

Лабораторная работа № 5

Построение графиков

Шияпова Д.И.

08 ноября 2025

Российский университет дружбы народов, Москва, Россия

- Шияпова Дарина Илдаровна
- Студентка
- Российский университет дружбы народов
- 1132226458@pfur.ru



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

Выполнение лабораторной работы

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

Показать скрытые выходные данные

```
# задание функции:
f(x) = (2x.^2 + 6x - 9).*exp.(-0.1x)

f (generic function with 1 method)
```

```
# генерирование массива значений x в диапазоне от -5 до 10 с шагом 0,1
# (var задан через указание длины массива):
x = collect(range(-5,10,length=101))
```

Показать скрытые выходные данные

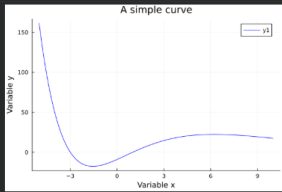
```
# генерирование массива значений y:
y = f(x)
```

Показать скрытые выходные данные

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

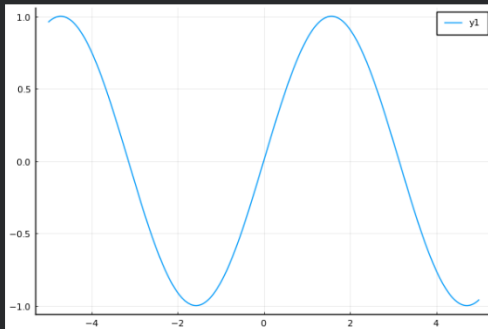
```
Plots.grBackend()
```

```
# задание опций при построении графика
# (название кризиса, подпись по осям, цвет графика):
plot(x,y,
title="A simple curve",
xlabel="Variable x",
ylabel="Variable y",
color="blue")
```



Выполнение лабораторной работы

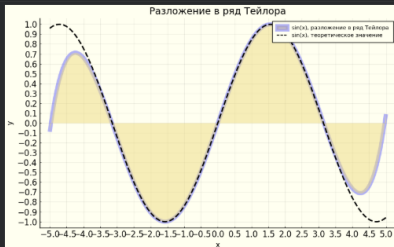
```
[ ]  
  
# указывается, что для построения графика используется pyplot():  
pyplot()  
  
# задание функции sin(x):  
sin_theor(x) = sin(x)  
  
# построение графика функции sin(x):  
plot(sin_theor)
```



```
[ ]  
  
# задание функции разложения исходной функции в ряд Тейлора:  
sin_taylor(x) = [((-1)**i*x**(2*i+1)/factorial(2*i+1) for i in 0:4] |> sum  
  
sin_taylor (generic function with 1 method)
```

Выполнение лабораторной работы

```
plot(  
# функции sin(x):  
sin_taylor,  
# подписи в легенде, цвет и тип линии:  
label = "sin(x), разложение в ряд Тейлора",  
lines=(blue, 0.3, 6, :solid),  
# размер графика:  
size=(800, 500),  
# параметры отображения значений по осям  
xticks = (-5.0:0.5),  
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, заданных 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!  
# функции sin_theor:  
sin_theor,  
# подписи в легенде, цвет и тип линии:  
label = "sin(x), теоретическое значение",  
lines=(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.

Выполнение лабораторной работы

```
# 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: Точечный график

Выполнение лабораторной работы

```
# параметры распределения точек на плоскости:
```

```
n = 50
```

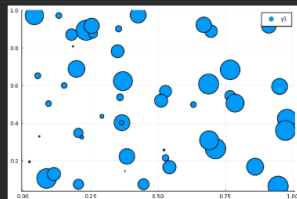
```
x = rand(n)
```

```
y = rand(n)
```

```
sz = rand(50) * 30
```

```
# параметры построения графика:
```

```
scatter(x, y, markersize=sz)
```



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

```
# параметры распределения точек в пространстве:
```

```
n = 50
```

```
x = rand(n)
```

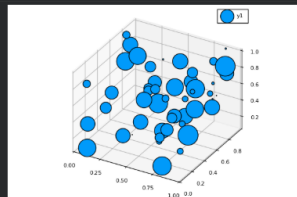
```
y = rand(n)
```

```
z = rand(n)
```

```
sz = rand(50) * 30
```

