

Математическое моделирование

Лабораторная работа № 1

Чилеше Лупупа

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 сбор 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

Рассмотреть модель экспоненциального роста, раскрыть её формальное описание, получить аналитическое решение соответствующего дифференциального уравнения и проанализировать влияние коэффициента роста на поведение системы. Освоить приёмы параметрического исследования и интерпретации результатов моделирования.

2 Задание

В рамках лабораторной работы необходимо исследовать модель экспоненциального роста: проанализировать её математическую формулировку, изучить решение дифференциального уравнения, а также выполнить параметрический анализ влияния коэффициента роста на динамику процесса, время удвоения и вычислительные характеристики.

3 Теоретическое введение

Экспоненциальный рост представляет собой процесс, при котором скорость изменения величины пропорциональна её текущему значению. Иными словами, чем больше значение функции в данный момент времени, тем интенсивнее происходит её дальнейшее увеличение.

В качестве наглядных примеров можно привести начисление сложных процентов или увеличение снежного кома: по мере роста объекта его прирост становится всё более существенным.

Математическая модель описывается дифференциальным уравнением:

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

где:

—

u

— текущее значение исследуемой величины (численность популяции, объём капитала, количество заражённых и т.п.);

—

t

— время;

—

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

— мгновенная скорость изменения;

—

$$\alpha$$

— постоянный коэффициент роста (мальтузианский параметр).

При

$$\alpha > 0$$

система демонстрирует рост, при

$$\alpha < 0$$

наблюдается экспоненциальное убывание.

Смысл модели заключается в прямой зависимости темпа изменения величины от её текущего состояния.

Решение данного уравнения имеет 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 значения в два раза происходит за фиксированный промежуток времени.

— Время удвоения определяется формулой:

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

— Длительность периода удвоения зависит исключительно от коэффициента роста и не определяется текущим значением функции.

3.2 Области применения

— Биология: размножение микроорганизмов при отсутствии ограничений по ресурсам.

— Финансы: начисление сложных процентов по банковским вкладам.

— Эпидемиология: начальная фаза распространения инфекционных заболеваний.

— Демография: рост населения в определённые исторические периоды.

— Физика: развитие цепных ядерных реакций.

— Информационные технологии: увеличение вычислительных мощностей и сетевого трафика.

3.3 Ограничения модели

Экспоненциальная модель носит идеализированный характер. В реальных условиях ресурсы ограничены, поэтому бесконечный рост невозможен. По мере исчерпания ресурсов динамика изменяется и система переходит к логистическому (S-образному) типу роста.

4 Выполнение лабораторной работы

В процессе выполнения работы была реализована модель экспоненциального роста на основе аналитического решения дифференциального уравнения.

На первом этапе проведён базовый эксперимент при фиксированном значении коэффициента

$$\alpha = 0.3$$

. Построенный график отражает зависимость величины от времени и иллюстрирует характерную ускоряющуюся динамику.

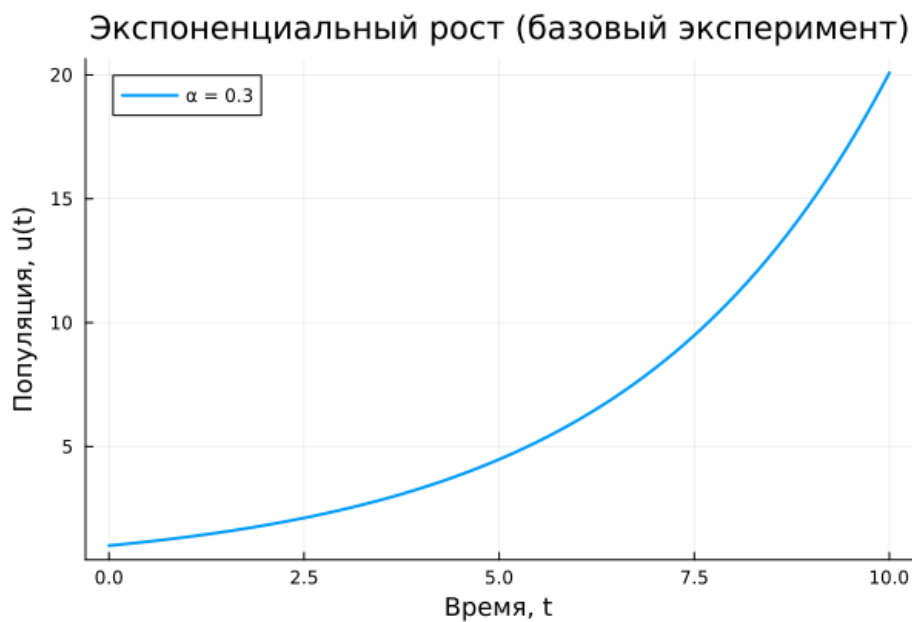


Рисунок 4.1: Экспоненциальный рост (базовый эксперимент)

В начале интервала функция изменяется сравнительно медленно, однако по

мере увеличения времени темп роста возрастает, и к концу рассматриваемого промежутка наблюдается резкое увеличение значения.

