

# **Математическое моделирование**

**Лабораторная работа № 1**

Хамди Мохаммад

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Ключевые характеристики модели . . . . .	8
3.2	Области применения . . . . .	9
3.3	Ограничения модели . . . . .	9
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>5</b>	<b>Экспоненциальный рост</b>	<b>14</b>
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
<b>6</b>	<b>Параметрическое исследование экспоненциального роста</b>	<b>17</b>
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 сбор результатов . . . . .	21
6.9	Анализ и визуализация результатов сканирования . . . . .	22
6.10	Бенчмаркинг с разными параметрами . . . . .	25
6.11	Сохранение всех результатов . . . . .	26
<b>7</b>	<b>Выводы</b>	<b>28</b>
	<b>Список литературы</b>	<b>29</b>

## Список иллюстраций

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

## **Список таблиц**

# 1 Цель работы

Изучение модели экспоненциального роста, её математического описания, аналитического решения дифференциального уравнения и влияния параметра роста на поведение системы. Освоение методов параметрического исследования и анализа полученных результатов.

## 2 Задание

В качестве примера выполнения лабораторной работы рассмотреть модель экспоненциального роста. Изучить её математическое описание, исследовать решение дифференциального уравнения, а также провести параметрический анализ влияния коэффициента роста на динамику системы, время удвоения и вычислительные характеристики.

### 3 Теоретическое введение

Экспоненциальный рост — это процесс увеличения величины, при котором скорость её изменения в каждый момент времени пропорциональна текущему значению этой величины. Чем больше система, тем быстрее она растёт.

Примеры аналогий: начисление сложных процентов в банке или снежный ком, который, увеличиваясь в размерах, начинает расти всё быстрее.

Математически модель описывается дифференциальным уравнением:

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

где:

—

$u$

— текущее значение величины (численность популяции, капитал, число заражённых и т.д.);

—

$t$

— время;

—

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

— скорость роста;

—

$$\alpha$$

— константа роста (мальтузианский параметр).

Если

$$\alpha > 0$$

, наблюдается рост. Если

$$\alpha < 0$$

, происходит экспоненциальное затухание.

Смысл уравнения состоит в том, что скорость изменения величины напрямую зависит от её текущего значения.

Решение данного дифференциального уравнения имеет вид:

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

где

$$u_0$$

— начальное значение величины.

### 3.1 Ключевые характеристики модели

— Удвоение за постоянное время.

— Время удвоения определяется формулой:

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

— Время удвоения не зависит от текущего значения величины, а определяется только параметром роста.



## 3.2 Области применения

- Биология: рост популяции бактерий при избытке ресурсов.
- Финансы: сложные проценты по вкладам.
- Эпидемиология: начальная стадия распространения инфекции.
- Демография: рост численности населения в определённые периоды.
- Физика: цепные ядерные реакции.
- Информационные технологии: рост вычислительной мощности, сетевого трафика.

## 3.3 Ограничения модели

Экспоненциальный рост является идеализированной моделью. В реальности он не может продолжаться бесконечно из-за ограниченности ресурсов. Со временем рост замедляется и переходит в логистический (S-образный).

## 4 Выполнение лабораторной работы

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

Сначала был рассмотрен базовый эксперимент при фиксированном значении коэффициента роста

$$\alpha = 0.3$$

. Полученный график показывает зависимость величины от времени и демонстрирует характерный ускоряющийся рост.

