**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 1**

Тема: Построение изображений 2D-кривых

Студент: Лошманов Юрий Андреевич

Группа: 80-306

Преподаватель: Чернышов Л.Н.

Дата: 21.09.2021

Оценка:

Москва, 2021

1. Постановка задачи

Написать и отладить программу, строящую изображение заданной замечательной кривой.

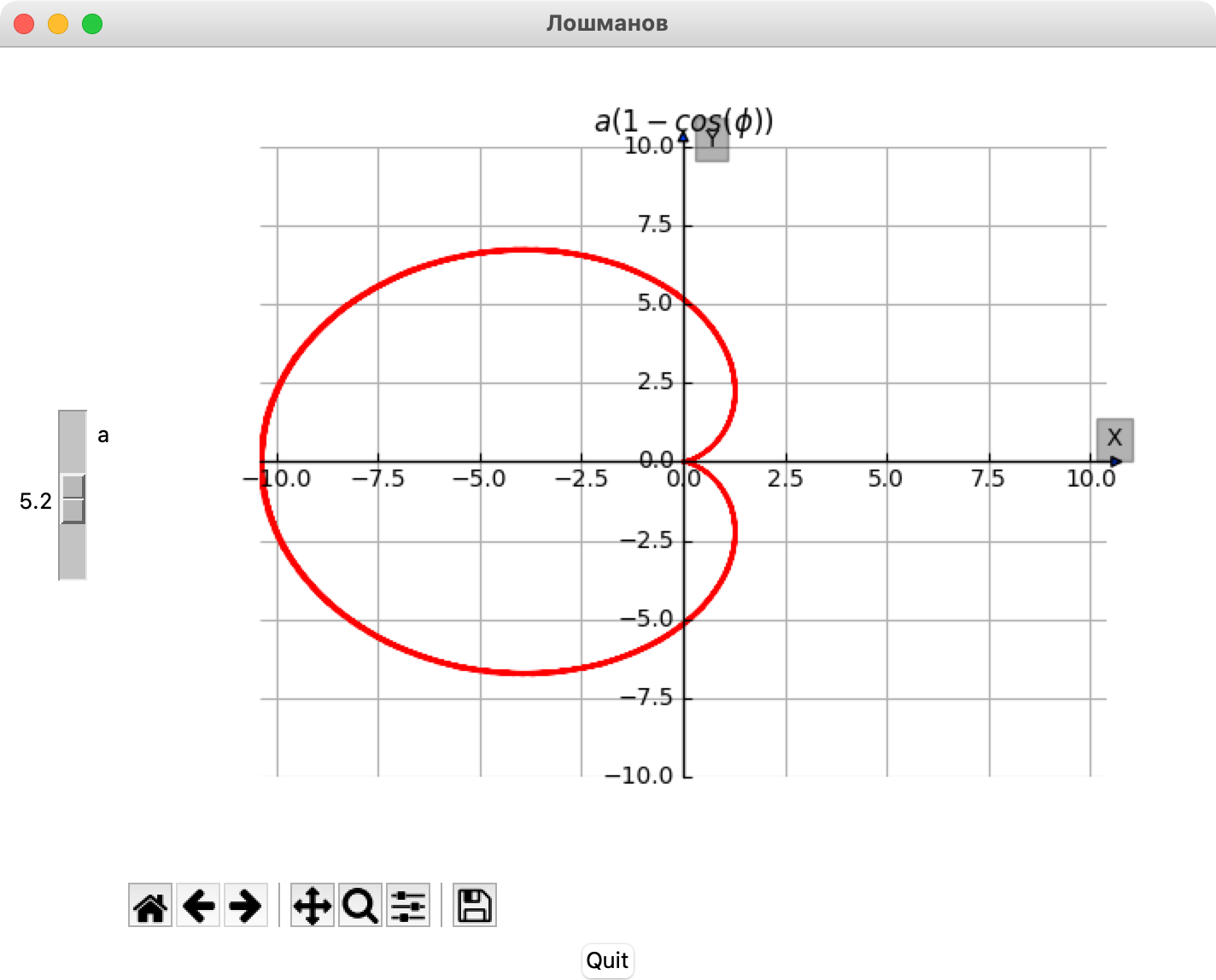
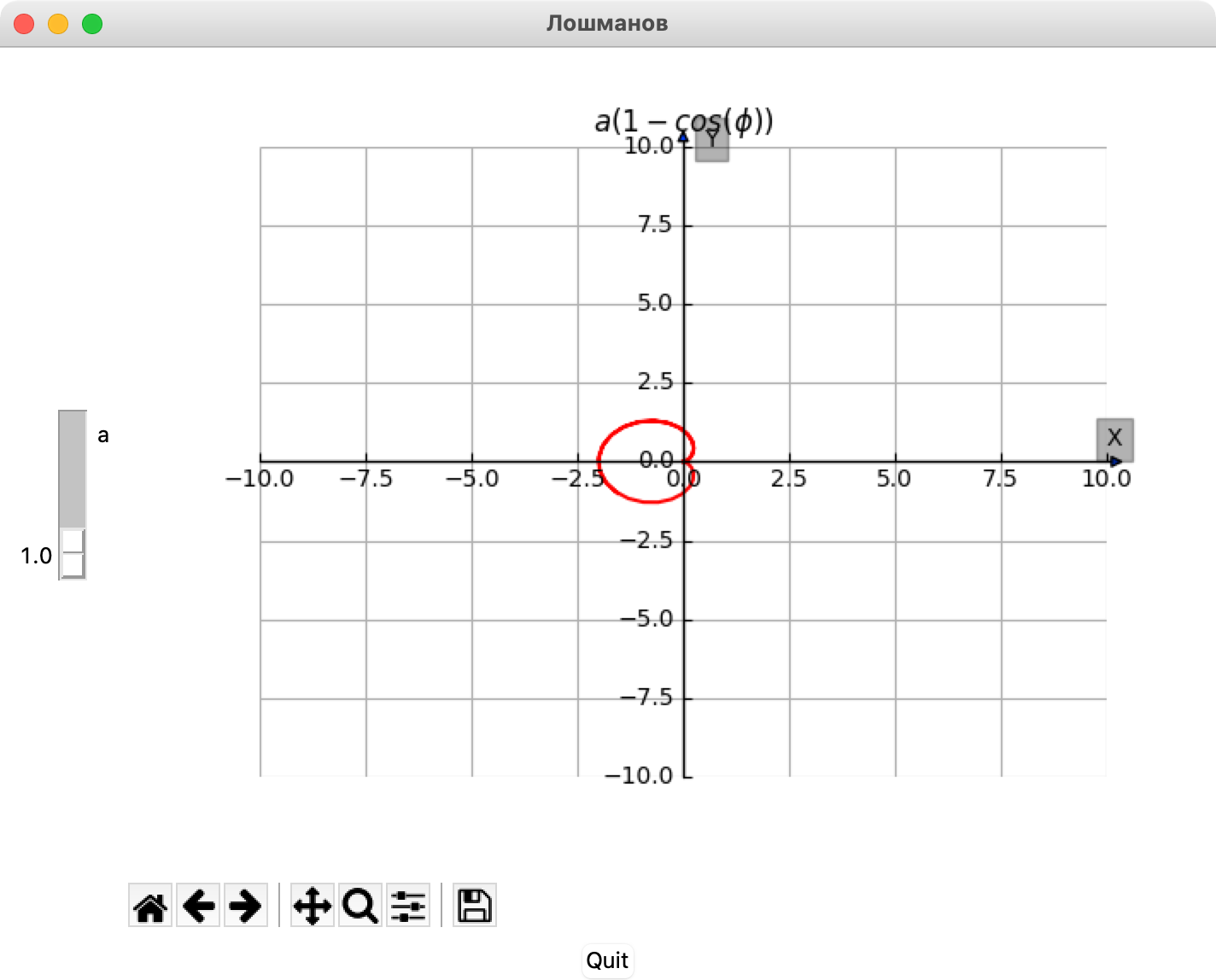
Вариант 14:

1. Описание программы

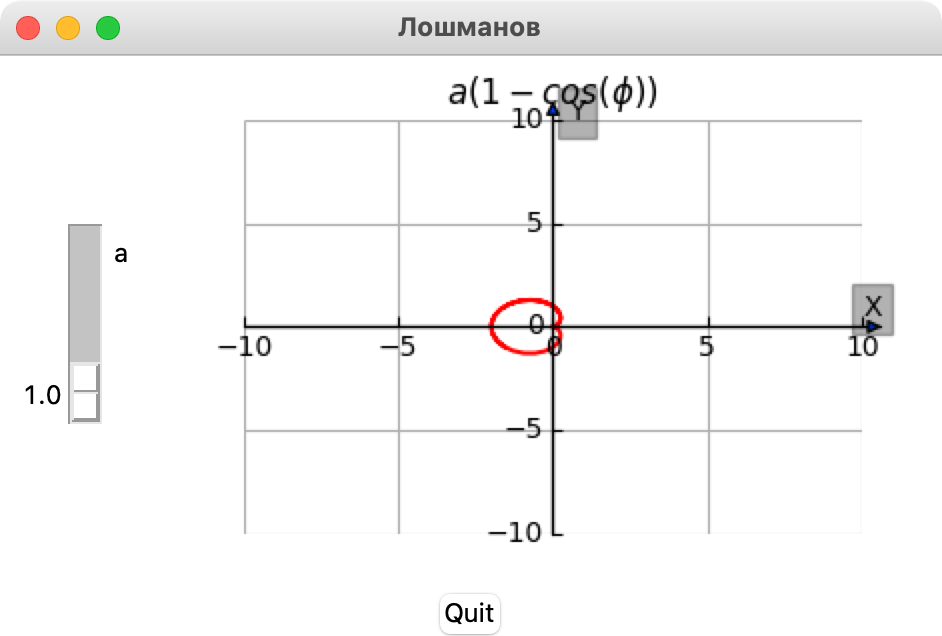
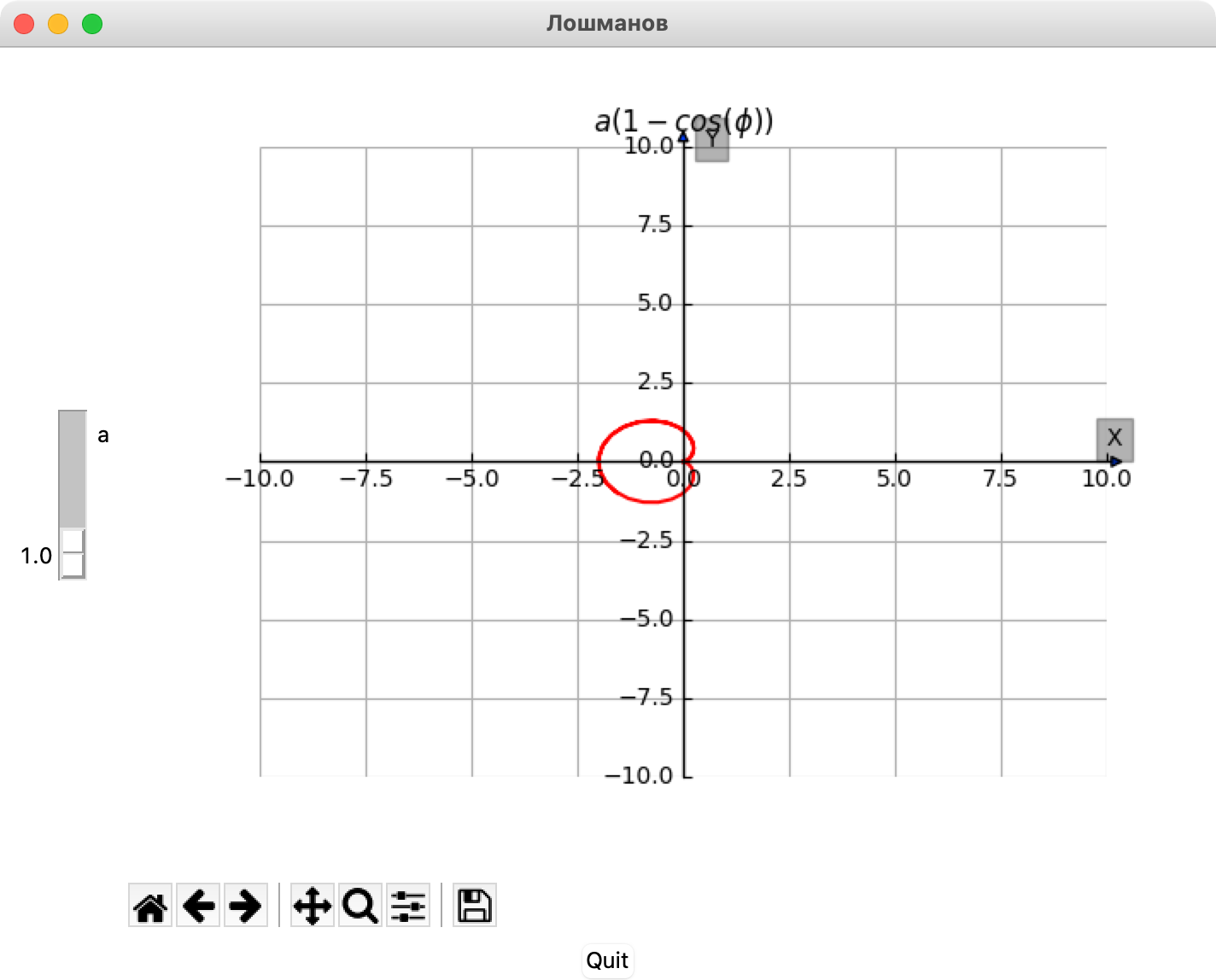
Программа состоит из слайдера для регулировки параметра a, из координатных осей с графиком и из навигационной панели для перемещения по графику.

