Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий Высшая школа «Киберфизических систем и управления»

УДК

УТВЕРЖДАЮ



« » г.

ОТЧЕТ

по дисциплине «Практикум по программированию»

**Лабораторная работа № 1**

# Выполнил:

студент гр. 3530902/00002 А. В. Антонов

# Проверил

доцент А. А. Жиленков

Санкт-Петербург 2021г.

**Задание:**

задание 2 для самостоятельного выполнения:

• чекбоксами или в выпадающем списке выбираем одну из кривых задания 1

• с помощью слайдера перемещаем точку на выбранной кривой, при этом: из точки выходит вектор скорости и единичный вектор нормали, слайдер перемещается на интервале [min, max] заданной функции, при перемещении подсчитывается и показывается длина пройденной кривой (интеграл на заданном интервале от корня из скалярного произведения от вектора скорости

**Код программы:**

*from IPython.display import display*

*import ipywidgets as widgets*

*import numpy as np*

*import matplotlib.pyplot as plt*

*from matplotlib.widgets import Slider, Button, RadioButtons*

*%matplotlib notebook*

*fig, ax = plt.subplots(figsize=(10, 12), dpi=100)*

*plt.subplots\_adjust(left=0.2, bottom=0.3)*

*def get\_func\_value\_bot(x):*

*result = np.sqrt(np.abs(x))-np.sqrt(1-x\*x)*

*return result*

*def get\_func\_value\_top(x):*

*result = np.sqrt(np.abs(x))+np.sqrt(1-x\*x)*

*return result*

*x = np.linspace(-1, 1,1000)*

*line\_bot, = plt.plot(x, get\_func\_value\_bot(x), color='red')*

*line\_top, = plt.plot(x, get\_func\_value\_top(x), color='red')*

*t = np.linspace(-2, 2, 1000)*

*x0 = -1*

*y0 = get\_func\_value\_bot(x0)*

*point, = plt.plot(x0,y0,'ro', markersize=10)*

*delta = 0.0001*

*k=(get\_func\_value\_bot(x0+delta)-get\_func\_value\_bot(x0))/((x0+delta)-x0)*

*b=get\_func\_value\_bot(x0+delta)-(get\_func\_value\_bot(x0+delta)-get\_func\_value\_bot(x0))/((x0+delta)-x0)\*(x0+delta)*

*xk=t*

*yk=k\*t+b*

*tangent, = plt.plot(xk,yk)*

*t1 = np.linspace(-1, 1, 1000)*

*xn=t1*

*yn=get\_func\_value\_bot(x0)-(1/k)\*(t1-x0)*

*normal, = plt.plot(xn,yn)*

*plt.axis([-2, 2, -2, 2])*

*ax.grid(axis = 'both')*

*axcolor = 'lightgray'*

*axt1 = plt.axes([0.25, 0.2, 0.65, 0.03], facecolor=axcolor)*

*st1 = Slider(axt1, 'X', -1, 1, valinit=x0)*

*reset\_ax = plt.axes([0.02, 0.025, 0.1, 0.04])*

*reset\_button = Button(reset\_ax, 'Reset', color=axcolor, hovercolor='0.975')*

*def reset(event):*

*st1.reset()*

*reset\_button.on\_clicked(reset)*

*def update\_top(val):*

*x0 = st1.val*

*point.set\_data(x0, get\_func\_value\_top(x0))*

*k=(get\_func\_value\_top(x0+delta)-get\_func\_value\_top(x0))/((x0+delta)-x0)*

*b=get\_func\_value\_top(x0+delta)-(get\_func\_value\_top(x0+delta)-get\_func\_value\_top(x0))/((x0+delta)-x0)\*(x0+delta)*

*yk = k\*t+b*

*tangent.set\_ydata(yk)*

*yn = get\_func\_value\_top(x0)-(1/k)\*(t1-x0)*

*normal.set\_ydata(yn)*

*fig.canvas.draw\_idle()*

*def update\_bot(val):*

*x0 = st1.val*

*point.set\_data(x0, get\_func\_value\_bot(x0))*

*k=(get\_func\_value\_bot(x0+delta)-get\_func\_value\_bot(x0))/((x0+delta)-x0)*

*b=get\_func\_value\_bot(x0+delta)-(get\_func\_value\_bot(x0+delta)-get\_func\_value\_bot(x0))/((x0+delta)-x0)\*(x0+delta)*

*yk = k\*t+b*

*tangent.set\_ydata(yk)*

*yn = get\_func\_value\_bot(x0)-(1/k)\*(t1-x0)*

*normal.set\_ydata(yn)*

*fig.canvas.draw\_idle()*

*button\_top = widgets.Button(description="top")*

*output\_top = widgets.Output()*

*display(button\_top, output\_top)*

*def on\_button\_clicked\_top(b):*

*with output\_top:*

*line\_bot.set\_color('red')*

*line\_top.set\_color('green')*

*fig.canvas.draw\_idle()*

*st1.on\_changed(update\_top)*

*button\_top.on\_click(on\_button\_clicked\_top)*

*button\_bot = widgets.Button(description="bot")*

*output\_bot = widgets.Output()*

*display(button\_bot, output\_bot)*

*def on\_button\_clicked\_bot(b):*

*with output\_bot:*

*line\_bot.set\_color('green')*

*line\_top.set\_color('red')*

*fig.canvas.draw\_idle()*

*st1.on\_changed(update\_bot)*

*button\_bot.on\_click(on\_button\_clicked\_bot)*

*plt.show()*

# Пример работы программы:



