

Лабораторная работа №5

Построение графиков

Легиньких Г.А.

Российский университет дружбы народов, Москва, Россия

Информация

- Легиньких Галина Андреевна
- НФИбд-02-21
- Российский университет дружбы народов
- 1032216447@pfur.ru
- <https://github.com/galeginkikh>

Основная информация

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

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

Выполнение лабораторной работы

Для начала я повторила примеры и добавила, где это было необходимо, обозначения осей координат, легенду с названиями траекторий, названия графиков и т.п. Больше я это нигде прописывать не буду.

Julia поддерживает несколько пакетов для работы с графиками. Использование того или иного пакета зависит от целей, преследуемых пользователем при построении. Стандартным для Julia является пакет `Plots.jl`.

Основные пакеты для работы с графиками в Julia

Основные пакеты для работы с графиками в Julia

```
[12]: f(x) = (3x.^2 + 6x - 9).*exp.(-0.3x)
      x = collect(range(-5,10,length=151))
      y = f(x)
      gr()
      plot(x,y, title="A simple curve", xlabel="Variable x", ylabel="Variable y", color="blue")
```

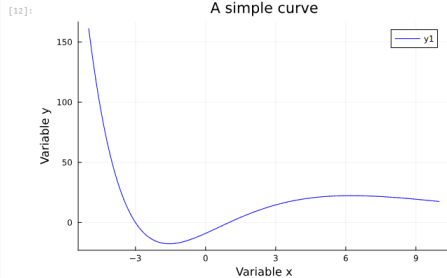


Рис. 1: Способ 1

Опции при построении графика

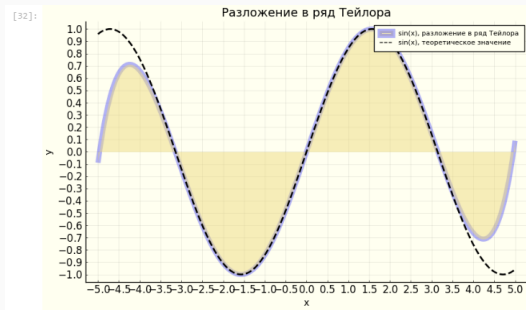


Рис. 2: Графики исходной функции и её разложения в ряд Тейлора с опциями

Точечный график

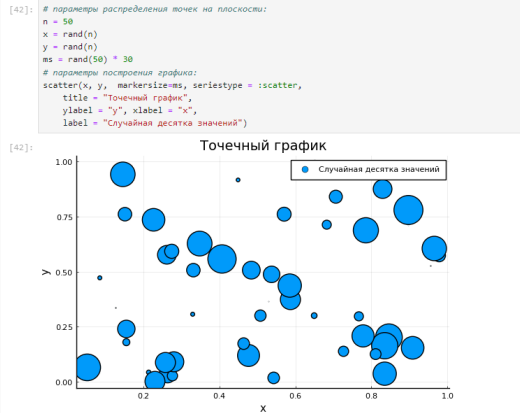


Рис. 3: График пятидесяти случайных значений на плоскости с различными опциями отображения

Точечный график

```
[48]: # параметры распределения точек в пространстве:
n = 50
x = rand(n)
y = rand(n)
z = rand(n)
ms = rand(50) * 30
# параметры построения графика:
scatter(x, y, z, markersize=ms, seriestype = :scatter,
        title = "Точечный график",
        ylabel = "y", xlabel = "x", zlabel = "z",
        label = "Случайные 50 значений")
```

[48]:

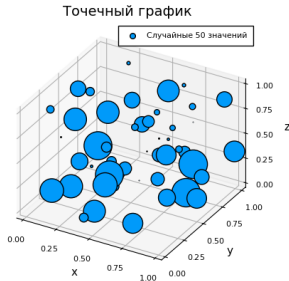
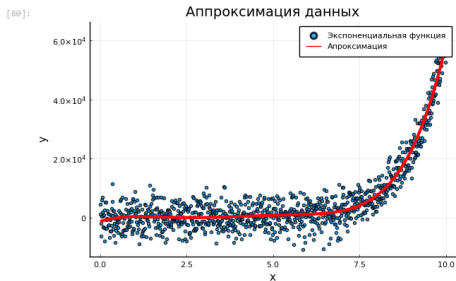


Рис. 4: График пятидесяти случайных значений в пространстве с различными опциями отображения

Аппроксимация — научный метод, состоящий в замене объектов их более простыми аналогами, сходными по своим свойствам.