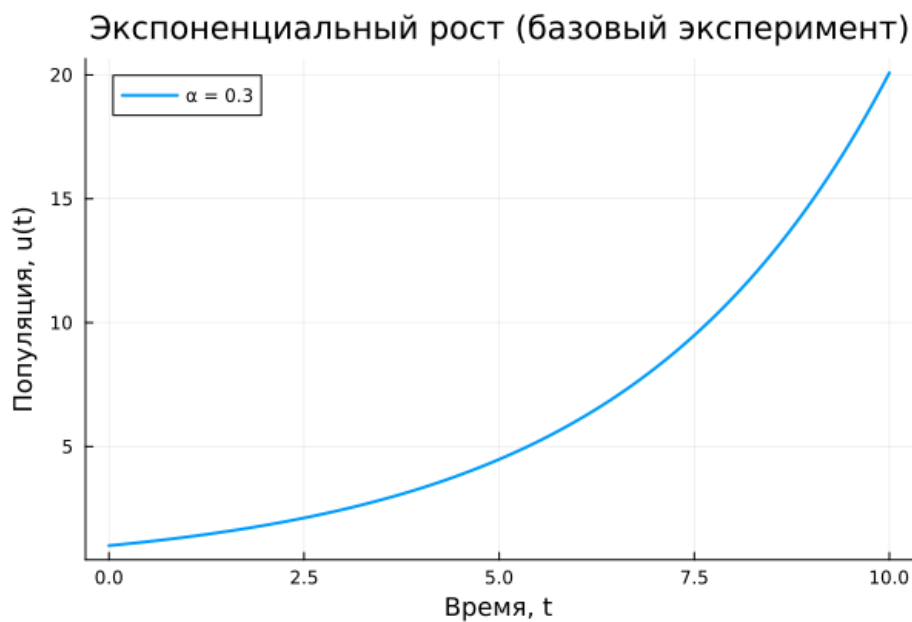


Рисунок 4.1: Экспоненциальный рост (базовый эксперимент)

На начальном этапе увеличение происходит медленно, затем скорость роста

возрастает, и к концу интервала функция резко увеличивается.

Далее было проведено параметрическое исследование, позволяющее оценить влияние параметра

$$\alpha$$

на динамику системы. Были построены графики для нескольких значений коэффициента роста.

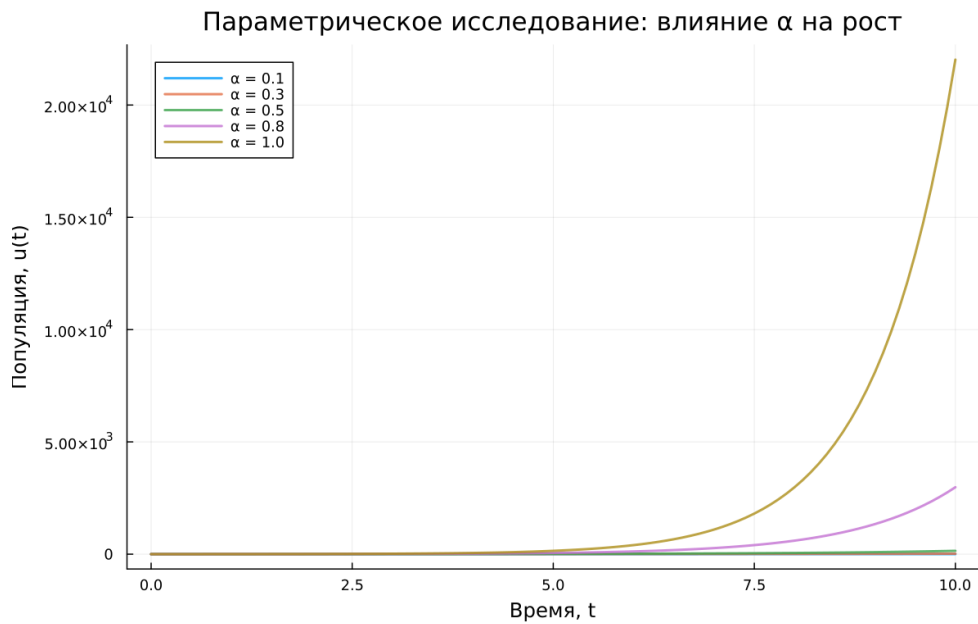


Рисунок 4.2: Параметрическое исследование: влияние  $\alpha$  на рост

Результаты показывают, что увеличение параметра

$$\alpha$$

существенно ускоряет рост. При малых значениях коэффициента функция возрастает плавно, при больших — резко уходит в область больших значений.

Отдельно была исследована зависимость времени удвоения от коэффициента роста. Теоретически время удвоения определяется выражением

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

. Численные результаты подтверждают данную зависимость.

