Лабораторная работа № 5

Построение графиков

Шияпова Дарина Илдаровна

Содержание

Список иллюстраций

1 Цель работы

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

2 Задание

1. Используя JupyterLab, повторите примерыи. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.

2. Выполните задания для самостоятельной работы.

3 Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычисления. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia.

4 Выполнение лабораторной работы

Выполним примеры из лабораторной работы для знакомства с пакетами по отрисовки графиков и их функциями (рис. 4.1-4.19).

```
using Pkg
Pkg.add("Plots")
Pkg.add("PyPlot")
Pkg.add("Plotly")
Pkg.add("UnicodePlots")
# подключаем для использования Plots:
using Plots
```

Показать скрытые выходные данные

```
# задание функции:
f(x) = (3x.^2 + 6x .- 9).*exp.(-0.3x)
```

```
f (generic function with 1 method)
```

```
# 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))
```

Показать скрытые выходные данные

```
# генерирование массива значений y:
y = f(x)
```

Показать скрытые выходные данные

```
# указывается, что для построения графика используется gr():
gr()
```

```
Plots.GRBackend()
```

```
# 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")
```

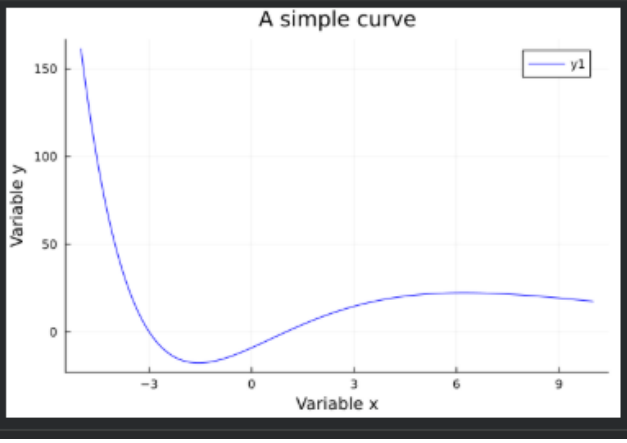


Рис. 4.1: Основные пакеты для работы с графиками в Julia

```
# задание опций при построении графика
# (название кривой, подписи по осям, цвет графика):
plot(x,y,
title="Простая кривая",
xlabel="Переменная x",
ylabel="Переменная y",
color="blue")
```

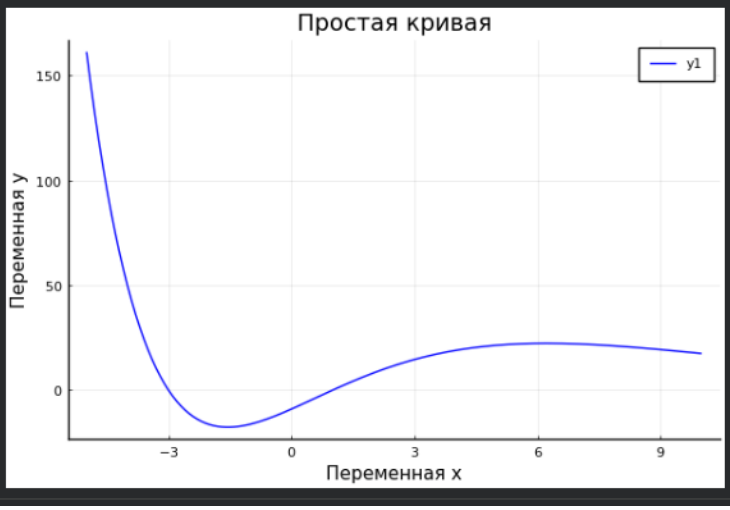


Рис. 4.2: Основные пакеты для работы с графиками в Julia

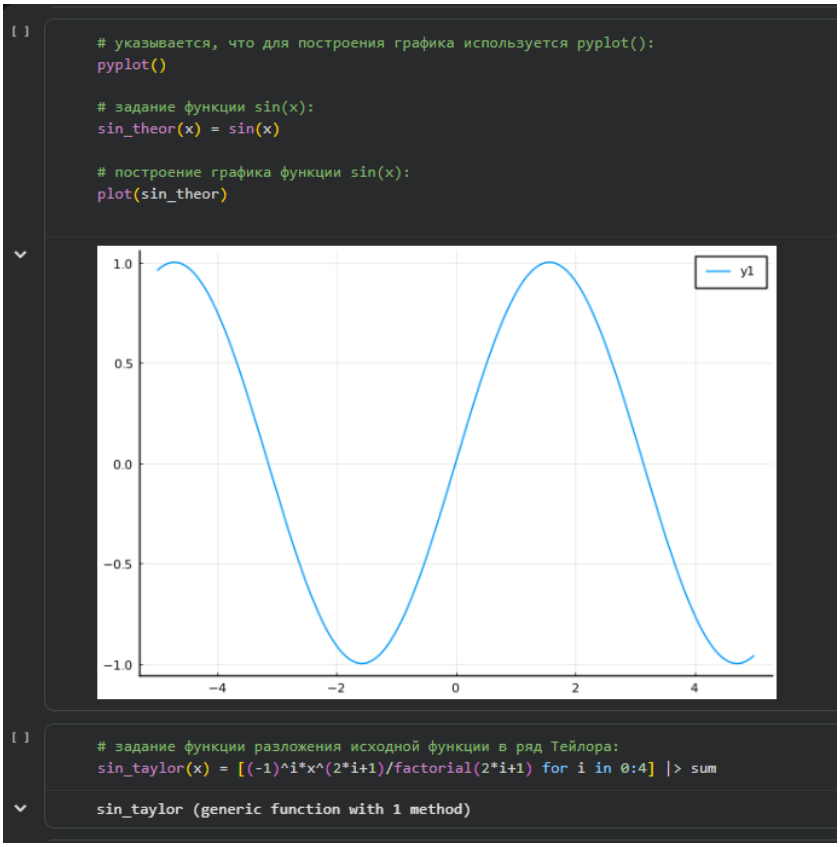
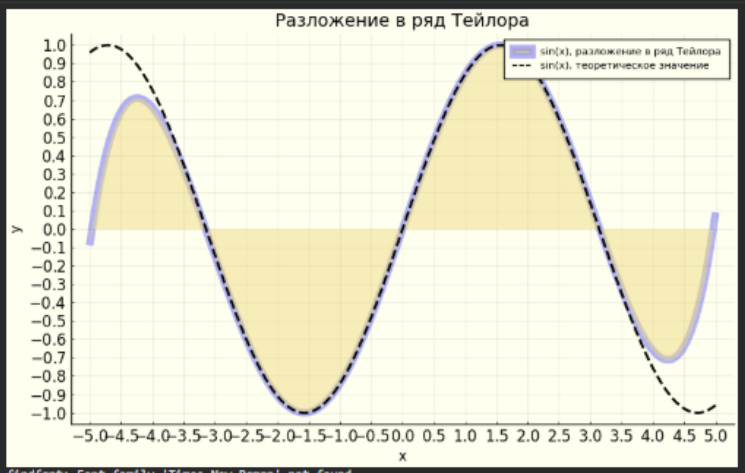


Рис. 4.3: Опции при построении графика

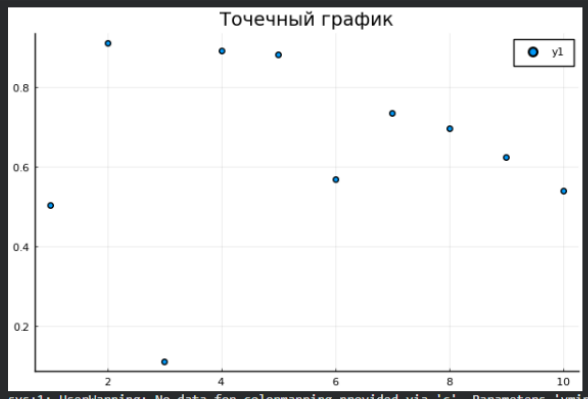
```
plot(
# функция sin(x):
sin_taylor,
# подпись в легенде, цвет и тип линии:
label = "sin(x), разложение в ряд Тейлора",
line=(:blue, 0.3, 6, :solid),
# размер графика:
size=(800, 500),
# параметры отображения значений по осям
xticks = (-5:0.5:5),
yticks = (-1:0.1:1),
xtickfont = font(12, "Times New Roman"),
ytickfont = font(12, "Times New Roman"),
# подписи по осям:
ylabel = "y",
xlabel = "x",
# название графика:
title = "Разложение в ряд Тейлора",
# поворот значений, заданный по оси x:
xrotation = rad2deg(pi/4),
# заливка области графика цветом:
fillrange = 0,
fillalpha = 0.5,
fillcolor = :lightgoldenrod,
# задание цвета фона:
background_color = :ivory
)
plot!(
# функция sin_theor:
sin_theor,
# подпись в легенде, цвет и тип линии:
label = "sin(x), теоретическое значение",
line=(:black, 1.0, 2, :dash))
```




```
findfont: Font family 'Times New Roman' not found.
findfont: Font family ['Times New Roman'] not found. Falling back to DejaVu Sans.
```