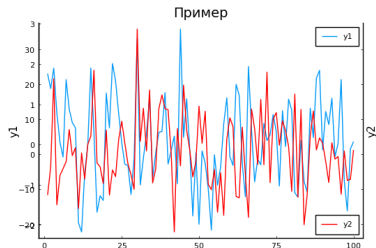
```
[80]: # массив данных от 0 до 10 с шагом 0.01:
x = collect(0:0.01:9.99)
# экспоненциальная функция со случайным сдвигом значений:
y = exp.(ones(1000)*x) + 4000*randn(1000)
# построение графика:
scatter(x,y,markersize=3,alpha=.8,
ylabel = "y", xlabel = "x", title = "Аппроксимация данных", label = "Экспоненциальная функция")
# определение массива для нахождения коэффициентов полинома:
A = [ones(1000) x x.^2 x.^3 x.^4 x.^5]
# решение матричного уравнения:
c = A\y
# построение полинома:
f_approx = 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,f_approx, linewidth=3, color=:red, label = "Аппроксимация")
```



Две оси ординат

Иногда требуется на один график вывести несколько траекторий с существенными отличиями в значениях по оси ординат.

```
[106]: p1 = plot(randn(100),
ylabel="y1",
title = "Пример",
leg=:topright,
grid = :off,
size=(600, 400)
)
plot!(twinx(), randn(100)*10,
c=:red,
ylabel="y2",
leg=:bottomright,
grid = :off,
box = :on
)
display(p1)
```



Полярные координаты

```
[112]: # 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),  
title = "Полярные координаты",  
label = "y1"  
)
```

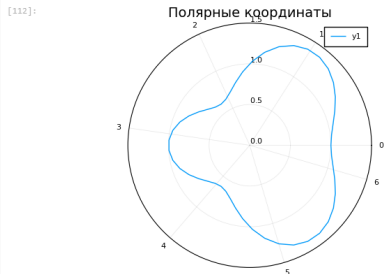


Рис. 7: График функции, заданной в полярных координатах

Параметрический график кривой на плоскости

•[126]: *# параметрическое уравнение:*

```
x_t(t) = sin(t)
```

```
y_t(t) = sin(2t)
```

построение графика:

```
plot(x_t, y_t, 0, 2π, fill=(0,:orange), title = "кривая на плоскости",
```

```
      ylabel = "y", xlabel = "x", label = "График")
```

[126]:

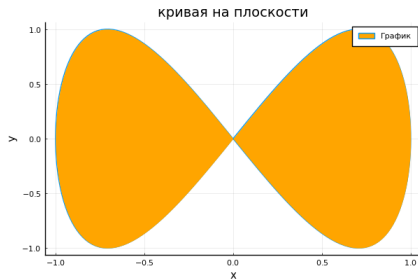


Рис. 8: Параметрический график кривой на плоскости

Параметрический график кривой в пространстве

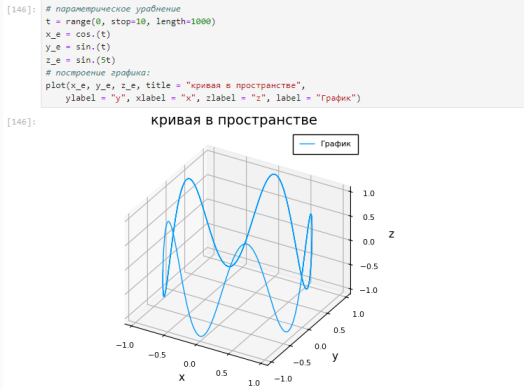


Рис. 9: Параметрический график кривой в пространстве

График поверхности (использована функция surface())

