состояние системы.

5 Выполнение лабораторной работы

5.1 Построение математической модели. Решение с помощью программ

5.1.1 Julia

Код программы для нестационарного состояния:

```
using Plots
using DifferentialEquations

x0 = 13
y0 = 18

a = 0.19
b = 0.048
c = 0.39
d = 0.036


function ode_fn(du, u, p, t)
    x, y = u
    du[1] = -a*u[1] + b * u[1] * u[2]
    du[2] = c * 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,
  color=:blue)

savefig(plt, "lab5_julia_1.png")

plt2 = plot(
  dpi=300,
  legend=true)

plot!(
  plt2,
  T,
  X,
  label="Численность жертв",
  color=:red)

plot!(
  plt2,
  T,
  Y,
  label="Численность хищников",
  color=:green)

savefig(plt2, "lab5_julia_2.png")
```

Код программы для стационарного состояния:

```
using Plots
using DifferentialEquations

a = 0.19
b = 0.048
c = 0.39
d = 0.036

x0 = c / d
y0 = a / b

function ode_fn(du, u, p, t)
    x, y = u
    du[1] = -a*u[1] + b * u[1] * u[2]
    du[2] = c * 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=:red)

plot!(
  plt2,
  T,
  Y,
  label="Численность хищников",
  color=:green)

savefig(plt2, "lab5_julia_3.png")
```

В стационарном состоянии решение вида $y(x) = somefunction$ будет представлять собой точку.

