

Отчёт по лабораторной работе №5

Построение графиков

Козлов Всеволод Павлович НФИбд-02-22

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Выводы	20
5 Список литературы	21

Список иллюстраций

3.1 Задание 5.2.1	7
3.2 Задание 5.2.1	8
3.3 Задание 5.2.2	8
3.4 Задание 5.2.3	9
3.5 Задание 5.2.3	9
3.6 Задание 5.2.4	10
3.7 Задание 5.2.5	10
3.8 Задание 5.2.6	11
3.9 Задание 5.2.7	11
3.10 Задание 5.2.8	12
3.11 Задание 5.2.9	12
3.12 Задание 5.2.12	13
3.13 Задание 5.2.13	13
3.14 Задание 5.2.14	14
3.15 Задание 5.4.1	14
3.16 Задание 5.4.2	15
3.17 Задание 5.4.3	15
3.18 Задание 5.4.4	16
3.19 Задание 5.4.5	16
3.20 Задание 5.4.6	17
3.21 Задание 5.4.7	17
3.22 Задание 5.4.8	18
3.23 Задание 5.4.9	18
3.24 Задание 5.4.10	19
3.25 Задание 5.4.11	19

Список таблиц

1 Цель работы

Освоить синтаксис языка Julia для построения графиков

2 Задание

1. Используя Jupyter Lab, повторите примеры из раздела 5.2. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.
2. Выполните задания для самостоятельной работы (раздел 5.4).

3 Выполнение лабораторной работы

Задание 5.2.1 (рис. 3.1)

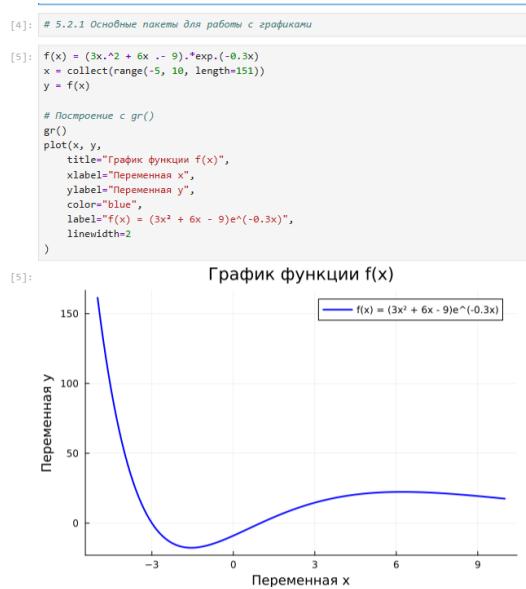


Рис. 3.1: Задание 5.2.1

Задание 5.2.1 (рис. 3.2)

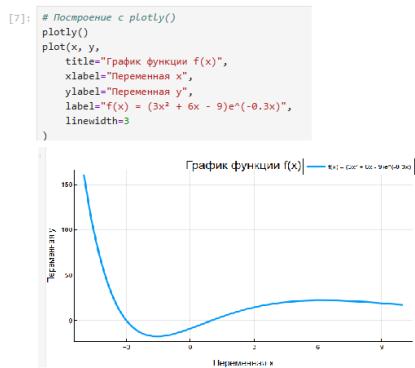


Рис. 3.2: Задание 5.2.1

Задание 5.2.2 (рис. 3.3)

```
[12]: # 5.2.2 Сравнить sin(x) с разложением в ряд Тейлора
sin_theor(x) = sin(x)
sin_taylor(x) = sum([(-1)^i * x^(2*i+1) / factorial(2*i+1) for i in 0:4])
x_taylor = range(-2π, 2π, length=100)
plot(x_taylor, sin_theor.(x_taylor), label="sin(x) теор", linewidth=2)
plot!(x_taylor, sin_taylor.(x_taylor), label="ряд Тейлора", linestyle=:dash)
```