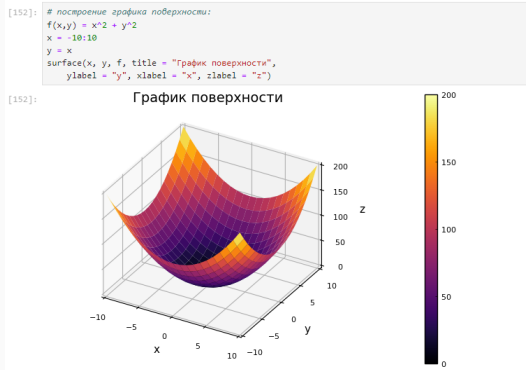


Рис. 10: График поверхности (использована функция surface())

График поверхности (использована функция plot())



Рис. 11: График поверхности (использована функция plot())

Сглаженный график поверхности

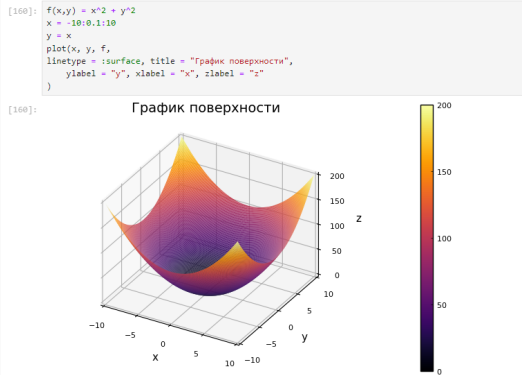


Рис. 12: Сглаженный график поверхности

График поверхности с изменённым углом зрения

```
[162]: x=range(-2,stop=2,length=100)
        y=range(sqrt(2),stop=2,length=100)
        f(x,y) = x*y-x-y+1
        plot(x,y,f,
              linetype=:surface,
              c=cgrad([:red,:blue]),
              camera=(-30,30),
              title="График поверхности",
              ylabel="y", xlabel="x", zlabel="z"
        )
```

[162]:

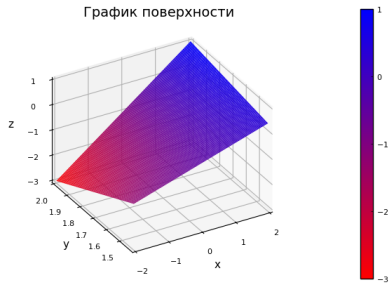


Рис. 13: График поверхности с изменённым углом зрения

```
[170]: 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, title = "График поверхности",  
      ylabel = "y", xlabel = "x", zlabel = "z"  
)
```

[170]:

График поверхности

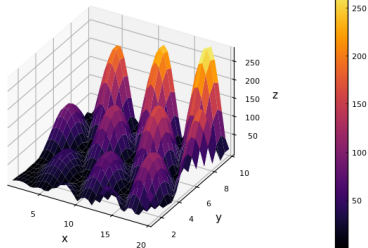


Рис. 14: График поверхности, заданной функцией

```
[180]: 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,  
)  
p = contour(x, y, g, title = "Линии уровня",  
ylabel = "y", xlabel = "x", fill=true)  
plot(p)
```

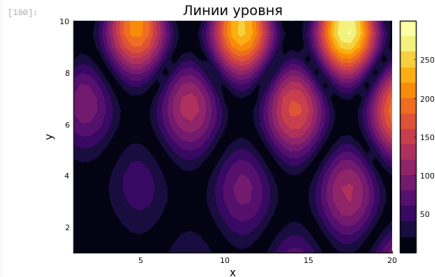


Рис. 15: Линии уровня с заполнением

```
[186]: # 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, title = "График функции",  
      ylabel = "y", xlabel = "x", zlabel = "z"  
)
```

[186]:

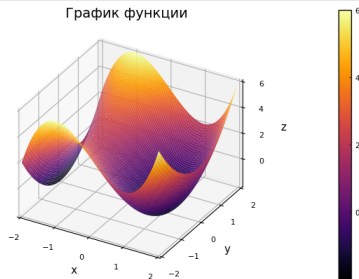


Рис. 16: График функции


```
[214]: # 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)
# 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, title = "Линии уровня",
ylabel = "y", xlabel = "x")
# коррекция области видности графика:
xlims!(-2, 2)
ylims!(-2, 2)
```

