

Лабораторная работа №1

Модель экспоненциального роста

Чувакина Мария Владимировна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Подготовка	6
3.2	Создание проекта DrWatson для лабораторных	6
3.3	Добавление необходимых пакетов	6
3.4	Проверка установки	8
3.5	Реализация модели	8
4	Выводы	21
	Список литературы	22

Список иллюстраций

3.1	Создадим проект DrWatson	6
3.2	Добавление необходимых пакетов	8
3.3	Проверка установки	8
3.4	Скрипт	9
3.5	Модель экспоненциального роста 1	9
3.6	Запуск скрипта	9
3.7	График	10
3.8	Производные форматы	12
3.9	Проверка	12
3.10	Запуск	12
3.11	Параметрический скрипт	19
3.12	Производные форматы	19

1 Цель работы

Целью работы являлось изучение модели экспоненциального роста, её программная реализация на языке Julia, проведение параметрического исследования и освоение принципов литературного программирования.

2 Задание

1. Выполнить задания из лабораторной работы
2. Оформить отчет по лабораторной работе с помощью Markdown


```
cat > add_packages.jl << 'EOF'
#!/usr/bin/env julia
using Pkg
Pkg.activate(".")

packages = [
    "DrWatson",
    "DifferentialEquations",
    "Plots",
    "DataFrames",
    "CSV",
    "JLD2",
    "Literate",
    "IJulia",
    "BenchmarkTools"
]

println("Установка пакетов...")
Pkg.add(packages)
println("✓ Все пакеты установлены!")
EOF
```

```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ cat > add_packages.jl << 'EOF'
#!/usr/bin/env julia
using Pkg
Pkg.activate(".")

packages = [
    "DrWatson",
    "DifferentialEquations",
    "Plots",
    "DataFrames",
    "CSV",
    "JLD2",
    "Literate",
    "IJulia",
    "BenchmarkTools"
]

println("Установка пакетов...")
Pkg.add(packages)
println("✓ Все пакеты установлены!")
EOF

```

Рис. 3.2: Добавление необходимых пакетов

3.4 Проверка установки

```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ julia --project=. scripts/test_setup.jl
✓ Проект активирован: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project

Проверка пакетов:
✓ DrWatson
✓ DifferentialEquations
✓ Plots
✓ DataFrames
✓ CSV
✓ JLD2
✓ Literate
✓ IJulia
✓ BenchmarkTools

Структура проекта:
Корень: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project
Данные: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/data
Скрипты: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/src
Графики: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/plots
mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $

```

Рис. 3.3: Проверка установки

3.5 Реализация модели

Создадим папку для скриптов

Создадим проверочный скрипт


```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ cat > scripts/test_setup.jl
<< 'EOF'
#!/usr/bin/env julia
using DrWatson
@quickactivate "project"

println("✓ Проект активирован: ", projectdir())

packages = [
    "DrWatson", "DifferentialEquations", "Plots",
    "DataFrames", "CSV", "JLD2", "Literate", "IJulia", "BenchmarkTools"
]

println("\nПроверка пакетов:")
for pkg in packages
    try
        eval(Meta.parse("using $pkg"))
        println(" ✓ $pkg")
    catch e
        println(" ✗ $pkg: Ошибка загрузки")
    end
end

println("\nСтруктура проекта:")
println(" Корень: ", projectdir())
println(" Данные: ", datadir())
println(" Скрипты: ", srcdir())

```

Рис. 3.4: Скрипт

Создадим первый скрипт (модель экспоненциального роста)

```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ cat > scripts/01_exponential_growth.jl << 'EOF'
# Экспоненциальный рост
# **Цель:** Исследовать решение уравнения  $du/dt = \alpha u$ .

using DrWatson
@quickactivate "project"
using DifferentialEquations
using Plots
using DataFrames
using JLD2

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

function exponential_growth!(du, u, p, t)
    α = p
    du[1] = α * u[1]
end

u0 = [1.0]
α = 0.3
tspan = (0.0, 10.0)

prob = ODEProblem(exponential_growth!, u0, tspan, α)

```

Рис. 3.5: Модель экспоненциального роста 1

Запустим модель

```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ julia --project=. scripts/02_exponential_growth.jl
Базовые параметры эксперимента:
α = 0.3
u0 = [1.0]
saveat = 0.1
solver = Tsit5{typeof(OrdinaryDiffEqCore.trivial_limiter!), typeof(OrdinaryDiffEqCore.trivial_limiter!), Static.False}(OrdinaryDiffEqCore.trivial_limiter!, OrdinaryDiffEqCore.trivial_limiter!, static(false))
tspan = (0.0, 10.0)

Результаты базового эксперимента:
Финальная популяция: 20.085448516186737
Время удвоения: 2.31
Файл результатов: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/data/02_exponential_growth/single/exp_growth_saveat=0.1_α=0.3.jld2

```

Рис. 3.6: Запуск скрипта

Посмотрим на график который создался

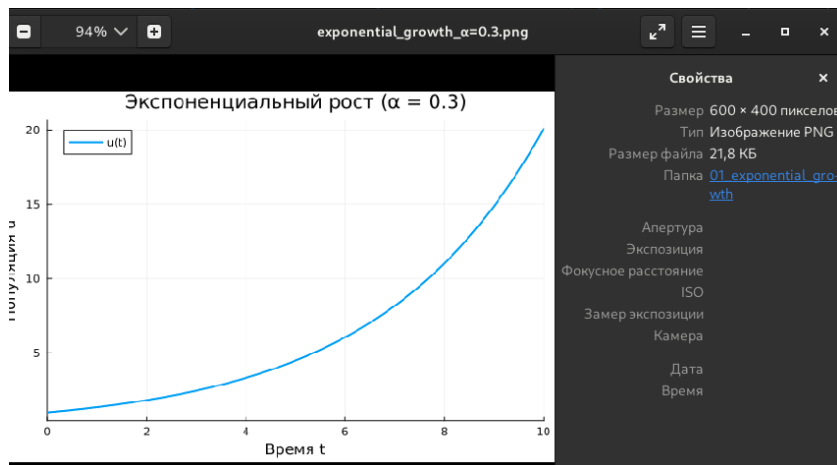


Рис. 3.7: График

Создадим скрипт для генерации производных форматов

```
cat > scripts/tangle.jl << 'EOF'
#!/usr/bin/env julia
using DrWatson
@quickactivate
using Literate

function main()
    if length(ARGS) == 0
        println("Использование: julia tangle.jl <путь_к_скрипту>")
        return
    end

    script_path = ARGS[1]
    script_name = splitext(basename(script_path))[1]

    println("Генерация из: $script_path")

    # Чистый скрипт (без комментариев)
```

```

scripts_dir = scriptsdir(script_name)
Literate.script(script_path, scripts_dir; credit=false)
println("  ✓ Чистый скрипт создан")

# Quarto-документ
quarto_dir = projectdir("markdown", script_name)
mkpath(quarto_dir)
Literate.markdown(script_path, quarto_dir;
                  flavor=Literate.QuartoFlavor(),
                  name=script_name, credit=false)
println("  ✓ Quarto документ создан")

# Jupyter notebook
notebooks_dir = projectdir("notebooks", script_name)
mkpath(notebooks_dir)
Literate.notebook(script_path, notebooks_dir, name=script_name;
                  execute=false, credit=false)
println("  ✓ Jupyter notebook создан")

println("\nГотово!")
end

if abspath(PROGRAM_FILE) == @__FILE__
    main()
end
EOF

```

Сгенерируем производные форматы

```

mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ julia --project=. scripts/t
angle.jl scripts/01_exponential_growth.jl
Генерация из: scripts/01_exponential_growth.jl
[ Info: generating plain script file from `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scr
ipts/01_exponential_growth.jl`
[ Info: writing result to `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/01_exponentia
l_growth/01_exponential_growth.jl`
✓ Чистый скрипт создан
[ Info: generating markdown page from `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/0
1_exponential_growth.jl`
[ Info: writing result to `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/markdown/01_exponentia
l_growth/01_exponential_growth.qmd`
✓ Quarto документ создан
[ Info: generating notebook from `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/01_exp
ponential_growth.jl`
[ Info: writing result to `~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/notebooks/01_exponent
ial_growth/01_exponential_growth.ipynb`
✓ Jupyter notebook создан
Готово!

