

Математическое моделирование

Лабораторная работа № 1

Максим Новичков

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
3.1 Ключевые характеристики модели	8
3.2 Области применения	9
3.3 Ограничения модели	9
4 Выполнение лабораторной работы	10
5 Экспоненциальный рост	14
5.1 Инициализация проекта и загрузка пакетов	14
5.2 Определение модели	14
5.3 Первый запуск с параметрами по умолчанию	15
5.4 Визуализация результатов	15
5.5 Анализ результатов	15
5.6 Сохранение всех результатов	16
6 Параметрическое исследование экспоненциального роста	17
6.1 Активация проекта и загрузка пакетов	17
6.2 Определение модели	17
6.3 Определение параметров в Dict	18
6.4 Функция-обертка для запуска одного эксперимента	18
6.5 Запуск базового эксперимента	19
6.6 Визуализация базового эксперимента	20
6.7 Параметрическое сканирование	20
6.8 Запуск всех экспериментов и собор результатов	21
6.9 Анализ и визуализация результатов сканирования	22
6.10 Бенчмаркинг с разными параметрами	25
6.11 Сохранение всех результатов	26
7 Выводы	28
Список литературы	29

Список иллюстраций

4.1	Экспоненциальный рост (базовый эксперимент)	10
4.2	Параметрическое исследование: влияние α на рост	11
4.3	Зависимость времени удвоения от α	12
4.4	Зависимость времени вычисления от α	13

Список таблиц

1 Цель работы

Освоение модели экспоненциального роста, изучение её математической формализации, аналитического решения соответствующего дифференциального уравнения, а также исследование влияния коэффициента роста на динамику системы. Дополнительно предполагается приобретение навыков параметрического анализа и интерпретации полученных результатов.

2 Задание

В рамках лабораторной работы необходимо рассмотреть модель экспоненциального роста в качестве демонстрационного примера. Требуется изучить её математическое описание, проанализировать решение дифференциального уравнения и выполнить параметрическое исследование, показывающее, как коэффициент роста влияет на поведение системы, время удвоения величины и вычислительные характеристики.

3 Теоретическое введение

Экспоненциальный рост представляет собой процесс изменения величины, при котором скорость её увеличения в каждый момент времени прямо пропорциональна текущему значению. Это означает, что чем больше становится система, тем быстрее происходит её дальнейшее увеличение.

В качестве наглядных аналогий можно привести начисление сложных процентов в банковской сфере или увеличение снежного кома, который с ростом массы начинает набирать её всё быстрее.

С математической точки зрения модель описывается дифференциальным уравнением:

$$\frac{du}{dt} = \alpha u$$

где:

—

u

— текущее значение исследуемой величины (например, численность популяции, объём капитала или число инфицированных);

—

t

— время;

$$\frac{du}{dt}$$

— скорость изменения величины;

—

$$\alpha$$

— коэффициент роста (мальтузианский параметр).

При

$$\alpha > 0$$

система демонстрирует рост. Если же

$$\alpha < 0$$

, наблюдается экспоненциальное убывание.

Смысл уравнения заключается в том, что скорость изменения величины определяется её текущим значением.

Аналитическое решение данного уравнения имеет вид:

$$u(t) = u_0 e^{\alpha t}$$

где

$$u_0$$

— начальное значение функции.

3.1 Ключевые характеристики модели

- Увеличение в два раза происходит за фиксированный промежуток времени.
- Время удвоения определяется выражением:

$$T_2 = \frac{\ln(2)}{\alpha} \approx \frac{0.693}{\alpha}$$

- Этот показатель зависит исключительно от коэффициента роста и не связан с текущим значением величины.

3.2 Области применения

- Биология: увеличение численности бактерий при наличии достаточного количества ресурсов.
- Финансовая сфера: рост капитала при начислении сложных процентов.
- Эпидемиология: начальный этап распространения инфекционных заболеваний.
- Демография: рост населения в определённые исторические периоды.
- Физика: процессы цепных ядерных реакций.
- Информационные технологии: увеличение вычислительных мощностей и сетевого трафика.

3.3 Ограничения модели

Экспоненциальная модель является упрощённой и не учитывает реальных ограничений среды. На практике бесконечный рост невозможен из-за исчерпания ресурсов. Со временем динамика замедляется и чаще всего переходит к логистическому, S-образному характеру развития.

