

Лабораторная работа №1

Андрей Геннадьевич Софич

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Выводы	15

Список иллюстраций

3.1	Настройка git	7
3.2	Подключение устройства к аккаунтам	7
3.3	Создание gpg ключа	8
3.4	Создание рабочего пространства	8
3.5	Создание ключа для клонирования репозитория	9
3.6	Завершение настройки git	9
3.7	Создание среды для работы	10
3.8	Настройка julia	10
3.9	Исполняемый файл №1	11
3.10	Создание производных форматов	12
3.11	Исполняемый файл №2	12
3.12	Проверка создания отчета	13
3.13	Правильность компиляции отчета	14

Список таблиц

1 Цель работы

Подготовить среду для работы с моделированием, подключить необходимое окружение, провести пару пробных тестов.

2 Задание

Подготовить среду для работы с моделированием, подключить необходимое окружение, провести пару пробных тестов.

3 Выполнение лабораторной работы

Базово настраиваем git, чтобы подключить наши репозитории к устройству (рис. 3.1).

```
PS C:\Users\Andrew> git config --global user.name "AndreySofich"
PS C:\Users\Andrew> git config --global user.email "andrejsofic2@yandex.ru"
PS C:\Users\Andrew> git config --global core.quotePath false
PS C:\Users\Andrew> git config --global init.defaultBranch master
PS C:\Users\Andrew> git config --global core.safecrlf warn
PS C:\Users\Andrew>
```

Рисунок 3.1: Настройка git

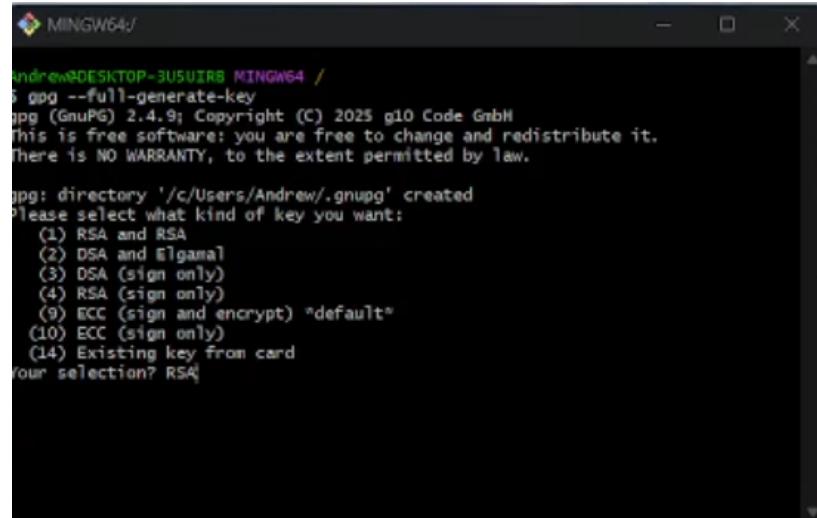
Логинимся к нашим аккантам на github и gitverse (рис. 3.2).

```
PS C:\Users\Andrew> gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? C:\Users\Andrew\.ssh\id_ed25519.pub
? Title for your SSH key: (GitHub CLI)

? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: C:\Users\Andrew\.ssh\id_ed25519.pub
✓ Logged in as AndreySofich
```

Рисунок 3.2: Подключение устройства к аккантам

Создаем gpg ключ, чтобы связать компьютер с приложениями, после чего в gihub и gitverse добавляем эти ключи (рис. 3.3).



```
Andrew@DESKTOP-BUSUIRB MINGW64 /  
$ gpg --full-generate-key  
gpg (GnuPG) 2.4.9; Copyright (C) 2025 g10 Code GmbH  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
  
gpg: directory '/c/Users/Andrew/.gnupg' created  
Please select what kind of key you want:  
(1) RSA and RSA  
(2) DSA and Elgamal  
(3) DSA (sign only)  
(4) RSA (sign only)  
(9) ECC (sign and encrypt) *default*  
(10) ECC (sign only)  
(14) Existing key from card  
Your selection? RSA
```

Рисунок 3.3: Создание gpg ключа

Далеем создаем репозитории на основе шаблона в gitverse и github (рис. 3.4).

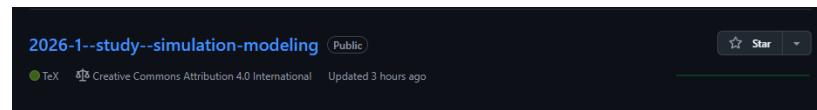


Рисунок 3.4: Создание рабочего пространства

Я дополнительно создал и подключил ssh ключ для gitverse, так как для github он создался автоматически (рис. 3.5).

