

# Лабораторная работа №5

## Построение графиков

---

Ким Реачна<sup>1</sup>

07 декабря, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи

---

# Цель лабораторной работы

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

1. Используя Jupyter Lab, повторите примеры.
2. Выполните задания для самостоятельной работы.

# Процесс выполнения лабораторной работы

---

# Основные пакеты для работы с графиками в Julia

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

# 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)

# 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")
```

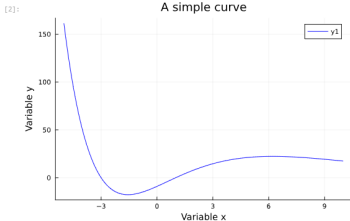


Рис. 1: График функции  $f(x) = (3x^2 + 6x - 9)e^{-0,3x}$ , при помощи gr()

# Опции при построении графика

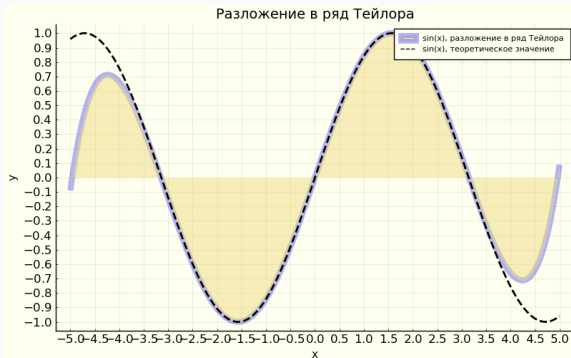


Рис. 2: Вид графиков после добавления опций при их построении

# Точечный график - Простой точечный график

```
[11]: # 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",  
      xlabel = "x",  
      ylabel = "y",  
      # подпись в легенде, цвет и тип линии:  
      label = "y1, теоретическое значение",  
      leg = :topright  
)
```

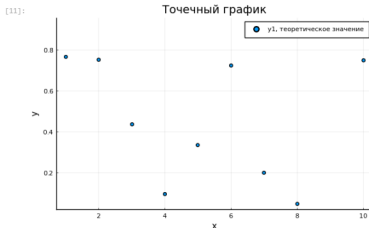


Рис. 3: График десяти случайных значений на плоскости



# Точечный график с кодированием значения размером точки

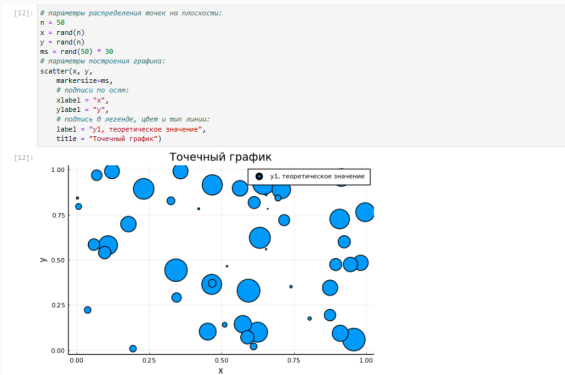


Рис. 4: График пятидесяти случайных значений на плоскости с различными опциями отображения

# 3-мерный точечный график с кодированием значения размером точки

```
[13]: # 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,  
        xlabel = "x",  
        ylabel = "y",  
        zlabel = "z",  
        # подписи в легенде, цвет и тип линии:  
        label = "sin(x), теоретическое значение",  
        title = "Точечный график")
```

```
[13]:
```

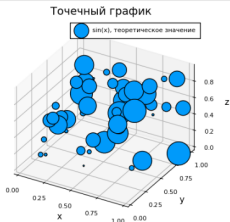


Рис. 5: График пятидесяти случайных значений в пространстве с различными опциями отображения

# Аппроксимация данных

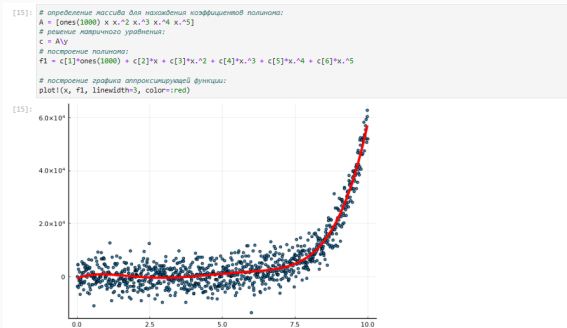


Рис. 6: Пример аппроксимации исходной функции полиномом 5-й степени

# Две оси ординат

