

Построение графиков

Лабораторная работа № 4

Шулуужук Айраана НПИбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
	2.1 Выполнение самостоятельной работы	6
3	Выводы	17

Список иллюстраций

2.1	построение графиков	6
2.2	различные типы оформлений графиков	7
2.3	Транспонирование, след, ранг, определитель и инверсия матрицы	8
2.4	построение графиков	9
2.5	результат	9
2.6	5 задание	10
2.7	5 задание	11
2.8	6 задание	12
2.9	результат	12
2.10	7 задание	13
2.11	8 задание	14
2.12	9 задание	15
2.13	10 задание	16

Список таблиц

1 Цель работы

Основная цель работы — освоить синтаксис языка Julia для построения графиков./

2 Выполнение лабораторной работы

2.1 Выполнение самостоятельной работы

1. Постройте все возможные типы графиков (простые, точечные, гistogramмы и т.д.) функции $y = \sin(x)$, $x = 0, 2\pi$. Отобразите все графики в одном графическом окне (рис. 2.1)

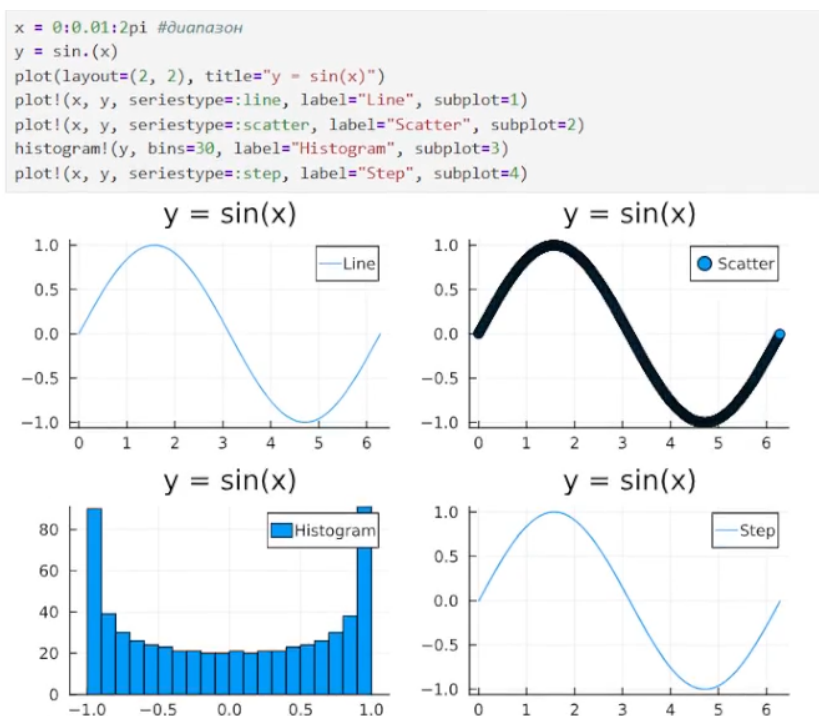


Рис. 2.1: построение графиков

2. Постройте графики функции $y = \sin(x)$, $x = 0, 2\pi$ со всеми возможными (сколько сможете вспомнить) типами оформления линий графика.

Отобразите все графики в одном графическом окне (рис. 2.2)



Рис. 2.2: различные типы оформлений графиков

3. Постройте график функции $y(x) = \pi \cdot x^2 \ln(x)$, назовите оси соответственно. Пусть цвет рамки будет зелёным, а цвет самого графика — красным. Задайте расстояние между надписями и осями так, чтобы надписи полностью уместались в графическом окне. Задайте шрифт надписей. Задайте частоту отметок на осях координат (рис. 2.3)

