

Лабораторная работа №6

Компьютерный практикум по статистическому анализу данных

Зиязетдинов Алмаз Радикович

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Цель лабораторной работы

- Освоить специализированные пакеты для решения задач в непрерывном и дискретном времени.

Выполнение лабораторной работы

Решение обыкновенных дифференциальных уравнений

Для решения обыкновенных дифференциальных уравнений (ОДУ) в Julia можно использовать пакет `differentialEquations.jl`.

Модель экспоненциального роста

1. Решение обыкновенных дифференциальных уравнений

1.1. Модель экспоненциального роста

```
[2]: # задаём описание модели с начальными условиями:  
a = 0.98  
f(u,p,t) = a*u  
u0 = 1.0  
# задаём интервал времени:  
tspan = (0.0,1.0)  
# решение:  
prob = ODEProblem(f,u0,tspan)  
sol = solve(prob)  
# строим графики:  
plot(sol, linewidth=5,title="Модель экспоненциального роста", xaxis="Время",yaxis="u(t)",label="u(t)")  
plot!(sol.t, t->1.0*exp(a*t),lin=:dash,label="Аналитическое решение")
```

[2]:

Модель экспоненциального роста

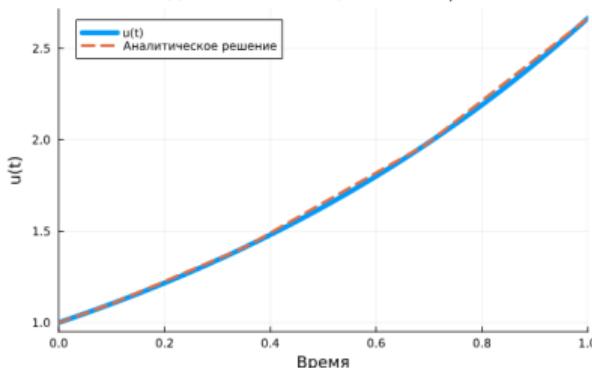


Рис. 1: График модели экспоненциального роста

Модель экспоненциального роста

```
[4]: # задаём точность решения:  
sol = solve(prob,abstol=1e-8,reltol=1e-8)  
# строим график:  
plot(sol, lw=2, color="black", title="Модель экспоненциального роста", xaxis="Время",yaxis="u(t)",label="Численное решение")  
plot!(sol.t, t->1.0*exp(a*t),lw=3,ls=:dash,color="red",label="Аналитическое решение")
```

[4]: Модель экспоненциального роста

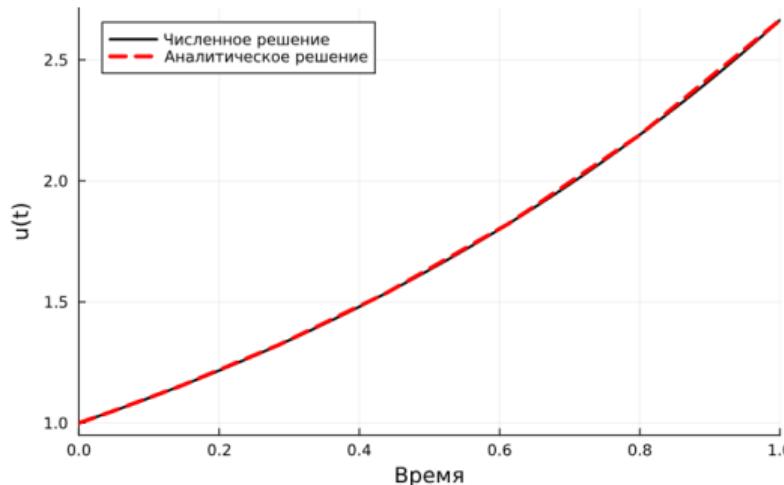


Рис. 2: График модели экспоненциального роста (задана точность решения)