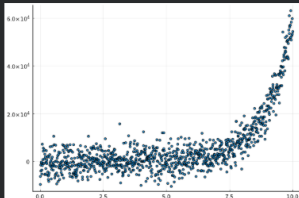
```
# параметры построения графика:
```

```
scatter(x, y, z, markersize=sz)
```



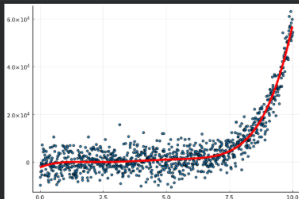
Выполнение лабораторной работы

```
# массив данных от 0 до 10 с шагом 0.01:  
x = collect(0:0.01:9.99)  
# экспоненциальная функция со случайным сдвигом значений:  
y = exp.(ones(1000)*x) + 4000*rand(1000)  
# построение графика:  
scatter(x,y,markersize=3,alpha=.8,legend=false)
```



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

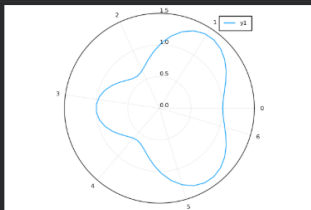
```
# определение массива для нахождения коэффициентов полинома:  
A = [ones(1000) x.^2 x.^3 x.^4 x.^5]  
# решение матричного уравнения:  
c = A\y  
# построение полинома:  
f1 = c[1]*ones(1000) + c[2]*x + c[3]*x.^2 + c[4]*x.^3 + c[5]*x.^4 + c[6]*x.^5  
# построение графика аппроксимирующей функции:  
plot(x,f1,linewidth=3, color='red')
```



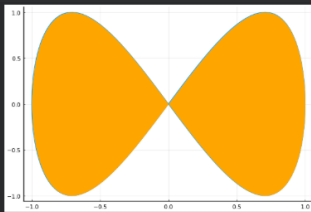
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored

Выполнение лабораторной работы

```
# функции 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)  
)
```

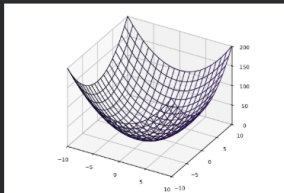


```
# параметрическое уравнение:  
x1(t) = sin(t)  
y1(t) = sin(2t)  
# построение графика:  
plot(x1, y1, θ, 2π, leg=false, fill=(0, :orange))
```

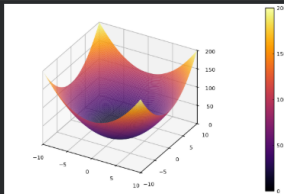


Выполнение лабораторной работы

```
# 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'  
)
```

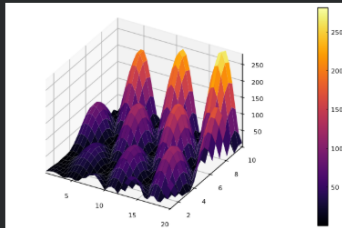


```
f(x,y) = x^2 + y^2  
x = -10:0.1:10  
y = x  
plot(x, y, f,  
linetype = 'surface'  
)
```

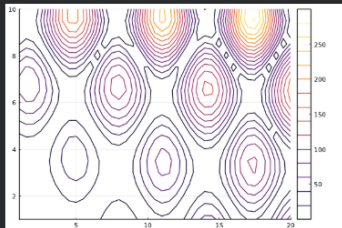


Выполнение лабораторной работы

```
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,  
      linestyle = :surface,  
      )
```



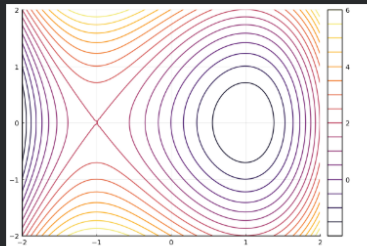
```
contour(x, y, g)
```



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

Выполнение лабораторной работы

```
# 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,  
      linestyle = :surface  
      )  
# построение линий уровня:  
contour(X, Y, h)
```



