

Лабораторная работа № 5

Ли Тимофей Александрович, НФИбд-01-18

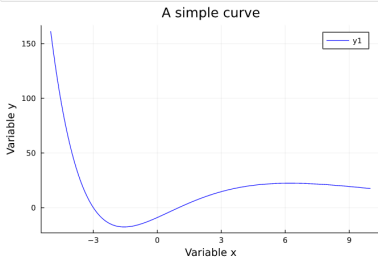
Освоить синтаксис языка Julia для построения графиков.

Ход работы. Примеры

```
f(x) = (3x.^2 + 6x - 9).*exp.(-0.3x)
x = collect(range(-5,10,length=151))
y=f(x)
```

```
151-element Vector{Float64}:
 161.34080653217032
```

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



```
pyplot()
plot(x,y,
      title="Простая кривая",
      xlabel="Переменная x",
      ylabel="Переменная y",
      color="blue")
```

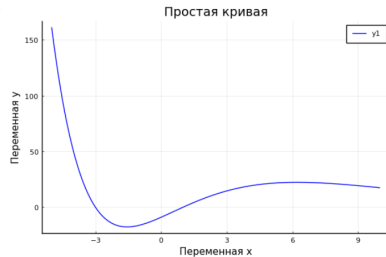
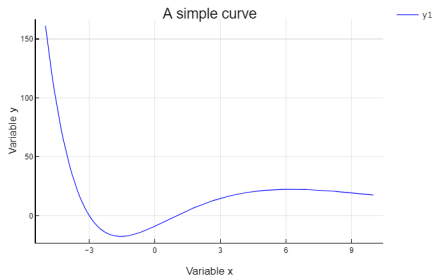


Рис. 1: примеры1

Ход работы. Примеры

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

[Info: For saving to png with the Plotly backend PlotlyBase has to be install
@ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\backends.jl:432



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

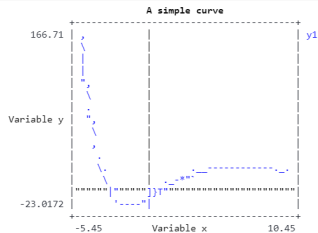


Рис. 2: примеры2

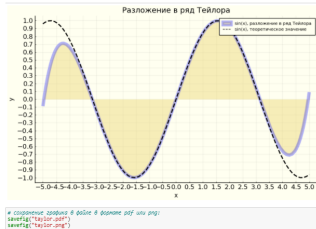
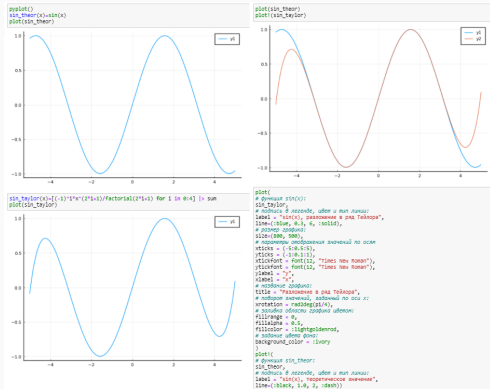


Рис. 3: примеры3

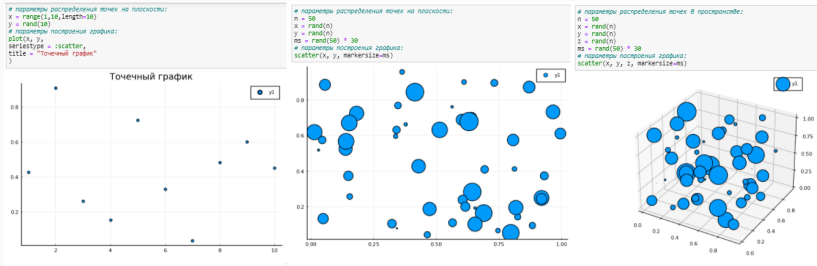
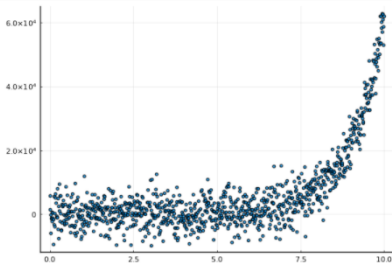


Рис. 4: примеры4

Ход работы. Примеры

```
# 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)
```



```
# 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:  
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)
```

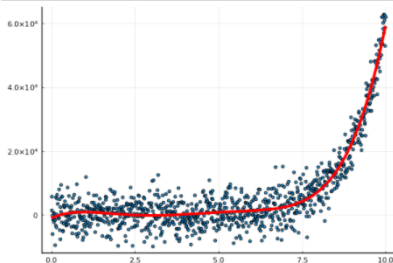
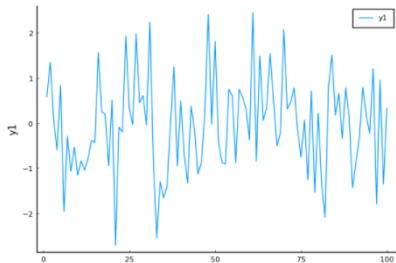


Рис. 5: примеры

Ход работы. Примеры

```
# пример случайной траектории
# (заданы обозначение траектории, легенда вверху справа, без сетки)
plot(randn(100),
      ylabel="y1",
      leg=:topright,
      grid = :off,
      )
```



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

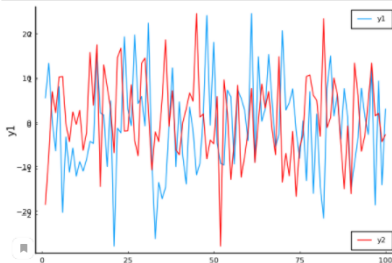


Рис. 6: примеры

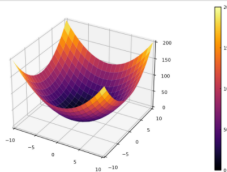
Ход работы. Примеры



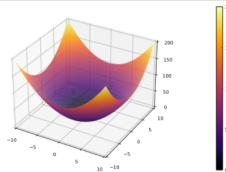
Рис. 7: примеры7

Ход работы. Примеры

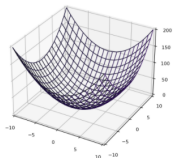
```
# построение графика поверхности:  
f(x,y) = x**2 + y**2  
x = -10:10  
y = x  
surface(x, y, f)
```



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



```
# построение графика поверхности:  
f(x,y) = x**2 + y**2  
x = -10:10  
y = x  
plot(x, y, f,  
linetype='wireframe'  
)
```



```
xrange(-2,stop=2,length=100)  
yrange(sqrt(2),stop=2,length=100)  
f(x,y) = x**y-x-y+1  
plot(x,y,f,  
linetype = 'surface',  
c=grad([red,blue]),  
camera=(-30,30),  
)
```

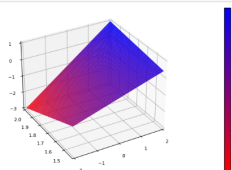
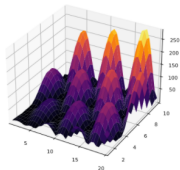


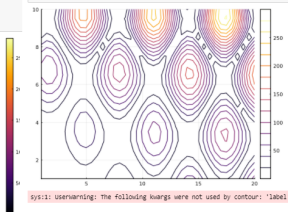
Рис. 8: примеры

Ход работы. Примеры

```
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)
```



```
p = contour(x, y, g,  
fill=True)  
plot(p)
```

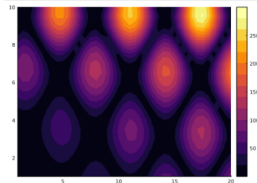


Рис. 9: примеры9

Ход работы. Примеры

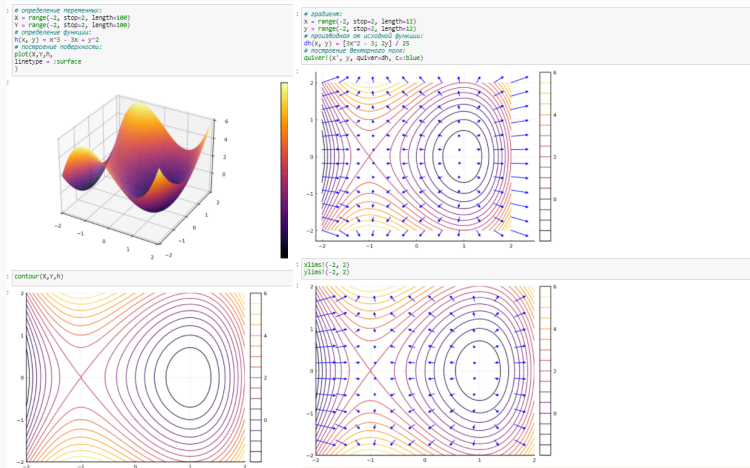


Рис. 10: примеры10

Ход работы. Примеры

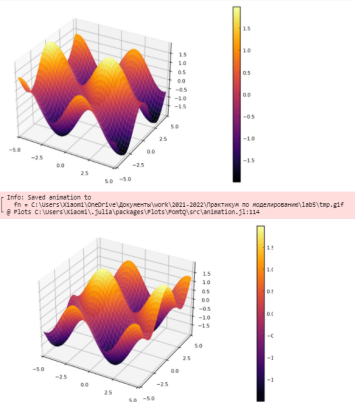
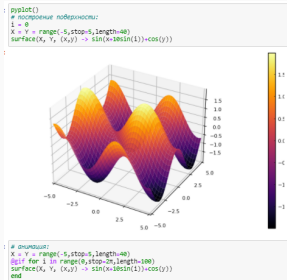


Рис. 11: примеры11

Ход работы. Примеры

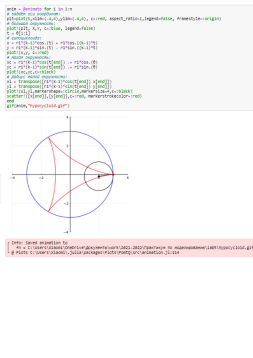
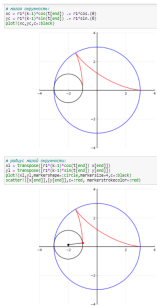
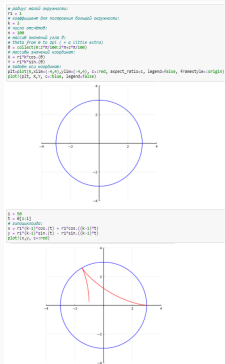


Рис. 12: примеры12

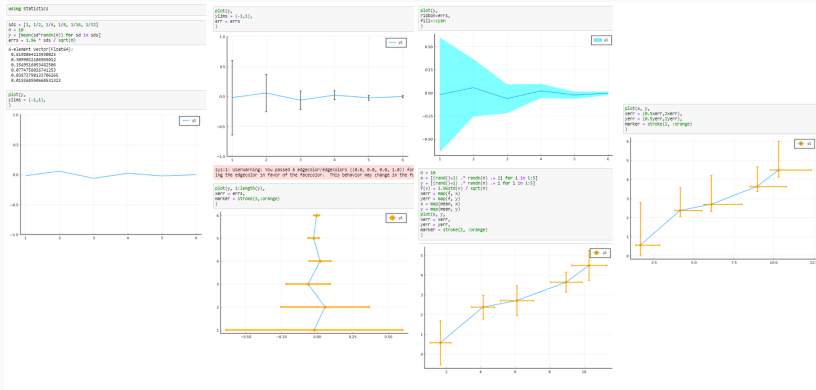


Рис. 13: примеры13

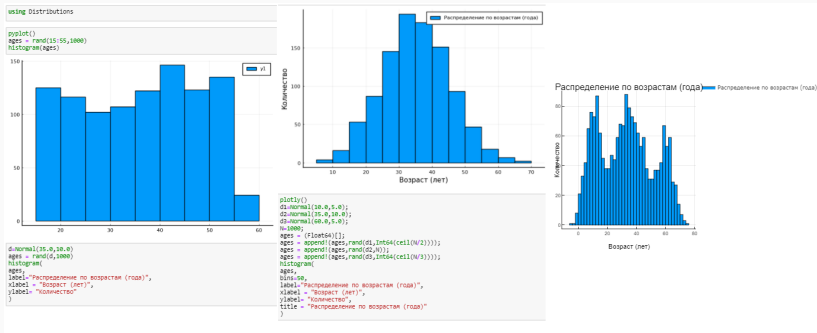


Рис. 14: примеры14

Ход работы. Примеры

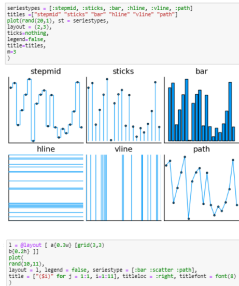


Рис. 15: примеры15

Ход работы. 1-3

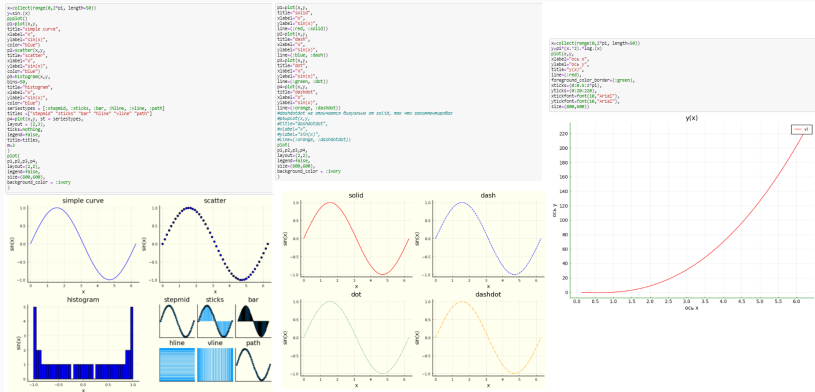


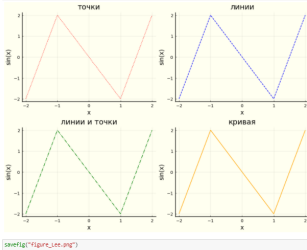
Рис. 16: номер1-3

Ход работы. 4-6

```

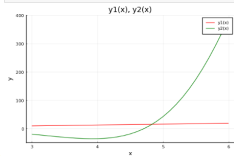
x=[-2, -1, 0, 1, 2]
y=[x**2 - 1, 3*x]
p1=plt.plot(x,y,
             title="TOWAZU",
             xlabel="x",
             ylabel="sin(x)",
             line=(red, :dot))
p2=plt.plot(x,y,
             title="KAWASA",
             xlabel="x",
             ylabel="sin(x)",
             line=(blue, :dash))
p3=plt.plot(x,y,
             title="KAWASA + TOWAZU",
             xlabel="x",
             ylabel="sin(x)",
             line=(green, :dashdot))
p4=plt.plot(x,y,
             title="KAWASA",
             xlabel="x",
             ylabel="sin(x)",
             line=(orange, :solid))
plt.show()
p1,p2,p3,p4
plt.rcParams["figure.figsize"]=(12,2)
legend=False,
size=(800,600)
background_color = 'white'

```

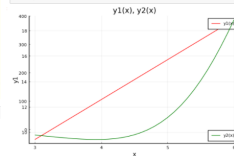


```
savefig("figure_Lee.png")
```

```
x=collect(range(3,6,step=0.1))
y1=pi*x
y2=exp.(x).*cos.(x)
plot(x,y1,
title="y1(x), y2(x)",
label="y1(x)",
xlabel="x",
ylabel="y",
line=:red, :solid))
plot(x,y2,
label="y2(x)",
line=:green, :solid))
```



```
plot(x,y1,  
title="y1(x) y2(x)",  
label="y1(x)",  
xlabel="x",  
ylabel="y1",  
leg="topright",  
line=(red, :solid))  
plot(twain(),x,y2,  
ylabel="y2",  
leg="bottomright",  
label="y2(x)",  
line=(green, :solid),  
bcoo=on)
```



