Лабораторная работа №2: Основные модели

Мария Улитина

Table of contents

1. **Введение** **1**

2. **Модель SIR** **1**

3. **Модель SIR (Susceptible-Infectious-Recovered)** **1**

4. **Параметрический анализ SIR** **2**

5. **Параметрическое исследование SIR** **2**

6. **Модель Лотки-Вольтерры** **3**

7. **Модель Лотки-Вольтерры** **3**

1. Введение

Исследуются модели SIR и Лотки-Вольтерры.

2. Модель SIR

3. Модель SIR (Susceptible-Infectious-Recovered)

Цель: Исследование динамики эпидемии. Уравнения:

$$\begin{cases} \frac{dS}{dt} = -\beta c \frac{I}{N} S \\ \frac{dI}{dt} = \beta c \frac{I}{N} S - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases}$$

```julia
using DrWatson
@quickactivate "project"
using DifferentialEquations, DataFrames, Plots, LaTeXStrings

script_name = splitext(basename(PROGRAM_FILE))[1]
mkpath(plotsdir(script_name))

function sir_ode!(du, u, p, t)
    S, I, R = u
    β, c, γ = p
    N = S + I + R
    du[1] = -β * c * I / N * S
    du[2] = β * c * I / N * S - γ * I
    du[3] = γ * I
    nothing
end

prob = ODEProblem(sir_ode!, [990.0, 10.0, 0.0], (0.0, 40.0), [0.05, 10.0, 0.2
sol = solve(prob, dt = 0.1)

df = DataFrame(t=sol.t, S=[u[1] for u in sol.u], I=[u[2] for u in sol.u], R=[
plt = plot(df.t, [df.S df.I df.R], label=[L"S" L"I" L"R"], title="Модель SIR"
savefig(plt, plotsdir(script_name, "sir_lit.png"))
```

```
"/home/mmulitina/work/study/2026-1/2026-1-study-simulation-
modeling/labs/lab02/project/plots/sir_lit.png"
```

4. Параметрический анализ SIR

5. Параметрическое исследование SIR

```julia
using DrWatson
@quickactivate "project"
using DifferentialEquations, DataFrames, Plots

script_name = splitext(basename(PROGRAM_FILE))[1]
mkpath(plotsdir(script_name))
```

```
function sir_ode!(du, u, p, t)
    S, I, R = u
    β, c, γ = p
    N = S + I + R
    du[1] = -β * c * I / N * S
    du[2] = β * c * I / N * S - γ * I
    du[3] = γ * I
    nothing
end

beta_values = [0.02, 0.05, 0.08, 0.12]
plt = plot(title="Влияние β на I(t)", xlabel="Время", ylabel="I(t)")

for β in beta_values
    prob = ODEProblem(sir_ode!, [990.0, 10.0, 0.0], (0.0, 40.0), [β, 10.0, 0
    sol = solve(prob, Tsit5())
    plot!(plt, sol.t, [u[2] for u in sol.u], label="β = $β", lw=2)
end

savefig(plt, plotsdir(script_name, "beta_sens.png"))
```

```
"/home/mmulitina/work/study/2026-1/2026-1-study-simulation-
modeling/labs/lab02/project/plots/beta_sens.png"
```

6. Модель Лотки-Вольтерры

7. Модель Лотки-Вольтерры

Уравнения:

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta x y \\ \frac{dy}{dt} = \delta x y - \gamma y \end{cases}$$

```
using DrWatson
@quickactivate "project"
using DifferentialEquations, DataFrames, Plots, LaTeXStrings

script_name = splitext(basename(PROGRAM_FILE))[1]
mkpath(plotsdir(script_name))
```

```
function lotka_volterra!(du, u, p, t)
    x, y = u
    α, β, δ, γ = p
    du[1] = α*x - β*x*y
    du[2] = δ*x*y - γ*y
    nothing
end

prob_lv = ODEProblem(lotka_volterra!, [40.0, 9.0], (0.0, 200.0), [0.1, 0.02,
sol_lv = solve(prob_lv, Tsit5(), dt=0.1)

df_lv = DataFrame(t=sol_lv.t, prey=[u[1] for u in sol_lv.u], pred=[u[2] for u

plt1 = plot(df_lv.t, [df_lv.prey df_lv.pred], label=[L"x" L"y"], title="Динам
savefig(plt1, plotsdir(script_name, "lv_dyn.png"))

plt2 = plot(df_lv.prey, df_lv.pred, title="Фазовый портрет", xlabel="Жертвы",
savefig(plt2, plotsdir(script_name, "lv_phase.png"))
```

```
"/home/mmulitina/work/study/2026-1/2026-1-study-simulation-modeling/labs/lab02/project/plots/lv_phase.png"
```