

Лабораторная работа № 5

Построение графиков

Старовойтов Е. С.

14 декабря 2024

Информация

Докладчик

- Старовойтов Егор Сергеевич
- студент кафедры ТВиК
- Российский университет дружбы народов
- 1032212281@pfur.ru

Вводная часть

Цели и задачи

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)

Результаты

Поставленные боевые задачи были выполнены, все цели достигнуты.

Выполнение лабораторной работы

```
Untitled3.jupyter      X +
+ - - - - - Code
Notebook  Julia 1.11.0-rc3

[1]: using Pkg
      Pkg.add("Plots")
      Pkg.add("PyPlot")
      Pkg.add("Plotly")
      Pkg.add("UnicodePlots")
      # Подключаем для использования Plots:
      using Plots

      Updating registry at '~/.julia/registries/General.toml'
      Resolving package versions...
      Installing LERC_jll v4.0.0+0
      Installing JpegTurbo_jll v3.0.4+0
      Installing GR_jll v0.73.9+0
      Installing libdecor_jll v0.2.2+0
      Installing libfdk_aac_jll v2.0.3+0
      Installing x265_jll v3.5.0+0
      Installing Libmount_jll v2.40.2+0
      Installing LoggingExtras v1.1.0
      Installing Xorg_xkbcomp_jll v1.4.6+1
      Installing RelocatableFolders v1.0.1
      Installing Opus_jll v1.3.3+0
      Installing Unitful v1.21.1
      Installing Measures v0.3.2
      Installing Contour v0.6.3
      Installing ConcurrentUtilities v2.4.3
      Installing Grisu v1.0.2
      Installing Xorg_xcb_util_wm_jll v0.4.1+1
      Installing Xorg_xcb_util_image_jll v0.4.0+1
      Installing PlotUtils v1.4.3
      Installing ColorSchemes v3.27.1
      Installing RecipesPipeline v0.6.12
      Installing GR v0.73.9
      Installing OpenSSL v1.4.3
      Installing Xorg_libSM_jll v1.2.4+0
      Installing Xorg_libpthread_stubs_jll v0.1.1+1
      Installing HTTP v1.10.14
      Installing DelimitedFiles v1.9.1
      Installing Cairo_jll v1.18.2+1
      Installing Xorg_xcb_util_jll v0.4.0+1
      Installing libpng_error_jll v1.50.0+0
      Installing Fontconfig_jll v2.15.0+0
      Installing EpollShim_jll v0.0.20230411+1
      Installing Xorg_libxkbfile_jll v1.1.2+1
      Installing Xorg_libXau_jll v1.0.11+1
      Installing Missings v1.2.0
      Installing Xorg_libXinerama_jll v1.1.5+0
      Installing FFMPEG v0.4.2
      Installing IrrationalConstants v0.2.2
      Installing PtrArrays v1.2.1
      Installing Pango_jll v1.54.1+0
      Installing Showoff v1.0.3
      Installing Bzip2_jll v1.0.8+2
      Installing xkbcommon_jll v1.4.1+1
      Installing Xorg_xcb_util_keysyms_jll v0.4.0+1
      Installing SimpleBufferStream v1.2.0
      Installing Pipe v1.3.0
      Installing XZ_jll v5.6.3+0
      Installing PlotThemes v3.3.0
      Installing NaNMath v1.0.2
      Installing HarfBuzz_jll v8.5.0+0
      Installing LZ0_jll v2.10.2+1
      Installing TranscodingStreams v0.11.3
      Installing FriBidi_jll v1.0.14+0
      Installing GLFW_jll v3.4.0+1
      Installing rtf_jll v0.56.3+0
      Installing UnicodeFun v0.4.1
      Installing DataStructures v0.18.20
      Installing FreeType2_jll v2.13.3+1
      Installing x264_jll v2021.5.5+0
      Installing JLFzf v0.1.9
      Installing StatsAPI v1.7.0
      Installing Compat v4.16.0
      Installing CodecZlib v0.7.6
      Installing StatsBase v0.34.4
      Installing libpng_jll v1.6.44+0
      Installing Xorg_libxcb_jll v1.17.0+1
      Installing EventLoggingNotifier v0.1.11
```

This may mean IJulia [7073ff75-c697-5162-941a-fcdaad2a7d2a] does not support precompilation but is imported by a module that does.