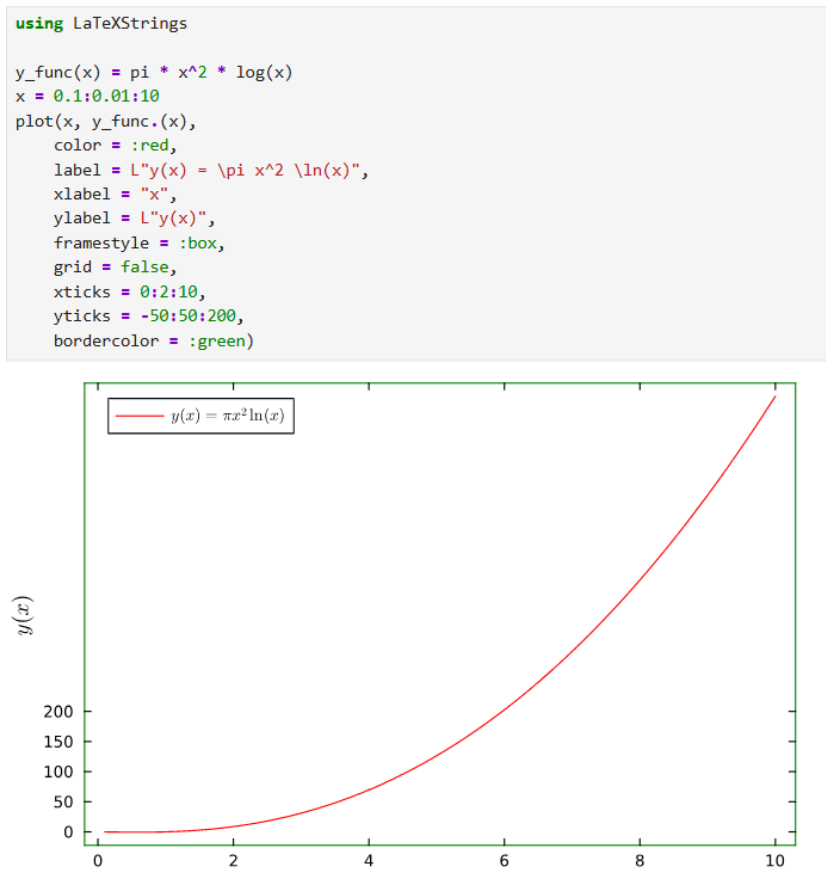


Рис. 2.3: Транспонирование, след, ранг, определитель и инверсия матрицы

4. Задайте вектор $x = (-2, -1, 0, 1, 2)$. В одном графическом окне (в 4-х подокнах) изобразите графически по точкам значения функции $y(x) = x^3 - 3x$ в виде: – точек, – линий, – линий и точек, – кривой. Сохраните полученные изображения в файле `figure_familiya.png`, где вместо `familiya` укажите вашу фамилию (рис. 2.4)


```

x = [-2, -1, 0, 1, 2]
y_f(x) = x.^3 .- 3 .* x
y_values = y_f.(x)
plt = plot(layout = (2, 2))
scatter!(plt[1], x, y_values, label = "Точки")
plot!(plt[2], x, y_values, label = "Линии")
plot!(plt[3], x, y_values, label = "Линии и точки")
scatter!(plt[3], x, y_values, label = "")
plot!(plt[4], x, y_values, smooth = true, label = "Кривая")
savefig(plt, "figure_shuluuzhuk.png")

```

"C:\\Users\\airan\\figure_shuluuzhuk.png"