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

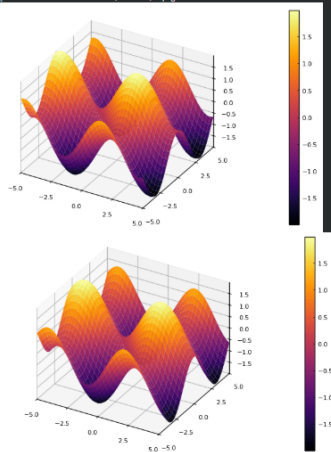
```
# градиент:  
x = range(-2, stop=2, length=12)  
y = range(-2, stop=2, length=12)  
# производная от исходной функции:  
dh(x, y) = [3x^2 - 3; 2y] / 25  
# построение векторного поля:  
quiver!(x, y', quiver=dh, c=:blue)  
# коррекция области видимости графика:  
xlims!(-2, 2)  
ylims!(-2, 2)
```

--- Показать скрытые выходные данные

Выполнение лабораторной работы

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

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



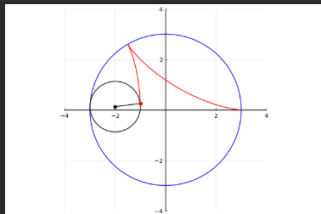
Выполнение лабораторной работы

```
# радиус малой окружности:
r_ = 1
# коэффициент для построения большой окружности:
k = 3
# число отсчётов:
n = 100
# Затем зададим массивы необходимых значений:
# массив значений угла  $\theta$ :
theta = from 0 to 2pi ( + a little extra)
 $\theta = \text{collect}(\theta:2*\pi/100:2*\pi+2*\pi/100)$ 
# массивы значений координат:
X = r_*k*cos.( $\theta$ )
Y = r_*k*sin.( $\theta$ )
# Построим оси координат:
# зададим оси координат:
plt=plot(t,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 =  $\theta[1:i]$ 
# гипотенуза:
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.( $\theta$ )
yc = r_*(k-1)*sin(t[end]) .+ r_*sin.( $\theta$ )
plot!(xc,yc,c=:black)

# радиус малой окружности:
x1 = transpose([r_*(k-1)*cos(t[end]) x[end]])
y1 = transpose([r_*(k-1)*sin(t[end]) y[end]])
Plots.plot!(x1,y1,markershape=:circle,markersize=4,c=:black)
Plots.scatter!([x[end]], [y[end]], c=:red, markerstrokecolor=:red)
```

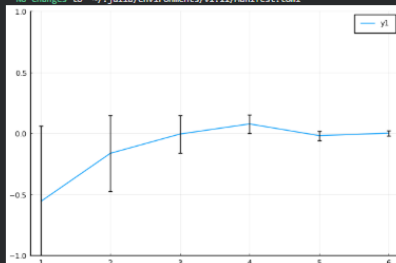


Выполнение лабораторной работы

```
# подключение пакета Statistics:
import Pkg
Pkg.add("Statistics")
using Statistics

# Заддим 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
     )
```

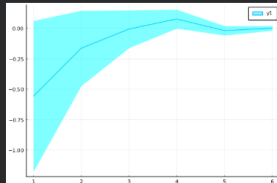
```
Resolving package versions...
No Changes to `~/julia/environments/v1.11/Project.toml`
No Changes to `~/julia/environments/v1.11/Manifest.toml`
```



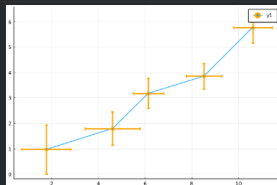
```
sys:1: UserWarning: You passed a edgecolor/edgecolors ((0.0, 0.0, 0.0, 1.0)) for an unfilled marker ('_'). Matplotlib is ig
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```


Выполнение лабораторной работы

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



```
n = 10  
x = [(rand()*1) .* randn(n) .* 21 for i in 1:5]  
y = [(rand()*1) .* randn(n) .* 1 for i in 1:5]  
f(v) = 1.96*std(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)  
 )
```

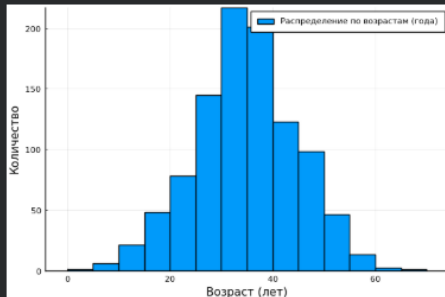


Выполнение лабораторной работы