Info: Skipping precompilation due to precompilable error. Importing IJuliaExt [2f4121a4-3b3a-5ce6-9c5e-1f2673ce168a].
exception = Error when precompiling module, potentially caused by a `__precompile__`(false) declaration in the module.

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

```
[2]: f (generic function with 1 method)
```

```
[6]: x = collect(range(-5,10,length=151))
```

```
[6]: 151-element Vector{Float64}:
```

```
-5.0  
-4.9  
-4.8  
-4.7  
-4.6  
-4.5  
-4.4  
-4.3  
-4.2  
-4.1  
-4.0  
-3.9  
-3.8  
⋮  
8.9  
9.0  
9.1  
9.2  
9.3  
9.4  
9.5  
9.6  
9.7  
9.8  
9.9  
10.0
```

```
[7]: y = f(x)
```

```
[7]: 151-element Vector{Float64}:
```

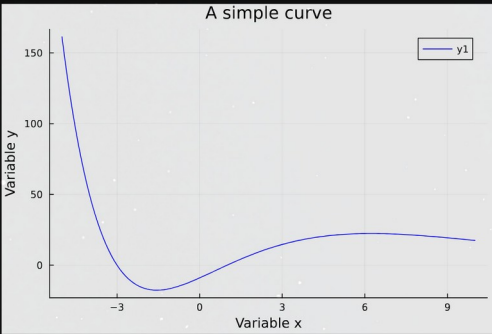
```
161.34080653217032  
146.26477779394003  
132.19219298833204  
119.06942359634911  
106.8453557470588  
95.47128188475011  
84.9007968362764  
75.08969810741056  
66.99589024347995  
57.579293005755176  
49.80175384104821  
42.62696260773204  
36.02037156694452  
⋮  
19.531205103994854  
19.355187669047936  
19.176427741119536  
18.995125520095375  
18.811475185747092  
18.625664977689375  
18.437877278854756  
18.248288702120195  
18.057070179740368  
17.86438705526336  
17.6703991776229  
17.475260997120245
```

```
[8]: gr()
```

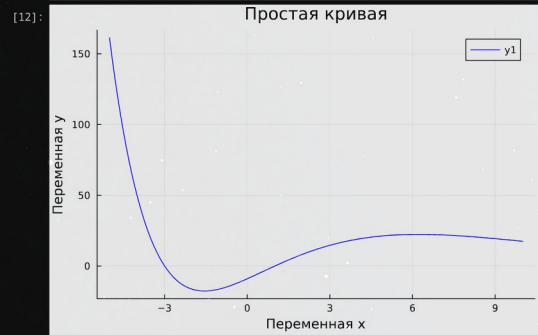
```
[8]: Plots.GRBackend()
```

```
[9]: plot(x,y,  
title="A simple curve",  
xlabel="Variable x",
```

```
title="A simple curve",
xlabel="Variable x",
ylabel="Variable y",
color="blue")
```



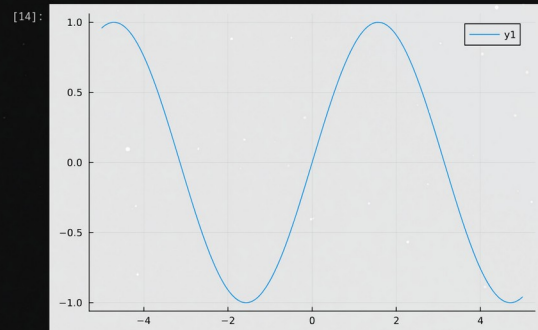
```
[12]: plot(x,y,  
         title="Простая кривая",  
         xlabel="Переменная x",  
         ylabel="Переменная y",  
         color="blue")
```



```
[13]: sin_theor(x) = sin(x)
```

[13]: sin_theor (generic function with 1 method)

```
[14]: plot(sin_theor)
```

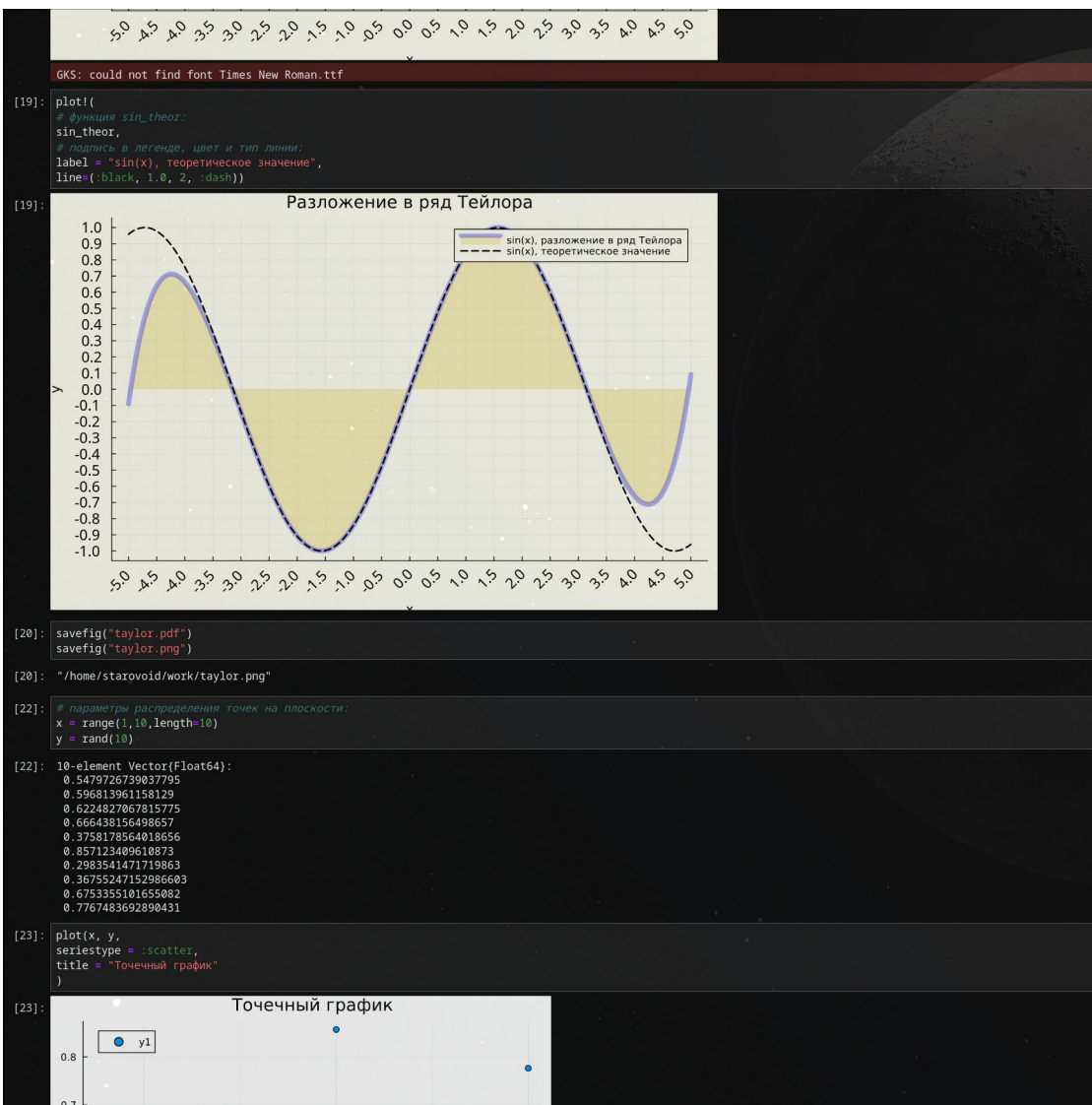


```
[15]: sin_taylor(x) = [(-1)**i*x**(2*i+1)/factorial(2*i+1) for i in 0:4] |> sum
```