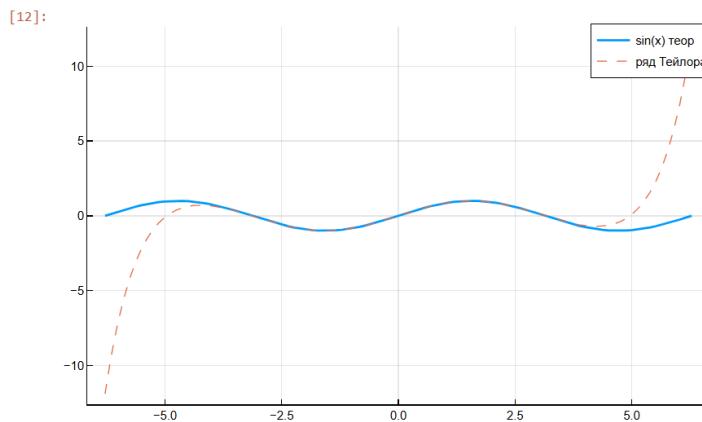


Рис. 3.3: Задание 5.2.2

Задание 5.2.3 (рис. 3.4)

```
[14]: # 5.2.3 Построить точечные графики 2D и 3D
x_scatter = range(1, 10, length=10)
y_scatter = rand(10)
scatter(x_scatter, y_scatter, title="Точечный график", xlabel="X", ylabel="Y")
```

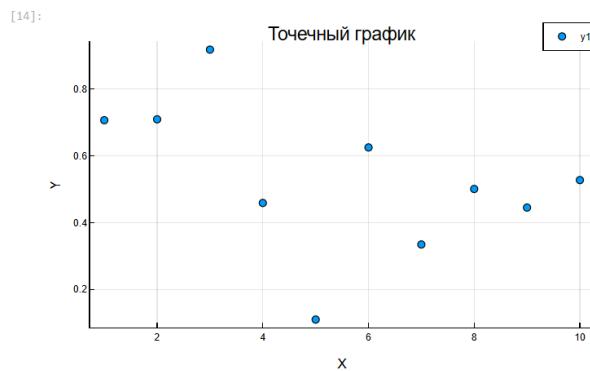


Рис. 3.4: Задание 5.2.3

Задание 5.2.3 (рис. 3.5)

[15]:

```
scatter(rand(50), rand(50), rand(50), title="3D точечный график")
```

[15]:

3D точечный график

A 3D scatter plot titled "3D точечный график". The plot shows 50 data points represented as blue spheres. The axes are labeled from 0.00 to 1.00. The x-axis ranges from 0.00 to 1.00, the y-axis from 0.00 to 1.00, and the z-axis from 0.00 to 1.00.

Рис. 3.5: Задание 5.2.3

Задание 5.2.4 (рис. 3.6)

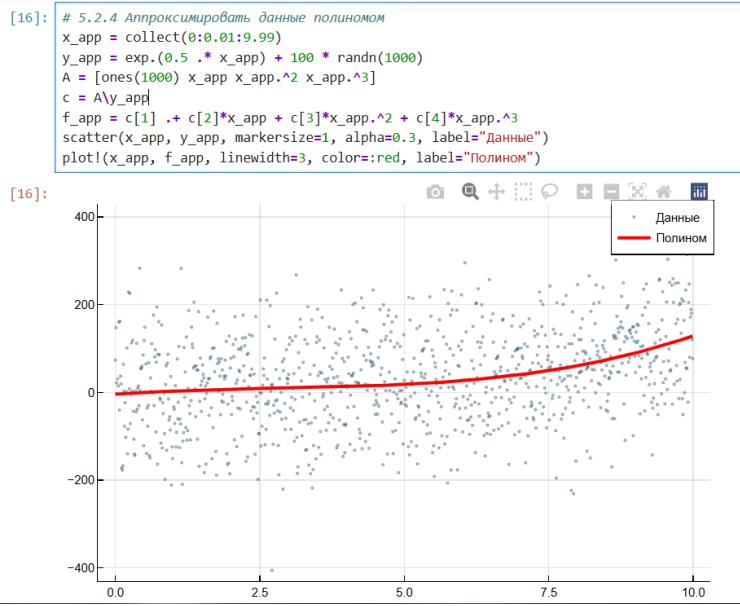


Рис. 3.6: Задание 5.2.4

Задание 5.2.5 (рис. 3.7)

