

# Лабораторная работа 5

## Построение графиков

---

Ланцова Я. И.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Ланцова Яна Игоревна
- студентка
- Российский университет дружбы народов

## Цель работы

---

Основная цель работы – освоить синтаксис языка Julia для построения графиков.

## Задание

---

1. Используя JupyterLab, повторите примеры. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.
2. Выполните задания для самостоятельной работы.

## Выполнение лабораторной работы

---

# Выполнение лабораторной работы

```
[*:]: using Pkg  
Pkg.add("Plots")  
Pkg.add("PyPlot")  
Pkg.add("Plotly")  
Pkg.add("UnicodePlots")  
# подключение для использования Plots:  
using Plots  
  
Updating registry at `~/julia/registries/General.toml'  
Resolving package versions...  
 Installed DataFrames_jll └── v0.28.1+0  
 Installed GL_jll └── v0.1.8+0  
 Installed libfdk_aac_jll └── v0.73.10+0  
 Installed JpegTurbo_jll └── v3.1.3+0  
 Installed Libmount_jll └── v2.41.2+0  
 Installed LERC_jll └── v4.0.1+0  
 Installed LLVM_jll └── v8.0.1+0  
 Installed LoggingExtras └── v1.2.0  
 Installed Opus_jll └── v1.5.2+0  
 Installed ConcurrentUtilities └── v2.5.0  
 Installed Measures └── v0.3.3  
 Installed NonExcludableFolders └── v0.1  
 Installed Xorg_xcb_compat_jll └── v1.4.7+0  
 Installed Contour └── v0.6.3  
 Installed Grisu └── v1.0.2  
 Installed Xorg_xcb_util_image_jll └── v0.4.1+0  
 Installed Xorg_xcb_util_wm_jll └── v0.4.1+0  
 Installed Pkg └── v1.4.3  
 Installed RecipePipeline └── v0.6.12  
 Installed OpenSSL └── v1.6.0  
 Installed DelimitedFiles └── v1.9.1
```

Рис. 1: Основные пакеты для работы с графиками в Julia

# Выполнение лабораторной работы

Основные пакеты для работы с графиками в Julia

```
[3]:  
# задание функции:  
f(x) = (3x+2 + 6x - 9).exp.(-x/3)  
# генерирование массива значений x в диапазоне от -5 до 10 с шагом 0,1  
# ( шаг задан через указание длины массива):  
x = collect(range(-5,10,length=151))  
# генерирование массива значений y:  
y = f(x);  
  
[4]:  
# указывается, что для построения графика используется gr();  
gr()  
# задание опций при построении графика  
# (название кривой, подписи по осям, цвет графика):  
plot(x,y, title="A simple curve", xlabel="Variable x", ylabel="Variable y", color="blue")
```