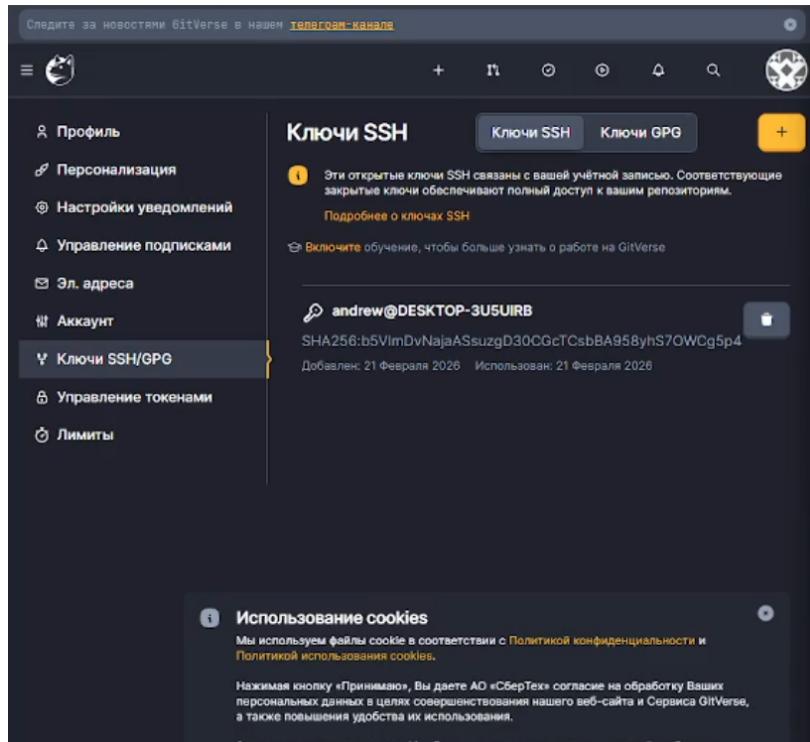


Рисунок 3.5: Создание ключа для клонирования репозитория

Далее клонируем репозиторий на устройство, и создаем курс, после чего отправляем все изменения на платформы (рис. 3.6).

```

create mode 100644 labs/lab08/report/_extensions/yamadharma/minted-quarto/minted-quarto.lua
create mode 100644 labs/lab08/report/_quarto.yaml
create mode 100644 labs/lab08/report/_resources/csl/post-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab08/report/_resources/tex/preamble.tex
create mode 100644 labs/lab08/report/bib/cite.bib
create mode 100644 labs/lab08/report/image/solvay.jpg
create mode 100644 labs/lab08/report/simulation-modeling-lab08-report.qmd
create mode 100644 prepare
```
WARNING: connection is not using a post-quantum key exchange algorithm.
This session may be vulnerable to "store now, decrypt later" attacks.
The server may need to be upgraded. See https://openSSH.com/pq.html
```
Enumerating objects: 64, done.
Counting objects: 100% (64/64), done.
Delta compression using up to 12 threads.
Compressing objects: 100% (47/47), done.
Writing objects: 100% (60/60), 702.27 KiB | 13.77 MiB/s, done.
Total 60 (delta 15), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Processing 1 references
remote: Processed 1 references in total
To ssh://gitverse.ru:2222/AndreySofich/2026-1---study---simulation-modeling.git
 ! 131b8e6..018678# master -> master
$ git push

