

# **Математическое моделирование**

## **Лабораторная работа № 1**

Заур Мустафаев

# Содержание

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Теоретическое введение</b>	<b>7</b>
3.1 Ключевые характеристики модели . . . . .	8
3.2 Области применения . . . . .	9
3.3 Ограничения модели . . . . .	9
<b>4 Выполнение лабораторной работы</b>	<b>10</b>
<b>5 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 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 собор 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 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**

Рассмотреть модель экспоненциального роста, изучить её формализацию в виде дифференциального уравнения, получить аналитическое решение и проанализировать влияние коэффициента роста на динамику системы. Освоить подходы к параметрическому исследованию и интерпретации полученных результатов.

## **2 Задание**

В рамках лабораторной работы необходимо на примере модели экспоненциального роста изучить её математическое представление, исследовать аналитическое решение соответствующего дифференциального уравнения, а также провести анализ влияния параметра роста на характер изменения системы, время удвоения и вычислительные особенности.

### 3 Теоретическое введение

Экспоненциальный рост представляет собой процесс, при котором скорость изменения величины в каждый момент времени определяется её текущим значением. Это означает, что по мере увеличения системы темп её роста также возрастает.

Подобный механизм можно наблюдать, например, при начислении сложных процентов или в случае увеличения снежного кома, который становится всё больше и начинает нарастать быстрее.

Математическая модель описывается дифференциальным уравнением:

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

где:

—

$u$

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

—

$t$

— время;

—

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

- скорость изменения величины;
- 
- $\alpha$
- коэффициент роста (мальтузианский параметр).

При

$$\alpha > 0$$

наблюдается рост, при

$$\alpha < 0$$

- экспоненциальное уменьшение.

Смысл уравнения заключается в том, что скорость изменения напрямую пропорциональна текущему значению величины.

Аналитическое решение имеет вид:

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

где

$$u_0$$

- начальное значение величины.

### **3.1 Ключевые характеристики модели**

- Увеличение в два раза происходит через равные промежутки времени.
- Время удвоения определяется формулой:

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

- Значение времени удвоения зависит только от параметра роста и не определяется текущим размером величины.

## **3.2 Области применения**

- Биология: увеличение численности микроорганизмов при наличии достаточных ресурсов.
- Финансы: накопление капитала при начислении сложных процентов.
- Эпидемиология: ранние стадии распространения инфекционных заболеваний.
- Демография: периоды интенсивного роста населения.
- Физика: развитие цепных ядерных реакций.
- Информационные технологии: рост производительности вычислительных систем и объёмов сетевого трафика.

## **3.3 Ограничения модели**

Экспоненциальная модель носит идеализированный характер. В реальных условиях бесконечный рост невозможен из-за ограниченности ресурсов. Со временем темп увеличения снижается, и поведение системы начинает описываться логистической (S-образной) зависимостью.

## 4 Выполнение лабораторной работы

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

Сначала рассмотрен базовый эксперимент при фиксированном значении коэффициента роста

$$\alpha = 0.3$$

. Построенный график отражает изменение величины во времени и демонстрирует типичный ускоряющийся характер роста.