A simple curve

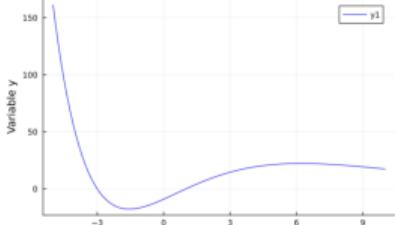


Рис. 2: Основные пакеты для работы с графиками в Julia

# Выполнение лабораторной работы



Рис. 3: Опции при построении графика

# Выполнение лабораторной работы

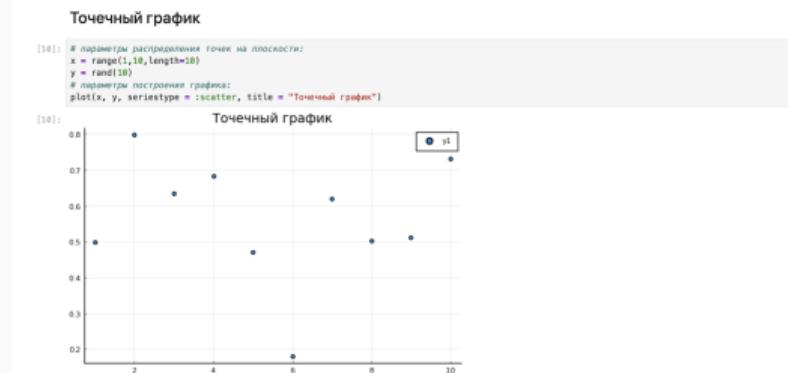


Рис. 4: Точечный график

# Выполнение лабораторной работы

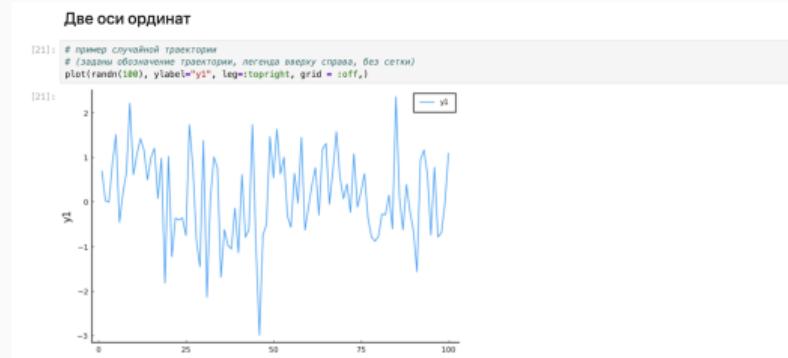


Рис. 5: Две оси ординат

# Выполнение лабораторной работы

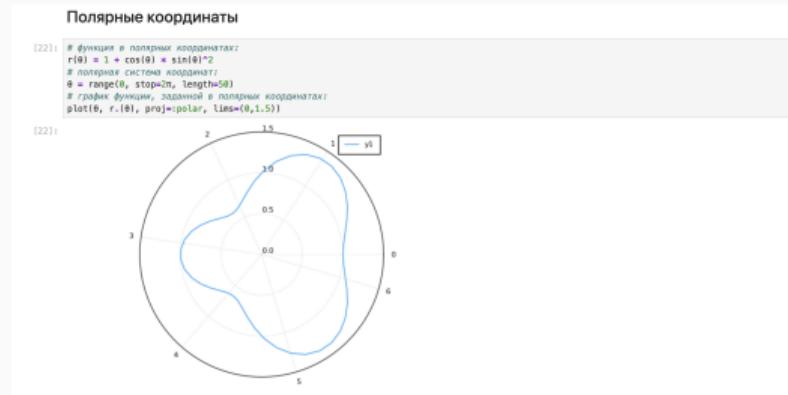


Рис. 6: Полярные координаты

# Выполнение лабораторной работы

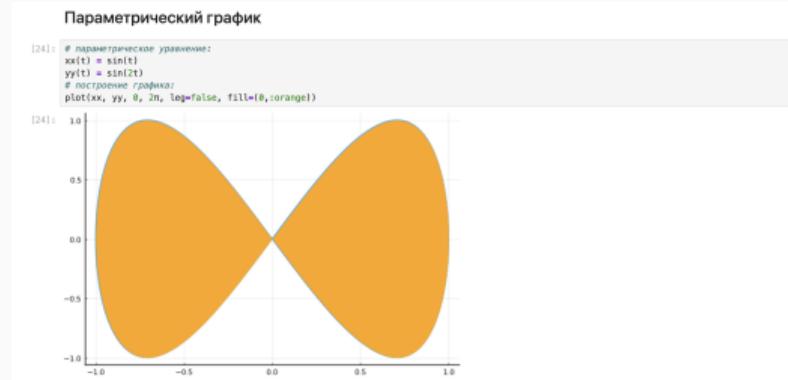


Рис. 7: Параметрический график

# Выполнение лабораторной работы

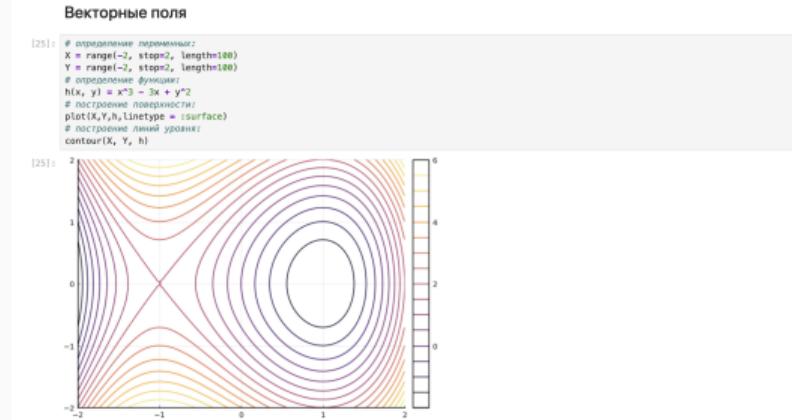


Рис. 8: Векторные поля

# Выполнение лабораторной работы

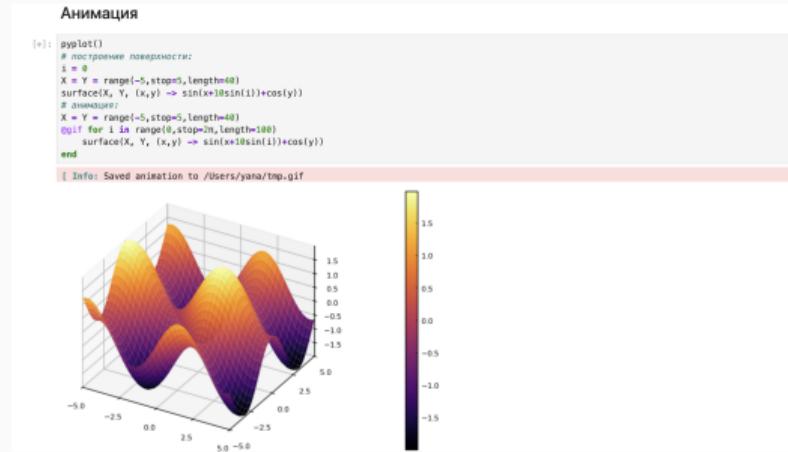


Рис. 9: Анимация

# Выполнение лабораторной работы

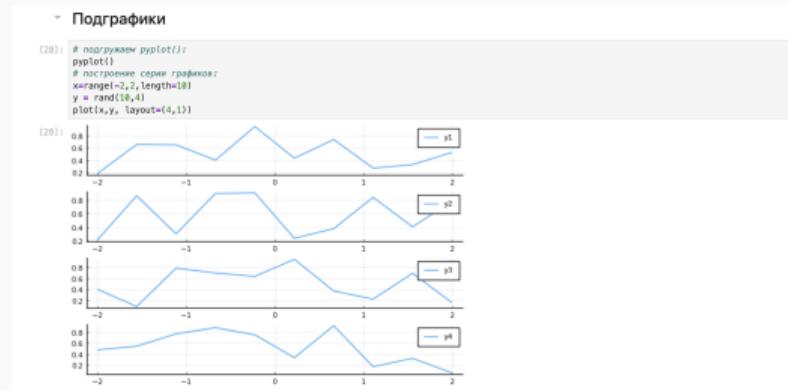


Рис. 10: Подграфики

## Задание 1

---

# Выполнение лабораторной работы

Задания для самостоятельного выполнения

1. Все возможные типы графиков функции  $y = \sin(x)$

```
[66]: using Plots  
[67]: x = collect(range(0, 2*pi, length=100))  
y = sin.(x);  
  
# Создаем некий XML для отображения всех графиков  
p1 = Plots.plot(x, y, title="Линейный график", xlabel="x", ylabel="sin(x)", legend=false)  
p2 = scatter(x, y, title="Точечный график", xlabel="x", ylabel="sin(x)", legend=false)  
p3 = Plots.plot(x, y, seriestype=:step, title="Ступенчатый график", xlabel="x", ylabel="sin(x)", legend=false)  
  
x_hist = range(0, 2*pi, length=20)  
y_hist = sin.(x_hist)  
p4 = histogram(y_hist, title="Гистограмма", xlabel="sin(x)", ylabel="Частота", legend=false)  
  
p5 = plot(x, y, seriestype=:path, fill=(0, :lightblue), title="График с областями", xlabel="x", ylabel="sin(x)", legend=false)  
p6 = plot(x, y, seriestype=:sticks, title="График с палочками", xlabel="x", ylabel="sin(x)", legend=false)  
  
Plots.plot(p1, p2, p3, p4, p5, p6, layout=(3,2), size=(1000, 700))
```

