Отчёт по лабораторной работе №5

Модель хищник-жертва

Голощапова Ирина Борисовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить модель “хищник-жертва” - модель Лотки-Вольтерры.

2 Библиография

1. Git - система контроля версий
2. Дифференциальные уравнения
3. Язык программирования - Julia
4. Решение ДУ на языке программирование Julia
5. Работа с OpenModelica
6. Julia и фазовые портреты динамических систем

3 Задачи лабораторной работы

1. Построить график зависимости численности хищников от численности жертв.
2. Построить графики изменения численности хищников и численности жертв при заданных начальных условиях.
3. Найти стационарное состояние системы.

4 Теоретическая справка

4.1 Модель хищник-жертва

Простейшая модель взаимодействия двух видов типа «хищник — жертва» - модель Лотки-Вольтерры. Данная двувидовая модель основывается на следующих предположениях:

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

Система уравнений(рис. 4.1):

$$\frac{dx}{dt} = ax(t) - bx(t)y(t)$$
$$\frac{dy}{dt} = -cy(t) + dx(t)y(t)$$

Рис. 4.1: Система уравнений

В этой модели x – число жертв, y - число хищников. Коэффициент a описывает скорость естественного прироста числа жертв в отсутствие хищников, c - 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).

4.2 Стационарное состояние

Математический анализ этой (жесткой) 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 система возвращается в состояние B.

Стационарное состояние системы (положение равновесия, не зависящее от времени решение) будет в точке:

x0 = c/d,

y0 = a/b.

Если начальные значения задать в стационарном состоянии

x(0) = x0,

y(0) = y0,

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

5 Условие задачи (вариант №7)

Для модели «хищник-жертва»(рис. 5.1):

$$\begin{cases} \dfrac{dx}{dt} = -0.18x(t) + 0.047x(t)y(t) \\ \dfrac{dy}{dt} = 0.38y(t) - 0.035x(t)y(t) \end{cases}$$

Рис. 5.1: Условие варианта №7

Постройте график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: x0 = 12, y0 = 17.

6 Выполнение лабораторной работы

6.1 Реализация в OpenModelica.

Для начала реализуем данную задачу в OpenModelica. Зададим начальные значения, параметры и системы уравнений.

Листинг программы:

```
model lab5

parameter Real a = 0.18; // коэффициент естественной смертности хищников
parameter Real b = 0.38; // коэффициент естественного прироста жертв
parameter Real c = 0.047; // коэффициент увеличения числа хищников
parameter Real d = 0.035; // коэффициент смертности жертв

parameter Real x0 = 12;
parameter Real y0 = 17;

Real x(start=x0);
Real y(start=y0);

equation
  der(x) = -a*x + c*x*y;
  der(y) = b*y - d*x*y;
```

```
annotation(experiment(StartTime = 0, StopTime = 70, Interval = 0.05));

end lab5;
```

В результате получим следующее решение (рис. 6.1):

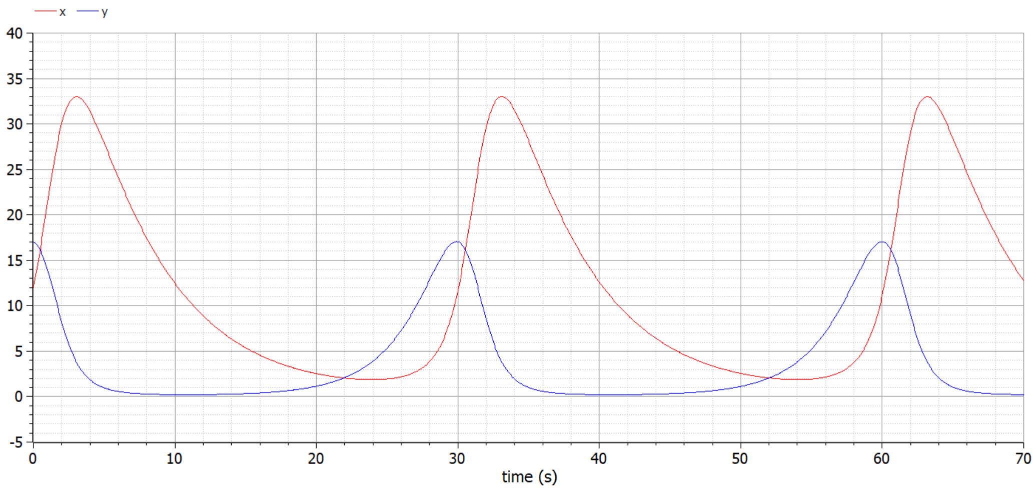

Рис. 6.1: Решение. OpenModelica

Фазовый портрет примет следующий вид (рис. 6.2):

