Методы математического моделирования в кибербезопасности

Лабораторная работа № 1

Андрей Грыцькив

Содержание

1 Цель работы **5**

2 Задание **6**

3 Теоретическое введение **7**
3.1 Ключевые характеристики модели . . . 9
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, её формального математического описания и аналитического решения соответствующего дифференциального уравнения. Анализ влияния параметра роста на поведение системы, а также получение навыков проведения параметрических исследований и интерпретации результатов.

2 Задание

В рамках лабораторной работы требуется рассмотреть модель экспоненциального роста в качестве демонстрационного примера. Необходимо изучить её математическую формулировку, проанализировать решение дифференциального уравнения и выполнить параметрическое исследование, позволяющее оценить влияние коэффициента роста на динамику системы, время удвоения и особенности вычислительного процесса.

3 Теоретическое введение

Экспоненциальный рост представляет собой процесс увеличения некоторой величины, при котором скорость её изменения в каждый момент времени пропорциональна текущему значению. Это означает, что по мере увеличения самой величины возрастает и скорость её роста.

В качестве наглядных примеров можно привести начисление сложных процентов в финансовой сфере или эффект снежного кома, который по мере увеличения начинает нарастать всё быстрее.

Математически данный процесс описывается дифференциальным уравнением:

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

где:

—

$$u$$

— текущее значение исследуемой величины (например, численность популяции, размер капитала или количество заражённых);

—

$$t$$

— время;

—

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

— скорость изменения величины;

—

$$\alpha$$

— постоянный коэффициент роста (мальтузианский параметр).

При

$$\alpha > 0$$

система демонстрирует рост, а при

$$\alpha < 0$$

наблюдается экспоненциальное убывание.

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

Аналитическое решение данного дифференциального уравнения имеет следующий вид:

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

где

$$u_0$$

— начальное значение величины в момент времени

$$t = 0$$

.

3.1 Ключевые характеристики модели

— Увеличение величины происходит с удвоением через равные промежутки времени.

— Время удвоения определяется формулой:

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

— Значение времени удвоения не зависит от текущего уровня величины и определяется исключительно коэффициентом роста.

3.2 Области применения

— Биология: увеличение численности микроорганизмов при достаточном количестве ресурсов.

— Финансы: рост вложений при начислении сложных процентов.

— Эпидемиология: распространение инфекции на ранних этапах.

— Демография: периоды активного роста населения.

— Физика: развитие цепных ядерных реакций.

— Информационные технологии: увеличение вычислительных мощностей и сетевого трафика.

3.3 Ограничения модели

Экспоненциальная модель является идеализированной и не учитывает ограниченность ресурсов. В реальных условиях бесконечный рост невозможен: со временем система сталкивается с ограничениями, что приводит к замедлению темпов увеличения и переходу к логистическому (S-образному) характеру развития.

4 Выполнение лабораторной работы

В процессе выполнения лабораторной работы было осуществлено моделирование экспоненциального роста на основе аналитического решения соответствующего дифференциального уравнения.

На первом этапе рассмотрен базовый эксперимент при фиксированном значении коэффициента роста

$$\alpha = 0.3$$

. Полученный график иллюстрирует зависимость исследуемой величины от времени и демонстрирует характерное ускорение роста.

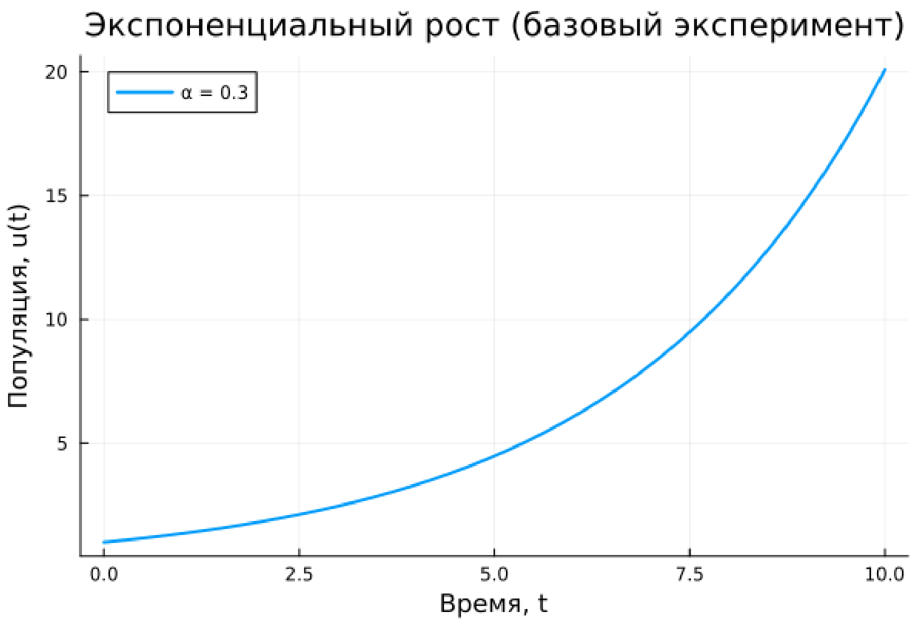

Рисунок 4.1: Экспоненциальный рост (базовый эксперимент)

В начальный период увеличение происходит сравнительно медленно, однако затем скорость возрастает, и к концу рассматриваемого интервала наблюдается резкий рост функции.