Рис. 11: Задание №1

# Выполнение лабораторной работы

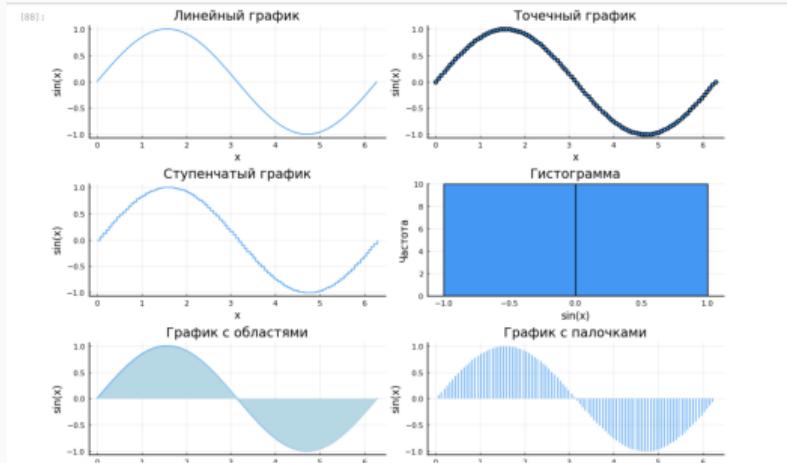


Рис. 12: Задание №1

## Задание 2

---

# Выполнение лабораторной работы

## 2. Все возможные типы оформления линий

```
[93]: line_styles = [:solid, :dash, :dot, :dashdot, :auto]
line_colors = [:blue, :red, :green, :orange, :purple]
line_widths = [1, 2, 3, 4, 5]
marker_shapes = [:circle, :rect, :diamond, :triangle, :star5]

[94]: plots_list = []
for i in 1:5
    p = plot(x, y,
              line_color=line_colors[i],
              linewidth=line_widths[i],
              markershape=marker_shapes[i],
              markersize=4,
              markercolor=marker_colors[i],
              title="Line: $(line_styles[i])",
              xlabel="x",
              ylabel="y",
              legend=false)
    push!(plots_list, p)
end
```