Система Лоренца

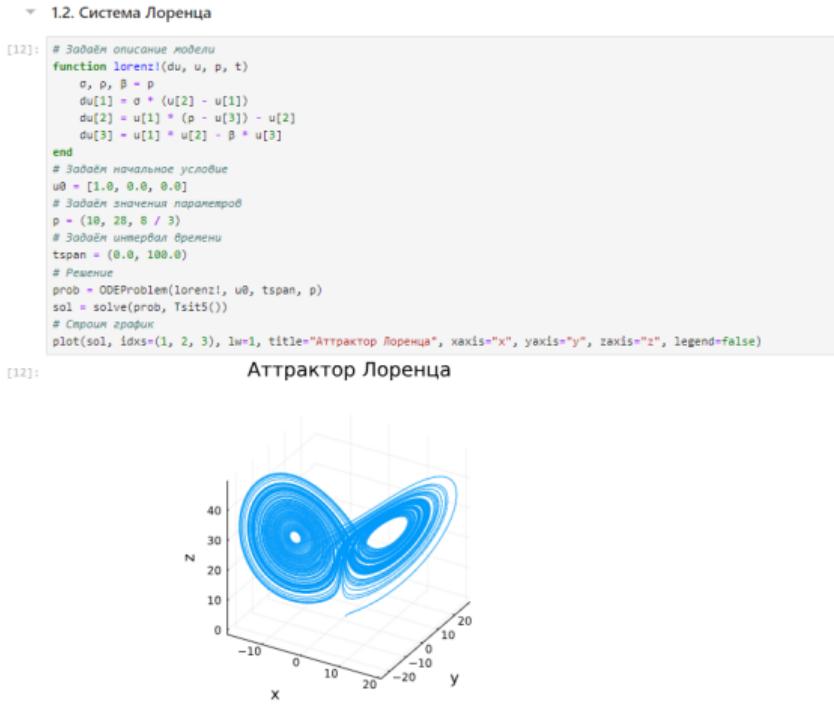


Рис. 3: Аттрактор Лоренца

Система Лоренца

```
[14]: # отключаем интерполяцию:  
plot(sol,vars=(1,2,3),denseplot=false, lw=1, title="Аттрактор Лоренца", xaxis="x",yaxis="y", zaxis="z", legend=false)
```

[14]:

Аттрактор Лоренца

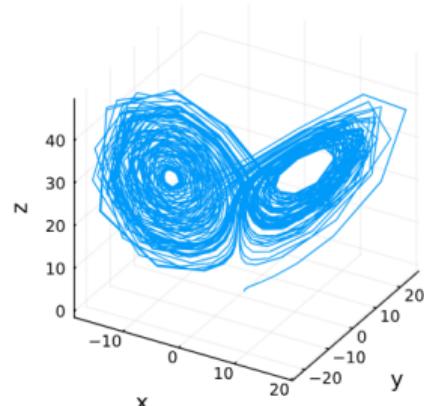


Рис. 4: Аттрактор Лоренца (интерполяция отключена)

Модель Лотки–Вольтерры

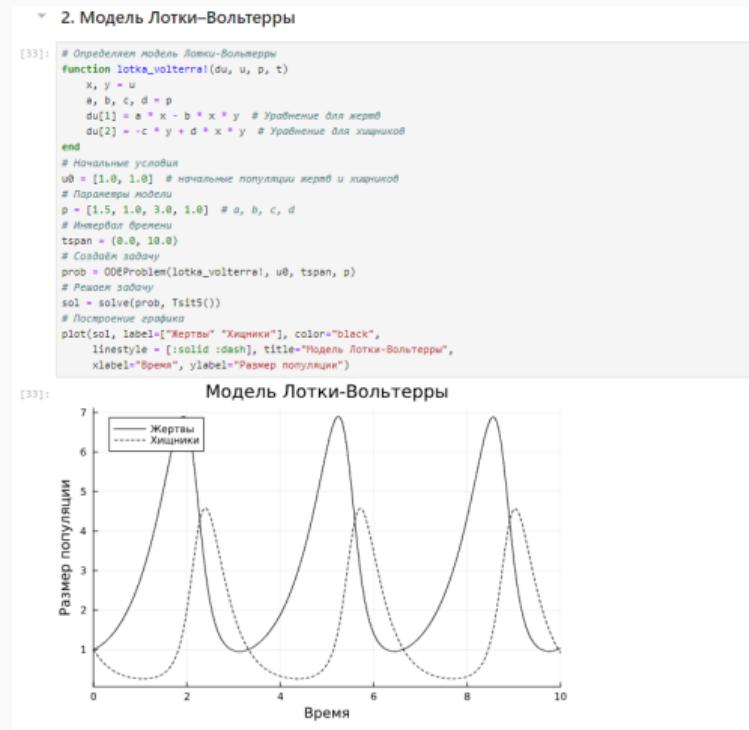


Рис. 5: Модель Лотки–Вольтерры: динамика изменения численности популяций

Модель Лотки–Вольтерры

```
[34]: # фазовый портрет:  
plot(sol,vars=(1,2), color="black", xaxis="Жертвы",yaxis="Хищники", legend=false)
```