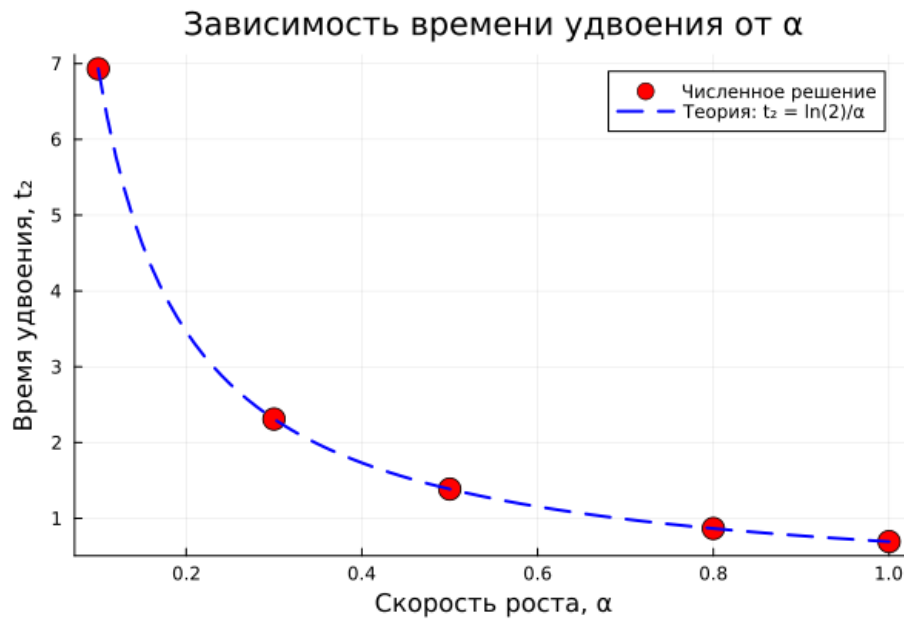


Рисунок 4.3: Зависимость времени удвоения от  $\alpha$

Из графика видно, что при увеличении

$\alpha$

время удвоения уменьшается, что соответствует теоретической модели.

Также была проанализирована зависимость времени вычисления от параметра роста.

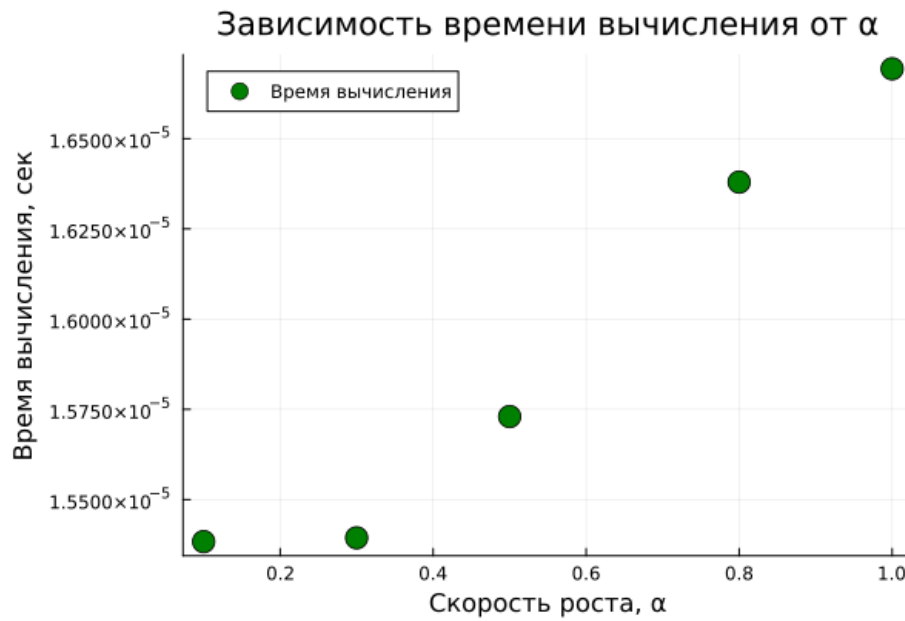


Рисунок 4.4: Зависимость времени вычисления от  $\alpha$

Наблюдается незначительное увеличение времени вычислений при росте параметра

$$\alpha$$

, что связано с увеличением значений функции и особенностями численной обработки данных.