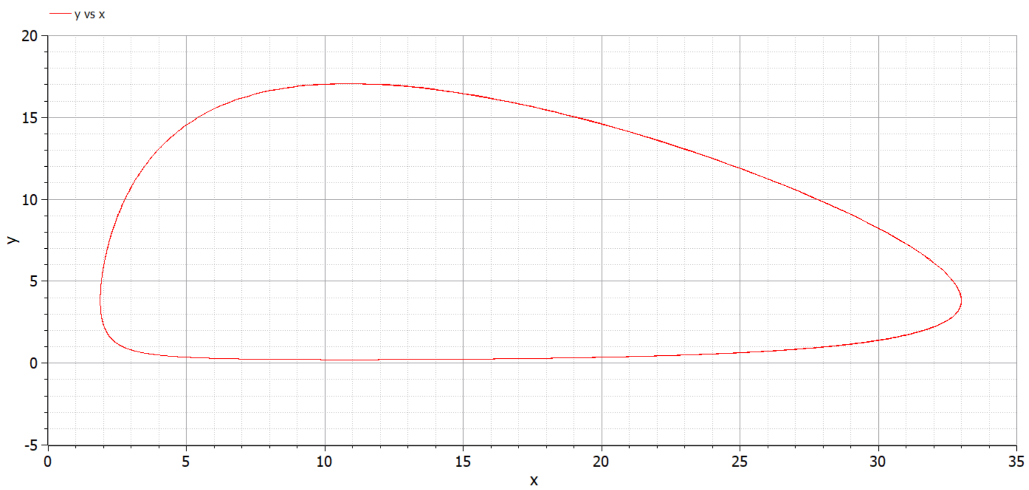

Рис. 6.2: Фазовый портрет. OpenModelica

6.2 Нахождение стационарного состояния системы в OpenModelica

Для того, чтобы найти стационарное состояние системы нужно только поменять начальные условия:

x0 = c/d

y0 = a/b

В результате получим следующий листинг программы:

```
model lab5_1
parameter Real a = 0.18; // коэффициент естественной смертности хищников
parameter Real b = 0.38; // коэффициент естественного прироста жертв
parameter Real c = 0.047; // коэффициент увеличения числа хищников
parameter Real d = 0.035; // коэффициент смертности жертв

parameter Real x0 = b / d;
parameter Real y0 = a / c;

Real x(start=x0);
Real y(start=y0);

equation
  der(x) = -a*x + c*x*y;
  der(y) = b*y - d*x*y;

annotation(experiment(StartTime = 0, StopTime = 70, Interval = 0.05));
end lab5_1;
```

Стационарное состояние системы (рис. 6.3):

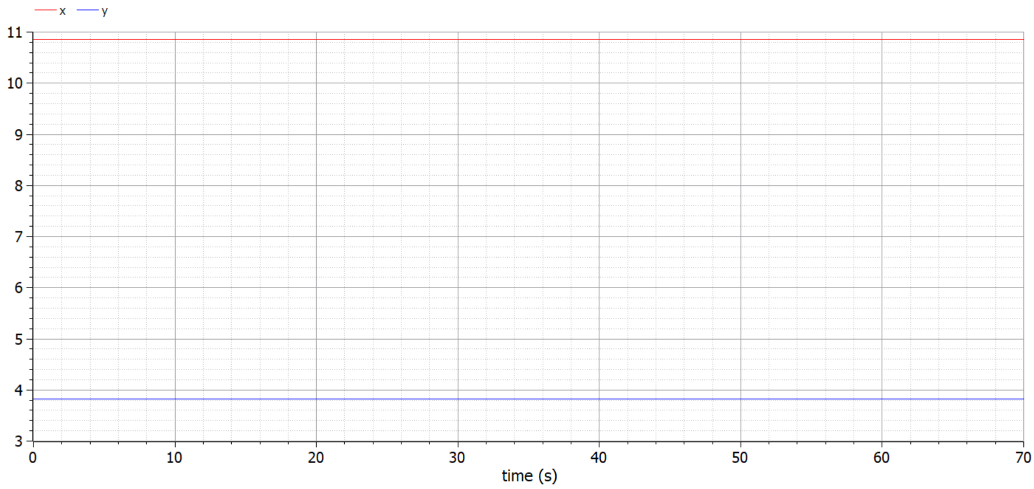

Рис. 6.3: Стационарное состояние системы. OpenModelica

x0 = 10.8571429

y0 = 3.82978723

6.3 Реализация на Julia.