Рис. 4.4: Опции при построении графика

```
# параметры распределения точек на плоскости:
x = range(1,10,length=10)
y = rand(10)
# параметры построения графика:
plot(x, y,
seriestype = :scatter,
title = "Точечный график"
)
```




```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.5: Точечный график

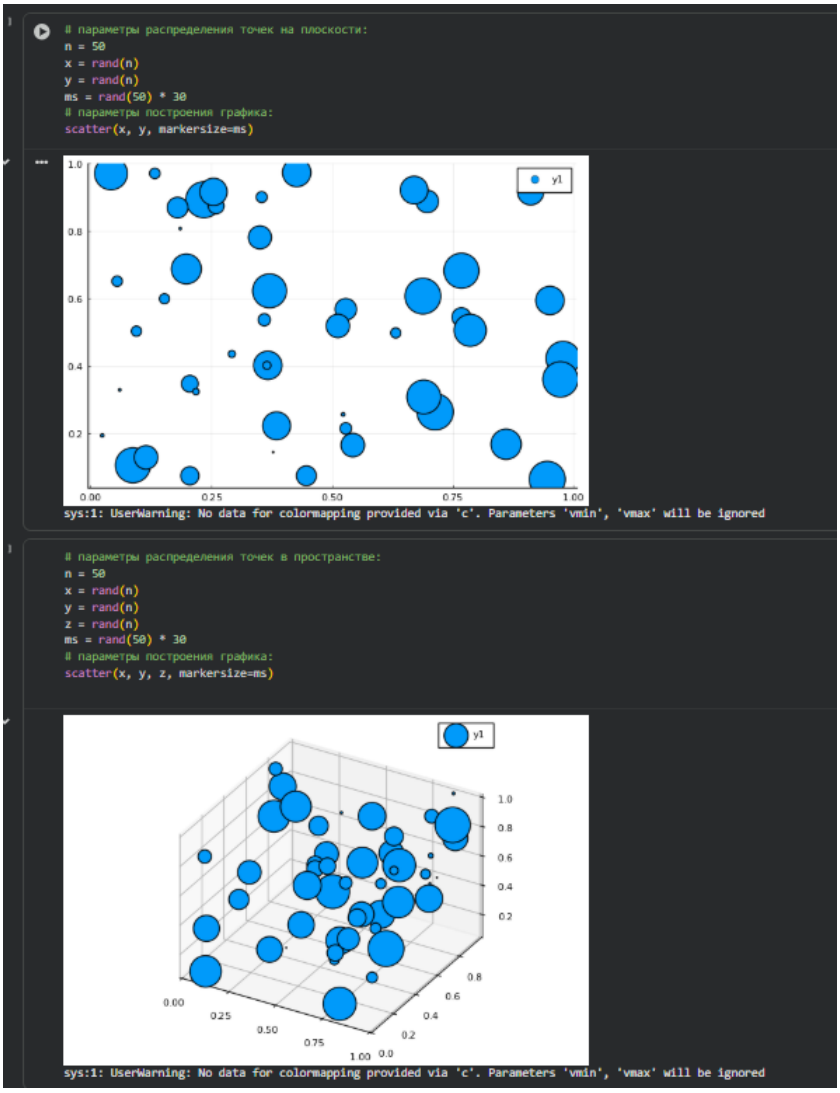
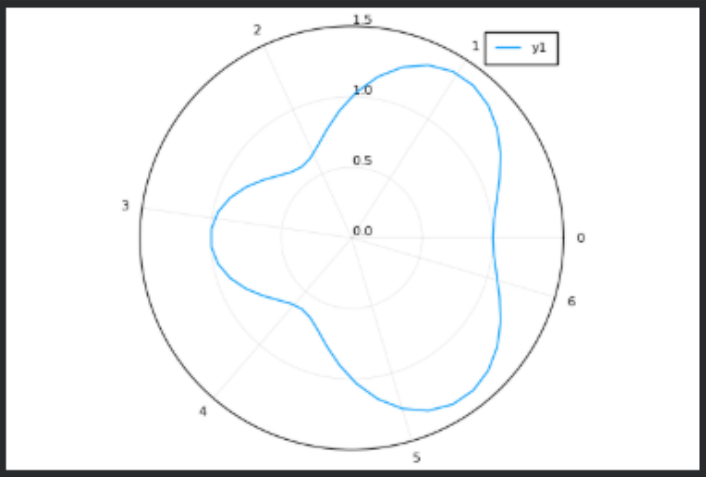


Рис. 4.6: Точечный график

Рис. 4.7: Аппроксимация данных

```
# функция в полярных координатах:
r(θ) = 1 + cos(θ) * sin(θ)^2
# полярная система координат:
θ = range(0, stop=2π, length=50)
# график функции, заданной в полярных координатах:
plot(θ, r.(θ),
proj=:polar,
lims=(0,1.5)
)
```




```
# параметрическое уравнение:
x1(t) = sin(t)
y1(t) = sin(2t)
# построение графика:
plot(x1, y1, 0, 2π, leg=false, fill=(0,:orange))
```

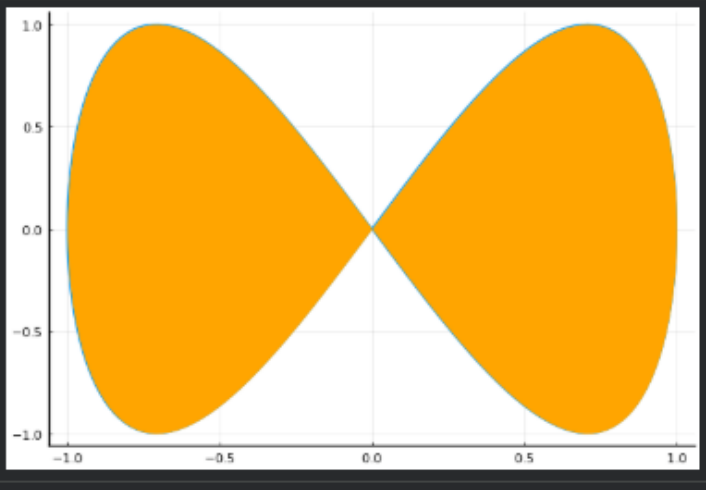
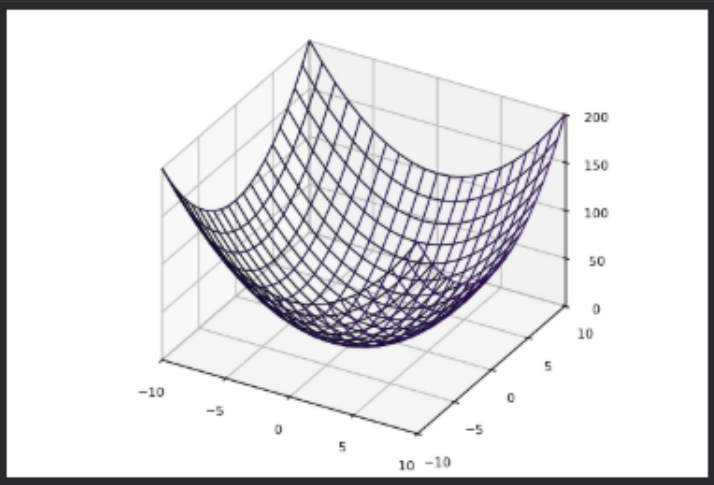


Рис. 4.8: Полярные координаты