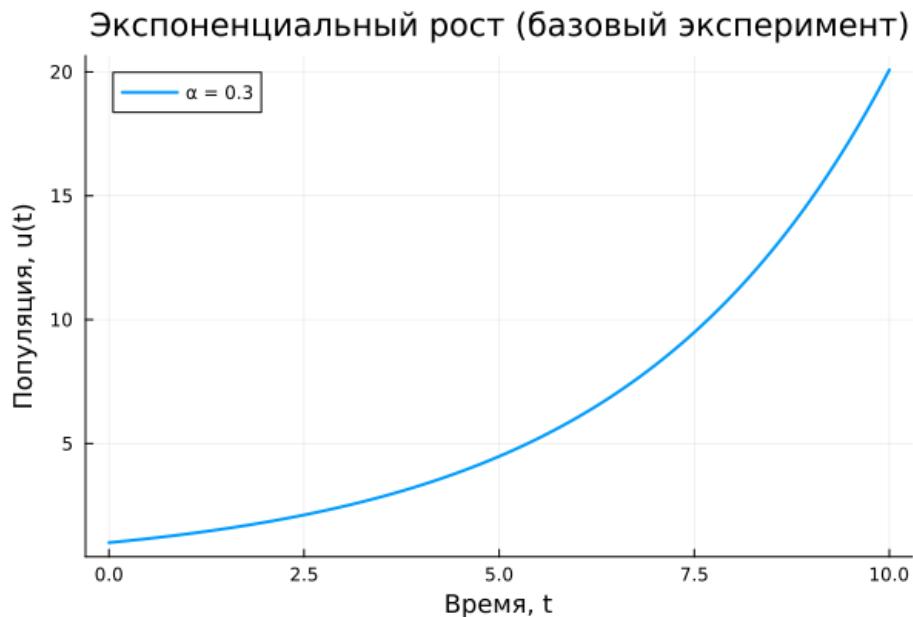


Рисунок 4.1: Экспоненциальный рост (базовый эксперимент)

На начальном участке рост происходит сравнительно медленно, затем темп увеличивается, и ближе к концу рассматриваемого интервала наблюдается резкое возрастание функции.

Далее выполнено параметрическое исследование, направленное на оценку влияния коэффициента

$$\alpha$$

на динамику системы. Были построены графики для ряда значений данного параметра.

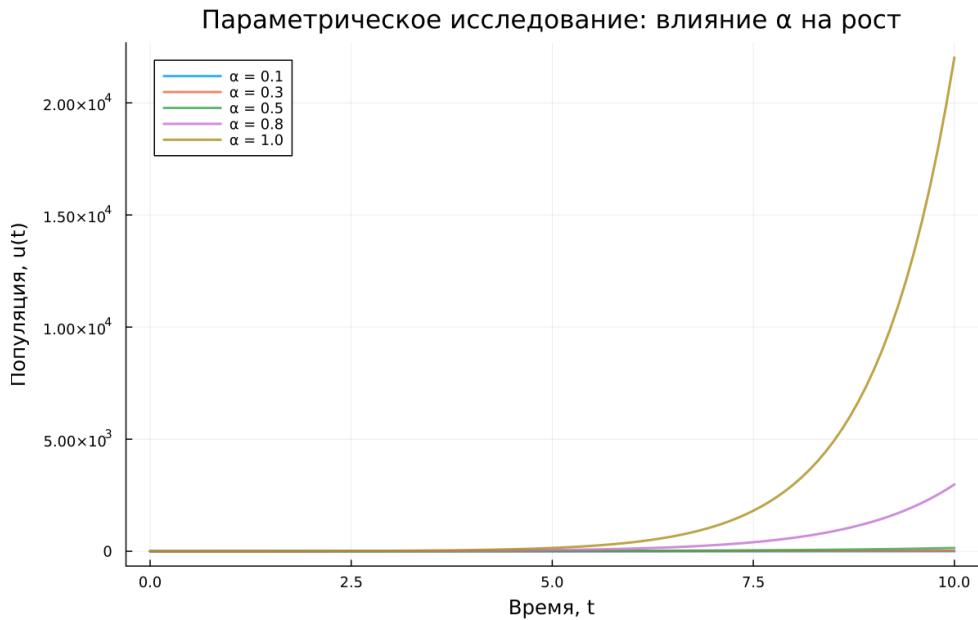


Рисунок 4.2: Параметрическое исследование: влияние  $\alpha$  на рост

Полученные результаты показывают, что увеличение параметра

$$\alpha$$

приводит к заметному ускорению роста. При небольших значениях коэффициента изменение величины происходит плавно, тогда как при больших — функция быстро принимает значительные значения.

Отдельно рассмотрена зависимость времени удвоения от коэффициента

роста. Теоретически оно определяется выражением

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

. Численные расчёты подтверждают справедливость данной зависимости.