Далее выполнено параметрическое исследование, направленное на анализ влияния коэффициента

$$\alpha$$

на поведение системы. Были построены графики для нескольких значений параметра.

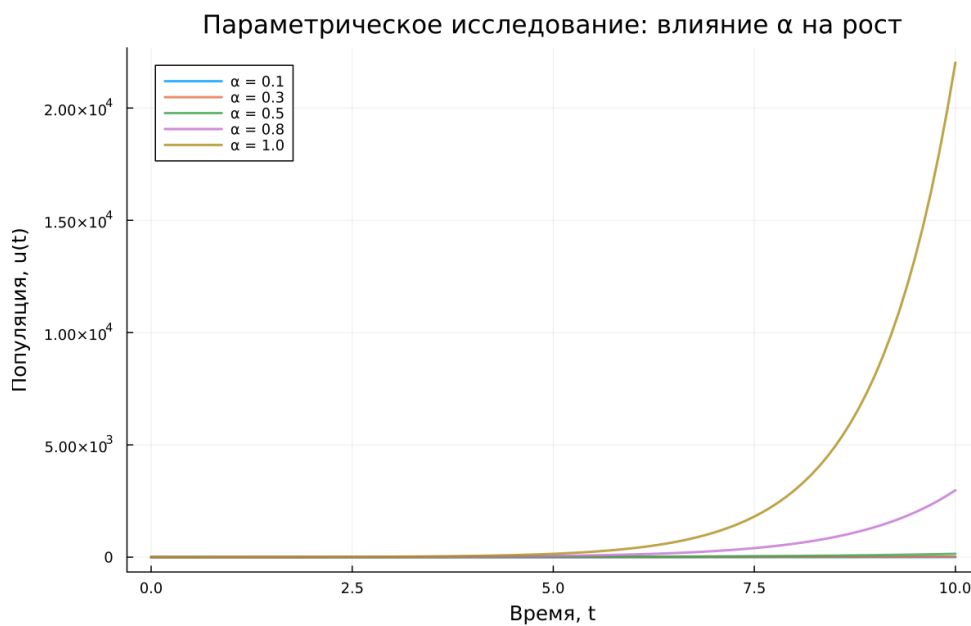


Рисунок 4.2: Параметрическое исследование: влияние α на рост

Полученные результаты демонстрируют, что увеличение

$$\alpha$$

приводит к существенному ускорению роста. При малых значениях параметра зависимость остаётся более полой, тогда как при больших — функция быстро достигает высоких значений.

Отдельно рассмотрена зависимость времени удвоения от коэффициента

роста. Теоретическое выражение

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

было сопоставлено с численными результатами.