[15]: sin_taylor (generic function with 1 method)

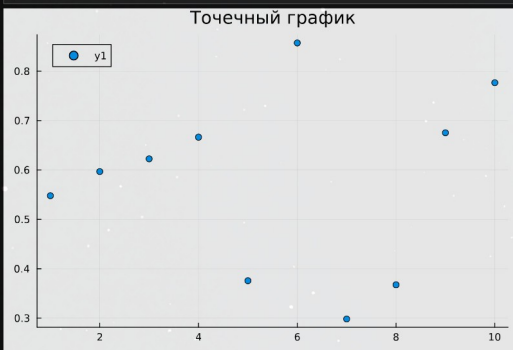
```
[16]: plot(sin_taylor)
```





```
title = "Точечный график"  
)
```

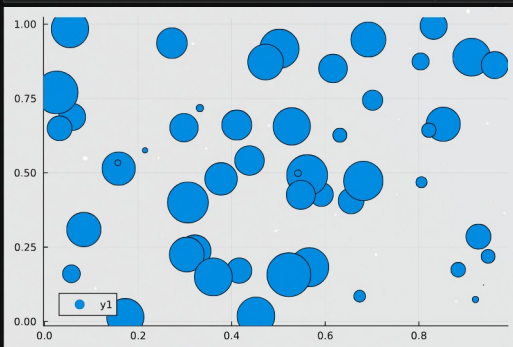
23]:



25]:

```
n = 50  
x = rand(n)  
y = rand(n)  
ms = rand(50) * 30  
# параметры построения графика:  
scatter(x, y, markersize=ms)
```

25]:



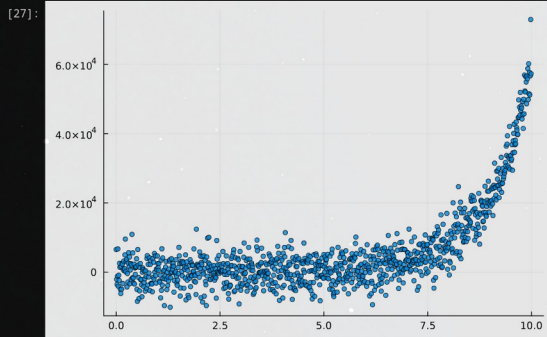
26]:

```
n = 50  
x = rand(n)  
y = rand(n)  
z = rand(n)  
ms = rand(50) * 30  
# параметры построения графика:  
scatter(x, y, z, markersize=ms)
```

26]:




```
[27]: # 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,legend=false)
```

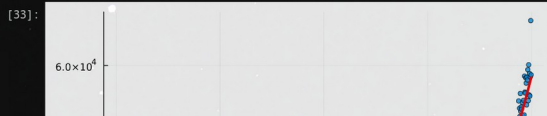


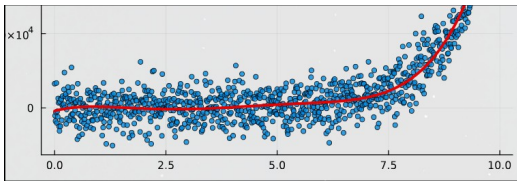
```
[32]: # определение массива для нахождения коэффициентов полинома:
A = [ones(1000) x x.^2 x.^3 x.^4 x.^5]
# решение матричного уравнения:
c = A\y
# построение полинома:
g = c[1]*ones(1000) + c[2]*x + c[3]*x.^2 + c[4]*x.^3 + c[5]*x.^4 + c[6]*x.^5
```

[32]: 1000-element Vector{Float64}:

```
-828.172038846246
-795.5142512340921
-763.4584919309991
-731.9085123204036
-701.1281000171448
-670.8410788616624
-641.1313087791918
-611.9926856929609
-583.4191414283861
-555.404643617268
-527.943195601988
-501.028836339705
-474.65564030655054
1
51385.44182856596
51835.821004672674
52289.4324781223
52746.29335141322
53206.420793411555
53669.83203144767
54136.5443514135
54606.57509785588
55079.941674074274
55556.661542216316
56036.75222337188
56520.23129767226
```