Рис. 13: Задание №2

# Выполнение лабораторной работы

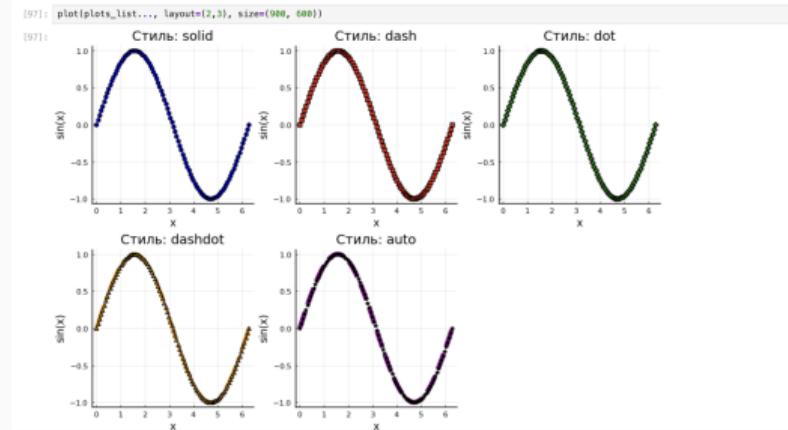


Рис. 14: Задание №2

## Задание 3

---

# Выполнение лабораторной работы

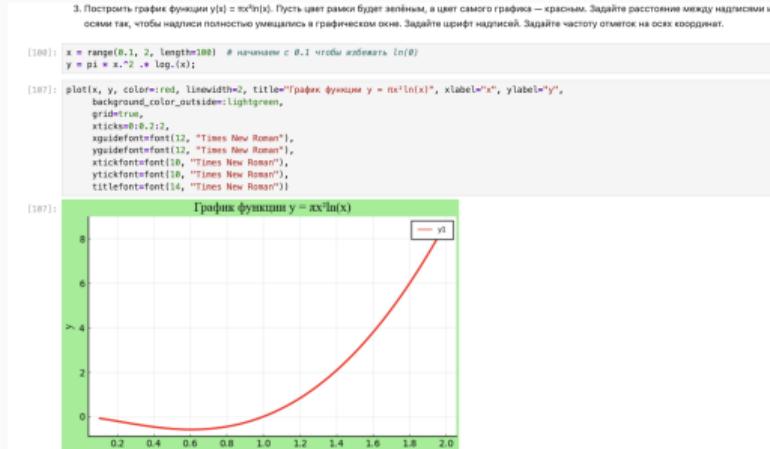


Рис. 15: Задание №3

## Задание 4

---

# Выполнение лабораторной работы

4. Задайте вектор  $x = [-2, -1, 0, 1, 2]$ . В одном графическом окне (в 4-х подокнах) изобразите графически по точкам х значения функции  $y(x) = x^3 - 3x$  в виде: точек, линий, линий и точек, кривой

```
[110]: x = [-2, -1, 0, 1, 2]
y = x.^3 - 3*x

[114]: # Создаем 4 разных типа графиков
p1 = scatter(x, y, title="Точкиный график", xlabel="x", ylabel="y", legend=False, markersize=6)
p2 = plot(x, y, title="Линейный график", xlabel="x", ylabel="y", legend=False, linewidth=2)
p3 = plot(x, y, markers=circle, markersize=6, title="Линия с точками", xlabel="x", ylabel="y", legend=False)
p4 = plot(x, y, seriestype=:path, linewidth=2, title="Кривая", xlabel="x", ylabel="y", legend=False);
```

Рис. 16: Задание №4

# Выполнение лабораторной работы

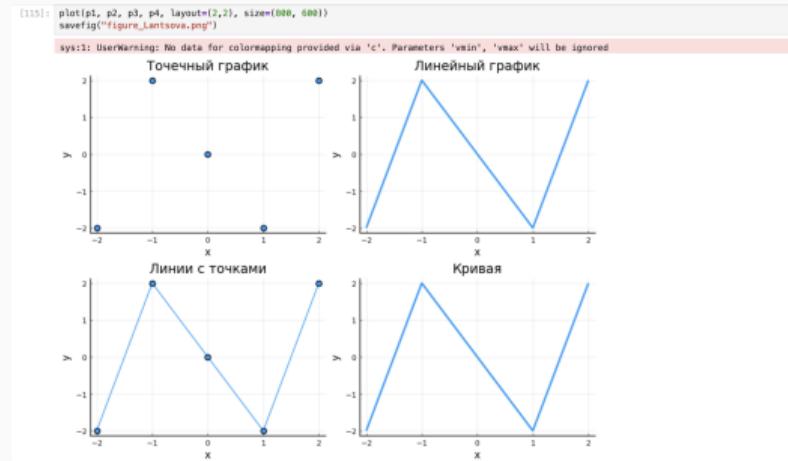


Рис. 17: Задание №4

## Задание 5

---

# Выполнение лабораторной работы

5. Задайте вектор  $x = (3, 3.1, 3.2, \dots, 6)$ . Постройте графики функций 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.

```
[118]: x = range(3, 6, length=100)
y1 = pi * x
y2 = exp(x) * cos(x);

[120]: p1 = plot(x, y1, linewidth=2, color='blue')
p2 = plot(x, y2, linewidth=2, color='red')
plot(p1, p2, layout=[1,2], size=(600, 400), legend=True, grid=True)
```