```

Рис. 3.8: Производные форматы

Проверим созданные файлы

```

Готово!
mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ # Проверьте чистый скрипт
ls -la scripts/01_exponential_growth/

# Проверьте Quarto документ
ls -la markdown/01_exponential_growth/

# Проверьте Jupyter notebook
ls -la notebooks/01_exponential_growth/
итого 5
drwxr-xr-x 2 mvchuvakina studsci 2048 фев 16 13:55 .
drwxr-xr-x 5 mvchuvakina studsci 2048 фев 16 14:24 ..
-rw-r--r-- 1 mvchuvakina studsci 974 фев 16 15:55 01_exponential_growth.jl
итого 6
drwxr-xr-x 2 mvchuvakina studsci 2048 фев 16 13:55 .
drwxr-xr-x 5 mvchuvakina studsci 2048 фев 16 14:24 ..
-rw-r--r-- 1 mvchuvakina studsci 1119 фев 16 15:55 01_exponential_growth.qmd
итого 8
drwxr-xr-x 3 mvchuvakina studsci 2048 фев 16 14:05 .
drwxr-xr-x 5 mvchuvakina studsci 2048 фев 16 14:24 ..
-rw-r--r-- 1 mvchuvakina studsci 1963 фев 16 15:55 01_exponential_growth.ipynb
drwxr-xr-x 2 mvchuvakina studsci 2048 фев 16 14:01 .ipynb_checkpoints
mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $

```

Рис. 3.9: Проверка

Запустим Jupyter notebook

```

@save datadir(script_name, all_results.jl02 ) dt

```

Первые 5 строк результатов:
5x2 DataFrame

Row	t	u
	Float64	Float64
1	0.0	1.0
2	0.1	1.03045
3	0.2	1.06184
4	0.3	1.09417
5	0.4	1.1275

Аналитическое время удвоения: 2.31

Рис. 3.10: Запуск

Создадим параметрический скрипт (02_exponential_growth.jl)

```
cat > scripts/02_exponential_growth.jl << 'EOF'
```

```
# # Параметрическое исследование экспоненциального роста
```

```
using DrWatson
```

```
@quickactivate "project"
```

```
using DifferentialEquations
```

```
using DataFrames
```

```
using Plots
```

```
using JLD2
```

```
using BenchmarkTools
```

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

```
mkpath(plotsdir(script_name))
```

```
mkpath(datadir(script_name))
```

```
function exponential_growth!(du, u, p, t)
```

```
    ρ = p
```

```
    du[1] = ρ * u[1]
```

```
end
```

```
# Базовый набор параметров
```

```
base_params = Dict{
```

```
    :u0 => [1.0],
```

```
    :ρ => 0.3,
```

```
    :tspan => (0.0, 10.0),
```

```
    :solver => Tsit5(),
```

```
    :saveat => 0.1
```

```
)
```

```
println("Базовые параметры эксперимента:")
```

```

for (key, value) in base_params
    println(" $key = $value")
end

# Функция для запуска одного эксперимента
function run_single_experiment(params::Dict)
    u0 = params[:u0]
    Δ = params[:Δ]
    tspan = params[:tspan]
    solver = params[:solver]
    saveat = params[:saveat]

    prob = ODEProblem(exponential_growth!, u0, tspan, Δ)
    sol = solve(prob, solver; saveat=saveat)

    final_population = last(sol.u)[1]
    doubling_time = log(2) / Δ

    return Dict(
        "solution" => sol,
        "time_points" => sol.t,
        "population_values" => first(sol.u),
        "final_population" => final_population,
        "doubling_time" => doubling_time,
        "parameters" => params
    )
end

# Запуск базового эксперимента