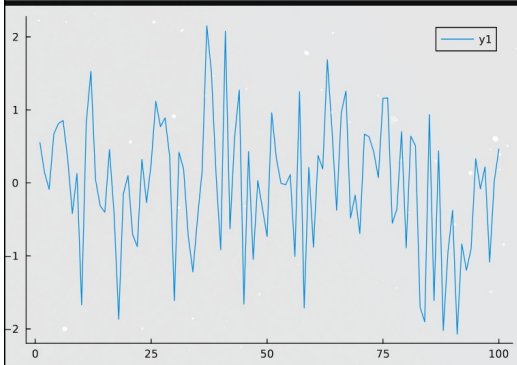
```
[33]: plot!(x,g,linewidth=3, color=:red)
```



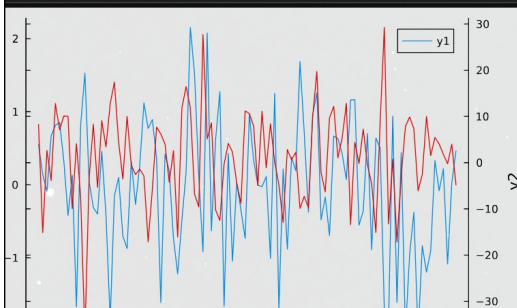


пример случайной траектории
заданы обозначение траектории, легенда вверху справа, без сетки)

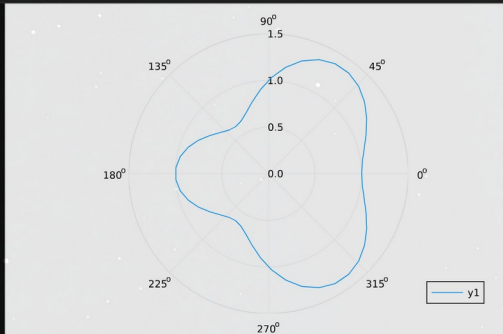
```
t(randn(100),  
  bel="y1",  
  pos=:topright,  
  d = :off,
```



```
t!(twinx(), randn(100)*10,  
    red,  
    bel="y2",  
    pos=:bottomright,  
    d = :off,  
    = :on)
```



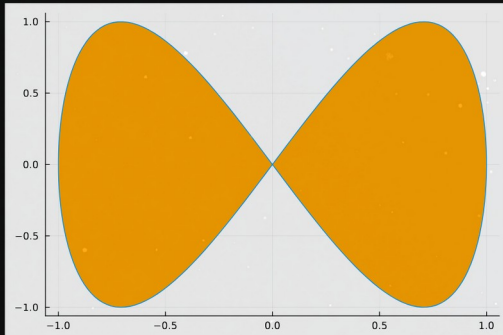
[36]:



[38]:

```
# параметрическое уравнение:  
a(t) = sin(t)  
b(t) = sin(2*t)  
# построение графика:  
plot(a, b, 0, 2*pi, leg=False, fill=(0,orange))
```

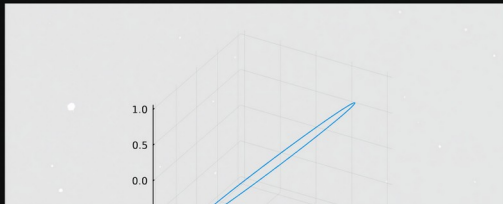
[38]:



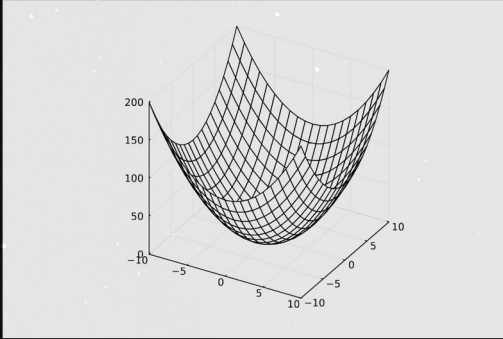
[39]:

```
# параметрическое уравнение  
t = range(0, stop=10, length=1000)  
x = cos(t)  
y = sin(t)  
z = sin(5*t)  
# построение графика:  
plot(x, y, x)
```