```
pyplot()
```

```
Plots.PyPlotBackend()
```

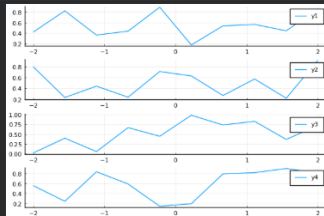
```
# Задаём массив случайных чисел:  
ages = rand(15:55,1000)  
# Строим гистограмму (рис. 5.41):  
histogram(ages)  
# Задаём нормальное распределение и строим гистограмму (рис. 5.42):  
d=Normal(35.0,10.0)  
ages = rand(d,1000)  
histogram(  
    ages,  
    label="Распределение по возрастам (года)",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



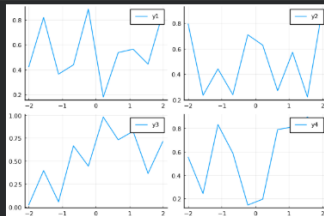
```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

Выполнение лабораторной работы

```
# подгружаем pyplot():  
pyplot()  
# построение серии графиков:  
x = range(-2, 2, length=10)  
y = rand(10, 4)  
plot(x, y,  
      layout=(4, 1)  
)
```

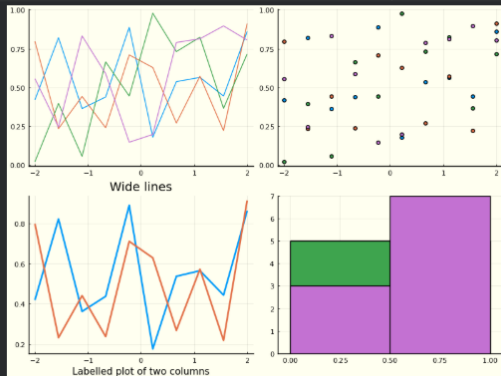


```
plot(x, y,  
      layout=4  
)
```



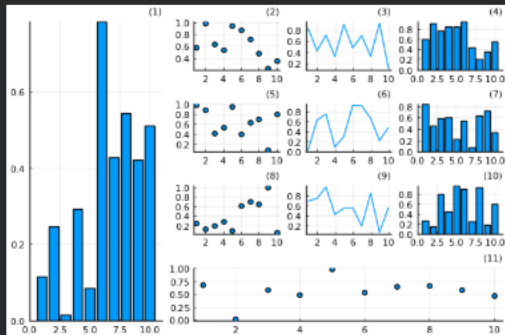
Выполнение лабораторной работы

```
# 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 colormap provided via 'c'. Parameters 'vmin', 'vmax' will be ignored

```
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. 18: Подграфики

Выполнение лабораторной работы

```
# Подключение необходимых пакетов
import numpy as np

# Задание функции и диапазона
x = range(0, stop=2n, 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=location)

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")

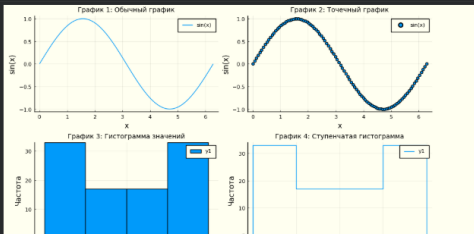
p4 = stephist(y, title="График 4: Ступенчатая гистограмма",
              xlabel="sin(x)", ylabel="Частота")

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=2n, 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=colormap,
             levels=20, colorbar=True)

# Объединяем все графики в одно окно
plot(p1, p2, p3, p4, p5, p6, layout=(3,2), size=(1000,800),
     background_color="ivory", titlefontsize=10)
```

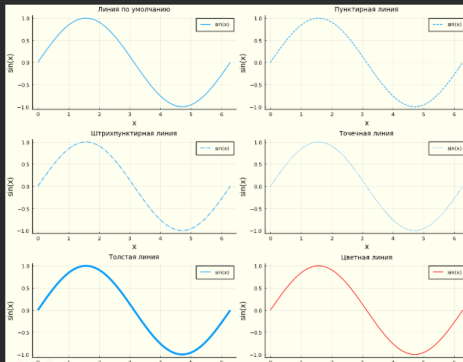


Выполнение лабораторной работы

```
# 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=(2,2),
     size=(1000, 800),
     background_color='ivory',
     titlefontsize=10,
     legend='topright')
```



Выполнение лабораторной работы

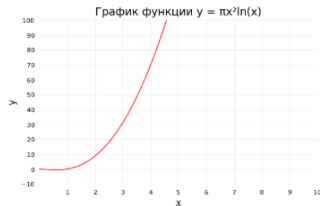
```
# 3
pyplot()