Рис. 2.4: построение графиков

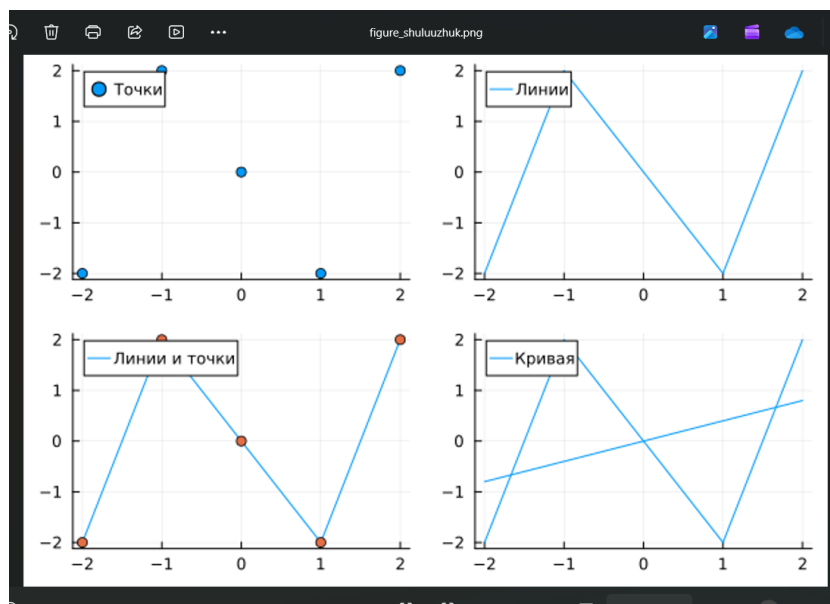


Рис. 2.5: результат

5. Задайте вектор $x = (3, 3.1, 3.2, \dots, 6)$. Постройте графики функций $y_1(x) = \pi x$ и $y_2(x) = \exp(x) \cos(x)$ в указанном диапазоне значений аргумента x следующим образом: – постройте оба графика разного цвета на одном рисунке, добавьте легенду и сетку для каждого графика; укажите недостатки у данного построения; – постройте аналогичный график с двумя осями ординат (рис. 2.6) (рис. 2.7)

```

x_1 = 3:0.1:6
y_1(x) = pi * x
y_2(x) = exp(x) * cos(x)

# Сначала создаем основной график
plot(x_1, y_1(x_1), label="y1(x) = π * x", color=:blue, grid=true, legend=:topright)

# Затем добавляем вторую функцию
plot!(x_1, y_2(x_1), label="y2(x) = exp(x) * cos(x)", color=:red)

# Добавляем подписи осей и заголовок
xlabel!("x")
ylabel!("y")
title!("Графики функций y1(x) и y2(x)")

```

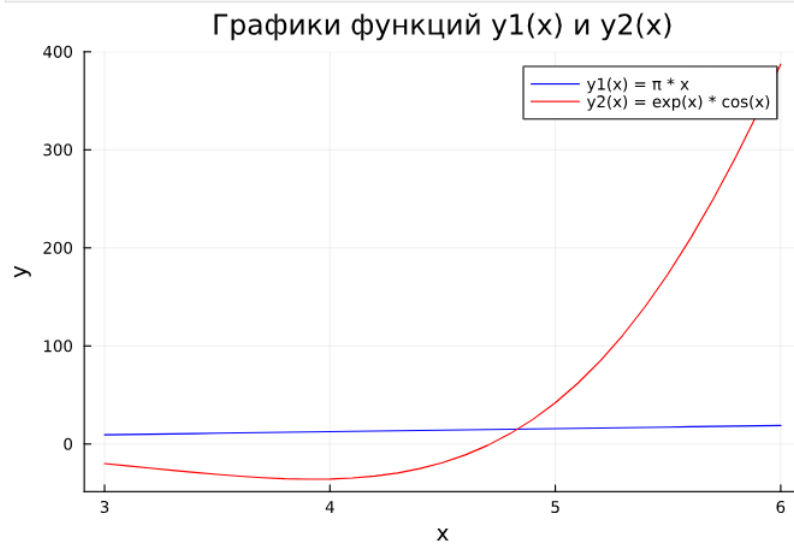


Рис. 2.6: 5 задание

```
# Построение графиков с двумя осями ординат
p = plot(x, y1(x), label="y1(x) = πx", color=:blue, linewidth=2, grid=true, legend=:to
plot!(p, x, y2(x), label="y2(x) = exp(x) * cos(x)", color=:red, linewidth=2)
# Добавляем вторую ось ординат для y2
plot!(p, secondary=true)
# Заголовок и сетка
xlabel!("x")
ylabel!("y1 (primary)", fontsize=10)
ylabel!(p, "y2 (secondary)", fontsize=10)
title!("Графики с двумя осями ординат")
```