Для моделирования процесса 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] = r * u[1]
end
```

## 5.3 Первый запуск с параметрами по умолчанию

Зададим начальные параметры:

```
u0 = [1.0] # начальная популяция
r = 0.3 # скорость роста
tspan = (0.0, 10.0) # временной интервал
prob = ODEProblem(exponential_growth!, u0, tspan, r)
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 (r = $r)", lw=2, legend=:topleft)
```

Сохраним график в папку plots

```
savefig(plotsdir(script_name, "exponential_growth_r=$r.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) /  $\lambda$   
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:  $\frac{dx}{dt} = \lambda x$

```

function exponential_growth!(du, u, p, t)
    r = p.r # **ИЗМЕНЕНИЕ:** Параметры теперь передаются как именованный кортеж
    du[1] = r * 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
    :r => 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, Δ, tspan, solver, saveat = params
prob = ODEProblem(exponential_growth!, u0, tspan, (Δ=Δ,)) # Создаем и решаем задачу
sol = solve(prob, solver; saveat=saveat)
final_population = last(sol.u)[1] # Анализ результатов
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 # 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="□ = $(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  $\alpha$  Сетка параметров для сканирования

```

param_grid = Dict(
:u0 => [[1.0]], # фиксируем начальное условие
:λ => [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 λ: ", param_grid[:λ])
println("="^60)

```

## 6.8 Запуск всех экспериментов и сбор 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)) | λ = $(params[:λ])")
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"],
  n = fill(params[:n], 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 результатов

```

results_df = DataFrame(all_results)
println("\nСводная таблица результатов:")
println(results_df[!, [:t, :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="□ = $(params[:r])",
        lw=2,
        alpha=0.8
    )
end
plot!(p2,
    xlabel="Время, t",
    ylabel="Популяция, u(t)",
    title="Параметрическое исследование: влияние r на рост",
    legend=:topleft,
    grid=true
)

```

Сохраним график в папку plots

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

График зависимости времени удвоения от  $\alpha$

```
p3 = plot(results_df.α, results_df.doubling_time,
          seriestype=:scatter,
          label="Численное решение",
          xlabel="Скорость роста, α",
          ylabel="Время удвоения, t₂",
          title="Зависимость времени удвоения от α",
          markersize=8,
          markercolor=:red,
          legend=:topright
        )
```

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

```
α_range = 0.1:0.01:1.0
plot!(p3, α_range, log(2) ./ α_range,
      label="Теория: t₂ = ln(2)/α",
      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  $\alpha$

```
println("\n" * "="^60)
println("Бенчмаркинг для разных значений  $\alpha$ ")
println("="^60)

benchmark_results = []
for  $\alpha$ _value in param_grid[: $\alpha$ ]
    bench_params = Dict{
        :u0 => [1.0],
        : $\alpha$  =>  $\alpha$ _value,
        :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],
            ( $\alpha$ =bench_params[: $\alpha$ ],))
        return solve(prob, bench_params[:solver];
            saveat=bench_params[:saveat])
    end
    println("\nБенчмарк для  $\alpha$  =  $\alpha$ _value:")
    b = @benchmark $benchmark_run() samples=100 evals=1 # Запуск бенчмарка
    push!(benchmark_results, ( $\alpha$ = $\alpha$ _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  $\alpha$

```
bench_df = DataFrame(benchmark_results)
p4 = plot(bench_df[1], bench_df.time,
          seriestype=:scatter,
          label="Время вычисления",
          xlabel="Скорость роста,  $\alpha$ ",
          ylabel="Время вычисления, сек",
          title="Зависимость времени вычисления от  $\alpha$ ",
          markersize=8,
          markercolor=: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 results
@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)")
```

Проведённое исследование позволило наглядно проследить влияние коэффициента роста на характер изменения величины, время удвоения и вычислительные характеристики.

## 7 Выводы

В ходе лабораторной работы была изучена модель экспоненциального роста и её математическое описание. Рассмотрено дифференциальное уравнение, определяющее динамику изменения величины, и его аналитическое решение.

Построен базовый график роста, демонстрирующий характерное ускоряющееся увеличение величины во времени. Проведено параметрическое исследование, показавшее, что коэффициент

$$\alpha$$

существенно влияет на скорость роста системы.

Подтверждена теоретическая зависимость времени удвоения от параметра роста: при увеличении

$$\alpha$$

время удвоения уменьшается. Также выполнен анализ вычислительных затрат, показавший слабую зависимость времени расчёта от значения параметра.

Полученные результаты согласуются с теоретическими представлениями об экспоненциальном росте и демонстрируют применимость 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.