```
# построение графика поверхности:
f(x,y) = x^2 + y^2
x = -10:10
y = x
surface(x, y, f)
#Также можно воспользоваться функцией plot() с заданными параметрами (рис. 5.18):
# построение графика поверхности:
f(x,y) = x^2 + y^2
x = -10:10
y = x
plot(x, y, f,
linetype=:wireframe
)
```




```
f(x,y) = x^2 + y^2
x = -10:0.1:10
y = x
plot(x, y, f,
linetype = :surface
)
```

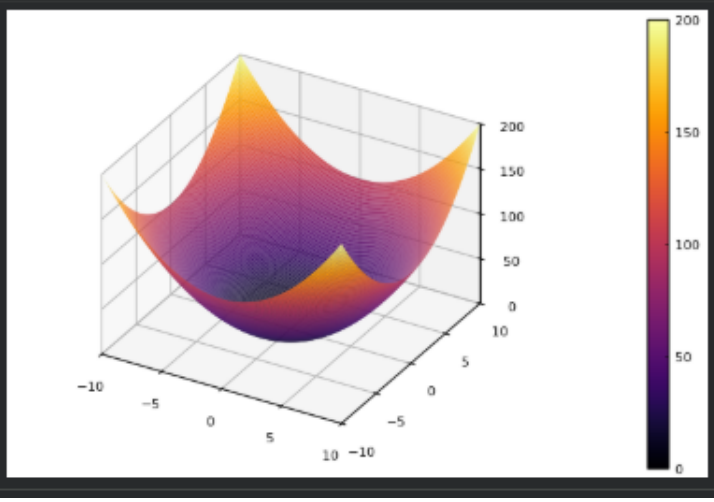
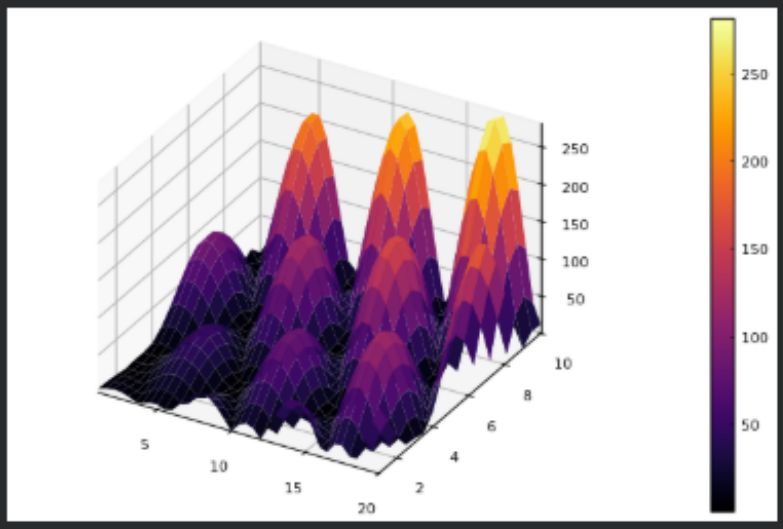
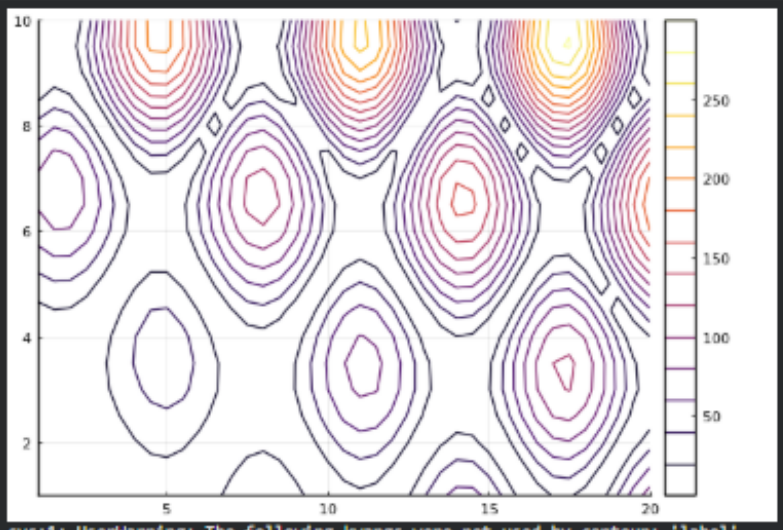


Рис. 4.9: График поверхности

```
x = 1:0.5:20
y = 1:0.5:10
g(x, y) = (3x + y ^ 2) * abs(sin(x) + cos(y))
plot(x,y,g,
linetype = :surface,
)
```



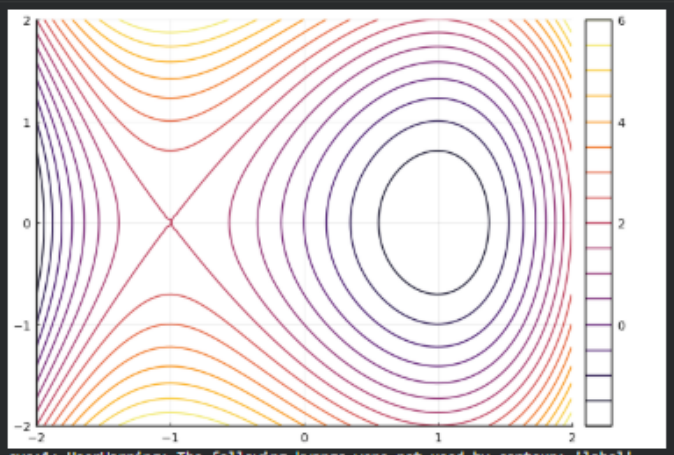

```
contour(x, y, g)
```




```
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
```

Рис. 4.10: Линии уровня

```
# определение переменных:
X = range(-2, stop=2, length=100)
Y = range(-2, stop=2, length=100)
# определение функции:
h(x, y) = x^3 - 3x + y^2
# построение поверхности:
plot(X,Y,h,
linetype = :surface
)
# построение линий уровня:
contour(X, Y, h)
```




```
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
```

```
# градиент:
x = range(-2, stop=2, length=12)
y = range(-2, stop=2, length=12)
# производная от исходной функции:
dh(x, y) = [3x^2 - 3; 2y] / 25
# построение векторного поля:
quiver!(x, y', quiver=dh, c=:blue)
# коррекция области видимости графика:
xlims!(-2, 2)
ylims!(-2, 2)
```

```
Показать скрытые выходные данные
```

Рис. 4.11: Векторные поля

```
# построение поверхности:
i = 0
X = Y = range(-5,stop=5,length=40)
surface(X, Y, (x,y) -> sin(x+10sin(i))+cos(y))
#Добавляем анимацию (рис. 5.29):
# анимация:
X = Y = range(-5,stop=5,length=40)
@gif for i in range(0,stop=2π,length=100)
surface(X, Y, (x,y) -> sin(x+10sin(i))+cos(y))
end
```