# Определение функции
f(x) = pi * 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 = pi*x^2*ln(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)
```



Выполнение лабораторной работы

```
from matplotlib import pyplot as plt

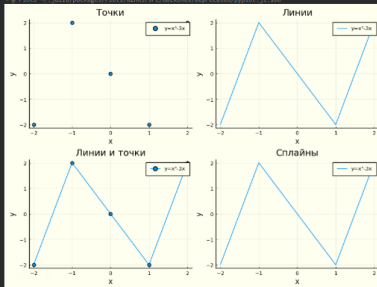
x = np.linspace(-2, 2, 100)
y = x**2

p1 = plot(x, y, 'x'),
title="Линия и точки",
label="y=x^2.3x",
markershape="circle",
markersize=5,
xlabel="x",
ylabel="y")

p2 = plot(x, y, 'x'),
title="Слайды",
label="y=x^2.3x",
linestyle="spline",
xlabel="x",
ylabel="y")

# Объединяем все графики в одно окно
plt.subplot(2,2)
plt.subplot(2,2)
plt.subplot(2,2)
plt.subplot(2,2)
plt.legend('topright')
plt.savefig("figure_00000.png")
```

```
Warning: linestyle spline is unsupported with Plots.PyPlotBackend(). Choose from: [:auto, :dash, :dashdot, :dot, :solid]
Warning: No data for colormap provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
Warning: Unknown linestyle spline
Warning: Unknown linestyle spline
```



Выполнение лабораторной работы

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

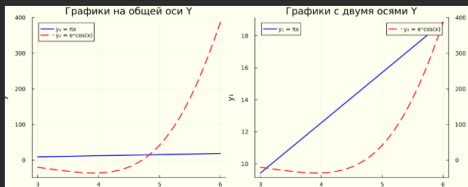
# Первый способ: общий график
p1 = plot(x, func1(x),
          label="y1 = mx",
          linestyle="solid",
          color="blue",
          linewidth=2,
          grid=True,
          legend="topleft")

plot(x, func2(x),
      label="y2 = excos(x)",
      linestyle="dash",
      color="red",
      linewidth=2,
      title="График на общей оси Y",
      xlabel="x",
      ylabel="y")

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

plot(twin(), x, func2(x),
      label="y2 = excos(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")
```



Выполнение лабораторной работы

```
using Plots
using Statistics
gr() #

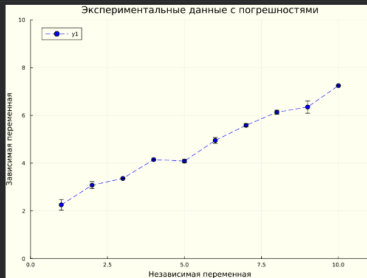
# Генерация экспериментальных данных
# Создаем массив значений x
x = 1:10

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

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

# Построение графика с погрешностями
plot(x, y_mean,
     yerr = sigma, # 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
```



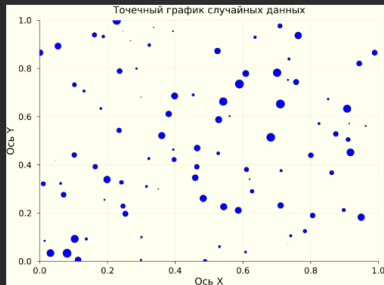
Выполнение лабораторной работы

```
% 7
using Plots
using Statistics
gr() # или другой backend

# 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
xlim!(0, 1) # установка границ по X
ylim!(0, 1) # установка границ по Y
```



Выполнение лабораторной работы

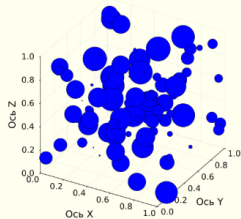
```
# В
using Plots
using Statistics
gr() # или другой backend

# 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
scatter3(x, y, z,
         markersize = s, # размер маркера
         markercolor = :blue, # цвет маркера
         markerstroke = :black, # 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
```

3D точечный график случайных данных



Выполнение лабораторной работы

```
# 9

pyplot()

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

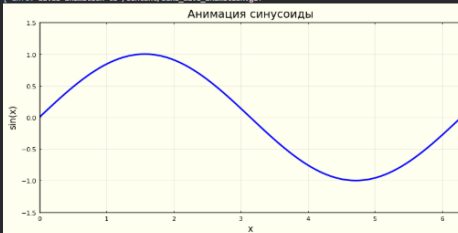
# Создаем диапазон значений x
x = range(0, stop=2π, length=100)

# 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



Выполнение лабораторной работы