```
# пример добавления на график второй случайной траектории
# (задано обозначение траектории и её цвет, легенда снизу справа, без сетки)
# задана рамка графика
plot(twine(), randn(100)*10,
     c='red',
     ylabel="y2",
     leg=:bottomright,
     grid=:off,
     box=:on,
     # size=(600, 400)
)
```

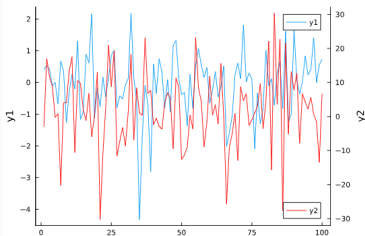


Рис. 7: Пример двух траекторий на одном графике с двумя осями ординат

# Полярные координаты

```
[18]: # 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)  
)
```

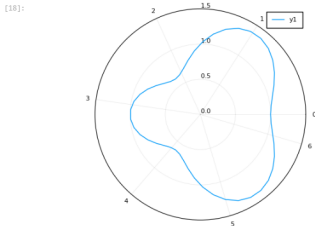


Рис. 8: График функции, заданной в полярных координатах

# Параметрический график кривой на плоскости

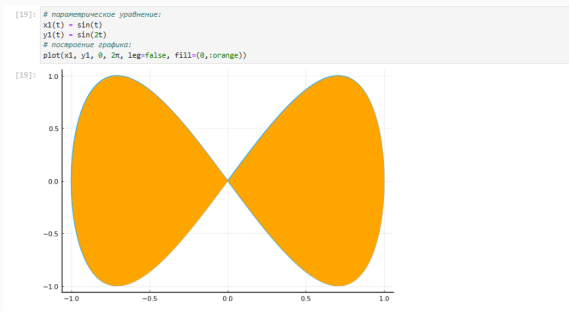


Рис. 9: Параметрический график кривой на плоскости

# Параметрический график кривой в пространстве

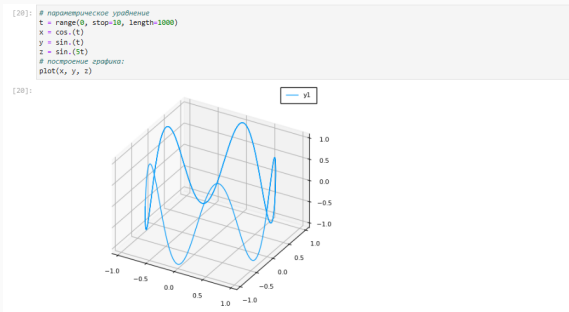


Рис. 10: Параметрический график кривой в пространстве

# График поверхности

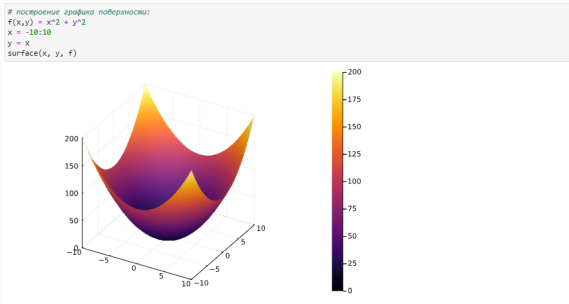


Рис. 11: График поверхности использована функция surface()



```
[27]: p = contour(x, y, g,  
fill=True)  
plot(p)
```

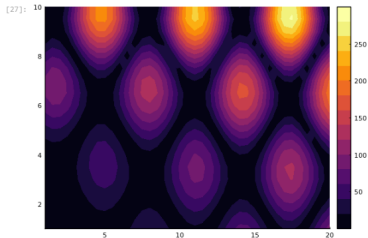


Рис. 12: Линии уровня с заполнением