```
[ Info: Saved animation to /content/tmp.gif
```

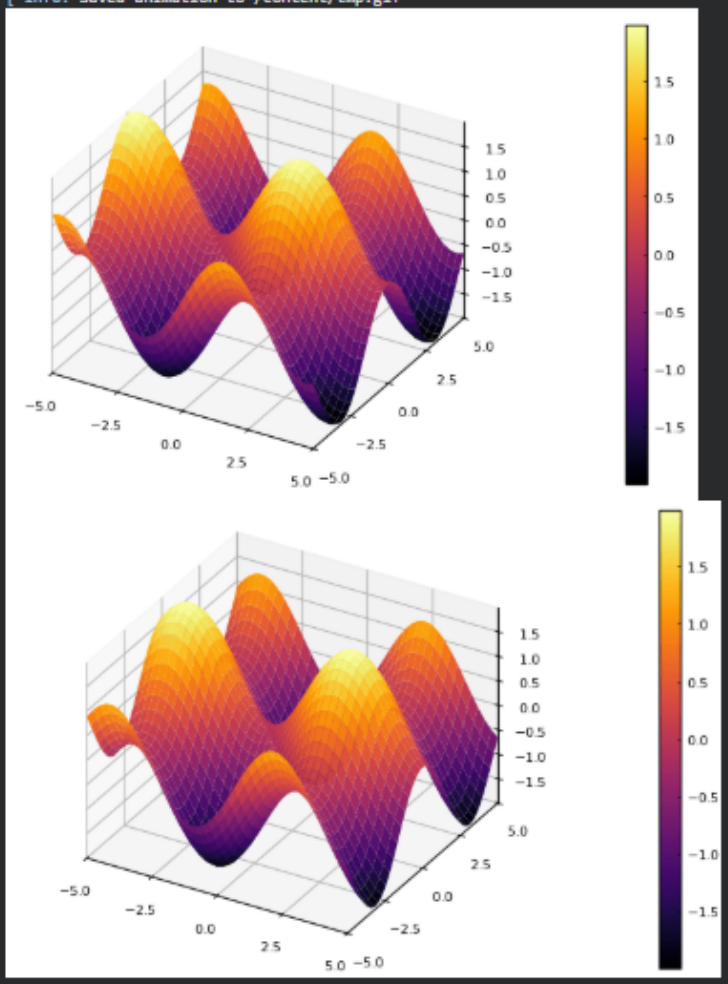
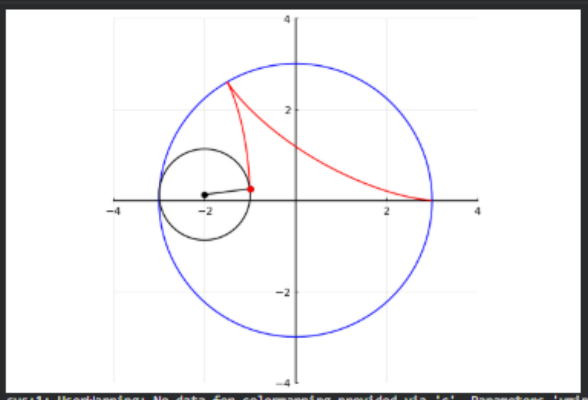


Рис. 4.12: Анимация

```
# радиус малой окружности:
r_ = 1
# коэффициент для построения большой окружности:
k = 3
# число отсчётов:
n = 100
# Затем зададим массивы необходимых значений:
# массив значений угла θ:
# theta from 0 to 2pi ( + a little extra)
θ = collect(0:2*π/100:2*π+2*π/100)
# массивы значений координат:
X = r_*k*cos.(θ)
Y = r_*k*sin.(θ)
# Построим оси координат:
# зададём оси координат:
plt=plot(5,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1, legend=false, framestyle=:origin)
# большая окружность:
plot!(plt, X,Y, c=:blue, legend=false)

i = 50
t = θ[1:i]
# гипоциклоида:
x = r_*(k-1)*cos.(t) + r_*cos.((k-1)*t)
y = r_*(k-1)*sin.(t) - r_*sin.((k-1)*t)
plot!(x,y, c=:red)

# малая окружность:
xc = r_*(k-1)*cos(t[end]) .+ r_*cos.(θ)
yc = r_*(k-1)*sin(t[end]) .+ r_*sin.(θ)
plot!(xc,yc,c=:black)

# радиус малой окружности:
xl = transpose([r_*(k-1)*cos(t[end]) x[end]])
yl = transpose([r_*(k-1)*sin(t[end]) y[end]])
Plots.plot!(xl,yl,markershape=:circle,markersize=4,c=:black)
Plots.scatter!([x[end]],[y[end]],c=:red, markerstrokecolor=:red)
```

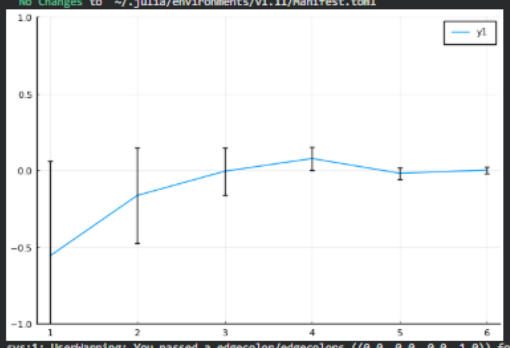



```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.13: Гипоциклоида

```
# подключение пакета Statistics:
import Pkg
Pkg.add("Statistics")
using Statistics
# Зададим массив значений:
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
# Затем сгенерируем массив ошибок (отклонений от исходных значений):
n = 10
y = [mean(sd*randn(n)) for sd in sds]
errs = 1.96 * sds / sqrt(n)
# Построим график исходных значений (рис. 5.35):
plot(y,
ylims = (-1,1),
)
# Построим график отклонений от исходных значений (рис. 5.36):
plot(y,
ylims = (-1,1),
err = errs
)
```

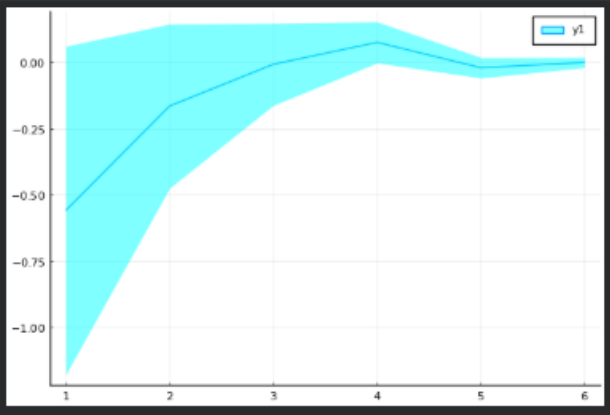
```
  Resolving package versions...
 No Changes to `~/.julia/environments/v1.11/Project.toml`
 No Changes to `~/.julia/environments/v1.11/Manifest.toml`
```




```
sys:1: UserWarning: You passed a edgecolor/edgecolors ((0.0, 0.0, 0.0, 1.0)) for an unfilled marker ('_').  Matplotlib is ig
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.14: Errorbars

```
plot(y, 1:length(y),
xerr = errs,
marker = stroke(3,:orange)
)
#Заполним область цветом (рис. 5.38):
plot(y,
ribbon=errs,
fill=:cyan
)
```




```
n = 10
x = [(rand()+1) .* randn(n) .+ 2i for i in 1:5]
y = [(rand()+1) .* randn(n) .+ i for i in 1:5]
f(v) = 1.96std(v) / sqrt(n)
xerr = map(f, x)
yerr = map(f, y)
x = map(mean, x)
y = map(mean, y)
plot(x, y,
xerr = xerr,
yerr = yerr,
marker = stroke(2, :orange)
)
```

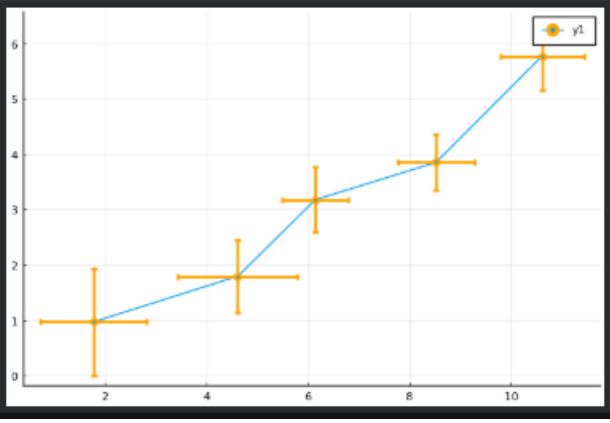
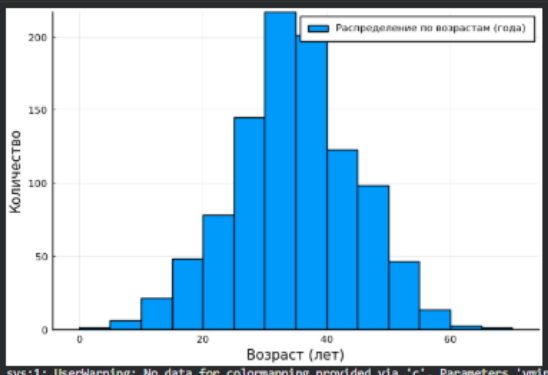


Рис. 4.15: Errorbars

```
pyplot()