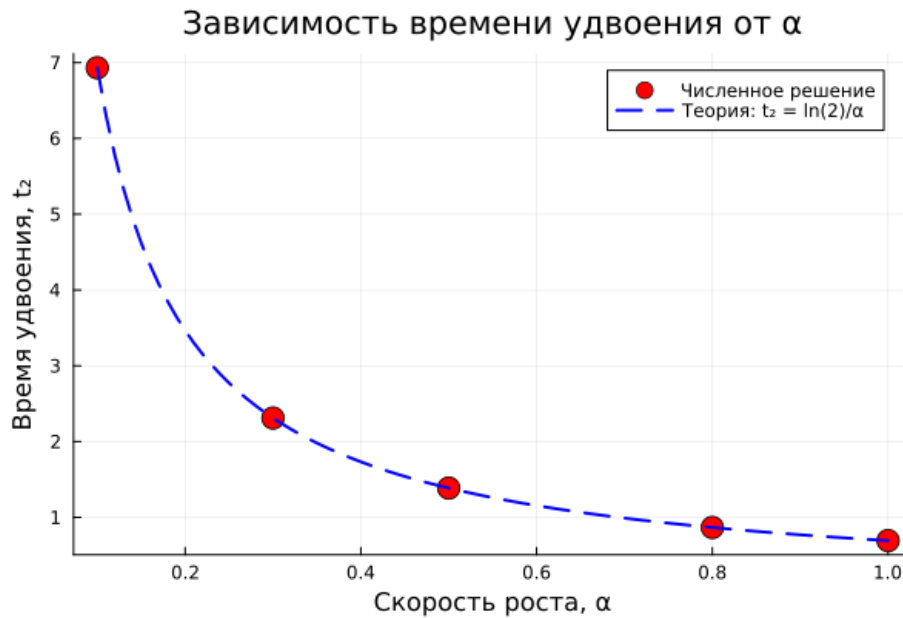


Рисунок 4.3: Зависимость времени удвоения от α

График подтверждает обратную зависимость: при увеличении

α

время удвоения сокращается, что полностью согласуется с аналитической формулой.

Дополнительно исследована зависимость времени вычисления от значения коэффициента роста.

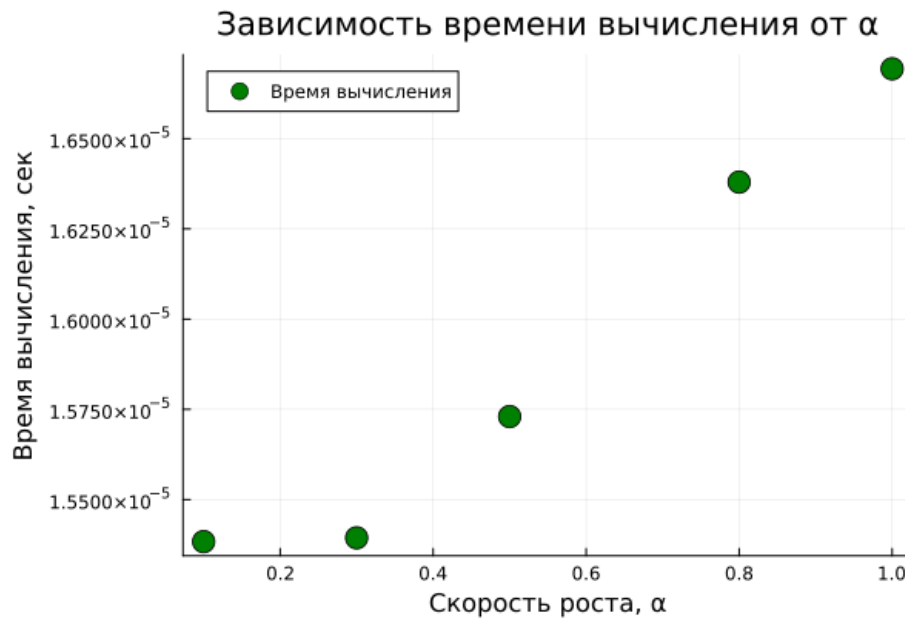


Рисунок 4.4: Зависимость времени вычисления от α

Зафиксировано незначительное увеличение вычислительных затрат при росте

$$\alpha$$

, что обусловлено увеличением диапазона значений функции и особенностями численной обработки данных.

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

5 Экспоненциальный рост

Цель: Исследовать решение уравнения $\dot{x}/x = p$.

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:

```
# $\dot{x}/x = p$ ,  $x(0) = x_0$ 
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 α 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"],
  I = fill(params[:I], 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[:λ])",
        lw=2,
        alpha=0.8
    )
end
plot!(p2,
    xlabel="Время, t",
    ylabel="Популяция, u(t)",
    title="Параметрическое исследование: влияние λ 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  $\alpha$ ")
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 α

```
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)")
```

Проведённый анализ позволил наглядно оценить влияние коэффициента роста на динамику изменения величины, время удвоения и вычислительные параметры модели.

7 Выводы

В рамках лабораторной работы детально рассмотрена модель экспоненциального роста и её математическая основа. Проанализировано дифференциальное уравнение, описывающее динамику процесса, и получено его аналитическое решение.

Построен базовый график, иллюстрирующий ускоряющийся характер изменения величины во времени. Параметрическое исследование показало, что коэффициент

$$\alpha$$

существенно определяет скорость развития системы.

Экспериментально подтверждена теоретическая формула для времени удвоения: с увеличением

$$\alpha$$

данный показатель уменьшается. Анализ вычислительных затрат выявил слабую зависимость времени расчёта от значения параметра.

Результаты моделирования соответствуют теоретическим положениям об экспоненциальной динамике и подтверждают применимость 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.