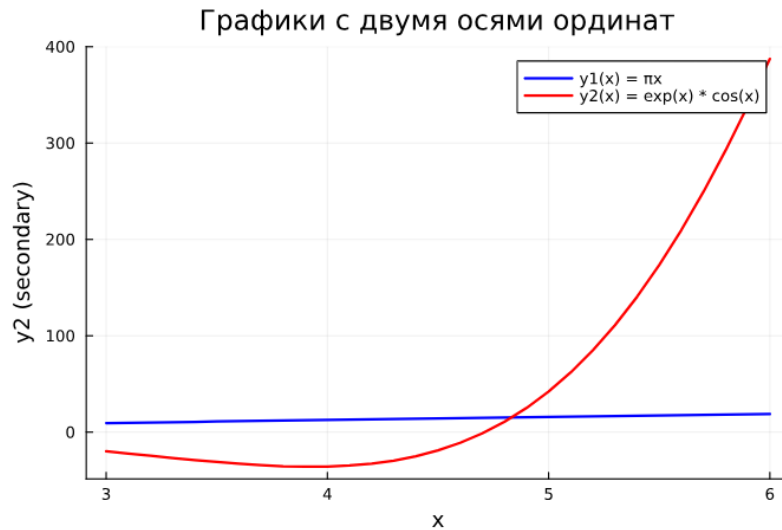


Рис. 2.7: 5 задание

6. Постройте график некоторых экспериментальных данных (придумайте сами), учитывая ошибку измерения (рис. 2.8) (рис. 2.9)

```

using Plots

# Данные по кинетике реакции (концентрация от времени)
time = 0:2:30 # время в минутах
concentration = [1.00, 0.85, 0.72, 0.61, 0.52, 0.44, 0.38, 0.32, 0.27, 0.23, 0.20, 0.17, 0.14,
                0.12, 0.10, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.00]

# Относительные ошибки измерения (5%)
relative_errors = 0.05 .* concentration
absolute_errors = relative_errors

# Построение графика
plot(time, concentration,
     ribbon=absolute_errors,
     fillalpha=0.3,
     fillcolor=:lightblue,
     label="Концентрация реагента",
     color=:blue,
     linewidth=2,
     xlabel="Время, мин",
     ylabel="Концентрация, моль/л",
     title="Кинетика химической реакции",
     legend=:topright,
     grid=true)

scatter!(time, concentration,
        label="Экспериментальные точки",
        color=:blue,
        markersize=4)

# Добавляем теоретическую кривую (экспоненциальный распад)
theoretical = exp.(-0.1 .* time)
plot!(time, theoretical,
     label="Теоретическая модель",
     color=:red,
     linestyle=:dash,
     linewidth=2)

annotate!(15, 0.7, text("Погрешность: ±5%", 10, :left))