Plots.PyPlotBackend()
```

```
# Задаём массив случайных чисел:
ages = rand(15:55,1000)
# Строим гистограмму (рис. 5.41):
histogram(ages)
# Задаём нормальное распределение и строим гистограмму (рис. 5.42):
d=Normal(35.0,10.0)
ages = rand(d,1000)
histogram(
ages,
label="Распределение по возрастам (года)",
xlabel = "Возраст (лет)",
ylabel= "Количество"
)
```

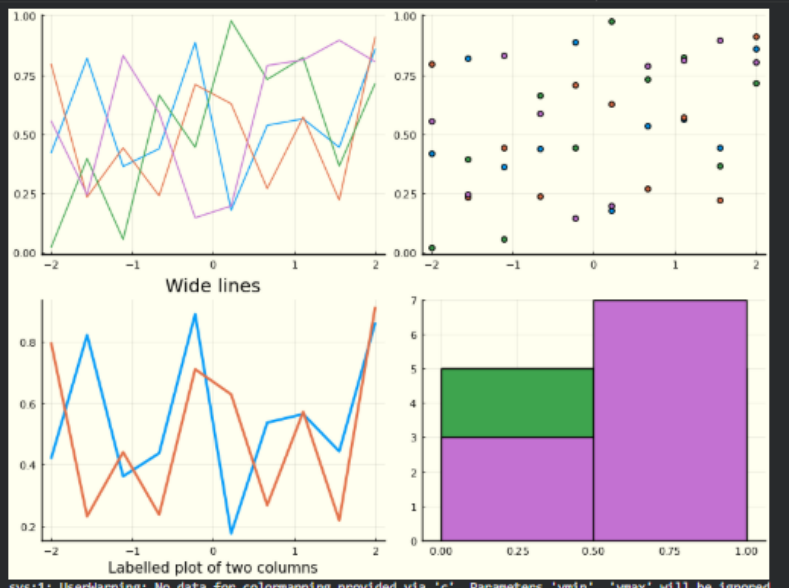



```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.16: Использование пакета Distributions

Рис. 4.17: Подграфики

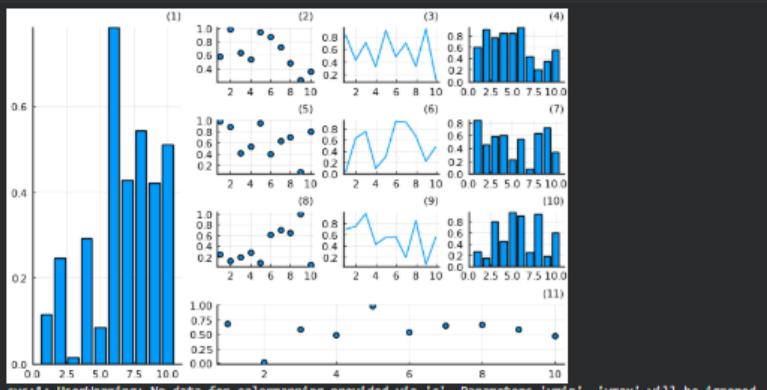
```
# график в виде линий:
p1 = plot(x,y)
# график в виде точек:
p2 = scatter(x,y)
# график в виде линий с оформлением:
p3 = plot(x,y[:,1:2],xlabel="Labelled plot of two columns",lw=2,title="Wide lines")
# 4 гистограммы:
p4 = histogram(x,y)
plot(
p1,p2,p3,p4,
layout=(2,2),
legend=false,
size=(800,600),
background_color = :ivory
)
```




```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.18: Подграфики

```
l = @layout [ a{0.3w} [grid(3,3)
b{0.2h} ]]
plot(
rand(10,11),
layout = l, legend = false, seriestype = [:bar :scatter :path],
title = ["($i)" for j = 1:1, i=1:11], titleloc = :right, titlefont = font(8)
)
```



```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Рис. 4.19: Подграфики

4.1 Задания для самостоятельного выполнения

Выполним задания (рис. 4.20-4.27).

```
# Подключение необходимых пакетов
pyplot()

# Задание функции и диапазона
x = range(0, stop=2π, length=100)
y = sin.(x)

# Создание макета для графиков
p1 = plot(x, y, title="График 1: Обычный график", label="sin(x)",
        xlabel="x", ylabel="sin(x)", legend=:topright)

p2 = scatter(x, y, title="График 2: Точечный график",
             label="sin(x)", xlabel="x", ylabel="sin(x)")

p3 = histogram(y, title="График 3: Гистограмма значений",
               xlabel="sin(x)", ylabel="Частота")

p4 = stephist(y, title="График 4: Ступенчатая гистограмма",
              xlabel="sin(x)", ylabel="Частота")

p5 = bar(x, y, title="График 5: Столбчатая диаграмма",
         xlabel="x", ylabel="sin(x)")

# Создаем двумерную сетку для контурного графика
X = Y = range(0, stop=2π, length=100)
# Формируем матрицы координат
xx = repeat(X, 1, length(Y))
yy = repeat(Y', length(X), 1)
# Создаем двумерную функцию
Z = sin.(xx) .* sin.(yy)

p6 = contour(xx, yy, Z, title="График 6: Контурный график",
             xlabel="x", ylabel="y", color=:viridis,
             levels=20, colorbar=true)

# Объединение всех графиков в одно окно
plot(p1, p2, p3, p4, p5, p6, layout=(3,2), size=(1000,800),
     background_color=:ivory, titlefontsize=10)
```

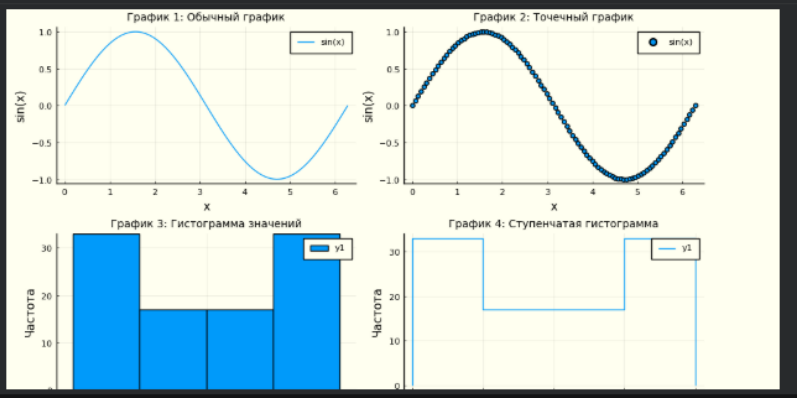


Рис. 4.20: Задание №1

```
# График 5: толстая линия
p5 = plot(x, y,
    title="Толстая линия",
    label="sin(x)",
    linewidth=3,
    xlabel="x",
    ylabel="sin(x)")

# График 6: цветная линия
p6 = plot(x, y,
    title="Цветная линия",
    label="sin(x)",
    color=:red,
    xlabel="x",
    ylabel="sin(x)")

# Объединение всех графиков в одно окно
plot(p1, p2, p3, p4, p5, p6,
    layout=(3,2),
    size=(1000,800),
    background_color=:ivory,
    titlefontsize=10,
    legend=:topright)
```

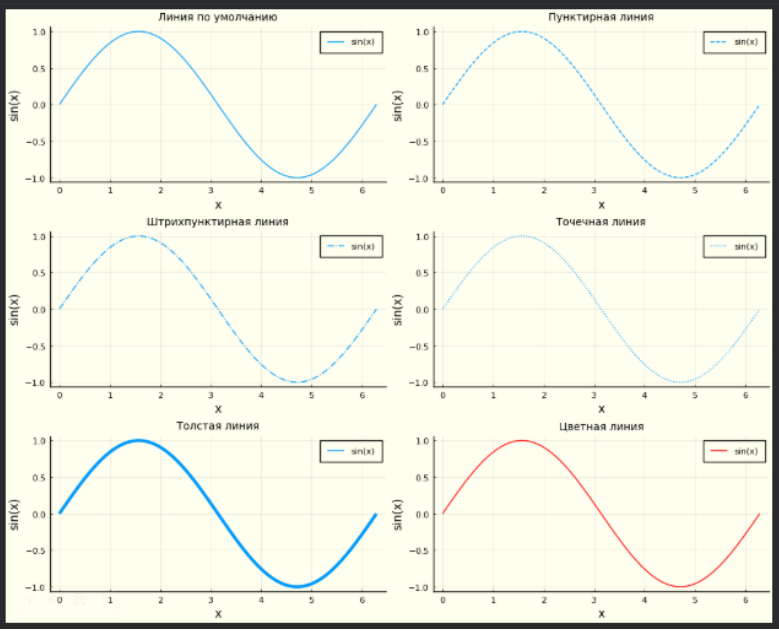


Рис. 4.21: Задание №2

```
# 3
pyplot()

# Определение функции
f(x) = π * x^2 * log(x)

# Задание диапазона значений x
x = range(1e-3, stop=10, length=1000)  # начинаем с малого положительного числа вместо 0

# Построение графика
plot(x, f.(x),
     title="График функции y = πx²ln(x)",
     xlabel="x",
     ylabel="y",
     color=:red,
     framestyle=:box,  # включение рамки
     frame_style=:grid,
     border_width=2,
     border_color=:green,
     tick_direction=:out,
     xtick=1:10,       # установка шага по оси x
     ytick=range(-10, stop=100, step=10),  # установка шага по оси y
     fontfamily="Times New Roman",
     tick_fontsize=10,
     guidefontsize=12,  # размер шрифта подписей осей
     titlefontsize=14,
     margin=5Plots.mm,  # отступы для надписей
     legend=false
)

# Дополнительные настройки для лучшей читаемости
xlims!(0, 10)
ylims!(-10, 100)
```