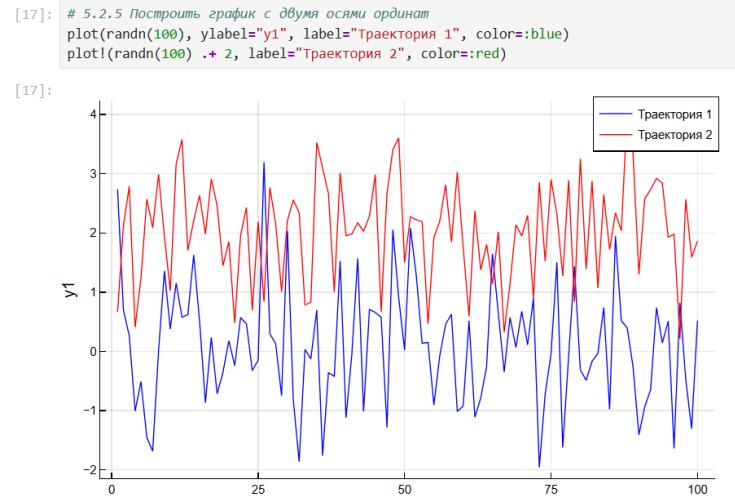


Рис. 3.7: Задание 5.2.5

Задание 5.2.6 (рис. 3.8)

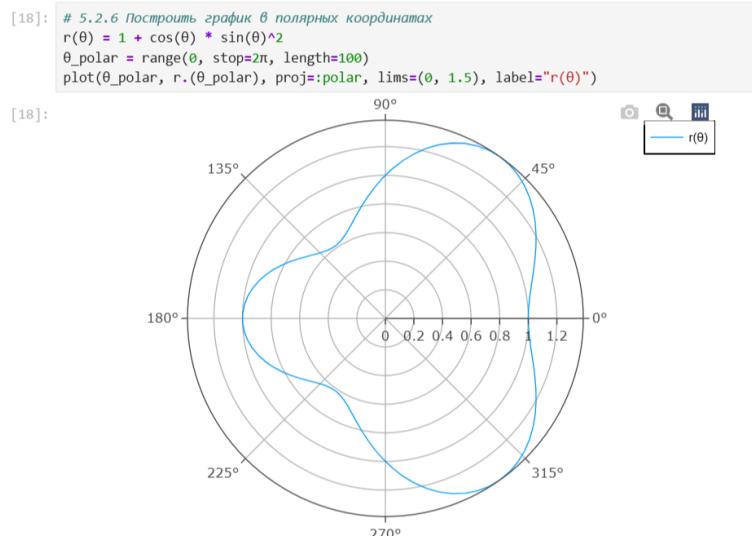


Рис. 3.8: Задание 5.2.6

Задание 5.2.7 (рис. 3.9)

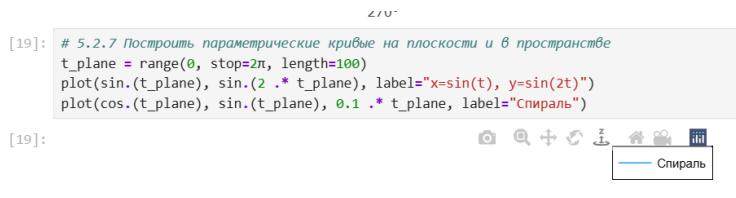


Рис. 3.9: Задание 5.2.7

Задание 5.2.8 (рис. 3.10)

```
[20]: # 5.2.8 Построить поверхности разными способами
f_surf(x,y) = x^2 + y^2
x_surf = -5:0.5:5
y_surf = -5:0.5:5
surface(x_surf, y_surf, f_surf, title="Поверхность")
plot(x_surf, y_surf, f_surf, linetype=:wireframe, title="Каркас")
```

[20]:
Каркас

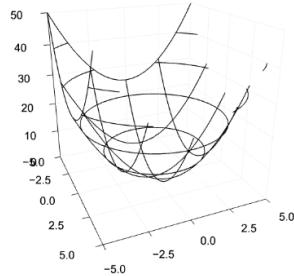


Рис. 3.10: Задание 5.2.8

Задание 5.2.9 (рис. 3.11)

```
[21]: # 5.2.9 Построить линии уровня с заполнением
g(x, y) = (3x + y^2) * abs(sin(x) + cos(y))
x_contour = 1:0.5:10
y_contour = 1:0.5:8
contour(x_contour, y_contour, g, fill=true, title="Линии уровня")
```