```

Рис. 2.8: 6 задание

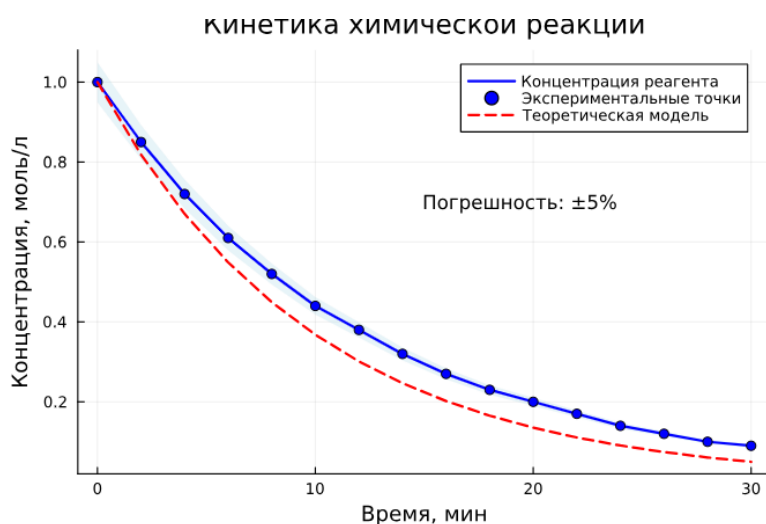


Рис. 2.9: результат

7. Постройте точечный график случайных данных. Подпишите оси,

легенду, название графика (рис. 2.10).

```
x = randn(100) # 100 случайных точек с нормальным распределением
y = randn(100)

# Строим точечный график
scatter(x, y,
        label="Случайные точки",
        color=:blue,
        markersize=6,
        alpha=0.7,
        xlabel="X координата",
        ylabel="Y координата",
        title="Точечный график случайных данных",
        legend=:topright,
        grid=true,
        framestyle=:box)

# Добавляем линию тренда (опционально)
using Statistics
if length(x) > 1
    A = [ones(length(x)) x]
    coeff = A \ y
    trend = A * coeff
    plot!(x, trend, label="Линия тренда", color=:red, linewidth=2)
end
```



Рис. 2.10: 7 задание

8. Постройте 3-мерный точечный график случайных данных. Подпишите оси, легенду, название графика. (рис. 2.11)

```

y = randn(n)
z = randn(n)

# Строим 3D точечный график
scatter(x, y, z,
        label="Случайные точки",
        color=:blue,
        markersize=4,
        alpha=0.7,
        xlabel="Ось X",
        ylabel="Ось Y",
        zlabel="Ось Z",
        title="3D точечный график случайных данных",
        legend=:topright,
        camera=(45, 30), # Угол обзора
        marker=:circle)

# Добавляем центр распределения
scatter!([mean(x)], [mean(y)], [mean(z)],
         label="Центр распределения",
         color=:red,
         markersize=8,
         marker=:diamond)

```

3D точечный график случайных данных

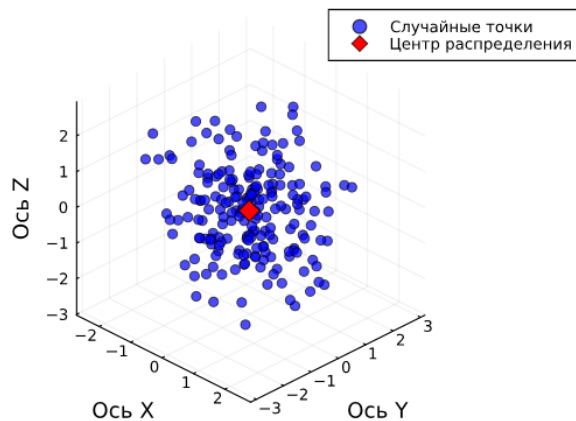


Рис. 2.11: 8 задание

9. Создайте анимацию с построением синусоиды. То есть вы строите последовательность графиков синусоиды, постепенно увеличивая значение аргумента. После соедините их в анимацию (рис. 2.12)

```

# Создаем анимацию
anim = @animate for n in 1:100
    x_range = range(0, stop=2π * n/100, length=1000)
    y = sin.(x_range)

    plot(x_range, y,
        linewidth=3,
        color=:blue,
        xlabel="x",
        ylabel="sin(x)",
        title="Построение синусоиды: шаг $n/100",
        xlims=(0, 2π),
        ylims=(-1.2, 1.2),
        legend=false,
        grid=true,
        framestyle=:box)

    vline!([2π * n/100], color=:red, linestyle=:dash, linewidth=2)
end

gif(anim, "sine_wave_animation.gif", fps=15)