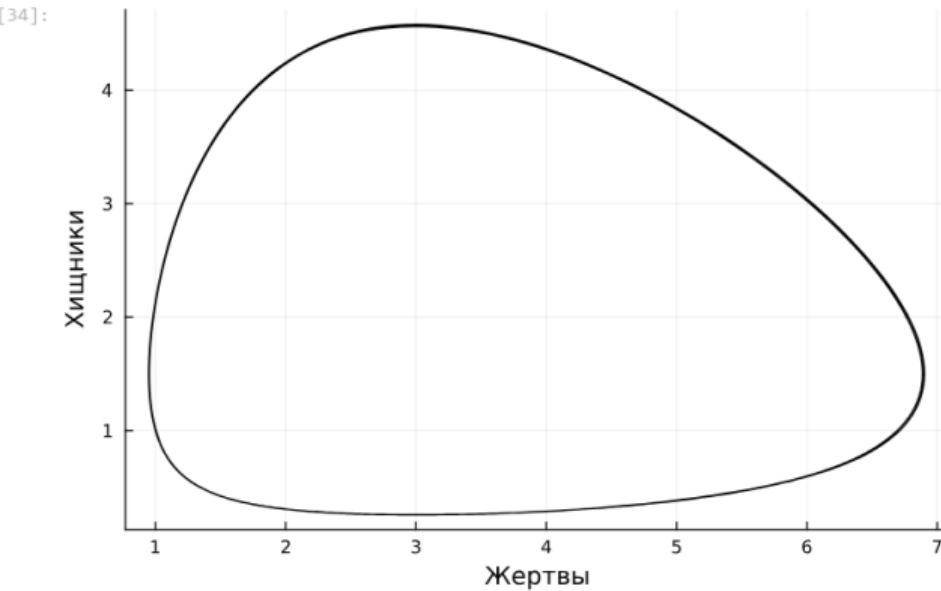


Рис. 6: Модель Лотки–Вольтерры: фазовый портрет

Самостоятельная работа

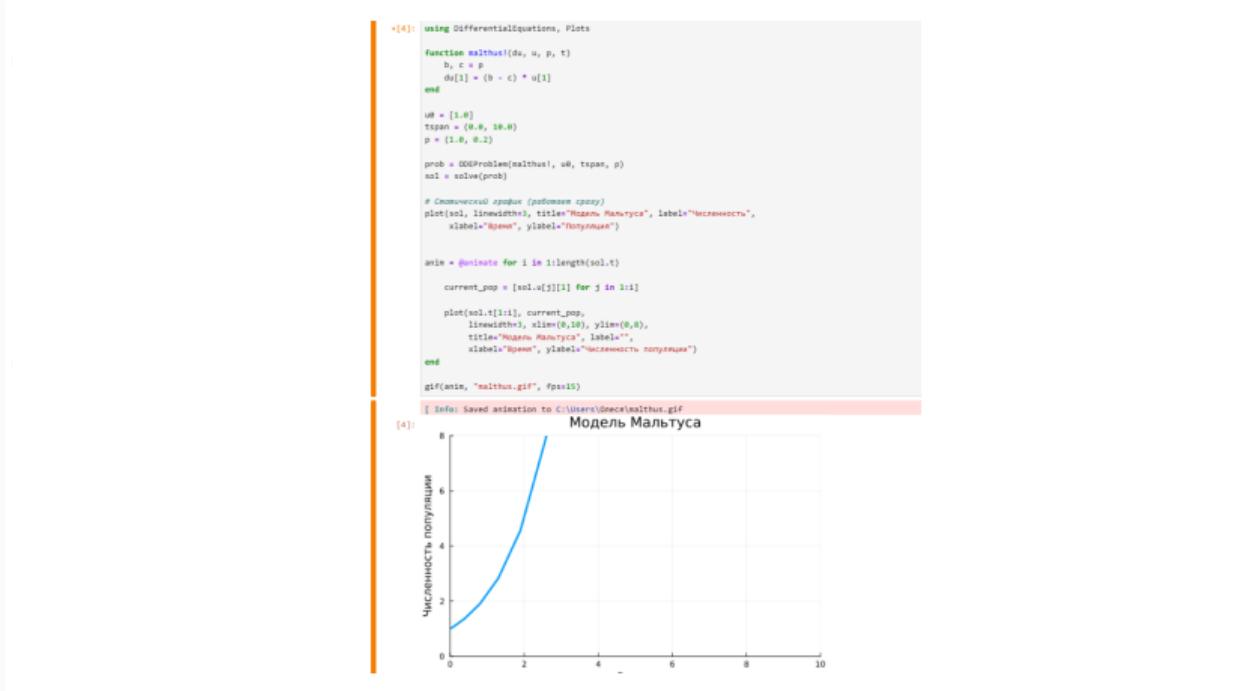


Рис. 7: Решение задания №1

Самостоятельная работа

```
[3]: using DifferentialEquations, Plots

# Логистическая модель роста
function logistic!(du, u, p, t)
    r, K = p
    du[1] = r * u[1] * (1 - u[1]/K)
end

# Параметры
u0 = [1.0] # начальная численность
tspan = (0.0, 20.0)
r = 0.5 # коэффициент роста
K = 10.0 # емкость экосистемы
p = (r, K)

# Задача и решение
prob = ODEProblem(logistic!, u0, tspan, p)
sol = solve(prob, Tsit5())

# Анимация
anim = @animate for i in 1:length(sol.t)
    plot(sol, index(1,i), tspan(0, sol.t[i]), linewidth=3, color=:blue,
        title="Логистическая модель роста", xlabel="Время", ylabel="Популяция",
        xlim=(0,20), ylim=(0,11), label="")
    scatter!([(sol.t[i]), [sol.u[i][1]]], color=:red, markersize=5,
        label="Добавляем текущую точку")
end

# Создание gif
gif(anim, "logistic_growth.gif", fps=18)
```

[Info: Saved animation to C:\Users\Olga\OneDrive\logistic_growth.gif
ЛОГИСТИЧЕСКАЯ МОДЕЛЬ РОСТА