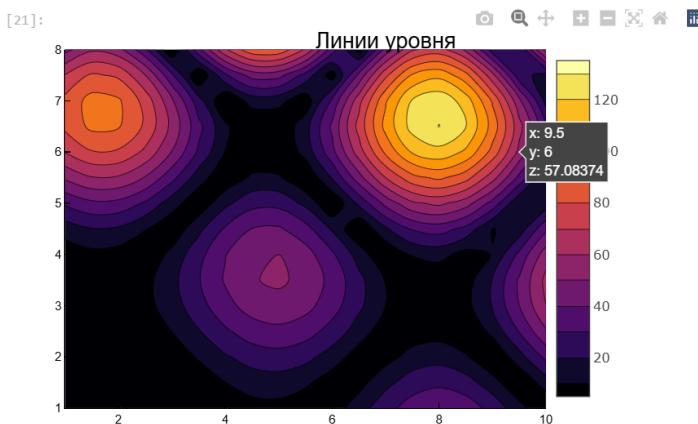


Рис. 3.11: Задание 5.2.9

Задание 5.2.12 (рис. 3.12)

```
[22]: # 5.2.12 Построить график с errorbars
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
y_err = [mean(sd * randn(10)) for sd in sds]
errs = 1.96 * sds / sqrt(10)
plot(y_err, errs, label="данные с ошибками", marker=stroke(3, :orange))
```

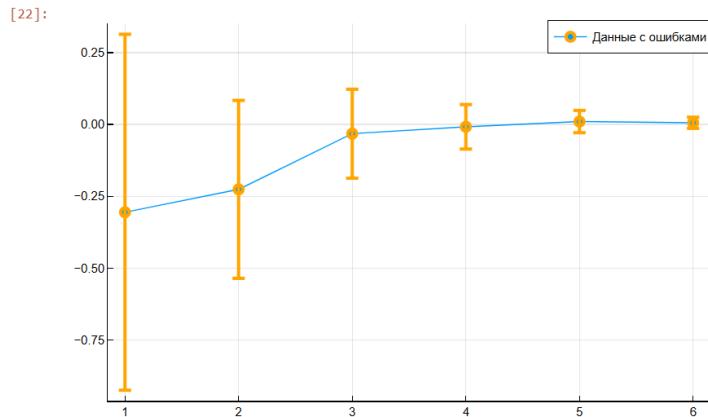


Рис. 3.12: Задание 5.2.12

Задание 5.2.13 (рис. 3.13)

```
[24]: # 5.2.13 Построить гистограмму нормального распределения
d = Normal(35.0, 10.0)
ages = rand(d, 1000)
histogram(ages, label="Распределение", xlabel="Возраст", ylabel="Количество")
```

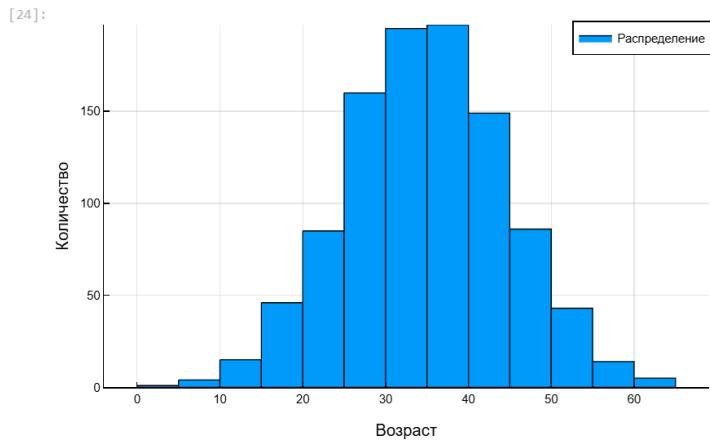


Рис. 3.13: Задание 5.2.13

Задание 5.2.14 (рис. 3.14)