```

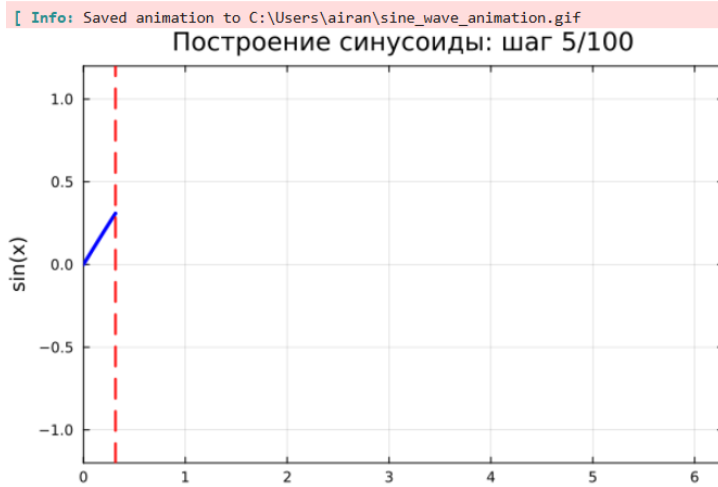


Рис. 2.12: 9 задание

10. Постройте анимированную гипоциклоиду для 2 целых значений модуля a и 2 рациональных значений модуля k

```

# Функция для вычисления координат гипоциклоиды
function hypocycloid(R, r, θ)
    x = (R - r) * cos.(θ) + r * cos.((R - r) / r * θ)
    y = (R - r) * sin.(θ) - r * sin.((R - r) / r * θ)
    return x, y
end
# Функция для создания анимации гипоциклоиды для целого значения k
function create_hypocycloid_animation(k)
    R = 10 # Радиус большой окружности
    r = R / k # Радиус маленькой окружности
    θ = 0:0.05:2 * π # Угол от 0 до 2π, шаг 0.05
    x, y = hypocycloid(R, r, θ)
    anim = @animate for i in 1:length(θ) # Итерация по всем углам
        plot(x[1:i], y[1:i], label="Гипоциклоид (k=$k)", color=:blue, linewidth=2,
            xlabel="x", ylabel="y", title="Анимация гипоциклоиды для k=$k", legend=:topright)
    end
    # Сохранение анимации в файл
    gif(anim, "hypocycloid_k_$(k).gif", fps=10)
end
# Создание анимации для целого значения k = 2
create_hypocycloid_animation(2)

```

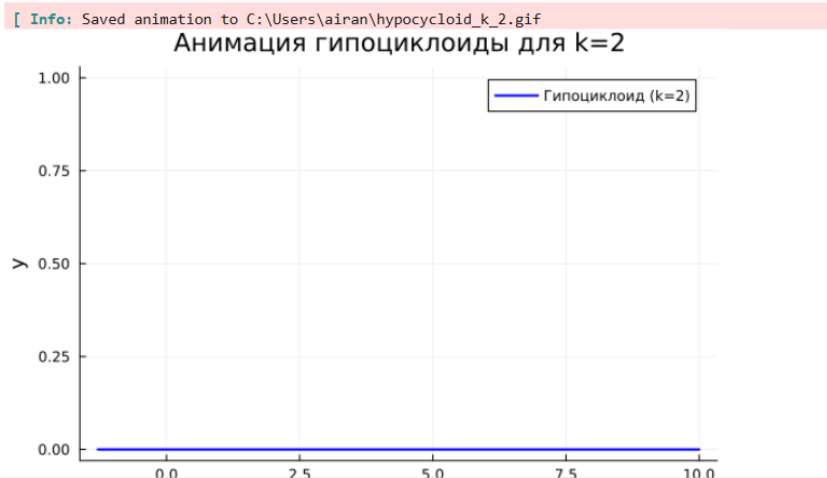


Рис. 2.13: 10 задание

3 Выводы

В результате выполнения лабораторной работы были изучены синтаксис языка Julia для построения графиков