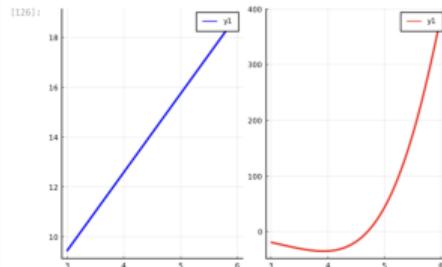


Рис. 18: Задание №5

# Выполнение лабораторной работы



Рис. 19: Задание №5

## Задание 6

---

# Выполнение лабораторной работы

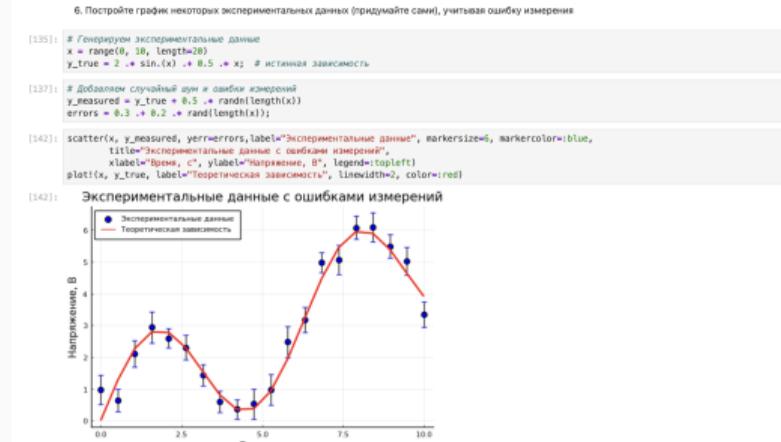


Рис. 20: Задание №6

## Задание 7

---

# Выполнение лабораторной работы

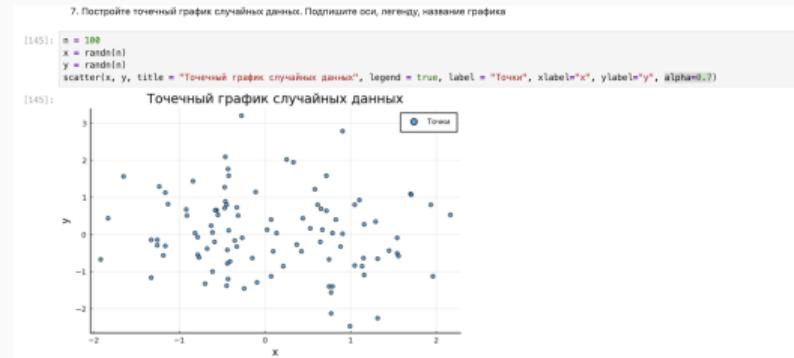


Рис. 21: Задание №7

## Задание 8

---

# Выполнение лабораторной работы

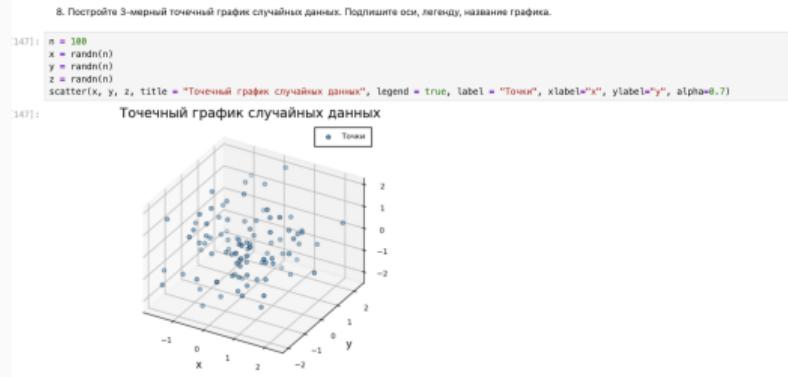


Рис. 22: Задание №8

## Задание 9

---

# Выполнение лабораторной работы

9. Создайте анимацию с построением синусоиды. То есть вы строите последовательность графиков синусоиды, постепенно увеличивая значение аргумента. После скончните их в анимации.

```
[1]: # Создаем анимацию синусоиды
anim = @animate for phase in range(0, 2π, length=100)
    x = range(0, 2π, length=100)
    y = sin.(x .+ phase)

    plot(x, y, linewidth=3, color=:blue, title="Анимация синусоиды: фаза = $(round(phase, digits=2))",
        xlabel="x",
        ylabel="sin(x + φ)",
        legend=false,
        ylim=(-1.5, 1.5),
        grid=true)
end;

[149]: gif(anim, "sine_animation.gif", fps=15)
```

[149]: **Info: Saved animation to /Users/yana/sine\_animation.gif**

АНИМАЦИЯ СИНУСОИДЫ: фаза = 5.08



Рис. 23: Задание №9

## Задание 10

---

# Выполнение лабораторной работы

## 10. Анимированная гипоциклоида

```
[*]: function plot_hypocycloid(r, k, n_frames=100)
    θ = range(0, 2π, length=n_frames)

    anim = @animate for i in 1:n_frames
        t = θ[1:i]

        # Координаты гипоциклоиды
        x = r * (k-1) * cos.(t) + r * cos.((k-1) * t)
        y = r * (k-1) * sin.(t) - r * sin.((k-1) * t)

        # Большая окружность
        X_big = r * k * cos.(θ)
        Y_big = r * k * sin.(θ)

        plot(X_big, Y_big, color=:blue, linewidth=1, label="Большая окружность",
              aspect_ratio=:equal,
              xlim=(-r*k-1, r*k+1),
              ylim=(-r*k-1, r*k+1))

        plot!(x, y, color=:red, linewidth=2, label="Гипоциклоида (k=$k)")

        title!("Гипоциклоида: k = $k")
    end

    return anim
end

# Целые значения k
anim1 = plot_hypocycloid(1, 3)
anim2 = plot_hypocycloid(1, 5)

# Рациональные значения k
anim3 = plot_hypocycloid(1, 2.5)
anim4 = plot_hypocycloid(1, 3.5)
```