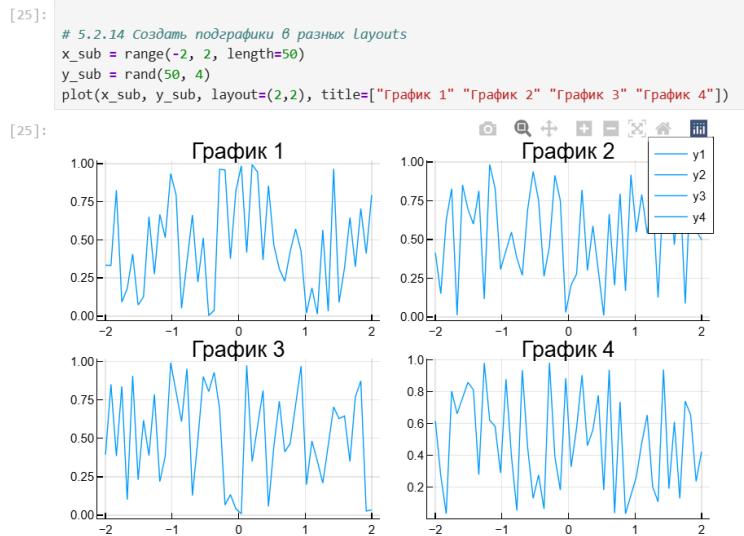


Рис. 3.14: Задание 5.2.14

Задание 5.4.1 (рис. 3.15)

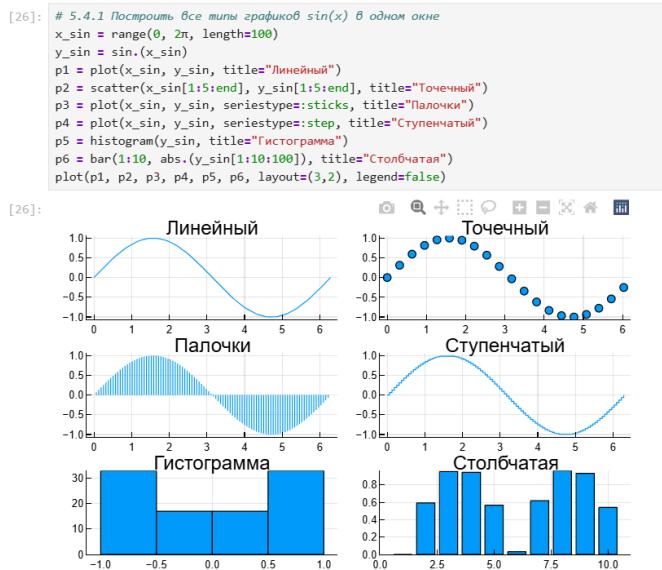


Рис. 3.15: Задание 5.4.1

Задание 5.4.2 (рис. 3.16)

```
[31]: # 5.4.2 Построить sin(x) со всеми типами оформления линий
plot(x_sin, y_sin, linestyle=:solid, label="Стиль 1", title="Разные стили линий")
plot!(x_sin, y_sin .+ 0.3, linestyle=:dash, label="Стиль 2")
plot!(x_sin, y_sin .+ 0.6, linestyle=:dot, label="Стиль 3")
plot!(x_sin, y_sin .+ 0.9, linestyle=:dashdot, label="Стиль 4")
```

