

Лабораторная работа №5

Модель хищник-жертва

Шубнякова Дарья НКНбд-01-22

1. Вводная часть

2. Основная часть

3. Результаты

1. Вводная часть

1.1 Цели и задачи

Ознакомиться с моделью “хищник-жертва”. Реализовать данную модель на языке Julia, а так же в среде OMEdit.

Построить график и фазовый портрет модели.

2. Основная часть

2.1 Выполнение лабораторной работы

2.1.1 Пишем код на языке Julia для реализации данной модели.

```
[1]: using DifferentialEquations, Plots

# Параметры модели для варианта 13
a = 0.41 # Коэффициент смертности хищников
b = 0.039 # Коэффициент взаимодействия хищников
c = 0.51 # Коэффициент прироста жертв
d = 0.019 # Коэффициент взаимодействия жертв

# Начальные условия
u0 = [7.0, 9.0] # x0 = 7 (жертвы), y0 = 9 (хищники)
tspan = (0.0, 100.0)

# Система уравнений Лотки-Вольтерры
function predator_prey!(du, u, p, t)
    x, y = u
    du[1] = -a*x + b*x*y # dx/dt = -0.41x + 0.039xy
    du[2] = c*y - d*x*y # dy/dt = 0.51y - 0.019xy
end

# Решение системы
prob = ODEProblem(predator_prey!, u0, tspan)
sol = solve(prob, Tsit5(), reltol=1e-6, abstol=1e-6)

# Построение графиков
plt1 = plot(sol, idxs=[1, 2], label=["Жертвы (x)" "Хищники (y)"],
            title="Изменение численности популяций", xlabel="Время", ylabel="Численнос",
            linewidth=2, legend=:right)

# Фазовый портрет
plt2 = plot(sol[1,:), sol[2,:), label="Фазовая траектория", linewidth=2,
            title="Фазовый портрет: хищники vs жертвы", xlabel="Численность жертв",
            ylabel="Численность хищников", legend=false)
scatter!(plt2, [7], [9], label="Начальная точка", color=:red)
```

2.2 Продолжение кода и анализ данных.

```
scatter!(plt2, [x_stationary], [y_stationary], label="Стационарное состояние", color=

# Отображение графиков
plot(plt1, plt2, layout=(2,1), size=(800, 600))

# Вывод результатов
println("Стационарное состояние системы:")
println("Численность жертв x* = ", round(x_stationary, digits=2))
println("Численность хищников y* = ", round(y_stationary, digits=2))
println("Начальные условия: x₀ = 7, y₀ = 9")

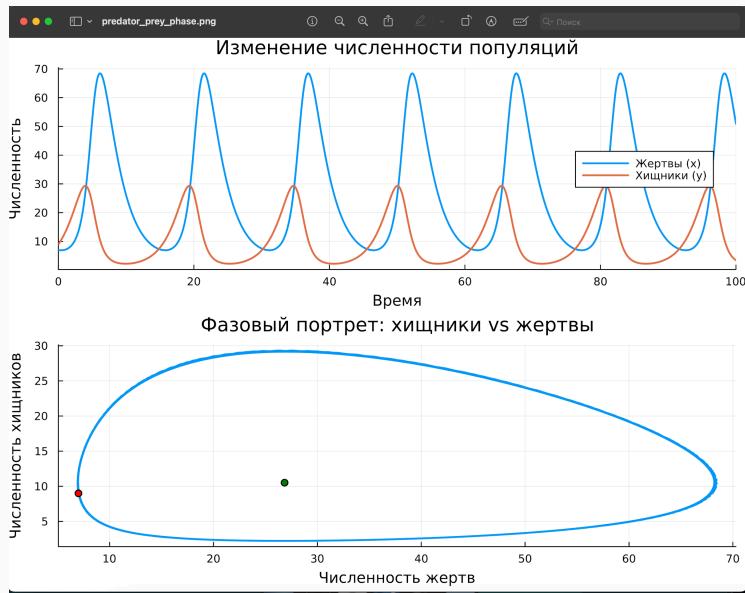
# Сохранение графиков
savefig("predator_preym_time.png")
savefig("predator_preym_phase.png")

println("\nГрафики сохранены в файлы:")
println("- predator_preym_time.png - временные зависимости")
println("- predator_preym_phase.png - фазовый портрет")
```

Стационарное состояние системы:
Численность жертв $x^* = 26.84$
Численность хищников $y^* = 10.51$
Начальные условия: $x_0 = 7$, $y_0 = 9$

Графики сохранены в файлы:
- predator_preym_time.png - временные зависимости
- predator_preym_phase.png - фазовый портрет

2.3 Полученные нами график и фазовый портрет модели.



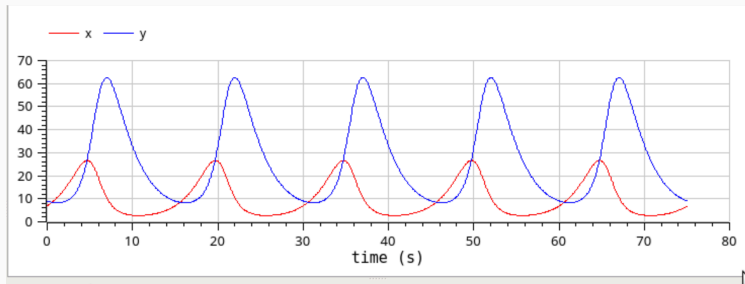
2.4 Прописываем код в OMEdit.

```
1 model PredatorPrey "Модель хищник-жертва для варианта 13 - ИСПРАВЛЕННАЯ"
2   parameter Real a = 0.41 "Коэффициент смертности хищников";
3   parameter Real b = 0.039 "Коэффициент взаимодействия (хищники питаются жертвами)";
4   parameter Real c = 0.51 "Коэффициент прироста жертв";
5   parameter Real d = 0.019 "Коэффициент взаимодействия (жертвы被杀)";
6
7   Real x(start = 7, fixed = true) "Численность жертв";
8   Real y(start = 9, fixed = true) "Численность хищников";
9   Real t "Время";
10
```

2.5 Продолжение кода.

```
10
11 equation
12 // ИСПРАВЛЕННЫЕ УРАВНЕНИЯ:
13 der(x) = c*x - d*x*y;    // dx/dt = 0.51x - 0.019xy (жертвы размножаются, но их
едят)
14 der(y) = -a*y + b*x*y;  // dy/dt = -0.41y + 0.039xy (хищники умирают, но
размножаются когда есть еда)
15
16 der(t) = 1;
17
18 annotation(experiment(StartTime=0, StopTime=100, Tolerance=1e-6, Interval=0.1));
19 end PredatorPrey;
```

2.6 Получаем аналогичный график.



3. Результаты

3. Результаты

Мы реализовали модель в OpenModelica и на языке Julia. На выходе получили две картинки: `predator_preymphase.png` и `predators_preymtime.png`. Итоговый файл `lab5.ibybn` с кодом на языке Julia в JupiterNotebook. А также файл для симуляции в OpenModelica: `PredatorPreym.mo`