# Выполнение лабораторной работы

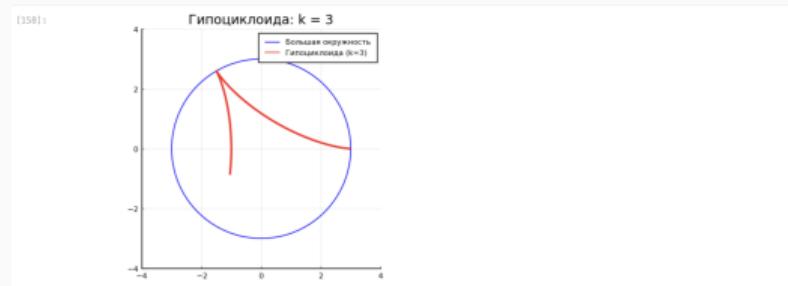


Рис. 24: Задание №10

# Выполнение лабораторной работы

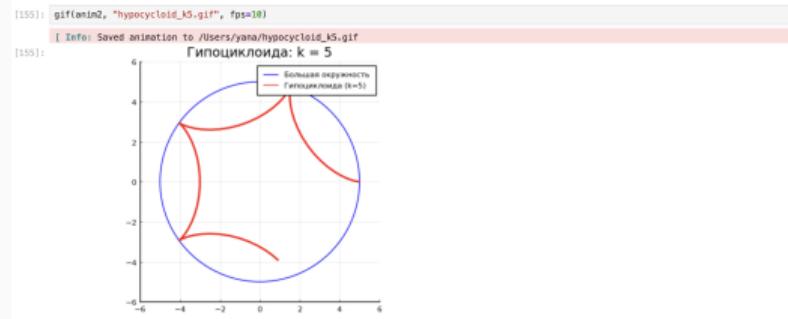


Рис. 25: Задание №10

# Выполнение лабораторной работы

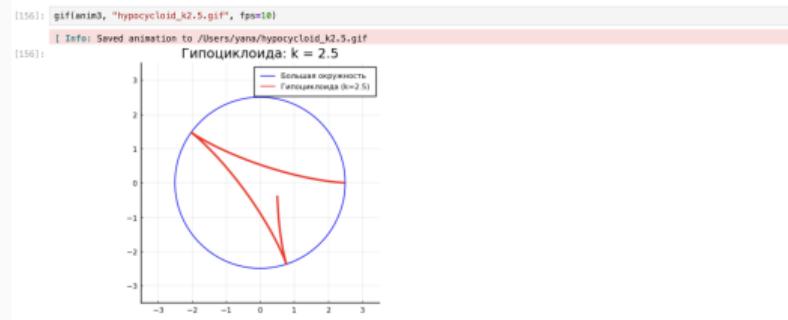


Рис. 26: Задание №10

# Выполнение лабораторной работы

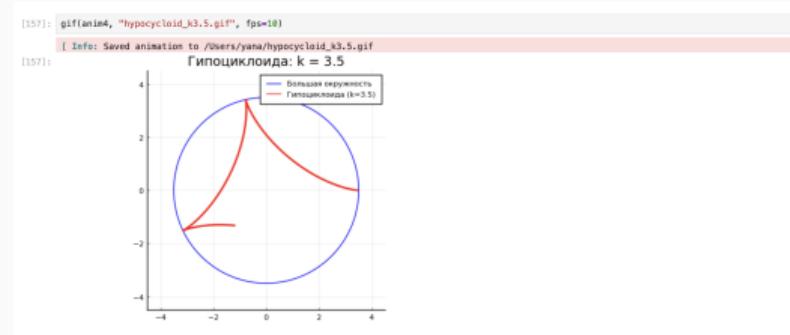


Рис. 27: Задание №10

## Задание 11

---

# Выполнение лабораторной работы

```
11. Анимированная эпиниклоида

[*]: function plot_epicycloid(r, k, n_frames=100)
    theta = range(0, 2pi, length(n_frames))
    anim =animate for i in 1:n_frames
        t = theta[i]
        # Координаты эпиниклоиды
        x = r * (k+1) * cos(t) - r * cos((k+1) * t)
        y = r * (k+1) * sin(t) - r * sin((k+1) * t)

        # Больших окружности
        X_big = r * k * cos(theta)
        Y_big = r * k * sin(theta)

        plot(X_big, Y_big, color=blue, linewidth=1, label="Небесная окружность", aspect_ratio=equal,
              xlim=(-r*(k+2), r*(k+2)),
              ylim=(-r*(k+2), r*(k+2)))

        plot(x, y, color=green, linewidth=2, label="Эпиниклоида (k=$k)")
        title!("Эпиниклоида: k = $k")
    end
    return anim
end

# Целые значения k
anim1 = plot_epicycloid(1, 2)
anim2 = plot_epicycloid(1, 4)

# Рациональные значения k
anim3 = plot_epicycloid(1, 1.5)
anim4 = plot_epicycloid(1, 2.5)
```