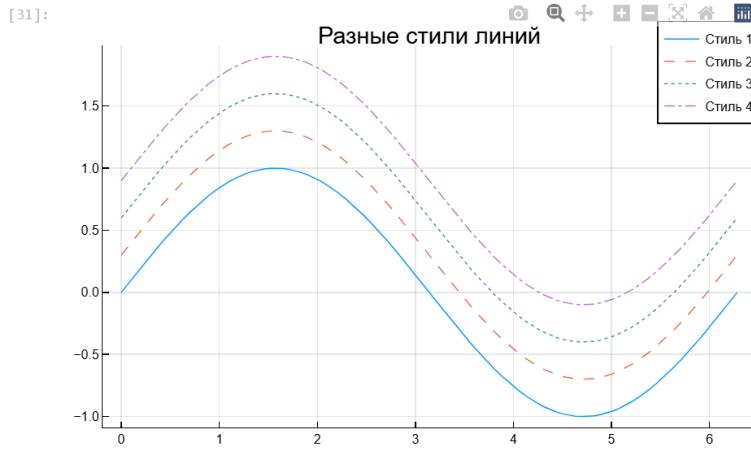


Рис. 3.16: Задание 5.4.2

Задание 5.4.3 (рис. 3.17)

```
[32]: # 5.4.3 Построить график с специальным оформлением
y_special(x) = π * x^2 * log(x + 0.1)
x_special = range(0.1, 2, length=100)
plot(x_special, y_special.(x_special), title="y = πx^2ln(x)", color=:red, linewidth=3, framestyle=:box)
```

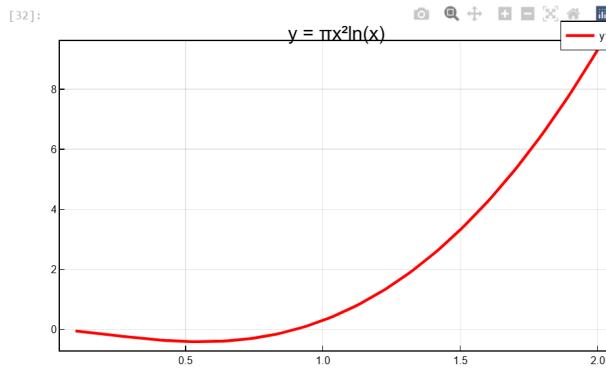


Рис. 3.17: Задание 5.4.3

Задание 5.4.4 (рис. 3.18)



Рис. 3.18: Задание 5.4.4

Задание 5.4.5 (рис. 3.19)

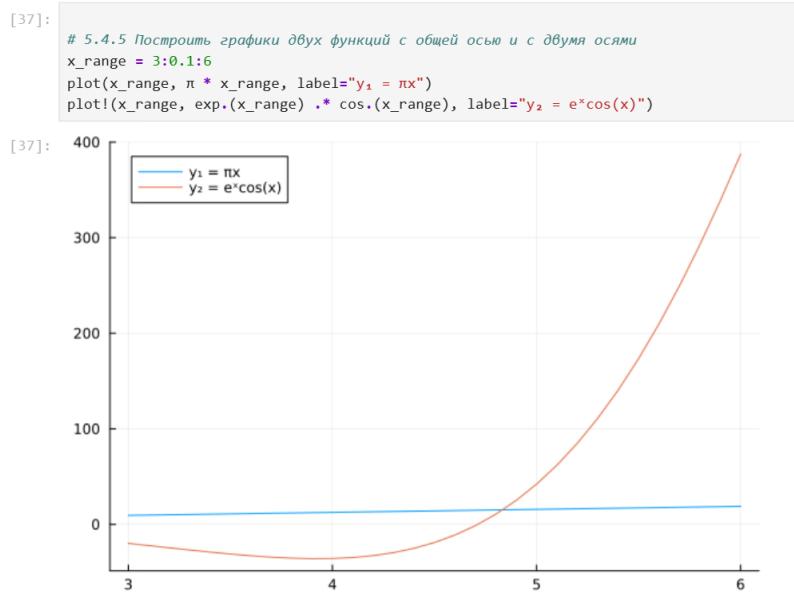


Рис. 3.19: Задание 5.4.5

Задание 5.4.6 (рис. 3.20)