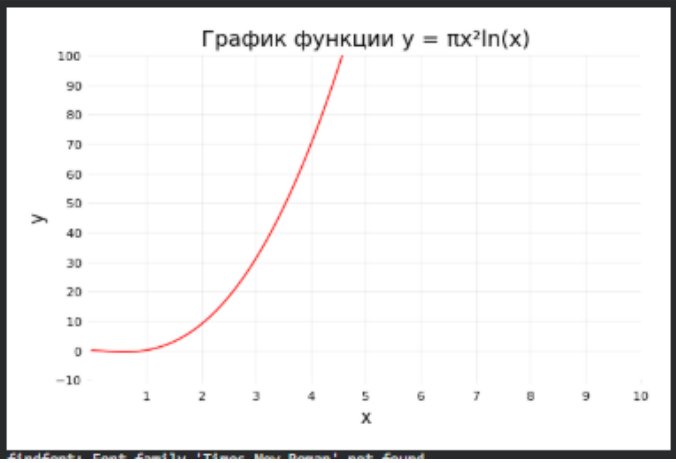
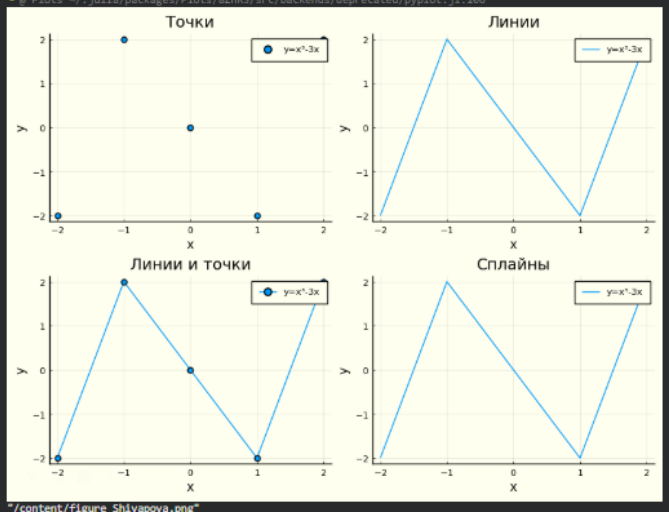


Рис. 4.22: Задание №3

```
    ylabel="y")

p3 = plot(x, y_.(x),
    title="Линии и точки",
    label="y=x³-3x",
    markershape=:circle,
    markersize=5,
    xlabel="x",
    ylabel="y")

p4 = plot(x, y_.(x),
    title="Сплайны",
    label="y=x³-3x",
    linestyle=:spline,
    xlabel="x",
    ylabel="y")

# Объединяем все графики в одно окно
plot(p1, p2, p3, p4,
    layout=(2,2),
    size=(800,600),
    legend=:topright,
    background_color=:ivory)

# Сохраняем результат
savefig("figure_Shiyapova.png")
```

```
┌ Warning: linestyle spline is unsupported with Plots.PyPlotBackend(). Choose from: [:auto, :dash, :dashdot, :dot, :solid]
└ @ Plots ~/.julia/packages/Plots/8ZnR3/src/args.jl:1579
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
┌ Warning: Unknown linestyle spline
└ @ Plots ~/.julia/packages/Plots/8ZnR3/src/backends/deprecated/pyplot.jl:106
┌ Warning: Unknown linestyle spline
└ @ Plots ~/.julia/packages/Plots/8ZnR3/src/backends/deprecated/pyplot.jl:106
```




```
"/content/figure_Shiyapova.png"
```

Рис. 4.23: Задание №4

```julia
# Создаем вектор x с шагом 0.1
x = 3:0.1:6

# Первый способ: общий график
p1 = plot(x, func1.(x),
    label="y₁ = πx",
    linestyle=:solid,
    color=:blue,
    linewidth=2,
    grid=true,
    legend=:topleft)

plot!(x, func2.(x),
    label="y₂ = e^cos(x)",
    linestyle=:dash,
    color=:red,
    linewidth=2,
    title="Графики на общей оси Y",
    xlabel="x",
    ylabel="y")

# Второй способ: две оси Y
p2 = plot(x, func1.(x),
    label="y₁ = πx",
    linestyle=:solid,
    color=:blue,
    linewidth=2,
    title="Графики с двумя осями Y",
    xlabel="x",
    ylabel="y₁",
    grid=true)

plot!(twinx(), x, func2.(x),
    label="y₂ = e^cos(x)",
    linestyle=:dash,
    color=:red,
    linewidth=2,
    ylabel="y₂",
    legend=:topright)

# Объединение графиков в одно окно
plot(p1, p2, layout=(1,2), size=(1000,400), background_color=:ivory)
```

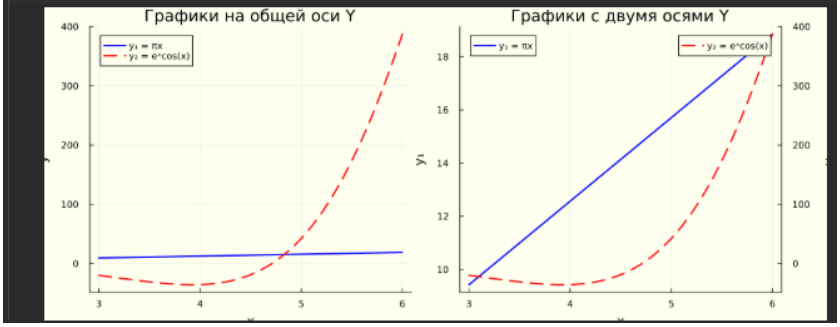


Рис. 4.24: Задание №5

```julia
using Plots
using Statistics
gr()  #

# Генерация экспериментальных данных
# Создаем массив значений x
x = 1:10

# Создаем массив средних значений y
y_mean = 2.0 .+ 0.5 .* x .+ 0.2 .* randn(length(x))

# Создаем массив стандартных отклонений
σ = 0.3 .* rand(length(x))  # случайная погрешность

# Построение графика с погрешностями
plot(x, y_mean,
     yerr = σ,  # указание погрешностей
     marker = :circle,  # тип маркера
     markersize = 5,    # размер маркера
     linestyle = :dash, # тип линии
     color = :blue,     # цвет графика
     title = "Экспериментальные данные с погрешностями",
     xlabel = "Независимая переменная",
     ylabel = "Зависимая переменная",
     legend = :topleft,
     grid = true,
     size = (800, 600),
     background_color = :ivory)

# Дополнительные настройки
xlims!(0, 11)  # установка границ по оси X
ylims!(0, 10)  # установка границ по оси Y
```