Согласно условию варианта №7 зададим начальные значения и реализуем решение поставленной задачи.

Листинг программы на языке Julia:

```
using DifferentialEquations

function lorenz!(du, u, p, t)
    a, b, c, d = p
    du[1] = -a * u[1] + c * u[1] * u[2]
    du[2] = b * u[2] - d * u[1] * u[2]
end

const x = 12
const y = 17
```

```
u0 = [x, y]

p = (0.18, 0.38, 0.047, 0.035)
tspan = (0.0, 50.0)
prob = ODEProblem(lorenz!, u0, tspan, p)
sol = solve(prob)

using Plots; gr()

#решение
plot(sol)
savefig("lab5_solution.png")

#фазовый портрет
plot(sol, vars=(2,1))
savefig("lab5.png")
```

В результате получим следующее решение (рис. 6.4):

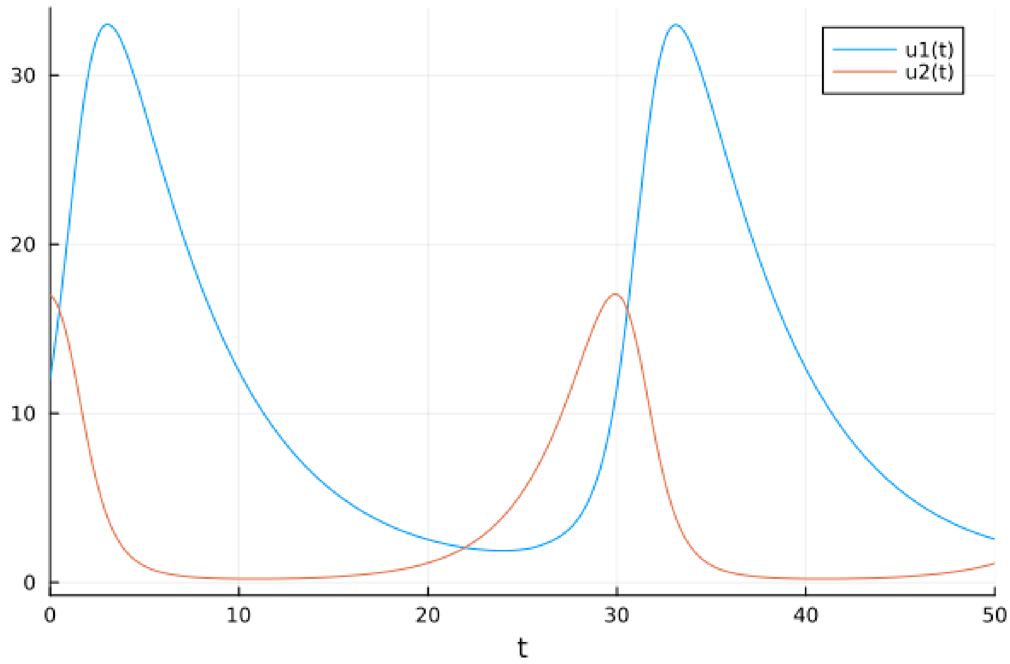

Рис. 6.4: Решение. Julia

Фазовый портрет примет следующий вид (рис. 6.5):