Далее было выполнено параметрическое исследование, направленное на изучение влияния коэффициента

$$\alpha$$

на динамику системы. Для этого были построены графики при различных значениях параметра роста.

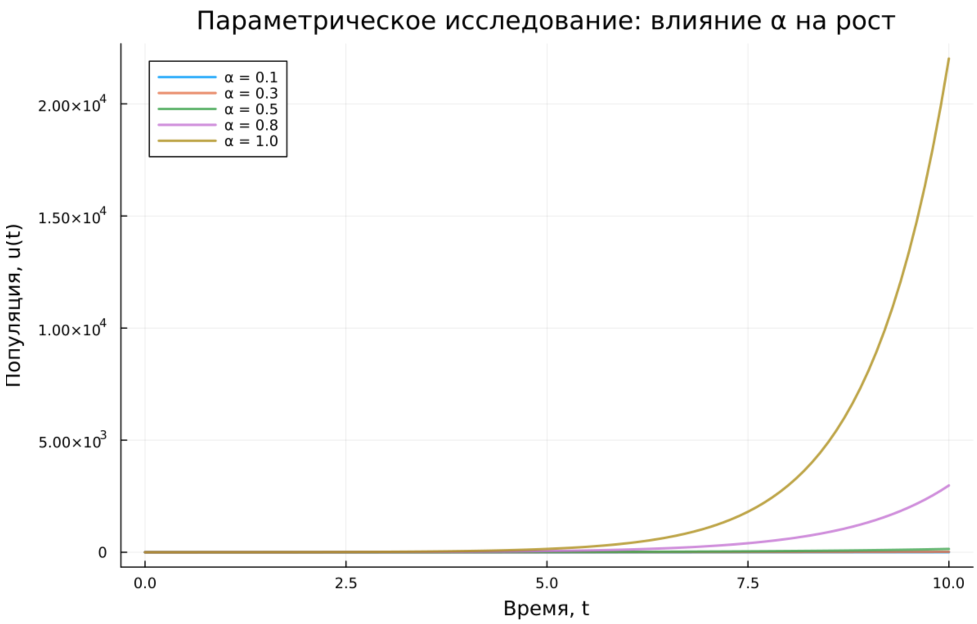

Рисунок 4.2: Параметрическое исследование: влияние α на рост

Полученные результаты показывают, что увеличение значения

$$\alpha$$

приводит к существенному ускорению роста. При малых значениях параметра функция возрастает постепенно, тогда как при больших — быстро достигает значительных величин.

Отдельное внимание было уделено исследованию зависимости времени удвоения от коэффициента роста. В теории это время определяется выражением

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

. Численные расчёты подтвердили справедливость данной зависимости.

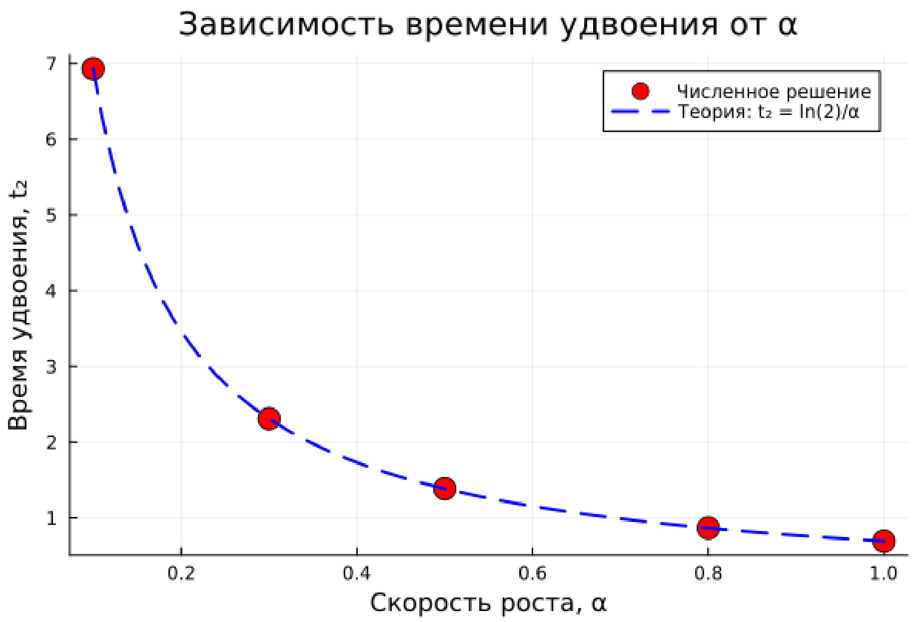

Рисунок 4.3: Зависимость времени удвоения от α

Из графика следует, что с увеличением

$$\alpha$$

время удвоения уменьшается, что полностью соответствует теоретическим ожиданиям.