```

Рисунок 3.6: Завершение настройки git

Скачиваем все необходимые плагины и пакеты, после чего переходим в консоль и запускаем julia (рис. 3.7).

```

PS C:\Users\12232\Documents\GitHub\2026-1---study---simulation-modeling\release> cd .\lab01\julia
PS C:\Users\12232\Documents\GitHub\2026-1---study---simulation-modeling> cd .\lab01\julia
julia> Documentation: https://docs.julialang.org
julia> Type "?" for help, "]??" for Pkg help.
julia> Version 1.12.0 (2025-10-07)
julia> Official https://julialang.org release

julia> using Pkg
julia> Pkg.add("DrWatson")
Resolving package versions...
  Project No packages added to or removed from 'C:\Users\12232\.julia\environments\v1.12\Project.toml'
  Manifest No packages added to or removed from 'C:\Users\12232\.julia\environments\v1.12\Manifest.toml'

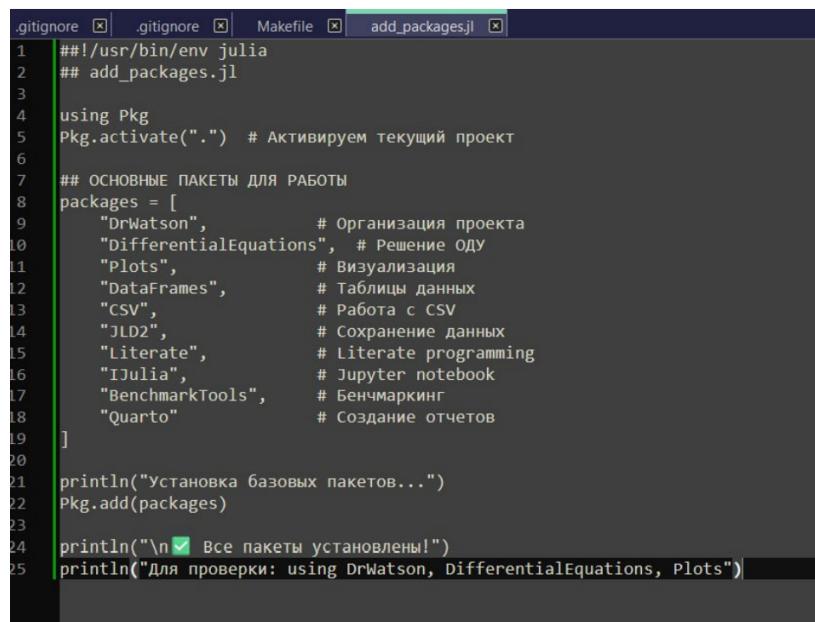
julia> using DrWatson

julia> initialize_project("project"; authors="Daniil Chistov", git=false)
Activating new project at 'C:\Users\12232\Documents\GitHub\2026-1---study---simulation-modeling\labs\lab01\project'
  Resolving package versions...
    Updating 'C:\Users\12232\Documents\GitHub\2026-1---study---simulation-modeling\labs\lab01\project\Project.toml'
      [634d3b9d] + DrWatson v2.19.1
      Updating 'C:\Users\12232\Documents\GitHub\2026-1---study---simulation-modeling\labs\lab01\project\Manifest.toml'
      [0b6fb169] + ChunkCodeCore v1.0.1
      [4c0bbbe4] + ChunkCodeLibZlib v1.0.0

```

Рисунок 3.7: Создание среды для работы

Создаем каталог и закачиваем все необходимые пакеты (рис. 3.8).



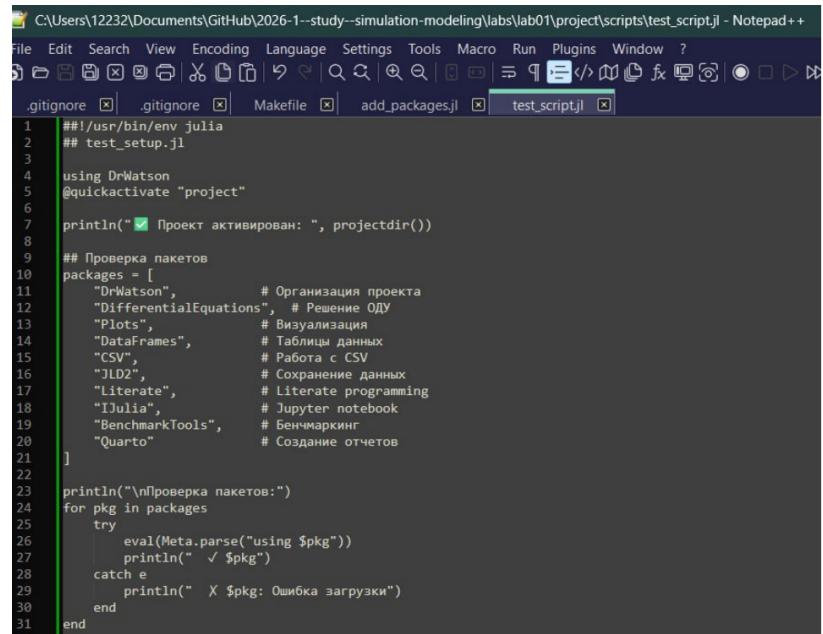
```

.gitignore .gitignore Makefile add_packages.jl
1  #!/usr/bin/env julia
2  ## add_packages.jl
3
4  using Pkg
5  Pkg.activate(".") # Активируем текущий проект
6
7  ## ОСНОВНЫЕ ПАКЕТЫ для РАБОТЫ
8  packages = [
9      "DrWatson",           # Организация проекта
10     "DifferentialEquations", # Решение ОДУ
11     "Plots",               # Визуализация
12     "DataFrames",          # Таблицы данных
13     "CSV",                  # Работа с CSV
14     "JLD2",                 # Сохранение данных
15     "Literate",              # Literate programming
16     "IJulia",                # Jupyter notebook
17     "BenchmarkTools",        # Бенчмаркинг
18     "Quarto"                 # Создание отчетов
19 ]
20
21 println("Установка базовых пакетов...")
22 Pkg.add(packages)
23
24 println("\n✅ Все пакеты установлены!")
25 println("Для проверки: using DrWatson, DifferentialEquations, Plots")