4 Выполнение лабораторной работы

В ходе выполнения работы было проведено моделирование процесса экспоненциального роста на основе аналитического решения соответствующего дифференциального уравнения.

Сначала был выполнен базовый эксперимент при фиксированном значении коэффициента роста

$$\alpha = 0.3$$

. Полученный график иллюстрирует зависимость величины от времени и демонстрирует характерную ускоряющуюся динамику.

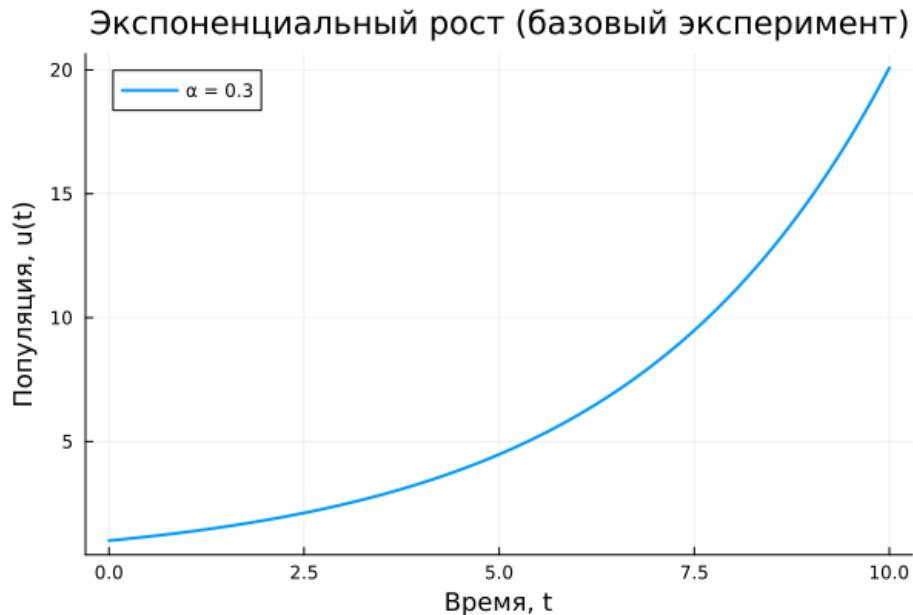


Рисунок 4.1: Экспоненциальный рост (базовый эксперимент)

На начальном участке наблюдается относительно медленное увеличение, однако по мере течения времени скорость роста возрастает, и к концу интервала функция резко увеличивается.

Далее было проведено параметрическое исследование, направленное на оценку влияния коэффициента

$$\alpha$$

на поведение системы. Для этого были построены графики при различных значениях данного параметра.