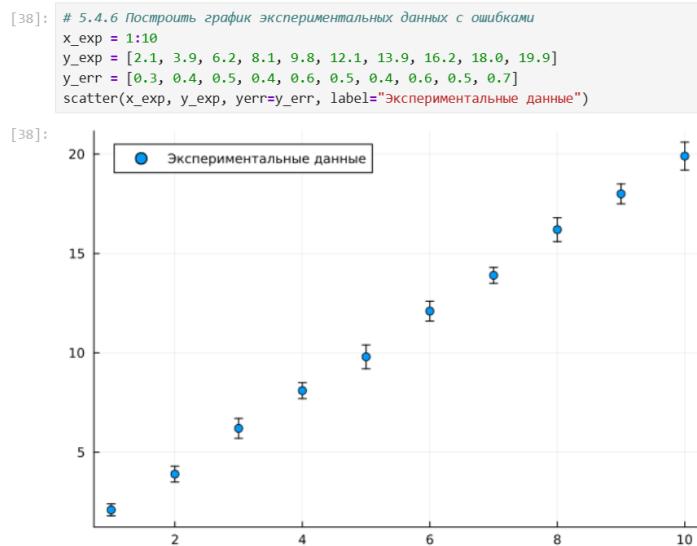


Рис. 3.20: Задание 5.4.6

Задание 5.4.7 (рис. 3.21)

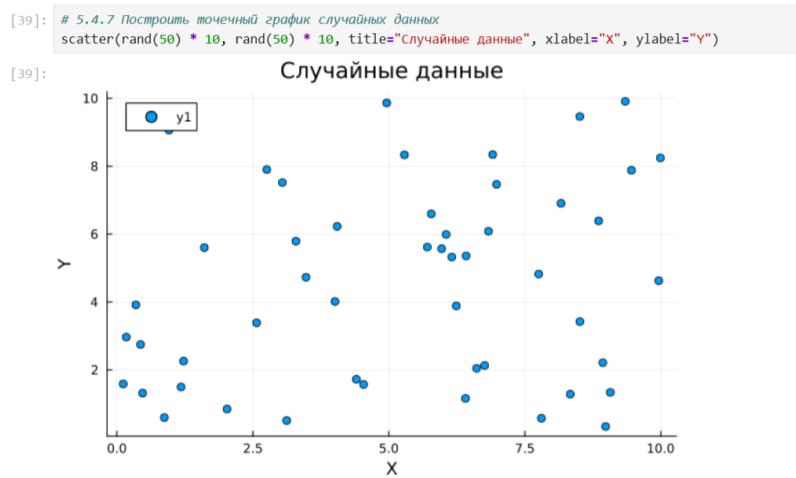


Рис. 3.21: Задание 5.4.7

Задание 5.4.8 (рис. 3.22)

```
[40]: # 5.4.8 Построить 3D точечный график случайных данных  
scatter(rand(100) * 10, rand(100) * 10, rand(100) * 10, title="3D случайные данные")
```

```
[40]: 3D случайные данные
```

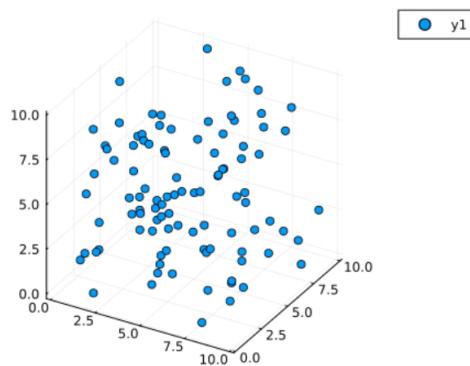


Рис. 3.22: Задание 5.4.8

Задание 5.4.9 (рис. 3.23)

```
11]: # 5.4.9 Создать анимацию построения синусоиды  
anim_sin = @animate for i in 1:100  
    x_anim = range(0, i/10, length=100)  
    plot(x_anim, sin.(x_anim), xlim=(0, 10), ylim=(-1.5, 1.5), title="Анимация синусоиды")  
end  
gif(anim_sin, "sine_animation.gif", fps=10)
```

```
[ Info: Saved animation to C:\studies\RUDN\linux_stoch\lab5\sine_animation.gif  
11]: Анимация синусоиды
```