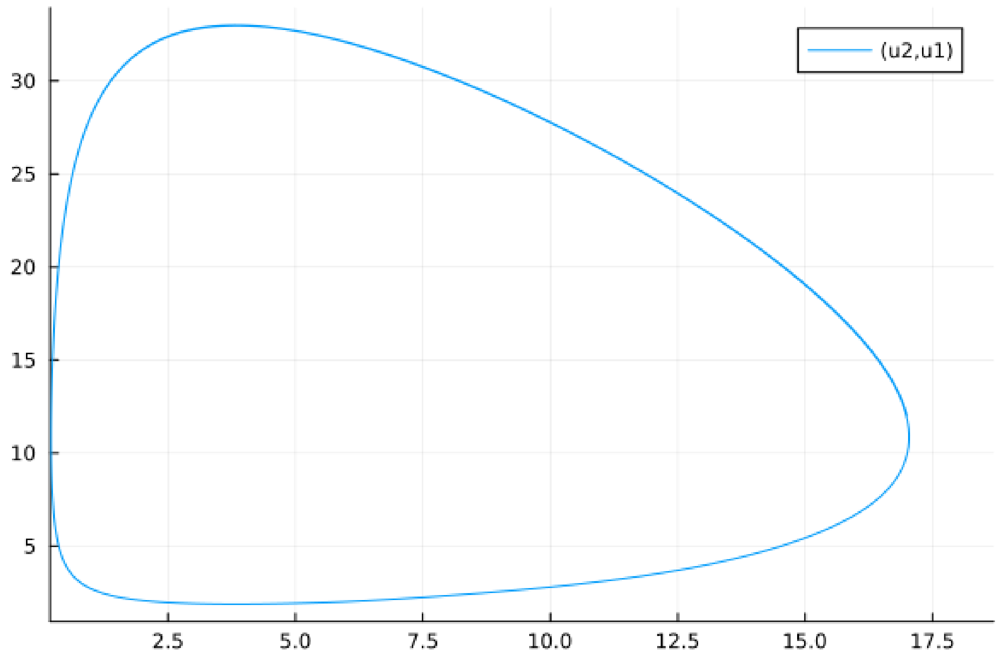

Рис. 6.5: Фазовый портрет. Julia

6.4 Нахождение стационарного состояния системы на Julia

Для того, чтобы найти стационарное состояние системы необходимо в начальные значения подставить следующие формулы:

x0 = c/d

y0 = a/b

Листинг программы на языке Julia:

```
using DifferentialEquations

function lorenz!(du, u, p, t)
    a, b, c, d = p
    du[1] = -a * u[1] + c * u[1] * u[2]
```

```
    du[2] = b * u[2] - d * u[1] * u[2]
end

const x = 0.38/0.035
const y = 0.18/0.047
u0 = [x, y]

p = (0.18, 0.38, 0.047, 0.035)
tspan = (0.0, 50.0)
prob = ODEProblem(lorenz!, u0, tspan, p)
sol = solve(prob)

using Plots; gr()
plot(sol)
savefig("lab5_1.png")
```

Стационарное состояние системы на Julia (рис. 6.6):

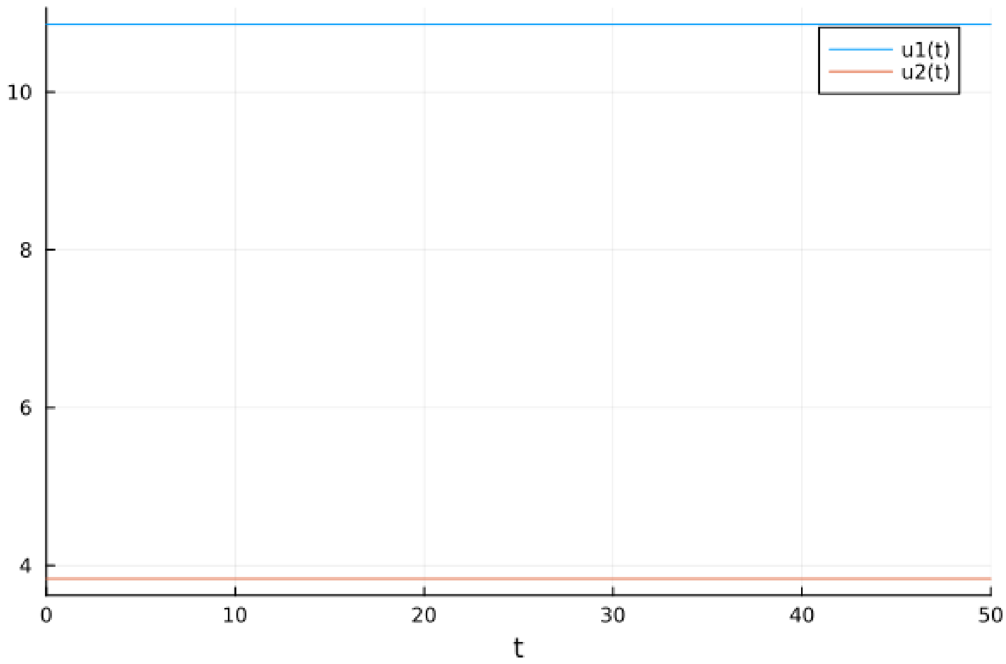

Рис. 6.6: Стационарное состояние системы. Julia

x0 = 10.8571429

y0 = 3.82978723

7 Выводы

В ходе лабораторной работы нам удалось познакомиться с простейшей моделью взаимодействия двух видов типа «хищник — жертва» - моделью Лотки-Вольтерры. Также мы реализовали решение данной задачи и построили графики зависимостей на языках: моделирования - OpenModelica и программирования - Julia.