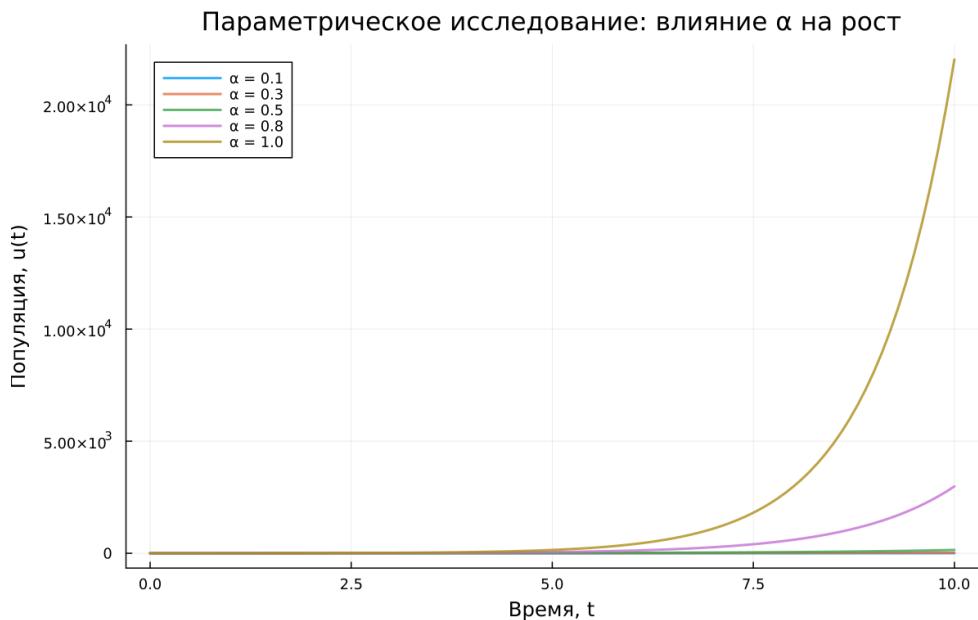


Рисунок 4.2: Параметрическое исследование: влияние α на рост

Полученные результаты показывают, что увеличение значения

$$\alpha$$

приводит к значительному ускорению роста. При малых значениях коэффициента функция изменяется плавно, тогда как при больших – быстро возрастает и достигает высоких значений за короткий промежуток времени.

Дополнительно была изучена зависимость времени удвоения от коэффици-

ента роста. В теории оно определяется формулой

$$T_2 = \ln(2)/\alpha$$

, что подтвердились и при численном моделировании.

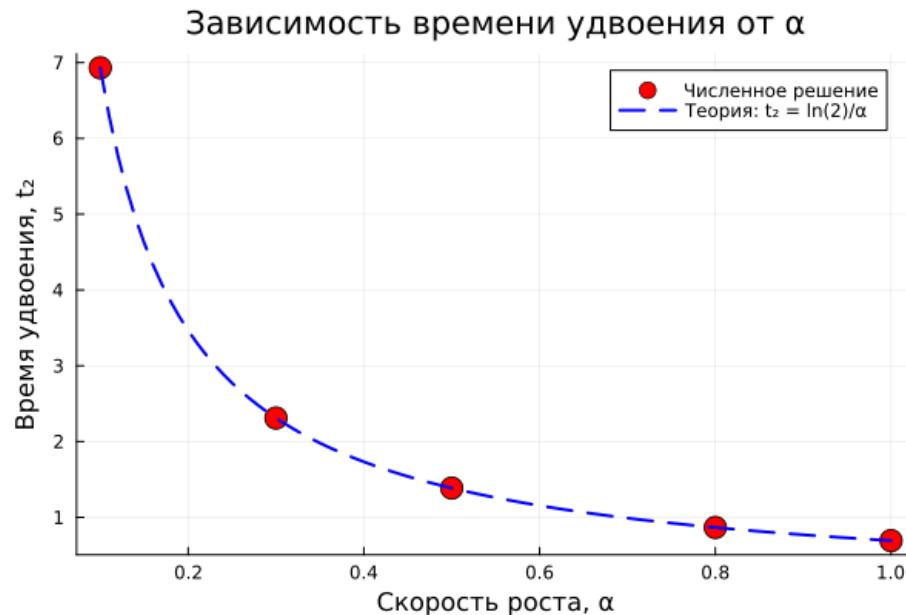


Рисунок 4.3: Зависимость времени удвоения от α

Из графика следует, что с увеличением

$$\alpha$$

время, необходимое для удвоения величины, сокращается, что полностью соответствует теоретическим ожиданиям.

Также был выполнен анализ зависимости времени вычислений от параметра роста.