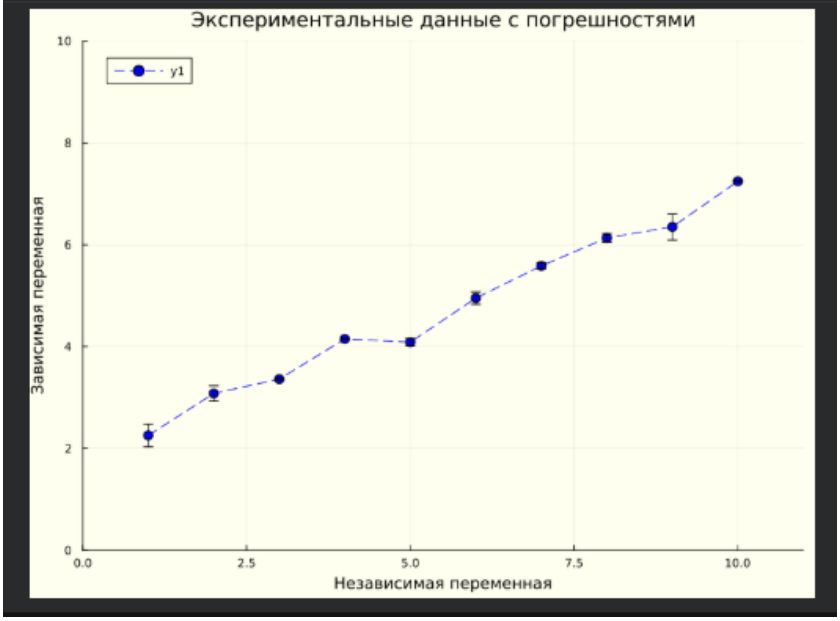


Рис. 4.25: Задание №6

```julia
# 7
using Plots
using Statistics
gr()  # или другой бэкенд

# Генерация случайных данных
n = 100  # количество точек
x = rand(n)  # случайные значения по оси X
y = rand(n)  # случайные значения по оси Y
z = rand(n)  # случайные значения размера маркера

# Построение точечного графика
scatter(x, y,
    markersize = z * 10,  # размер маркера зависит от z
    markercolor = :blue,  # цвет маркеров
    markerstrokewidth = 0.5,  # толщина обводки
    markerstrokecolor = :black,  # цвет обводки
    title = "Точечный график случайных данных",
    xlabel = "Ось X",
    ylabel = "Ось Y",
    legend = false,  # отключение легенды
    grid = true,  # включение сетки
    size = (800, 600),  # размер графика
    background_color = :ivory,  # цвет фона
    tick_direction = :out,  # направление меток наружу
    tickfont = font(12),  # размер шрифта меток
    guidefont = font(14)  # размер шрифта подписей осей
)

# Дополнительные настройки
xlims!(0, 1)  # установка границ по X
ylims!(0, 1)  # установка границ по Y
```

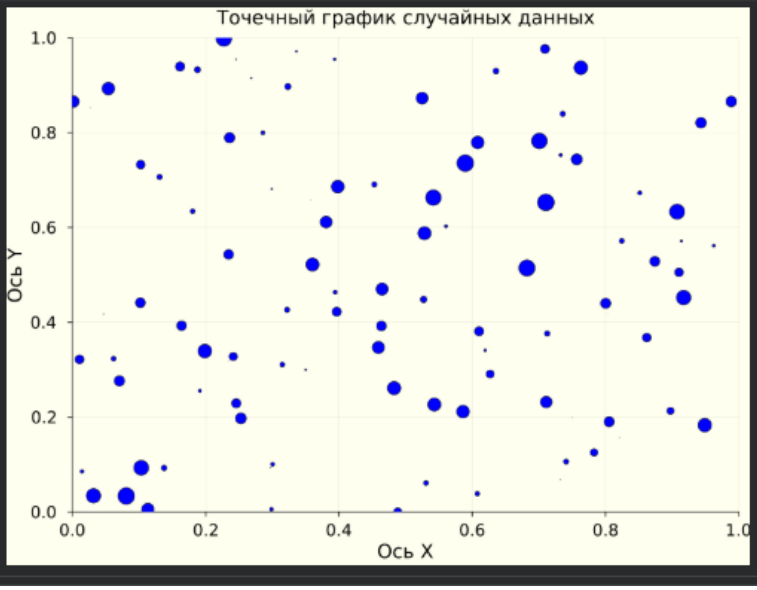


Рис. 4.26: Задание №7

```julia
# 8
using Plots
using Statistics
gr()  # или другой бэкенд

# Генерация случайных данных
n = 100  # количество точек
x = rand(n)  # случайные значения по оси X
y = rand(n)  # случайные значения по оси Y
z = rand(n)  # случайные значения по оси Z
s = rand(n) * 30  # размер маркеров

# Создание 3D точечного графика
scatter3d(x, y, z,
    markersize = s,  # размер маркеров
    markercolor = :blue,  # цвет маркеров
    markerstrokewidth = 0.5,  # толщина обводки
    markerstrokecolor = :black,  # цвет обводки
    title = "3D точечный график случайных данных",
    xlabel = "Ось X",
    ylabel = "Ось Y",
    zlabel = "Ось Z",
    legend = false,  # отключение легенды
    grid = true,  # включение сетки
    size = (800, 600),  # размер графика
    background_color = :ivory,  # цвет фона
    tick_direction = :out,  # направление меток наружу
    tickfont = font(12),  # размер шрифта меток
    guidefont = font(14)  # размер шрифта подписей осей
)

# Дополнительные настройки
xlims!(0, 1)  # установка границ по X
ylims!(0, 1)  # установка границ по Y
zlims!(0, 1)  # установка границ по Z
```

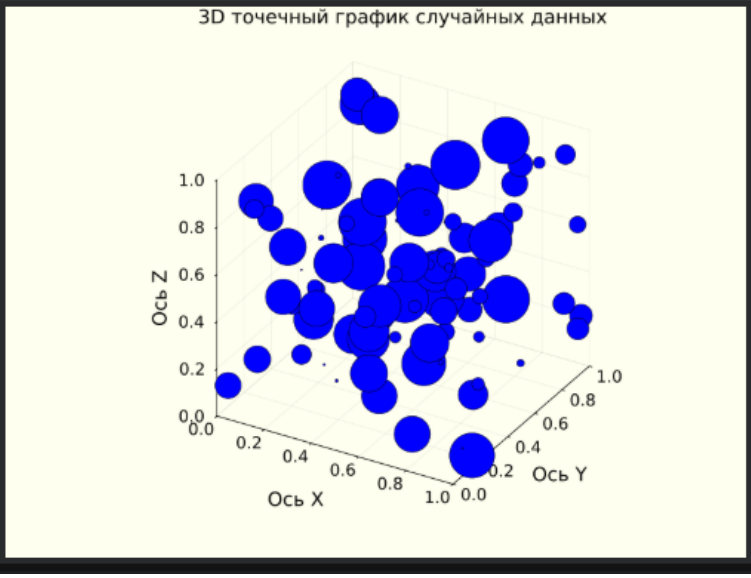


Рис. 4.27: Задание №8

```
# 9

pyplot()

# Создание анимации
anim = @animate for t in range(0, stop=2π, length=100)
    # Определяем функцию
    f(x) = sin(x + t)

    # Создаем диапазон значений x
    x = range(0, stop=2π, length=100)

    # Строим график
    plot(x, f.(x),
         title="Анимация синусоиды",
         xlabel="x",
         ylabel="sin(x)",
         legend=false,
         color=:blue,
         linewidth=2,
         grid=true,
         size=(800,400),
         background_color=:ivory,
         xlims=(0, 2π),
         ylims=(-1.5, 1.5),
         framestyle=:box)
end

# Сохранение анимации в файл
gif(anim, "sine_wave_animation.gif", fps=10)
```

```
[ Info: Saved animation to /content/sine_wave_animation.gif
```