[39]:

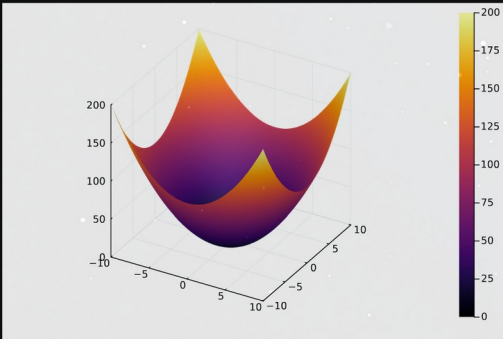


[41]:



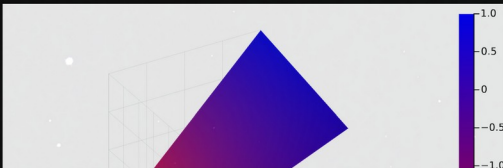
```
[42]: f(x,y) = x^2 + y^2
x = -10:0.1:10
y = x
plot(x, y, f,
      linestyle = :surface
    )
```

[42]:

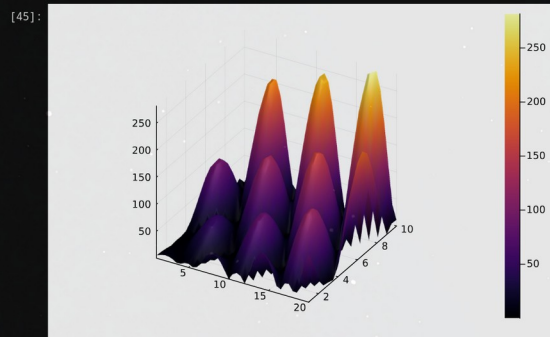


```
[43]: x=range(-2,stop=2,length=100)
y=range(sqrt(2),stop=2,length=100)
f(x,y) = x*y-x-y+1
plot(x,y,f,
      linestyle = :surface,
      c=cgrad([:red,:blue]),
      camera=(-30,30),
    )
```

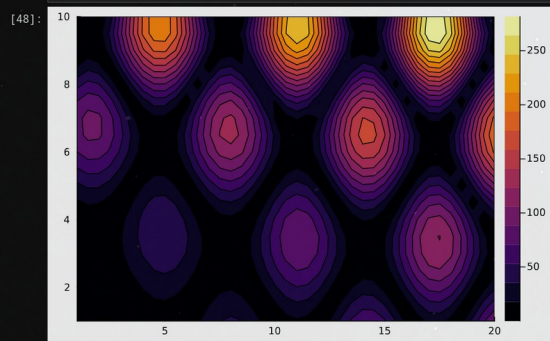
[43]:



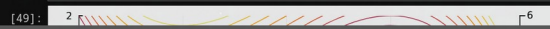
```
[45]: x = 1:0.5:20
y = 1:0.5:10
h(x, y) = (3x + y ^ 2) * abs(sin(x) + cos(y))
plot(x,y,h,
linetype = :surface,
)
```



```
[48]: p = contour(x, y, h,
fill=true)
plot(p)
```



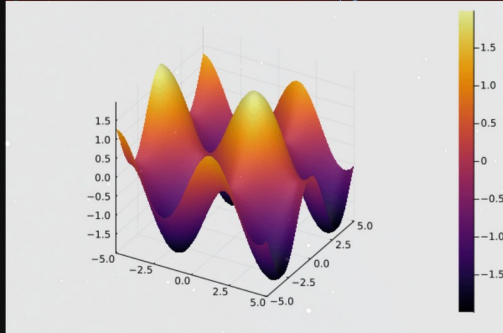
```
[49]: # определение переменных:
X = range(-2, stop=2, length=100)
Y = range(-2, stop=2, length=100)
# определение функции:
h(x, y) = x^3 - 3x + y^2
# построение поверхности:
plot(X,Y,h,
linetype = :surface
)
# построение линий уровня:
contour(X, Y, h)
```



```
# 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 /home/starovoid/work/tmp.gif

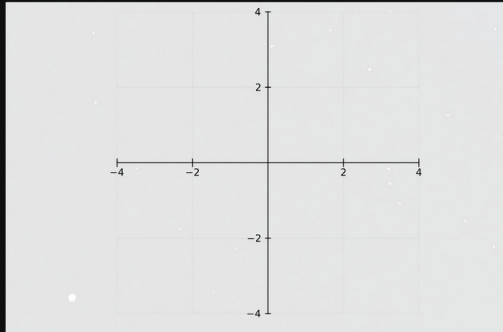
[52]:



[56]:

```
# радиус малой окружности:
rr = 1
# коэффициент для построения большой окружности:
k = 3
# число отсчетов:
n = 100
# массив значений угла θ:
# theta from 0 to 2π ( + a little extra)
θ = collect(0:2*π/100:2*π+2*π/100)
# массивы значений координат:
X = rr*k*cos. (θ)
Y = rr*k*sin. (θ)
# задаём оси координат:
plt=plot(S,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1,
legend=false, framestyle=:origin)
```

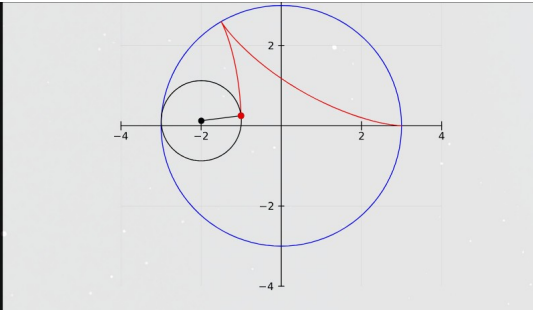
[56]:



[57]:

```
# большая окружность:
plot!(plt, X,Y, c=:blue, legend=false)
i = 50
```

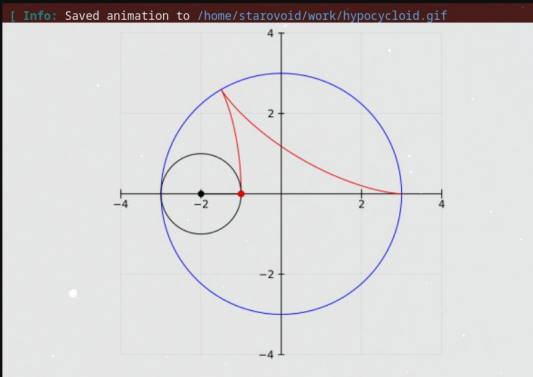


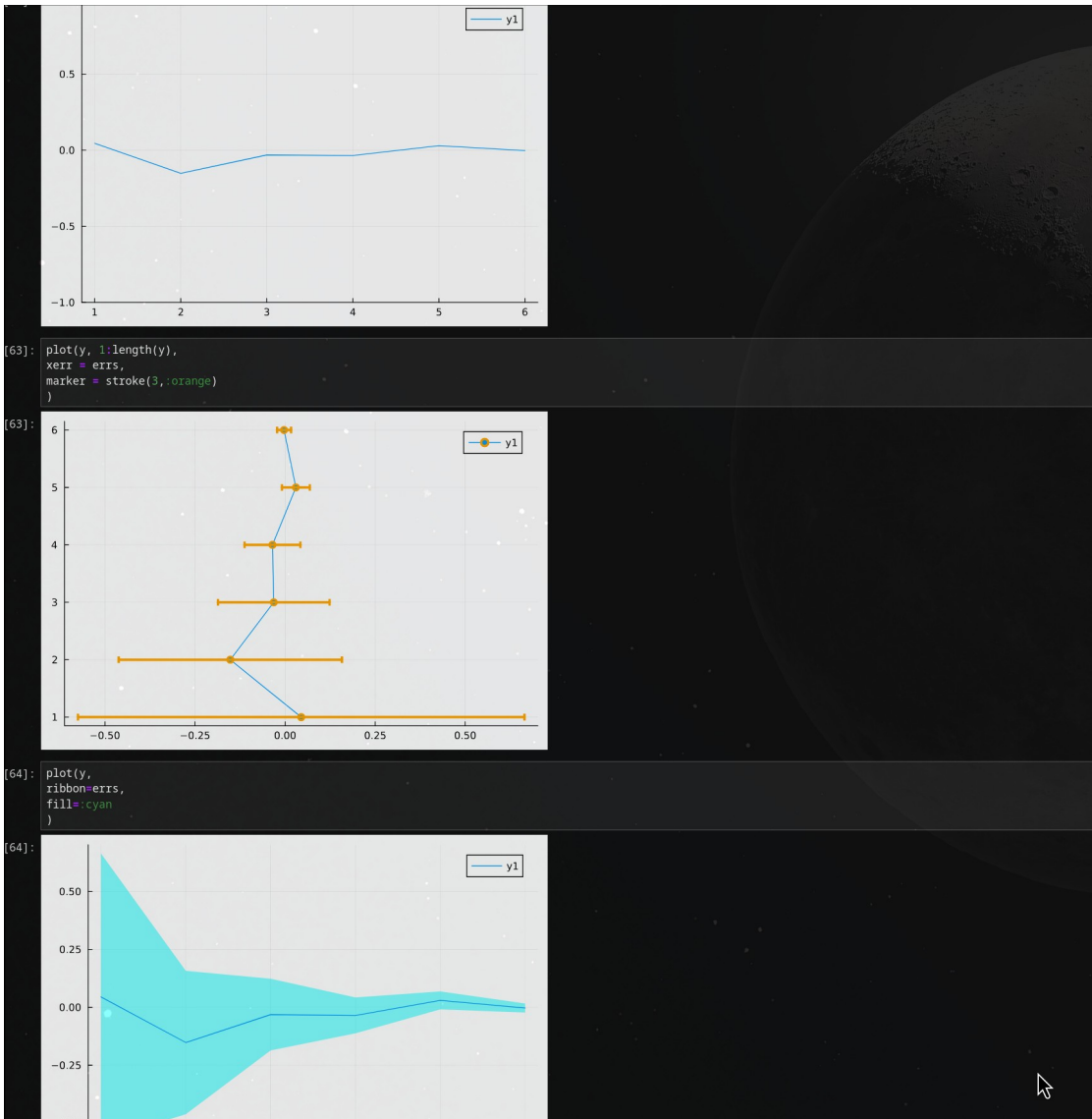


```
anim = @animate for i in 1:n
# задаём оси координат:
plt=plt.plot(5,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1,legend=false, framestyle=:origin)
# большая окружность:
plot!(plt, X,Y, c=:blue, legend=false)
t = 0[1:i]
# гипотенуза:
x = rr*(k-1)*cos.(t) + rr*cos.((k-1)*t)
y = rr*(k-1)*sin.(t) - rr*sin.((k-1)*t)
plot!(x,y, c=:red)
# малая окружность:
xc = rr*(k-1)*cos(t[end]) + rr*cos.(0)
yc = rr*(k-1)*sin(t[end]) - rr*sin.(0)
plot!(xc,yc,c=:black)
# радиус малой окружности:
x1 = transpose([rr*(k-1)*cos(t[end]) x[end]])
y1 = transpose([rr*(k-1)*sin(t[end]) y[end]])
plot!(x1,y1,markershape=:circle,markersize=4,c=:black)
scatter!([x[end]], [y[end]], c=:red, markerstrokecolor=:red)
end

Animation("/tmp/jl_EsAGCz", ["000001.png", "000002.png", "000003.png", "000004.png", "000005.png", "000006.png", "000007.png", "000008.png", "000009.png", "000010.png" - "000091.png", "000092.png", "000093.png", "000094.png", "000095.png", "000096.png", "000097.png", "000098.png", "000099.png", "000100.png"])

gif(anim,"hypocycloid.gif")
```





Выводы

Я освоил синтаксис языка Julia для построения графиков.

Список литературы