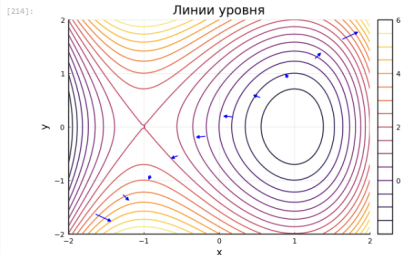


Рис. 17: Векторное поле функции

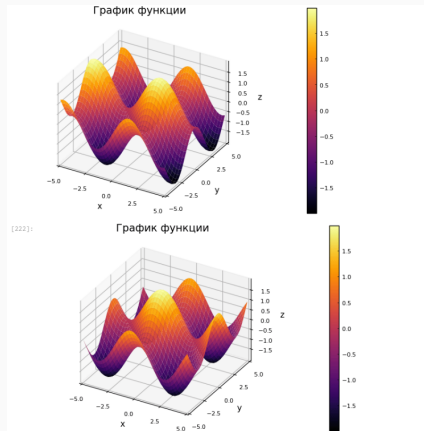


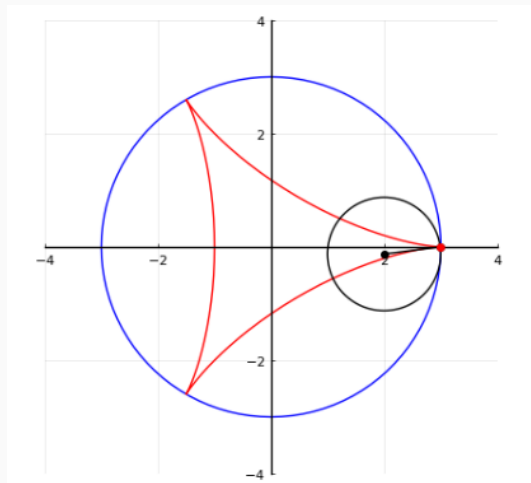
Рис. 18: Анимированный график поверхности

Гипоциклоида — плоская кривая, образуемая точкой окружности, катящейся по внутренней стороне другой окружности без скольжения.



Рис. 19: Малая окружность гипоциклоиды с добавлением радиуса

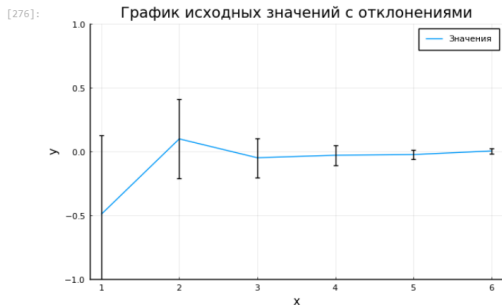
В конце сделаем анимацию получившегося изображения.



В исследованиях часто требуется изобразить графики погрешностей измерения.

Подключила пакет Statistics.

```
[276]: sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
n = 10
y = [mean(sd*randn(n)) for sd in sds]
errs = 1.96 * sds / sqrt(n)
plot(y,
      ylims = (-1,1),
      err = errs, title = "График исходных значений с отклонениями",
      ylabel = "y", xlabel = "x", label = "Значения")
```



Использование пакета Distributions.

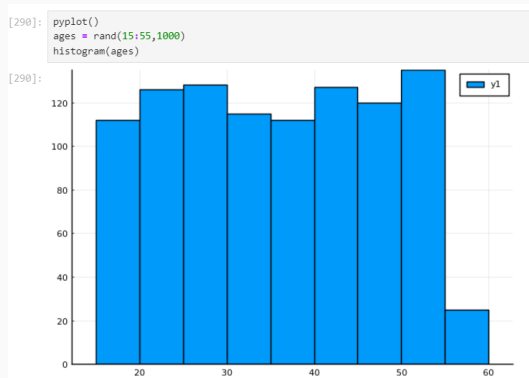


Рис. 22: Гистограмма, построенная по массиву случайных чисел

Определим макет расположения графиков. Команда `layout` принимает кортеж `layout = (N, M)`, который строит сетку графиков $N \times M$. Например, если задать `layout = (4,1)` на графике четыре серии, то получим четыре ряда графиков.

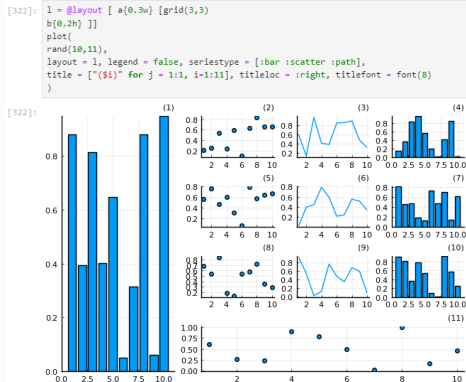


Рис. 23: Демонстрация применения сложного макета для построения графиков

Задания для самостоятельного выполнения

- Задание 1

```
[12]: x = 0:0.1:2π
sin_y = sin.(x)

plot( # Графики в одном окне
      plot(x, sin_y, title="Линия", lw=2), # Линейный график
      scatter(x, sin_y, title="Точки"), # Точечный график
      bar(x, abs.(sin_y), title="Гистограмма"), # Гистограмма (модули значений)
      histogram(sin_y, bins=10, title="Гистограмма значений"), # Гистограмма значений
      layout=(2, 2), xlabel = "x", ylabel = "y", label = "График"
    )
```

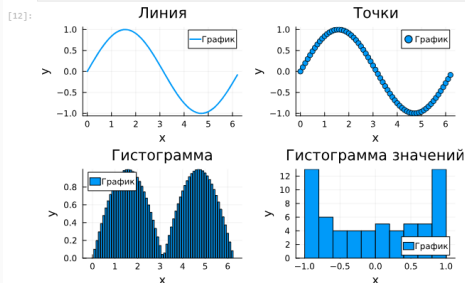


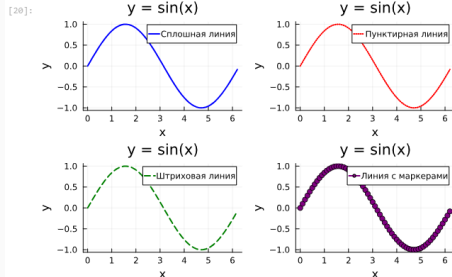
Рис. 24: Задание 1

Задания для самостоятельного выполнения

- Задание 2

```
[20]: x = 0:0.1:2π
      y = sin.(x)

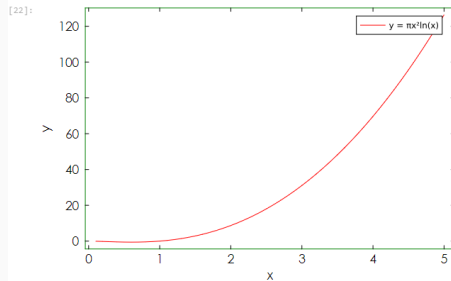
      # Создание сетки с графиками разных стилей линий
      plot(
        plot(x, y, label="Сплошная линия", color=:blue, linestyle=:solid, linewidth=2), # Сплошная
        plot(x, y, label="Пунктирная линия", color=:red, linestyle=:dot, linewidth=2), # Пунктирная
        plot(x, y, label="Штриховая линия", color=:green, linestyle=:dash, linewidth=2), # Штриховая
        plot(x, y, label="Линия с маркерами", color=:purple, marker=:circle, markersize=4,
              linewidth=2), # Линия с маркерами
        layout=(2, 2), # Макет 2x2
        xlabel="x", ylabel="y", title="y = sin(x)"
      )
```



Задания для самостоятельного выполнения

- Задание 3

```
[22]: x_vals = 0.1:0.1:5
      y_vals = pi .* x_vals.^2 .* log.(x_vals)
      plot(
          x_vals, y_vals,
          label="y =  $\pi x^2 \ln(x)$ ",
          xlabel="x",
          ylabel="y",
          framestyle=box,
          legend=:topright,
          grid=false,
          color=:red,
          foreground_color_border=:green # Задание зелёной рамки
      )
      # Настройка расстояний до осей и шрифта
      xaxis!(font(12, "Arial"))
      yaxis!(font(12, "Arial"))
```



Задания для самостоятельного выполнения

- Задание 4

```
[30]: x_points = [-2, -1, 0, 1, 2]
      y_points = x_points.^3 .- 3 .* x_points

      p1 = scatter(x_points, y_points, label="Точки", title="График точек")
      p2 = plot(x_points, y_points, label="Линии", title="График линий")
      p3 = plot(x_points, y_points, marker=:circle, label="Линии и точки", title="График линии и точки")
      p4 = plot(x_points, y_points, label="Кривая", title="Кривая")

      plot(p1, p2, p3, p4, layout=(2, 2), xlabel="x", ylabel="y")
      savefig("figure_leginkikh.png")
```