```

Рисунок 3.8: Настройка julia

Создаем файл, вписываем в него решение задачи о популяции(сначала был файл без текстовых вставок, после чего его нужно поменять) (рис. 3.9).



The screenshot shows a Notepad++ window with the title bar "C:\Users\12232\Documents\GitHub\2026-1--study--simulation-modeling\abs\lab01\project\scripts\test_script.jl - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. Below the menu is a toolbar with various icons. The main editor area contains the following code:

```
1  ##!/usr/bin/env julia
2  ## test_setup.jl
3
4  using DrWatson
5  @quickactivate "project"
6
7  println("✓ Проект активирован: ", projectdir())
8
9  ## Проверка пакетов
10 packages = [
11     "DrWatson",           # Организация проекта
12     "DifferentialEquations", # Решение ОДУ
13     "Plots",               # Визуализация
14     "DataFrames",          # Таблицы данных
15     "CSV",                  # Работа с CSV
16     "JLD2",                 # Сохранение данных
17     "Literate",              # Literate programming
18     "IJulia",                # Jupyter notebook
19     "BenchmarkTools",        # Бенчмаркинг
20     "Quarto"                  # Создание отчетов
21 ]
22
23 println("\nПроверка пакетов:")
24 for pkg in packages
25     try
26         eval(Meta.parse("using $pkg"))
27         println(" ✓ $pkg")
28     catch e
29         println(" ✗ $pkg: Ошибка загрузки")
30     end
31 end
```

Рисунок 3.9: Исполняемый файл №1

Создаем скрипт для генерации производных форматов и создаем эти форматы, результат можно увидеть например в visual studio, когда наш основной файл перешел в формат jupyter (рис. 3.10).

```

01_exponential_growth.pyml • ! .quarto.yml •
C: > Users > Andrew > work > study > 2026-1 > 2026-1-study--simulation-modeling > 2026-1-study--simulation-modeling > labs > lab01 > report > ! _quarto.yml
  2 project:
  3   standalone: true
  4   self-contained: true
  5
  6   ## Julia support
  7   engine: julia
  8   julia:
  9     exeflags: ["--project=../project"]
 10
 11   ## Generic options
 12   lang: ru-RU
 13   number-sections: true
 14   toc: true
 15   toc-title: "Содержание"
 16   toc-depth: 2
 17
 18   ## Crossref customization
 19   crossref:
 20     lof-title: "Список иллюстраций"
 21     lot-title: "Список таблиц"
 22     lol-title: "Листинги"
 23
 24   ## Bibliography
 25   bibliography:
 26     cs1: _resources/csl/gost-r-7-0-5-2008-numeric.csl
 27
 28   ## Formats
 29   format:
 30     docx:
 31       ## Docx output format
 32       toc: true
 33       number-sections: true
 34       toc-depth: 2
 35
 36     pdf:
 37       ## Pdf output format
 38       # pdf-engine: xelatex
 39       ## minted support
 40       # filters:
 41       # - minted-quarto
 42       toc: true
 43       number-sections: true
 44       colorlinks: false
 45       toc-depth: 2
 46       lof: true # List of figures
 47       lot: true # List of tables
 48
 49   #### Document

```

Рисунок 3.10: Создание производных форматов

Создаем второй файл и делаем все тоже самое с ним (рис. 3.11).

```

[редактор]: C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\labs\lab01\project\plots
PS C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project> julia --project= scripts/01_exponential_growt
Первые 5 строк результатов:
5x2 DataFrame
Row | t      u
     | Float64 Float64
1   | 0.0    1.0
2   | 0.1    1.03045
3   | 0.2    1.06184
4   | 0.3    1.09117
5   | 0.4    1.12705

Анализическое время выполнения: 2.31
PS C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project> julia --project= scripts/01_exponential_growt
Первые 5 строк результатов:
5x2 DataFrame
Row | t      u
     | Float64 Float64
1   | 0.0    1.0
2   | 0.1    1.03045
3   | 0.2    1.06184
4   | 0.3    1.09117
5   | 0.4    1.12705

Анализическое время выполнения: 2.31
PS C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project> julia --project= scripts/tangle.jl scripts/01_exponential_growth.jl
[ Info: generating plain script file from C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\scripts\01_exponential_growth.jl
] Info: writing result to C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\tangle\01_exponential_growth.jl
[ Info: generating markdown page from C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\scripts\01_exponential_growth.md
] Info: writing result to C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\tangle\01_exponential_growth.md
[ Info: generating notebook from C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\scripts\01_exponential_growth.ipynb
] Info: writing result to C:\Users\Andrew\work\study\2026-1\2026-1-study--simulation-modeling\2026-1-study--simulation-modeling\lab01\project\tangle\01_exponential_growth.ipynb

```

Рисунок 3.11: Исполняемый файл №2

Переходим в `report` и создаем отчет (рис. 3.12).

```

variables:
  graphics: true
  tables: true
  default-image-extension: pdf
  number-sections: true
  toc: true
  toc-depth: 2
  cite-method: bblatex

metadata
  documentclass: scrreprt
  classoption:
    - DIV=11
    - numbers=nonendperiod
  paper-size: a4
  header-includes:
    - \KOMAoption{captions}{tableheading}
  block-headings: true
  lang: ru-RU
  toc-title: Содержание
  crossref:
    lof-title: Список иллюстраций
    lot-title: Список таблиц
    lol-title: Листинги
  bibliography:
    - bib/cite.bib
  csl: _resources/csl/gost-r-7-0-5-2008-numeric.csl
  colorlinks: false
  lof: true
  lot: true
  fontsize: 12pt
  linestretch: 1.5
  babel-lang: russian
  babel-otherlangs: english
  biblio-style: gost-numeric
  bblatexoptions:
    - backend=biber
    - langhook-extras
    - autolang=other"
  csquotes: true
  indent: true
  author:
    name: Андрей Геннадьевич Софич
    degrees: DSc
    orcid: 0000-0002-0877-7063
    email: kulyabov-ds@rudn.ru
    affiliation:
      - name: Российский университет дружбы народов
        country: Российская Федерация
        postal-code: 117198
        city: Москва
        address: ул. Миклухо-Маклая, д. 6
    title: Шаблон отчёта по лабораторной работе
    subtitle: Простейший вариант
    license: CC BY

Rendering PDF
running lualatex - 1

No TeX installation was detected.

Please run 'quarto install tinytex' to install TinyTeX.
If you prefer, you may install TeXLive or another TeX distribution.

WARN: Error encountered when rendering files

```

Рисунок 3.12: Проверка создания отчета

Переходим в консоль, создаем отчет и проверяем правильность компиляции (рис. 3.13).

The screenshot shows a Jupyter Notebook cell with the following content:

```
function exponential_growth!(du, u, p, t)
    α = p
    du[1] = α * u[1]
end

# ## Первый запуск с параметрами по умолчанию
# Зададим начальные параметры:
u₀ = [1.0]           # начальная популяция
α = 0.3              # скорость роста
tspan = (0.0, 10.0) # временной интервал

prob = ODEProblem(exponential_growth!, u₀, tspan, α)
sol = solve(prob, Tsit5(), saveat=0.1)

# ## Визуализация результатов
# Построим график решения:
plot(sol, label="u(t)", xlabel="Время t", ylabel="Популяция u",
     title="Экспоненциальный рост (α = $α)", lw=2, legend=:topleft)

# Сохраним график в папку plots
savefig(plotsdir(script_name, "exponential_growth_α=$α.png"))

# ## 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) / α
println("\nАналитическое время удвоения: ", round(doubling_time; digits=2))

# ## Сохранение всех результатов
@save datadir(script_name, "all_results.jld2") df
```

Страна 55, столбец 48 1 506 символов Обычный текст 100% Windows (CRLF) UTF-8

Рисунок 3.13: Правильность компиляции отчета

4 Выводы

В данной работе мы создали необходимое окружение и подключили нужные функции для будущих работ, а так же попробовали создать некоторые примеры.