```

```

data, path = produce_or_load(
    datadir(script_name, "single"),
    base_params,
    run_single_experiment,
    prefix = "exp_growth",
    tag = false,
    verbose = true
)

println("\nРезультаты базового эксперимента:")
println("  Финальная популяция: ", data["final_population"])
println("  Время удвоения: ", round(data["doubling_time"]; digits=2))
println("  Файл результатов: ", path)

# Визуализация базового эксперимента
p1 = plot(data["time_points"], data["population_values"],
    label="□ = $(base_params[:□])",
    xlabel="Время t", ylabel="Популяция u(t)",
    title="Экспоненциальный рост (базовый эксперимент)",
    lw=2, legend=:topleft, grid=true)

savefig(plotsdir(script_name, "single_experiment.png"))

# Параметрическое сканирование
param_grid = Dict(
    :u0 => [[1.0]],
    :□ => [0.1, 0.3, 0.5, 0.8, 1.0],
    :tspan => [(0.0, 10.0)],
    :solver => [Tsit5()],

```

```

        :saveat => [0.1]
    )

using DrWatson: dict_list
all_params = dict_list(param_grid)

println("\n" * "="^60)
println("ПАРАМЕТРИЧЕСКОЕ СКАНИРОВАНИЕ")
println("Всего комбинаций параметров: ", length(all_params))
println("Исследуемые значения  $\theta$ : ", param_grid[: $\theta$ ])
println("="^60)

# Запуск всех экспериментов
all_results = []
all_dfs = []

for (i, params) in enumerate(all_params)
    println("Прогресс:  $i / \text{length}(\text{all\_params})$  |  $\theta = \text{params}[:\theta]$ ")

    data, path = produce_or_load(
        datadir(script_name, "parametric_scan"),
        params,
        run_single_experiment,
        prefix = "scan",
        tag = false,
        verbose = false
    )

    result_summary = merge(params, Dict(

```



```

        :final_population => data["final_population"],
        :doubling_time => data["doubling_time"],
        :filepath => path
    ))

    push!(all_results, result_summary)

    df = DataFrame(
        t = data["time_points"],
        u = data["population_values"],
        v = fill(params[:,1], length(data["time_points"]))
    )
    push!(all_dfs, df)
end

# Анализ результатов
results_df = DataFrame(all_results)
println("\nСводная таблица результатов:")
println(results_df[:, [:1, :final_population, :doubling_time]])

# Сравнительный график
p2 = plot(size=(800, 500), dpi=150)

for params in all_params
    data, _ = produce_or_load(
        datadir(script_name, "parametric_scan"),
        params,
        run_single_experiment,
        prefix = "scan"
    )
end

```

```

    )
    plot!(p2, data["time_points"], data["population_values"],
          label=" $\alpha = \$(params[:\alpha])$ ", lw=2, alpha=0.8)
end

plot!(p2, xlabel="Время t", ylabel="Популяция u(t)",
      title="Параметрическое исследование: влияние  $\alpha$  на рост",
      legend=:topleft, grid=true)

savefig(plotsdir(script_name, "parametric_scan_comparison.png"))

# График зависимости времени удвоения от  $\alpha$ 
p3 = scatter(results_df. $\alpha$ , results_df.doubling_time,
             label="Численное решение",
             xlabel="Скорость роста  $\alpha$ ",
             ylabel="Время удвоения  $t_2$ ",
             title="Зависимость времени удвоения от  $\alpha$ ",
             markersize=8, markercolor=:red, legend=:topright)

 $\alpha$ _range = 0.1:0.01:1.0
plot!(p3,  $\alpha$ _range, log.(2) ./  $\alpha$ _range,
      label="Теория:  $t_2 = \ln(2)/\alpha$ ",
      lw=2, linestyle=:dash, linecolor=:blue)

savefig(plotsdir(script_name, "doubling_time_vs_alpha.png"))

# Сохранение всех результатов
@save datadir(script_name, "all_results.jld2") base_params param_grid all_params resul
@save datadir(script_name, "all_plots.jld2") p1 p2 p3