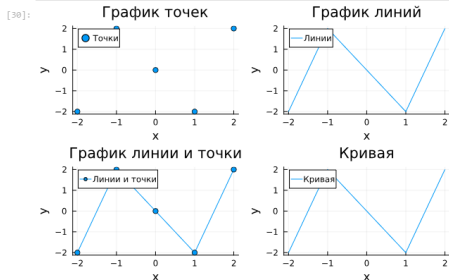


Рис. 27: Задание 4

Задания для самостоятельного выполнения

- Задание 5

```
[34]: x = 3:0.1:6
```

```
# Функции y1 и y2
```

```
y1(x) = π * x
```

```
y2(x) = exp(x) * cos(x)
```

```
# Построение графиков на одном рисунке
```

```
plot(x, y1(x), label="y1(x) = πx", color=:blue, xlabel="x", ylabel="y",  
      title="Графики функций y1(x) и y2(x)", grid=true)
```

```
plot!(x, y2(x), label="y2(x) = exp(x)cos(x)", color=:red)
```

```
[34]:
```

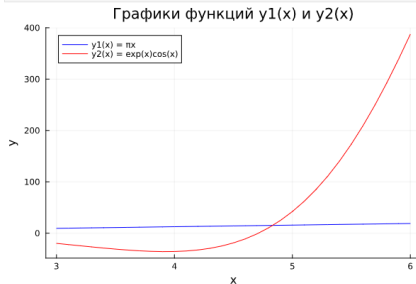


Рис. 28: Задание 5.1

Задания для самостоятельного выполнения

[46]: `x = 3:0.1:6`

```
# Функции y1 и y2
y1(x) = π * x
y2(x) = exp(x) * cos(x)

# Построение графиков с двумя осями ординат
p1 = plot(x, y1(x), label="y1(x) = πx", color=:blue, xlabel="x", ylabel="y1(x)",
          grid=true)
p2 = plot(x, y2(x), label="y2(x) = exp(x)cos(x)", color=:red, xlabel="x", ylabel="y2(x)",
          secondary = true)

# Отображаем оба графика
plot(p1, p2, title="Графики")
```

[46]:

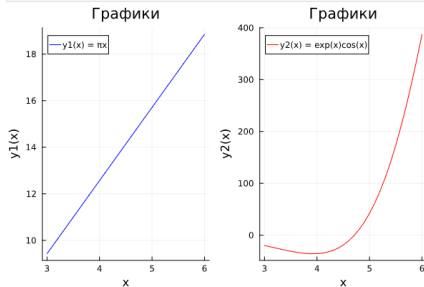


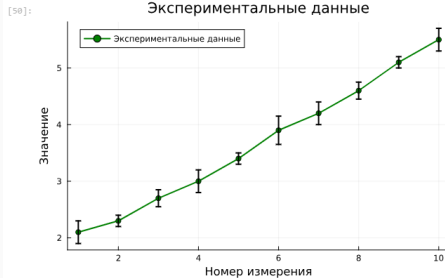
Рис. 29: Задание 5.2

Задания для самостоятельного выполнения

- Задание 6

```
[50]: x_data = 1:10
      y_data = [2.1, 2.3, 2.7, 3.0, 3.4, 3.9, 4.2, 4.6, 5.1, 5.5]
      errors = [0.2, 0.1, 0.15, 0.2, 0.1, 0.25, 0.2, 0.15, 0.1, 0.2]

      plot(
        x_data, y_data,
        yerror=errors, # Ошибки измерений
        label="Экспериментальные данные",
        xlabel="Номер измерения",
        ylabel="Значение",
        legend=:topleft,
        color=:green,
        marker=:circle,
        lw=2,
        title = "Экспериментальные данные"
      )
```