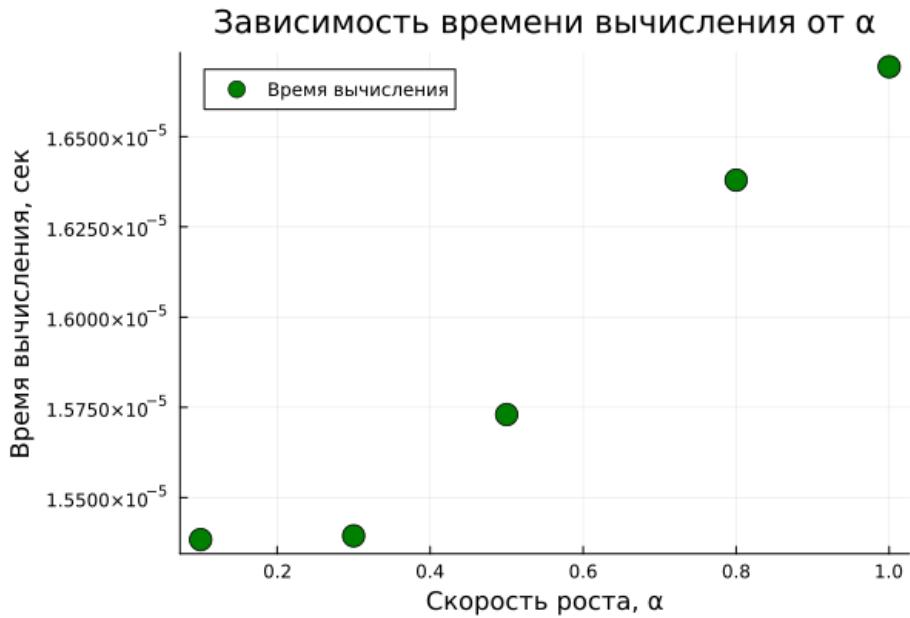


Рисунок 4.4: Зависимость времени вычисления от α

Отмечается небольшое увеличение времени расчёта при росте значения

$$\alpha$$

, что объясняется увеличением величин функции и особенностями численной обработки данных.

Для проведения моделирования и построения графиков использовались внешние файлы с программным кодом:

5 Экспоненциальный рост

Цель: Исследовать решение уравнения $\frac{du}{dt} = pu$.

5.1 Инициализация проекта и загрузка пакетов

```
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))
```

5.2 Определение модели

Уравнение экспоненциального роста:

```
# $\frac{du}{dt} = pu$ ,  $u(0) = u_0$ 
function exponential_growth!(du, u, p, t)
    du = p
```

```
du[1] = Δ * u[1]
end
```

5.3 Первый запуск с параметрами по умолчанию

Зададим начальные параметры:

```
u0 = [1.0] # начальная популяция
Δ = 0.3 # скорость роста
tspan = (0.0, 10.0) # временной интервал
prob = ODEProblem(exponential_growth!, u0, tspan, Δ)
sol = solve(prob, Tsit5(), saveat=0.1)
```

5.4 Визуализация результатов

Построим график решения:

```
plot(sol, label="u(t)", xlabel="Время t", ylabel="Популяция u",
      title="Экспоненциальный рост (Δ = $Δ)", lw=2, legend=:topleft)
```

Сохраним график в папку plots

```
savefig(plotsdir(script_name, "exponential_growth_Δ=$Δ.png"))
```

5.5 Анализ результатов

Создадим таблицу с данными:

```
df = DataFrame(t=sol.t, u=first.(sol.u))
println("Первые 5 строк результатов:")
println(first(df, 5))
```

Вычислим удвоение популяции:

```
u_final = last(sol.u)[1]
doubling_time = log(2) / □
println("\nАналитическое время удвоения: ", round(doubling_time; digits=2))
```

5.6 Сохранение всех результатов

```
@save datadir(script_name, "all_results.jld2") df
```

6 Параметрическое исследование экспоненциального роста

6.1 Активация проекта и загрузка пакетов

ИЗМЕНЕНИЕ: Добавлен DrWatson для управления проектом и параметрами

```
using DrWatson  
@quickactivate "project" # Активация проекта DrWatson  
using DifferentialEquations  
using DataFrames  
using Plots  
using JLD2  
using BenchmarkTools
```

Установка каталогов

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

6.2 Определение модели

Модель: $\text{□□}/\text{□□} = \text{□□□}$

```

function exponential_growth!(du, u, p, t)
     $\square = p.\square$  # **ИЗМЕНЕНИЕ:** Параметры теперь передаются как именованный кортеж
    du[1] =  $\square * u[1]$ 
end

```

6.3 Определение параметров в Dict

ОСНОВНОЕ ИЗМЕНЕНИЕ: Все параметры собраны в Dict для систематизации
Базовый набор параметров (один эксперимент)

```

base_params = Dict(
    :u0 => [1.0], # начальная популяция
    : $\square$  => 0.3, # скорость роста
    :tspan => (0.0, 10.0), # интервал времени
    :solver => Tsit5(), # метод решения
    :saveat => 0.1, # шаг сохранения результатов
    :experiment_name => "base_experiment"
)
println("Базовые параметры эксперимента:")
for (key, value) in base_params
    println("$key = $value")
end