Рис. 28: Задание №11

# Выполнение лабораторной работы

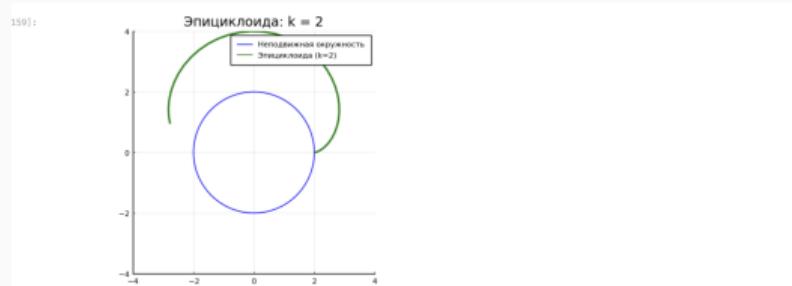


Рис. 29: Задание №11

# Выполнение лабораторной работы

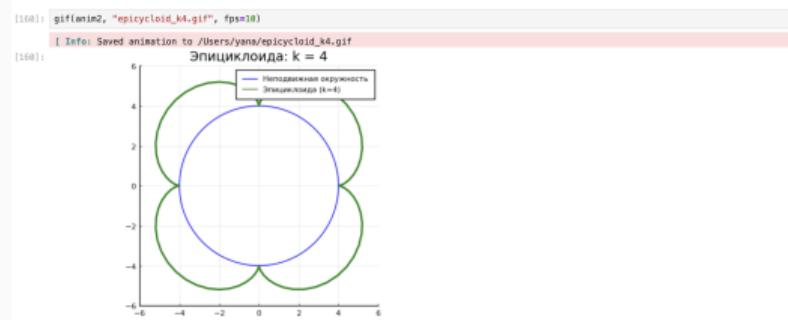


Рис. 30: Задание №11

# Выполнение лабораторной работы

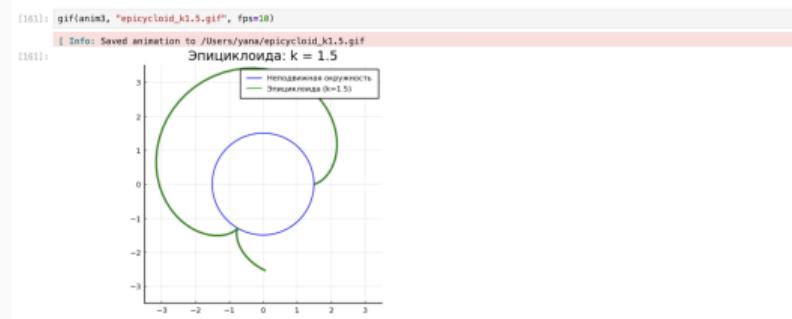


Рис. 31: Задание №11

# Выполнение лабораторной работы

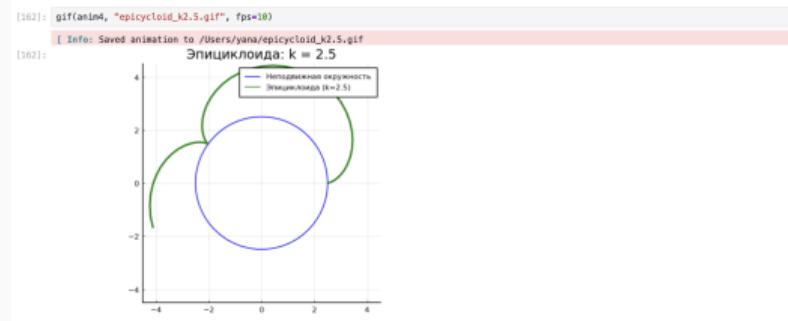


Рис. 32: Задание №11

## Выводы

---

## Выводы

В результате выполнения данной лабораторной работы я освоила синтаксис языка Julia для построения графиков.