Задания для самостоятельного выполнения

- Задание 7

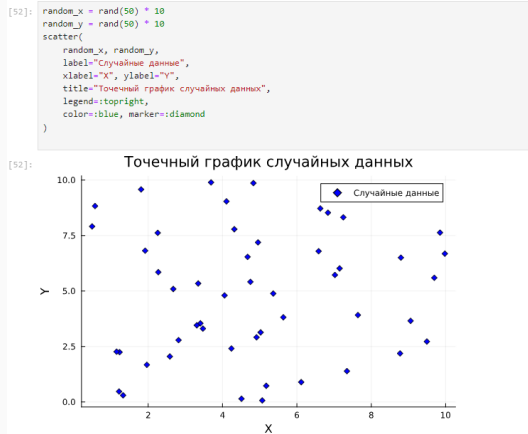


Рис. 31: Задание 7

Задания для самостоятельного выполнения

- Задание 8

```
[56]: random_x3 = rand(50) * 10  
      random_y3 = rand(50) * 10  
      random_z3 = rand(50) * 10  
      plot(  
          random_x3, random_y3, random_z3,  
          seriestype=:scatter, marker=:circle, color=:red,  
          xlabel="X", ylabel="Y", zlabel="Z",  
          label="Случайные данные",  
          title="3D точечный график случайных данных",  
          legend=false  
      )
```

[56]:

3D точечный график случайных данных

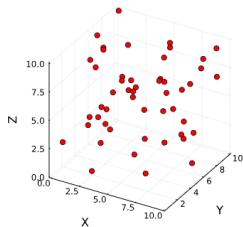


Рис. 32: Задание 8

Задания для самостоятельного выполнения

- Задание 9

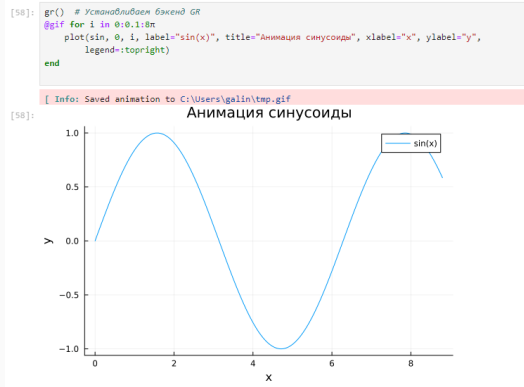


Рис. 33: Задание 9

Задания для самостоятельного выполнения

- Задание 10

```
[60]: @gif for k in [3, 4, 5, 1/2, 3/2] # Целые и рациональные значения для k
      θ = 0:0.1:2π
      x(t) = (3 - k) * cos(t) + k * cos((3 - k) / k * t)
      y(t) = (3 - k) * sin(t) - k * sin((3 - k) / k * t)

      plot(x.(θ), y.(θ), label="Гипоциклоид, k=$k", title="Анимация гипоциклоиды", xlabel="x",
            ylabel="y", legend=:topright)
    end
```

[Info: Saved animation to C:\Users\galin\tmp.gif

[60]:

Анимация гипоциклоиды

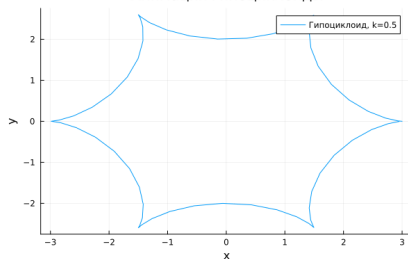


Рис. 34: Задание 10

Задания для самостоятельного выполнения

- Задание 11

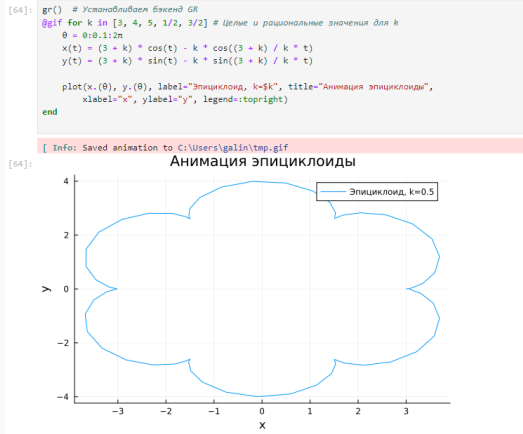


Рис. 35: Задание 11

Вывод

Освоила синтаксис языка Julia для построения графиков.