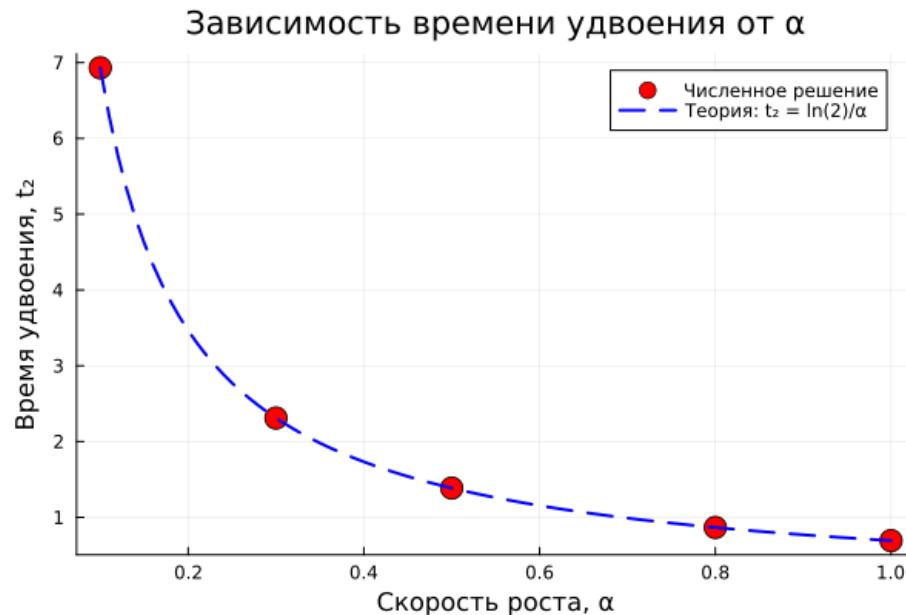


Рисунок 4.3: Зависимость времени удвоения от  $\alpha$

Из графика следует, что при увеличении

$\alpha$

время, необходимое для удвоения величины, уменьшается, что соответствует теоретическим ожиданиям.

Также была исследована связь между временем вычислений и значением параметра роста.

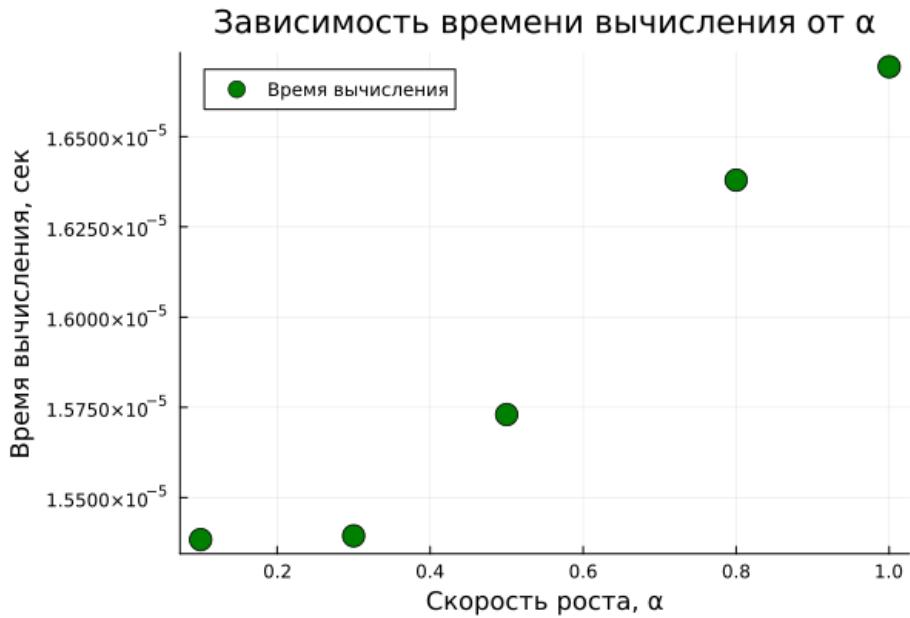


Рисунок 4.4: Зависимость времени вычисления от  $\alpha$

Наблюдается небольшое увеличение времени обработки при росте

$$\alpha$$

, что связано с увеличением численных значений функции и особенностями вычислительных процедур.

Для моделирования и построения графиков применялись 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  $\alpha$  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 результатов

```

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  $\alpha$

```
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  $\alpha$

```
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  $\alpha$

```
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 проследить, как коэффициент роста влияет на поведение системы, время удвоения и вычислительные характеристики.

## 7 Выводы

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

Построен базовый график, демонстрирующий ускоряющийся характер роста во времени. Выполнено параметрическое исследование, показавшее, что коэффициент

$$\alpha$$

существенно влияет на скорость изменения системы.

Подтверждена теоретическая зависимость времени удвоения от параметра роста: при увеличении

$$\alpha$$

время удвоения уменьшается. Также проведён анализ вычислительных затрат, который показал слабую зависимость времени расчёта от значения коэффициента.

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