5.1.2 Результаты работы кода на Julia

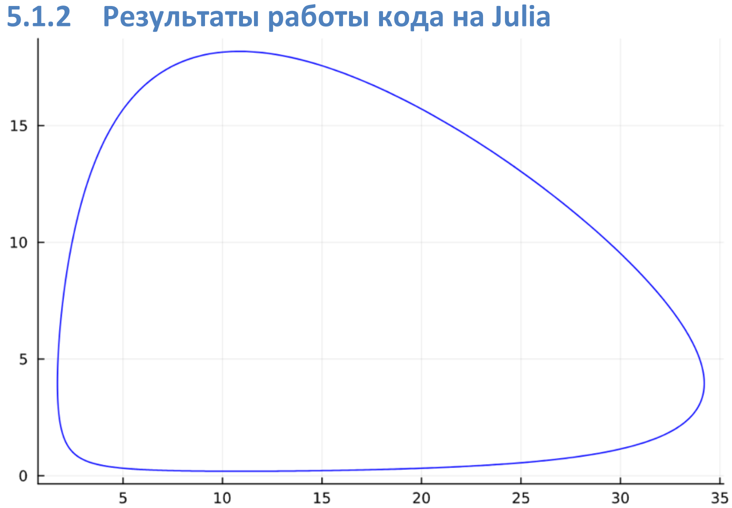

Рис. 1: График численности хищников от численности жертв

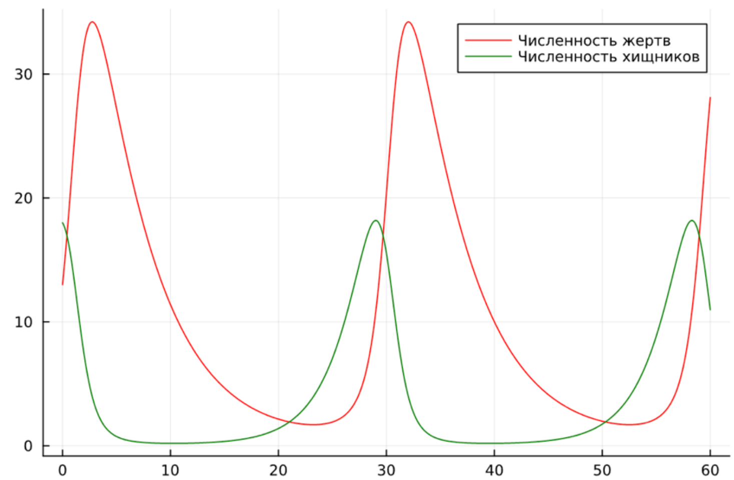

Рис. 2: График численности жертв и хищников от времени

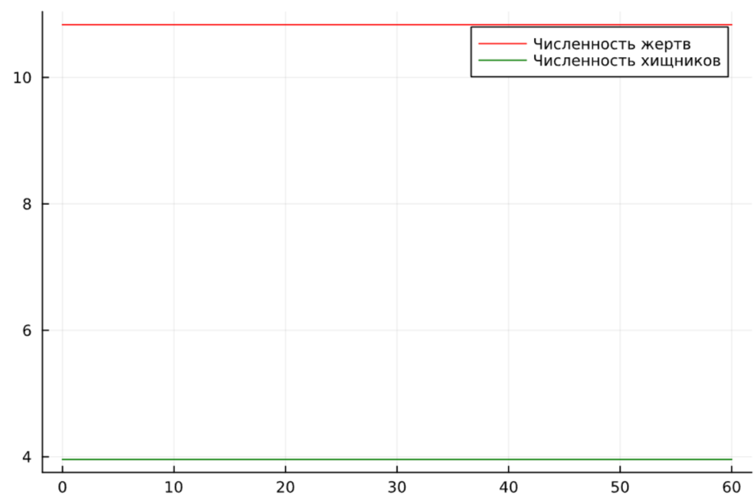

Рис. 3: Стационарное состояние

5.2 OpenModelica

Код программы для нестационарного состояния:

```
model lab5_1
Real a = 0.19;
Real b = 0.048;
Real c = 0.39;
Real d = 0.036;
Real x;
Real y;
initial equation
x = 13;
y = 18;
equation
der(x) = -a*x + b*x*y;
der(y) = c*y - d*x*y;
end lab5_1;
```

Код программы для стационарного состояния:

```
model lab5_2
Real a = 0.19;
Real b = 0.048;
Real c = 0.39;
Real d = 0.036;
Real x;
Real y;
initial equation
x = c / d;
y = a / b;
equation
der(x) = -a*x + b*x*y;
der(y) = c*y - d*x*y;
end lab5_2;
```

В стационарном состоянии решение вида $y(x) = somefunction$ будет представлять собой точку.

5.2.1 Результаты работы кода на OpenModelica

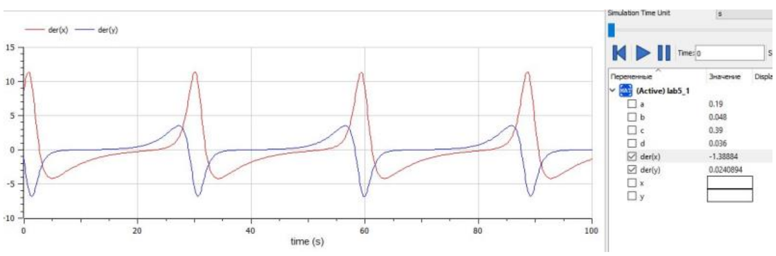

Рис. 4: График численности жертв и хищников от времени

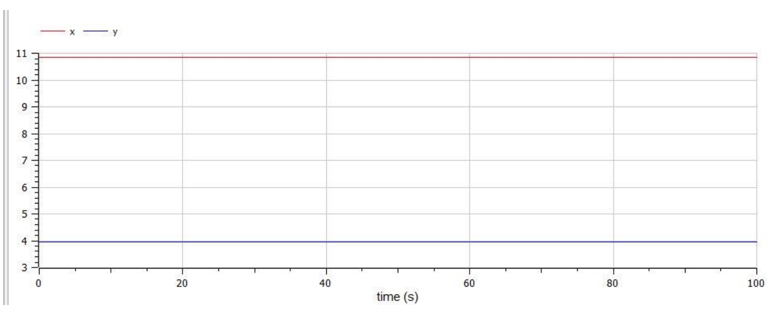

Рис. 5: Стационарное состояние

6 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.

7 Вывод

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

8 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