```
# Рассчитаем координаты для малой окружности (двукратный)
xс = r_ * (k - 1) * cos(t[end]) .* r_ .* cos(Θ)
yс = r_ * (k - 1) * sin(t[end]) .* r_ .* sin(Θ)

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

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

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

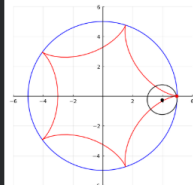
# Добавим малую окружность (черный цвет)
Plots.plot!(xc, yc, color=:black)

# Отобразим траекторию движения точки (черные кружки)
x1 = [r_ * (k - 1) * cos(t[end]) x[end]]
y1 = [r_ * (k - 1) * sin(t[end]) y[end]]
Plots.plot!(x1, y1, markershape=:circle, markersize=4, color=:black)

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

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

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



Выполнение лабораторной работы

```
# Координаты для малой окружности (дициклайса)
xc = r_ + (k - 1) * cos(t[end]) + r_ * cos(0)
yc = r_ + (k - 1) * sin(t[end]) + r_ * sin(0)

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

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

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

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

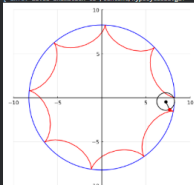
# Отобразим траектории движения точки (чёрные кружки)
x1 = transpose([r_ + (k - 1) * cos(t[end]) x[end]])
y1 = transpose([r_ + (k - 1) * sin(t[end]) y[end]])
Plots.plot!(x1, y1, markershape=:circle, markersize=4, color=:black)

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

end

# Сохраним 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
```



Выполнение лабораторной работы

```
# Большая окружность
X = r_1 * k .* cos(t)
Y = r_1 * k .* sin(t)

# Эпициклоида
x = r_1 * (k + 1) .* cos(t) .* r_1 .* cos((k + 1) .* t)
y = r_1 * (k + 1) .* sin(t) .* r_1 .* sin((k + 1) .* t)

# Малая окружность
xc = r_1 * (k + 1) * cos(t[end]) .* r_1 .* cos(0)
yc = r_1 * (k + 1) * sin(t[end]) .* r_1 .* sin(0)

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

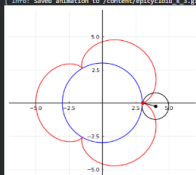
Plots.plot!(plt, X, Y, color=:blue, legend=false) # Большая окружность
Plots.plot!(x, y, color=:red) # Эпициклоида
Plots.plot!(xc, yc, color=:black) # Малая окружность

x1 = transpose([r_1 * (k + 1) * cos(t[end]) x[end]])
y1 = transpose([r_1 * (k + 1) * sin(t[end]) y[end]])

Plots.plot!(x1, y1, markershape=:circle, markersize=4, color=:black)
Plots.scatter!([k[end]], [y[end]], color=:red, markerstrokecolor=:red)
end

# Сохранение анимации
gif(anim, "epicycloid_k_5k.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
[ Info: Saved animation to /content/epicycloid_k_5k.gif
```



Выполнение лабораторной работы

```
# Координаты для малой окружности (диаметры=8)
xc = r_ * (k + 1) * cos(t[end]) - r_ * cos(8)
yc = r_ * (k + 1) * sin(t[end]) - r_ * sin(8)

# Создай основной график
plt = Plots.plotf;
xlim=(-10, 10),
ylim=(-10, 10),
color=:red,
aspect_ratio=1,
legend=false,
framestyles:origin

)

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

# Добавим эписцилоиду (красный цвет)
Plots.plotf(x, y, color=:red) # Эписцилоида

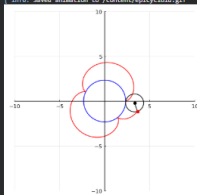
# Добавим малую окружность (чёрный цвет)
Plots.plotf(xc, yc, color=:black) # Малая окружность

# Отобразим траекторию движения точки (чёрные кружки)
x1 = transpose([r_ * (k + 1) * cos(t[end]) x[end]])
y1 = transpose([r_ * (k + 1) * sin(t[end]) y[end]])
Plots.plotf(x1, y1, markershape=:circle, markersize=4, color=:black)

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

# Сохраним 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
[ Info: Saved animation to /content/epicycloid.gif
```



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