```

6.4 Функция-обертка для запуска одного эксперимента

ИСПРАВЛЕНИЕ: Возвращаем Dict со строковыми ключами

```

function run_single_experiment(params::Dict)
    @unpack u0, □, tspan, solver, saveat = params
    prob = ODEProblem(exponential_growth!, u0, tspan, (□=□,)) # Создаем и решаем задачу
    sol = solve(prob, solver; saveat=saveat)
    final_population = last(sol.u)[1] # Анализ результатов
    doubling_time = log(2) / □
    return Dict(
        "solution" => sol,
        "time_points" => sol.t,
        "population_values" => first.(sol.u),
        "final_population" => final_population,
        "doubling_time" => doubling_time,
        "parameters" => params # Сохраняем исходные параметры
    ) # Используем строки как ключи для совместимости с DrWatson
end

```

6.5 Запуск базового эксперимента

ИЗМЕНЕНИЕ: Используем produce_or_load для автоматического кэширования

```

data, path = produce_or_load(
    datadir(script_name, "single"), # Папка для сохранения
    base_params, # Параметры эксперимента
    run_single_experiment, # Функция для выполнения
    prefix = "exp_growth", # Префикс имени файла
    tag = false, # Не добавлять git-тег
    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)
```

6.6 Визуализация базового эксперимента

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

Сохраним график в папку plots

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

6.7 Параметрическое сканирование

НОВАЯ СЕКЦИЯ: Исследование влияния параметра α Сетка параметров для сканирования

```

param_grid = Dict(
    :u0 => [[1.0]], # фиксируем начальное условие
    :l => [0.1, 0.3, 0.5, 0.8, 1.0], # исследуемые значения скорости роста
    :tspan => [(0.0, 10.0)], # фиксируем интервал времени
    :solver => [Tsit5()], # фиксируем метод решения
    :saveat => [0.1], # фиксируем шаг сохранения
    :experiment_name => ["parametric_scan"]
)

```

Генерация всех комбинаций параметров

```

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

```

6.8 Запуск всех экспериментов и собор результатов

НОВАЯ СЕКЦИЯ: Автоматический запуск и сохранение всех вариантов

```

all_results = []
all_dfs = []
for (i, params) in enumerate(all_params)
    println("Прогресс: $i/$(length(all_params)) | l = $(params[:l])")
    data, path = produce_or_load(
        datadir(script_name, "parametric_scan"), # Данные
        params, # Текущий набор параметров

```

```

run_single_experiment, # Функция для выполнения
prefix = "scan", # Префикс имени файла
tag = false,
verbose = false # Не выводить подробности для каждого запуска
) # Автоматическое сохранение/загрузка каждого эксперимента
result_summary = merge(
params,
Dict(
:final_population => data["final_population"],
:doubling_time => data["doubling_time"],
:filepath => path # Путь к сохраненным данным
)
) # Сохраняем сводные результаты (используем символы для параметров, но данные из data
push!(all_results, result_summary)
df = DataFrame(
t = data["time_points"],
u = data["population_values"],
□ = fill(params[:(□)], length(data["time_points"]))
) # Сохраняем полные данные для визуализации
push!(all_dfs, df)
end

```

6.9 Анализ и визуализация результатов сканирования

НОВАЯ СЕКЦИЯ: Сравнительный анализ всех экспериментов Сводная таблица результата

```

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

```

Сравнительный график всех траекторий

```

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"
    ) # Загружаем данные (они уже есть на диске)
    plot!(p2, data["time_points"], data["population_values"],
        label="t = $(params[:])",
        lw=2,
        alpha=0.8
    )
end
plot!(p2,
    xlabel="Время, t",
    ylabel="Популяция, u(t)",
    title="Параметрическое исследование: влияние t на рост",
    legend=:topleft,
    grid=true
)

```

Сохраним график в папку plots

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

График зависимости времени удвоения от α

```
p3 = plot(results_df[], results_df.doubling_time,
series_type=:scatter,
label="Численное решение",
xlabel="Скорость роста,  $\alpha$ ",
ylabel="Время удвоения,  $t_2$ ",
title="Зависимость времени удвоения от  $\alpha$ ",
marker_size=8,
marker_color=:red,
legend=:topright
)
```