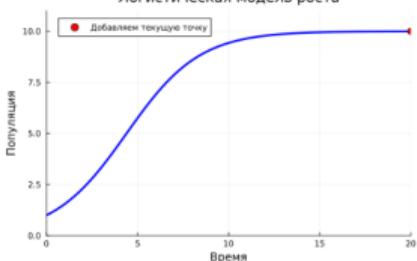


Рис. 8: Решение задания №2

Самостоятельная работа

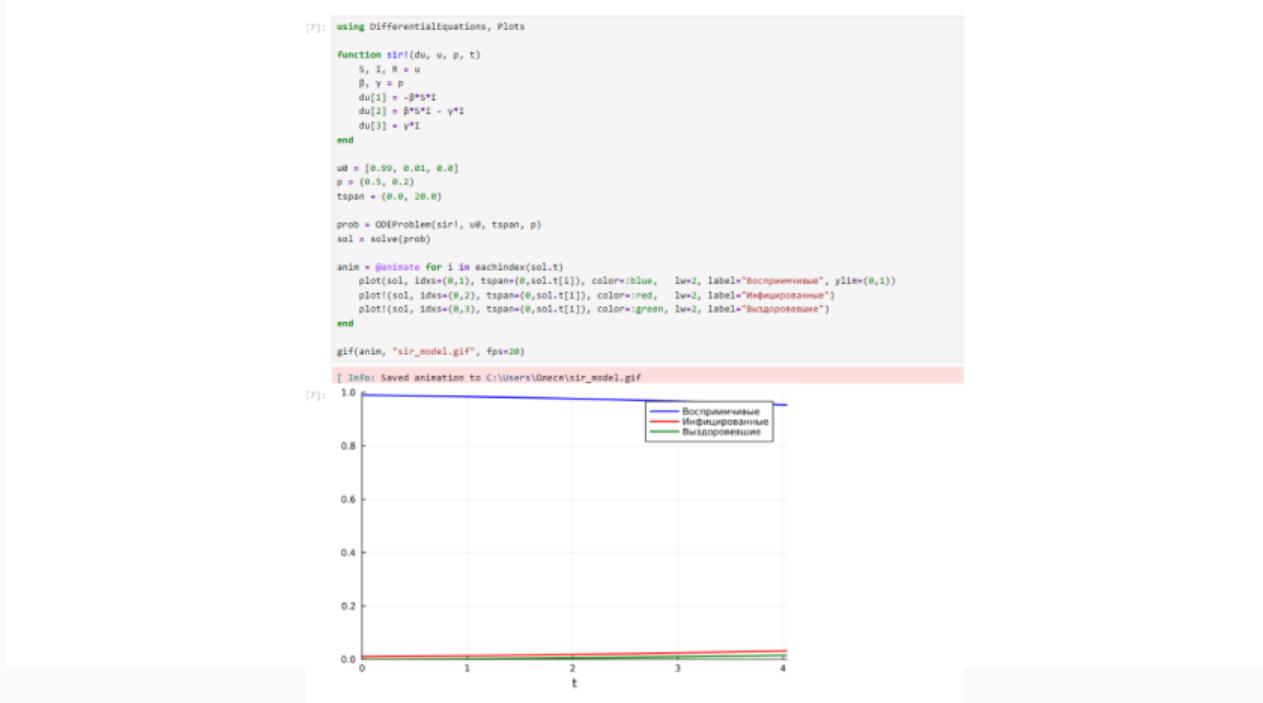


Рис. 9: Решение задания №3

Самостоятельная работа

```
[8]: using Plots  
  
a, b, c, d = 1.0, 0.1, 0.2, 0.05  
x, y = 10.0, 5.0  
times = 0:0.1:100  
X = zeros(length(times), 2)  
X[1,:] = [x, y]  
  
for i in 1:length(times)-1  
    x += (a*x - b*x*y) * 0.1  
    y += (-c*y + d*x*y) * 0.1  
    X[i+1,:] = [x, y]  
end  
  
plot(X[:,1], X[:,2], label="Фазовый портрет", xlabel="x1", ylabel="x2")
```

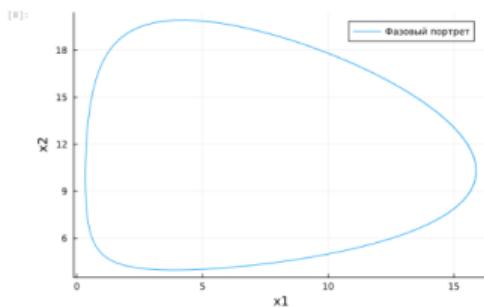


Рис. 10: Решение задания №4

Самостоятельная работа

[1]

```
[12]: using DifferentialEquations, Plots

function harmonic!(du, u, p, t)
    x, v = u
    du[1] = v
    du[2] = -u[1]
end

u0 = [0.0, 1.0]
tspan = (0.0, 50.0)
prob = ODEProblem(harmonic!, u0, tspan)
sol = solve(prob, Tsit5())

plot(sol, idxs=(1,2), label="Фазовый портрет", xlabel="Положение x", ylabel="Скорость v")