Дополнительно была изучена зависимость времени вычисления от значения параметра роста.

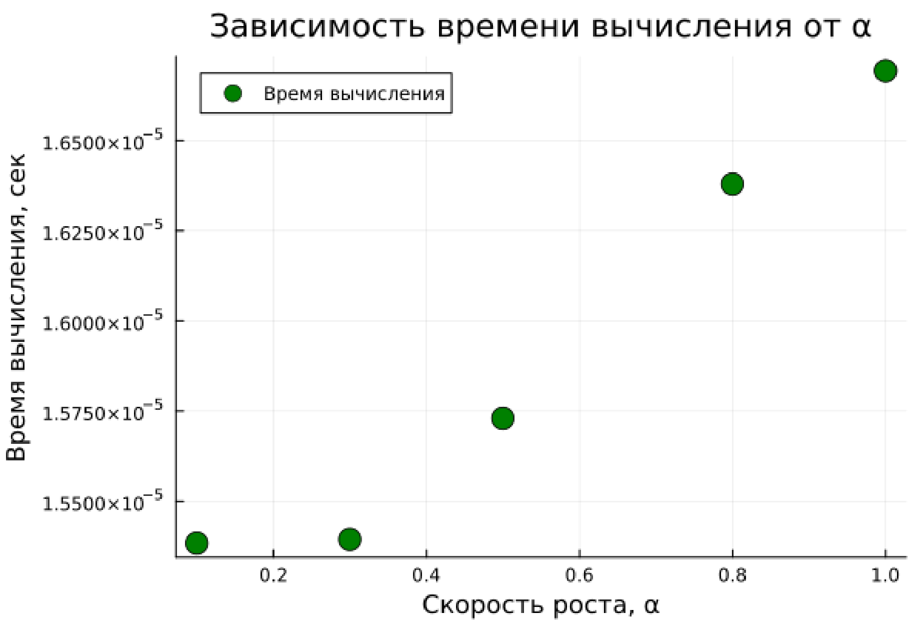

Рисунок 4.4: Зависимость времени вычисления от α

Отмечается незначительное увеличение времени расчётов при возрастании

$$\alpha$$

, что связано с ростом значений функции и особенностями обработки численных данных.

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

5 Экспоненциальный рост

Цель: Исследовать решение уравнения □□/□□ = □□.

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:

```
#□□/□□ = □□, □(0) = □0
function exponential_growth!(du, u, p, t)
□ = p
```

```
du[1] = □ * u[1]
end
```

5.3 Первый запуск с параметрами по умолчанию

Зададим начальные параметры:

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

Сохраним график в папку plots

```
savefig(plotsdir(script_name, "exponential_growth_□=$□.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: □□/□□ = □ □ □

```
function exponential_growth!(du, u, p, t)
□ = p.□ # **ИЗМЕНЕНИЕ:** Параметры теперь передаются как именованный кортеж
du[1] = □ * 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
:□ => 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=”□ = $(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
:□ => [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 □: ”, param_grid[:□])
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)) | □ = $(params[:□])”)
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 dat
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="□ = $(params[:□])",
lw=2,
alpha=0.8
)
end
plot!(p2,
xlabel="Время, t",
ylabel="Популяция, u(t)",
title="Параметрическое исследование: влияние □ 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.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 α

```
println(”\n” * ”=”^60)
println(”Бенчмаркинг для разных значений □”)
println(”=”^60)
benchmark_results = []
for □_value in param_grid[:□]
bench_params = Dict(
:u0 => [1.0],
:□ => □_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],
(□=bench_params[:□],))
return solve(prob, bench_params[:solver];
saveat=bench_params[:saveat])
end
println(”\nБенчмарк для □ = $□_value:”)
b = @benchmark $benchmark_run() samples=100 evals=1 # Запуск бенчмарка
push!(benchmark_results, (□=□_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,
seriestype=:scatter,
label="Время вычисления",
xlabel="Скорость роста, □",
ylabel="Время вычисления, сек",
title="Зависимость времени вычисления от □",
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 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 влияние коэффициента роста на динамику изменения величины, время её удвоения и вычислительные характеристики модели.

7 Выводы

В рамках лабораторной работы была подробно рассмотрена модель экспоненциального роста и её математическая постановка. Проанализировано дифференциальное уравнение, описывающее изменение величины во времени, а также получено его аналитическое решение.

Построен базовый график, демонстрирующий ускоряющийся характер увеличения величины. Выполнено параметрическое исследование, показавшее, что коэффициент

$$\alpha$$

оказывает значительное влияние на скорость развития системы.

Экспериментально подтверждена теоретическая зависимость времени удвоения от параметра роста: с увеличением

$$\alpha$$

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 подтверждают возможность применения данной модели для описания процессов в биологии, экономике, физике и сфере информационных технологий.

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.