```

```
println("\n" * "="^60)
println("ЛАБОРАТОРНАЯ РАБОТА ЗАВЕРШЕНА")
println("="^60)
println("\nРезультаты сохранены в:")
println("  - data/$(script_name)/single/ - базовый эксперимент")
println("  - data/$(script_name)/parametric_scan/ - параметрическое сканирование")
println("  - data/$(script_name)/all_results.jld2 - сводные данные")
println("  - plots/$(script_name)/ - все графики")
EOF
```

Запустим параметрический скрипт

```
EOF
mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ julia --project=. scripts/0
2_exponential_growth.jl
Базовые параметры эксперимента:
α = 0.3
u0 = [1.0]
saveat = 0.1
solver = Tsit5{typeof(OrdinaryDiffEqCore.trivial_limiter!), typeof(OrdinaryDiffEqCore.trivial_limiter!), Sta
tic.False}(OrdinaryDiffEqCore.trivial_limiter!, OrdinaryDiffEqCore.trivial_limiter!, static(false))
tspan = (0.0, 10.0)
[ Info: File /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/
project/data/02_exponential_growth/single/exp_growth_saveat=0.1_α=0.3.jld2 does not exist. Producing it now...
[ Info: File /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs/lab01/
project/data/02_exponential_growth/single/exp_growth_saveat=0.1_α=0.3.jld2 saved.

Результаты базового эксперимента:
Финальная популяция: 20.085448516186737
Время удвоения: 2.31
Файл результатов: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchuvakina/work/study/2026-1/2026-1--study--mathmod/labs
/lab01/project/data/02_exponential_growth/single/exp_growth_saveat=0.1_α=0.3.jld2
=====
ПАРАМЕТРИЧЕСКОЕ СКАНИРОВАНИЕ
```

Рис. 3.11: Параметрический скрипт

Сгенерируем производные форматы для параметрического скрипта

```
mvchuvakina@dk7n02 ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project $ julia --project=. scripts/t
angle.jl scripts/02_exponential_growth.jl
Генерация из: scripts/02_exponential_growth.jl
[ Info: generating plain script file from '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scr
ipts/02_exponential_growth.jl'
[ Info: writing result to '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/02_exponentia
l_growth/02_exponential_growth.jl'
✓ Чистый скрипт создан
[ Info: generating markdown page from '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/0
2_exponential_growth.jl'
[ Info: writing result to '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/markdown/02_exponenti
al_growth/02_exponential_growth.qmd'
✓ Quarto документ создан
[ Info: generating notebook from '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/scripts/02_exp
onential_growth.jl'
[ Info: writing result to '~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/project/notebooks/02_exponent
ial_growth/02_exponential_growth.ipynb'
✓ Jupyter notebook создан
Готово!
```

Рис. 3.12: Производные форматы

Интегрируем в отчёт. Для этого, мы

Перейдем в папку с отчётом

```
cd ~/work/study/2026-1/2026-1--study--mathmod/labs/lab01/report
```

Отредактируем файл `mathmod-lab01-report.qmd` и добавьте в него:

```
markdown
```

```
### Реализация модели
```

```
{{< include ../../project/markdown/01_exponential_growth/01_exponential_growth.qmd >}}
```

```
### Параметрическое исследование
```

```
{{< include ../../project/markdown/02_exponential_growth/02_exponential_growth.qmd >}}
```

Скомпилируем отчёт:

```
bash
```

```
make
```

После этого отправим изменения на GitHub

4 Выводы

В процессе выполнения данной лабораторной работы я изучила модели экспоненциального роста, её программную реализацию на языке Julia, проведение параметрического исследования и освоение принципов литературного программирования.

Список литературы