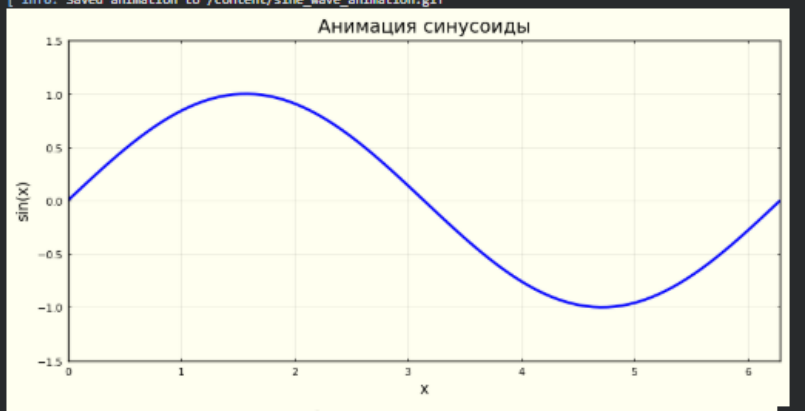


Рис. 4.28: Задание №9

```
    # Рассчитываем координаты для малой окружности (движущейся)
    xc = r_ * (k - 1) * cos(t[end]) .+ r_ .* cos.(θ)
    yc = r_ * (k - 1) * sin(t[end]) .+ r_ .* sin.(θ)

    # Создаём основной график
    plt = Plots.plot(;
        xlim=(-6, 6),
        ylim=(-6, 6),
        color=:red,
        aspect_ratio=1,
        legend=false,
        framestyle=:origin
    )

    # Добавляем большую окружность (синий цвет)
    Plots.plot!(plt, X, Y, color=:blue, legend=false)

    # Добавляем гипоциклоиду (красный цвет)
    Plots.plot!(x, y, color=:red)

    # Добавляем малую окружность (чёрный цвет)
    Plots.plot!(xc, yc, color=:black)

    # Отображаем траекторию движения точки (чёрные кружки)
    xl = [r_ * (k - 1) * cos(t[end]) x[end]]
    yl = [r_ * (k - 1) * sin(t[end]) y[end]]
    Plots.plot!(xl, yl, markershape=:circle, markersize=4, color=:black)

    # Выделяем последнюю точку траектории (красный кружок)
    Plots.scatter!([x[end]], [y[end]], color=:red, markerstrokecolor=:red)
end

# Сохраняем анимацию в файл GIF с частотой 20 кадров в секунду
gif(anim, "hypocycloid.gif", fps=20)
```

```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
[ Info: Saved animation to /content/hypocycloid.gif
```

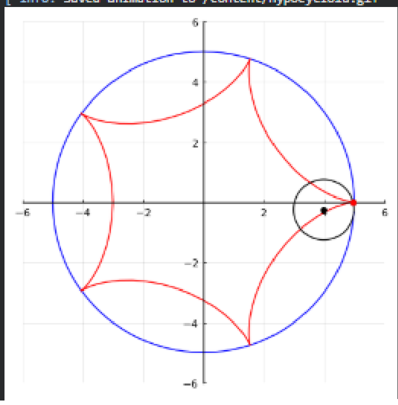


Рис. 4.29: Задание №10

```
    # Координаты для малой окружности (движущейся)
    xc = r_ * (k - 1) * cos(t[end]) .+ r_ .* cos.(θ)
    yc = r_ * (k - 1) * sin(t[end]) .+ r_ .* sin.(θ)

    # Создаём основной график
    plt = Plots.plot(;
        xlim=(-10, 10),
        ylim=(-10, 10),
        color=:red,
        aspect_ratio=1,
        legend=false,
        framestyle=:origin
    )

    # Добавляем большую окружность (синий цвет)
    Plots.plot!(plt, X, Y, color=:blue, legend=false)

    # Добавляем гипоциклоиду (красный цвет)
    Plots.plot!(x, y, color=:red)

    # Добавляем малую окружность (чёрный цвет)
    Plots.plot!(xc, yc, color=:black)

    # Отображаем траекторию движения точки (чёрные кружки)
    xl = transpose([r_ * (k - 1) * cos(t[end]) x[end]])
    yl = transpose([r_ * (k - 1) * sin(t[end]) y[end]])
    Plots.plot!(xl, yl, markershape=:circle, markersize=4, color=:black)

    # Выделяем последнюю точку траектории (красный кружок)
    Plots.scatter!([x[end]], [y[end]], color=:red, markerstrokecolor=:red)
end

# Сохраняем анимацию в файл GIF
gif(anim, "hypocycloid.gif", fps=20)
```

```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
[ Info: Saved animation to /content/hypocycloid.gif
```

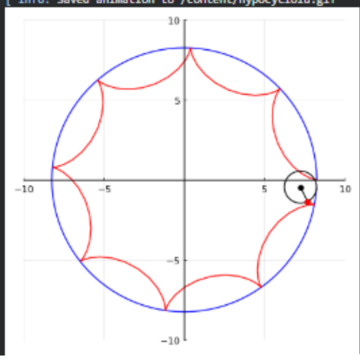


Рис. 4.30: Задание №10

```
    # Большая окружность
    X = r_ * k .* cos.(t)
    Y = r_ * k .* sin.(t)

    # Эпициклоида
    x = r_ * (k + 1) .* cos.(t) .- r_ .* cos.((k + 1) .* t)
    y = r_ * (k + 1) .* sin.(t) .- r_ .* sin.((k + 1) .* t)

    # Малая окружность
    xc = r_ * (k + 1) * cos(t[end]) .- r_ .* cos.(θ)
    yc = r_ * (k + 1) * sin(t[end]) .- r_ .* sin.(θ)

    # Построение графиков
    plt = Plots.plot(;
        xlim=(-7, 7),
        ylim=(-7, 7),
        color=:red,
        aspect_ratio=1,
        legend=false,
        framestyle=:origin
    )

    Plots.plot!(plt, X, Y, color=:blue, legend=false)  # Большая окружность
    Plots.plot!(x, y, color=:red)                       # Эпициклоида
    Plots.plot!(xc, yc, color=:black)                   # Малая окружность

    xl = transpose([r_ * (k + 1) * cos(t[end]) x[end]])
    yl = transpose([r_ * (k + 1) * sin(t[end]) y[end]])

    Plots.plot!(xl, yl, markershape=:circle, markersize=4, color=:black)
    Plots.scatter!([x[end]], [y[end]], color=:red, markerstrokecolor=:red)
end

# Сохранение анимации
gif(anim, "epicycloid_k_$k.gif", fps=20)
```

```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
[ Info: Saved animation to /content/epicycloid_k_3.gif
```

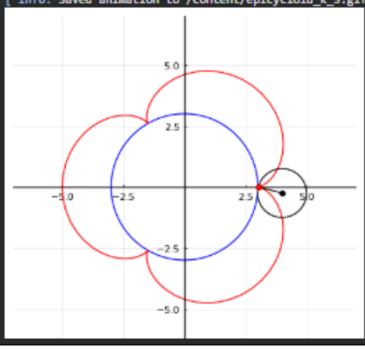


Рис. 4.31: Задание №11

```
    # Координаты для малой окружности (движущейся)
    xc = r_ * (k + 1) * cos(t[end]) .- r_ .* cos.(θ)
    yc = r_ * (k + 1) * sin(t[end]) .- r_ .* sin.(θ)

    # Создаём основной график
    plt = Plots.plot(;
        xlim=(-10, 10),
        ylim=(-10, 10),
        color=:red,
        aspect_ratio=1,
        legend=false,
        framestyle=:origin
    )

    # Добавляем большую окружность (синий цвет)
    Plots.plot!(plt, X, Y, color=:blue, legend=false)  # Большая окружность

    # Добавляем эпициклоиду (красный цвет)
    Plots.plot!(x, y, color=:red)                      # Эпициклоида

    # Добавляем малую окружность (чёрный цвет)
    Plots.plot!(xc, yc, color=:black)                  # Малая окружность

    # Отображаем траекторию движения точки (чёрные кружки)
    xl = transpose([r_ * (k + 1) * cos(t[end]) x[end]])
    yl = transpose([r_ * (k + 1) * sin(t[end]) y[end]])
    Plots.plot!(xl, yl, markershape=:circle, markersize=4, color=:black)

    # Выделяем последнюю точку траектории (красный кружок)
    Plots.scatter!([x[end]], [y[end]], color=:red, markerstrokecolor=:red)
end

# Сохраняем анимацию в файл GIF
gif(anim, "epicycloid.gif", fps=20)
```

```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
[ Info: Saved animation to /content/epicycloid.gif
```

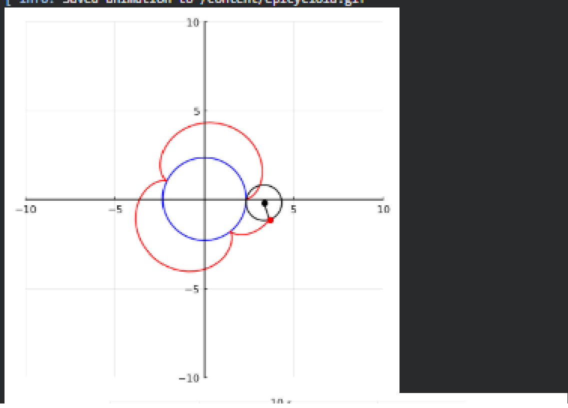


Рис. 4.32: Задание №11

5 Выводы

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

Список литературы