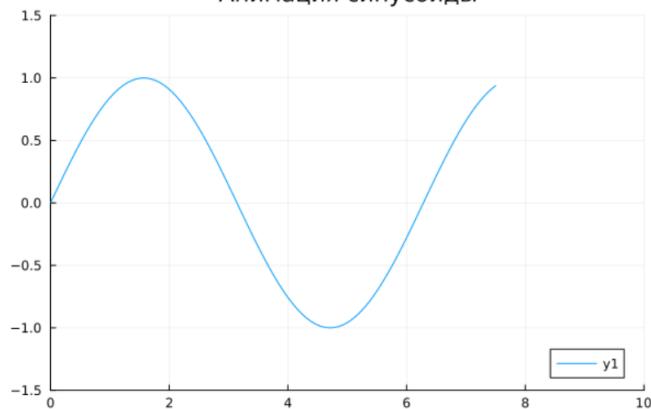


Рис. 3.23: Задание 5.4.9

Задание 5.4.10 (рис. 3.24)

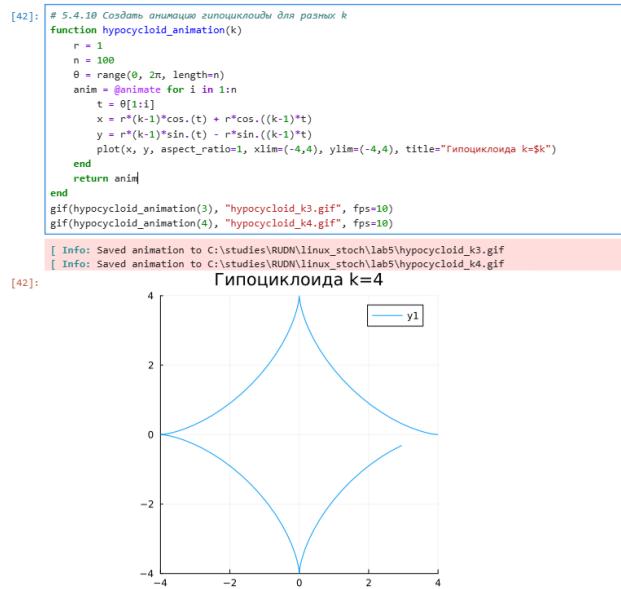


Рис. 3.24: Задание 5.4.10

Задание 5.4.11 (рис. 3.25)

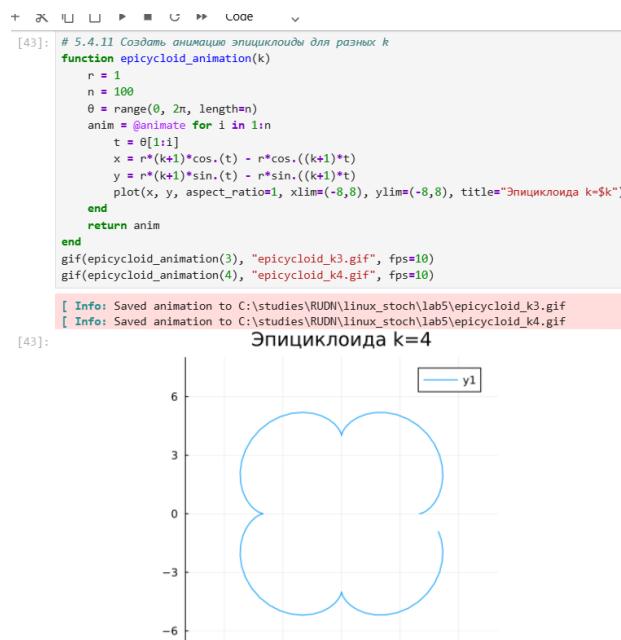


Рис. 3.25: Задание 5.4.11

4 Выводы

Освоил синтаксис языка Julia для построения графиков.

5 Список литературы

1. Julia 1.5 Documentation. — 2020. — URL: <https://docs.julialang.org/en/v1/>.
2. Klok H.,Nazarathy Y. Statistics with Julia: Fundamentals for Data Science,Machine Learning and Artificial Intelligence. — 2020. — URL: <https://statisticswithjulia.org/>.
3. Ökten G. First Semester in Numerical Analysis with Julia. — Florida State University, 2019. — DOI: 10.33009/jul.
4. Антонюк В. А. Язык Julia как инструмент исследователя. — М. : Физический факультет МГУ им. М. В. Ломоносова, 2019.
5. Шиндин А. В. Язык программирования математических вычислений Julia. Базовое руководство. — Нижний Новгород : Нижегородский госуниверситет, 2016.