```
[33]: # коррекция области видимости графика:  
xlims!(-2, 2)  
ylims!(-2, 2)  
quiver!(xxs, yys, quiver=dh, c=:blue)
```

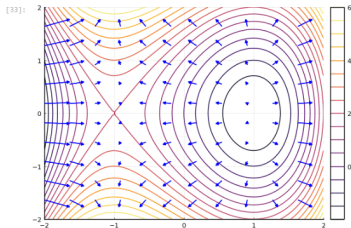


Рис. 13: Векторное поле функции  $h(x, y) = x^3 - 3x + y^2$  скорректирована область видимости

# Анимация - Gif-анимация

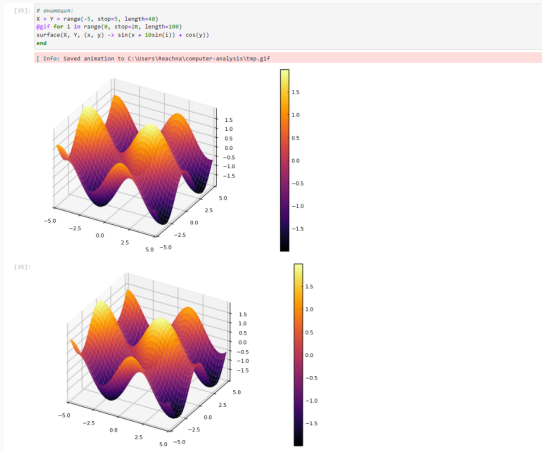


Рис. 14: Анимированный график поверхности

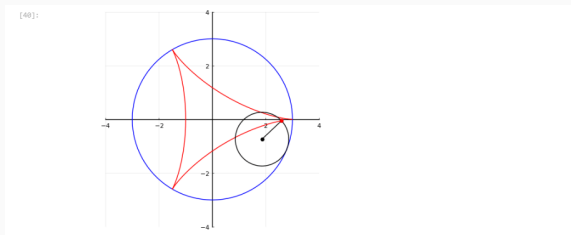


Рис. 15: Анимация движения гипоциклоиды

# Анимация - Синусоида

```
[71]: # Task9
n = 200
x = collect(0*pi : 2*pi/100 : 10*pi*pi/100)

anim = @animate for i in 1:n
    fig = plot(1,
               xlim=(0, 10),
               ylim=(-2, 2),
               cs=:purple,
               aspect_ratio=1,
               legend=false,
               framestyle=:origin)

    step = x[1 : i]
    y = sin.(step)
    plot!(step, y)
end
#Сохранила анимацию в gif-файл
gif(anim, "sinus.gif")
```

Info: Saved animation to C:\Users\Reachna\computer-analysis\sinus.gif



[71]:

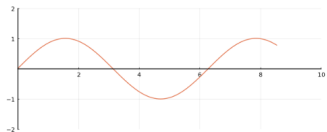


Рис. 16: Анимация Синусоида

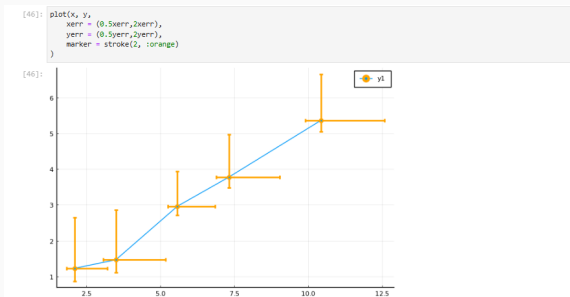


Рис. 17: График асимметричных ошибок по двум осям

# Использование пакета Distributions

```
plotly()  
d1=Normal(10.0,5.0);  
d2=Normal(35.0,10.0);  
d3=Normal(60.0,5.0);  
N=1000;  
ages = (Float64[]);  
ages = append!(ages,rand(d1,Int64(ceil(N/2))));  
ages = append!(ages,rand(d2,N));  
ages = append!(ages,rand(d3,Int64(ceil(N/3))));  
histogram(  
    ages,  
    bins=50,  
    label="Распределение по возрастам (года)",  
    xlabel = "Возраст (лет)",  
    ylabel = "Количество",  
    title = "Распределение по возрастам (года)"  
)
```

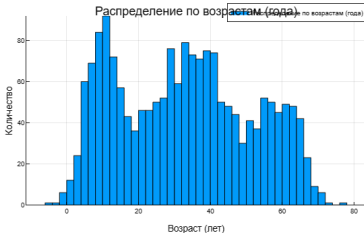


Рис. 18: Гистограмма распределения людей по возрастам



Рис. 19: Демонстрация применения сложного макета для построения графиков



## Выводы по проделанной работе

---

Освоила синтаксис языка Julia для построения графиков.