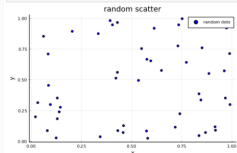
```
x = rand(10)
y = acos(x)
errs = 1.96 * x / sqrt(10)
plot(y,
      ribbon=errs,
      fill='cyan'
```



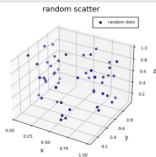
Рис. 17: номер4-6

Ход работы. 7-9

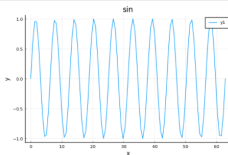
```
x = rand(50)
y = rand(50)
scatter(x,y,
    title="random scatter",
    xlabel="x",
    ylabel="y",
    label="random dots",
    color="blue")
```



```
x = rand(50)
y = rand(50)
scatter3(x,y,z,
    title="random scatter",
    xlabel="x",
    ylabel="y",
    zlabel="z",
    label="random dots",
    color="blue")
```



```
n=100
x=collect(range(0,2pi, length=n))
y=sin(x)
plot(title="sin",
    xlabel="x",
    ylabel="sin(x)",
    xs=[1:1],
    ys=[1:1],
    plot(x,y))
end
gif(anim,"sinusoid.gif")
```



Info: saved animation to
fn = C:\Users\VIKAS\OneDrive\Documents\work\2021-2022\графикy\sinusoid.gif
@ Plots C:\Users\VIKAS\julia\packages\Plots\src\animation\julia14

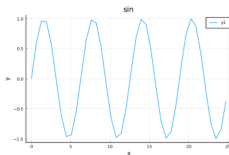


Рис. 18: номер7-9

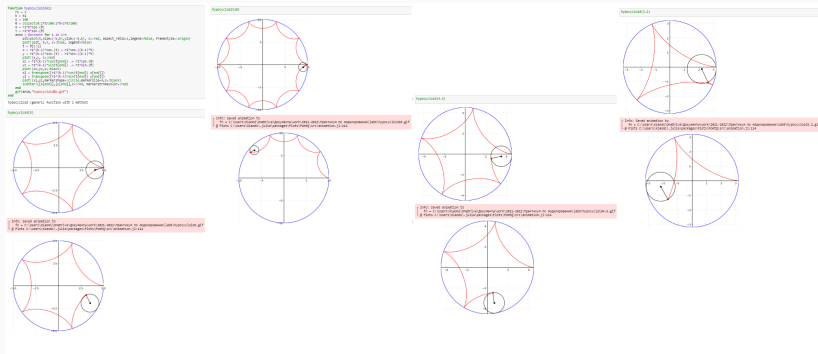


Рис. 19: номер10

Ход работы. 4

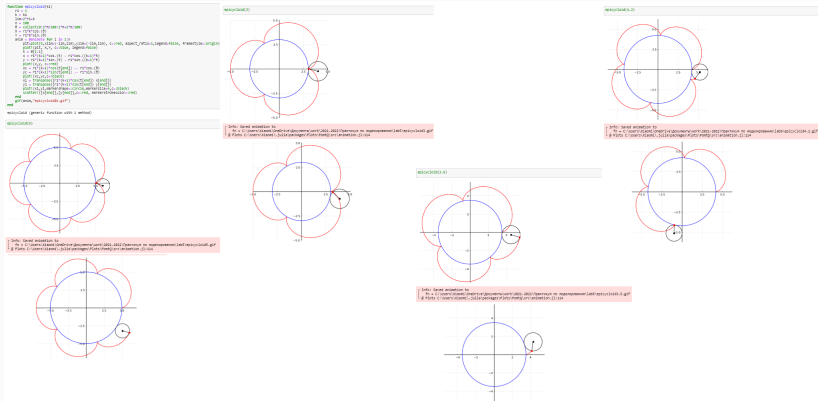


Рис. 20: номер11

Освоил синтаксис языка Julia для построения графиков.