Теоретическая кривая: $t_2 = \ln(2)/\alpha$

```
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
)
```

Сохраним график в папку plots

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

6.10 Бенчмаркинг с разными параметрами

ИЗМЕНЕНИЕ: Бенчмаркинг для разных значений α

```
println("\n" * "="^60)
println("Бенчмаркинг для разных значений α")
println("=".^60)
benchmark_results = []
for α_value in param_grid[:, :]
    bench_params = Dict(
        :u0 => [1.0],
        :tspan => (0.0, 10.0),
        :solver => Tsit5(),
        :saveat => 0.1
    ) # Подготавливаем параметры для бенчмарка
    function benchmark_run() # Функция для бенчмарка
        prob = ODEProblem(exponential_growth!,
            bench_params[:u0],
            bench_params[:tspan],
            (α=bench_params[:α],))
        return solve(prob, bench_params[:solver];
            saveat=bench_params[:saveat])
    end
    println("\nБенчмарк для α = $α_value:")
    b = @benchmark $benchmark_run() samples=100 evals=1 # Запуск бенчмарка
    push!(benchmark_results, (α=α_value, time=median(b).time/1e9)) # время в секундах
    println(" Среднее время: ", round(median(b).time/1e9; digits=4), " сек")
end
```

График зависимости времени вычисления от α

```
bench_df = DataFrame(benchmark_results)
p4 = plot(bench_df[], bench_df.time,
series_type=:scatter,
label="Время вычисления",
xlabel="Скорость роста, %",
ylabel="Время вычисления, сек",
title="Зависимость времени вычисления от %",
marker_size=8,
marker_color=:green,
legend=:topleft
)
```

Сохраним график в папку plots

```
savefig(plotsdir(script_name, "computation_time_vs_alpha.png"))
```

6.11 Сохранение всех результатов

НОВАЯ СЕКЦИЯ: Сохранение сводных данных для последующего анализа

```
@save datadir(script_name, "all_results.jld2") base_params param_grid all_params resu
@save datadir(script_name, "all_plots.jld2") p1 p2 p3 p4
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")
println(" • data/$(script_name)/all_plots.jld2 - объекты графиков")
println("\nДля анализа результатов используйте:")
println(" using JLD2, DataFrames")
println(" @load \"data/$(script_name)/all_results.jld2\"")
println(" println(results_df)")
```

Проведённый анализ позволил наглядно оценить, как коэффициент роста влияет на характер изменения величины, время её удвоения и вычислительные затраты.

7 Выводы

В процессе выполнения лабораторной работы была рассмотрена модель экспоненциального роста и её математическая основа. Проанализировано дифференциальное уравнение, описывающее динамику изменения величины, и изучено его аналитическое решение.

Построен базовый график, отражающий ускоряющийся характер роста во времени. Проведённое параметрическое исследование показало, что коэффициент

$$\alpha$$

оказывает существенное влияние на скорость увеличения системы.

Экспериментально подтверждена теоретическая зависимость времени удвоения от параметра роста: при увеличении

$$\alpha$$

оно уменьшается. Кроме того, был выполнен анализ вычислительных затрат, показавший слабую зависимость времени расчёта от величины параметра.

Полученные результаты согласуются с теоретическими представлениями об экспоненциальном росте и подтверждают возможность применения данной модели для описания процессов в биологии, экономике, физике и информационных технологиях.

Список литературы

1. A Multi-Language Computing Environment for Literate Programming and Reproducible Research / E. Schulte [et al.] // Journal of Statistical Software. — 2012. — Vol. 46, no. 3. — ISSN 1548-7660. — DOI: 10.18637/jss.v046.i03.
2. Knuth D. E. Literate Programming // The Computer Journal. — 1984. — Feb. — Vol. 27, no. 2. — P. 97–111. — ISSN 1460-2067. — DOI: 10.1093/comjnl/27.2.97.
3. The Story in the Notebook / M. B. Kery [et al.] // Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. — ACM, 04/2018. — P. 1–11. — DOI: 10.1145/3173574.3173748.