plot!(sol[:,1], sol[:,2], label="Траектория x(t), v(t)")
```

[1]

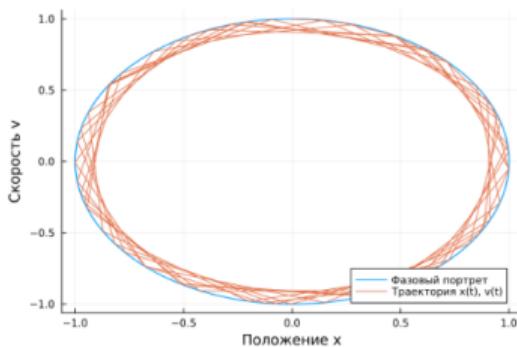


Рис. 11: Решение задания №5

Самостоятельная работа

```
[11]: using DifferentialEquations, Plots

function damped!(du, u, p, t)
    x, v = u
    y = 0.1
    du[1] = v
    du[2] = -2*y*v - u[1]
end

u0 = [1.0, 0.0]
tspan = (0.0, 50.0)
prob = ODEProblem(damped!, u0, tspan)
sol = solve(prob, Tsit5())

plot(sol, labels=["Положение x" "Скорость v"], title="Свободные колебания с потерей энергии",
      xlabel="Время", linewidth=2)
```

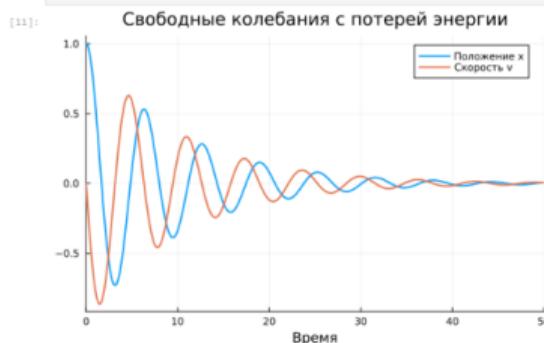


Рис. 12: Решение задания №6

Вывод

Вывод

- В ходе выполнения лабораторной работы были освоены специализированные пакеты для решения задач в непрерывном и дискретном времени.

Список литературы. Библиография

Список литературы. Библиография

[1] Julia Documentation: <https://docs.julialang.org/en/v1/>