1. Набор тестов
2. Изменение параметра a
3. Изменение размера окна
4. Перемотка графика
5. Изменение масштаба графика
6. Результаты выполнения тестов

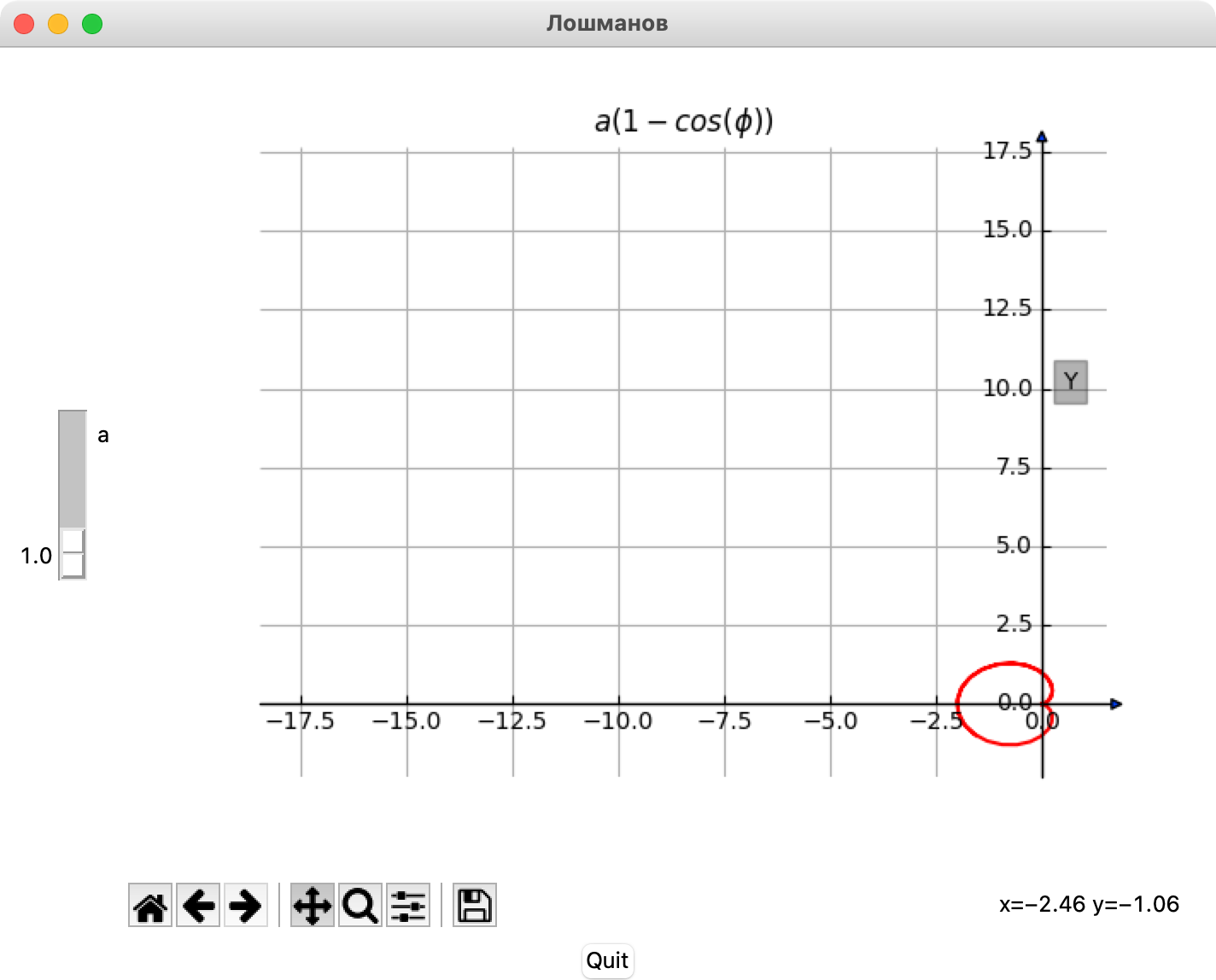
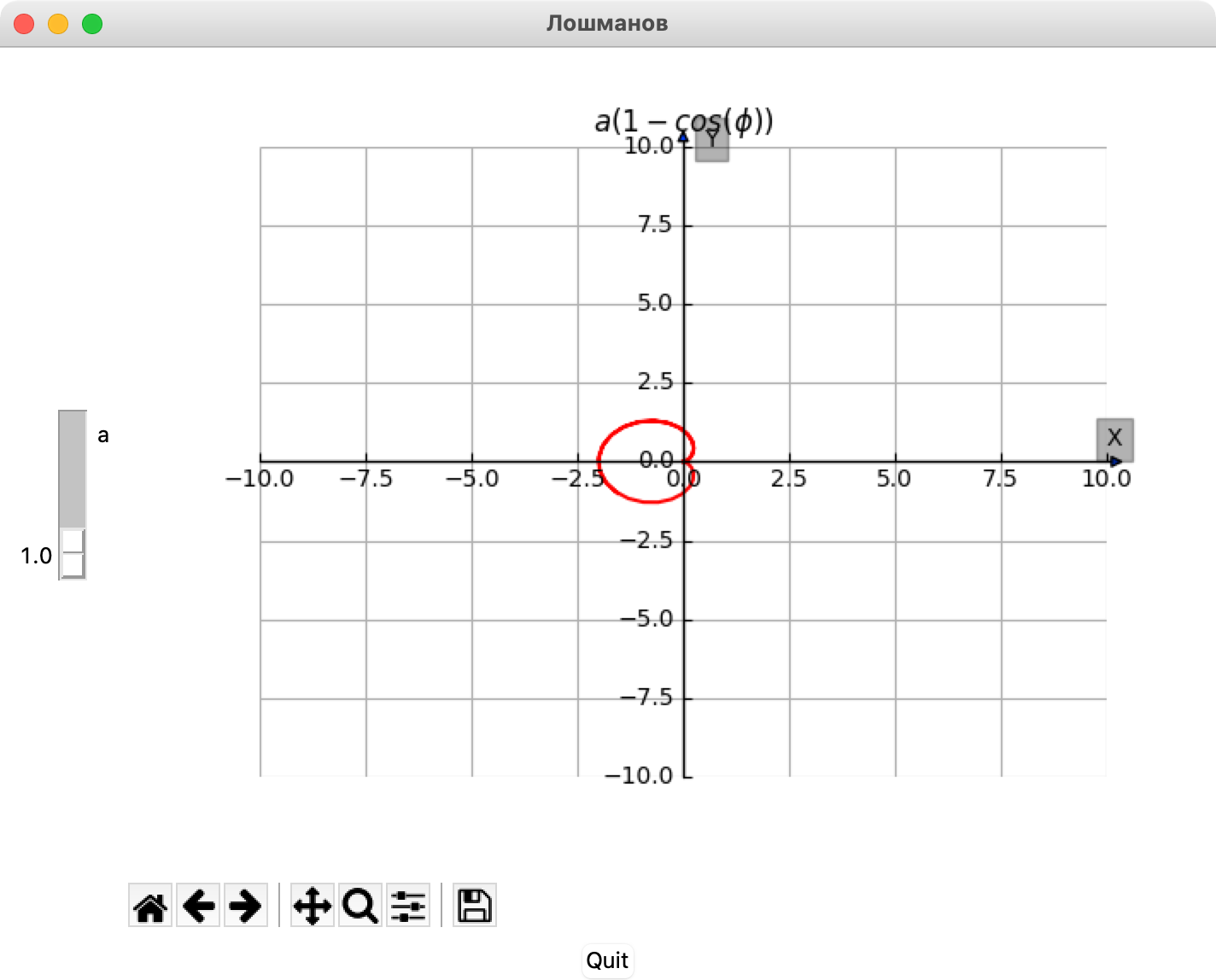
1.



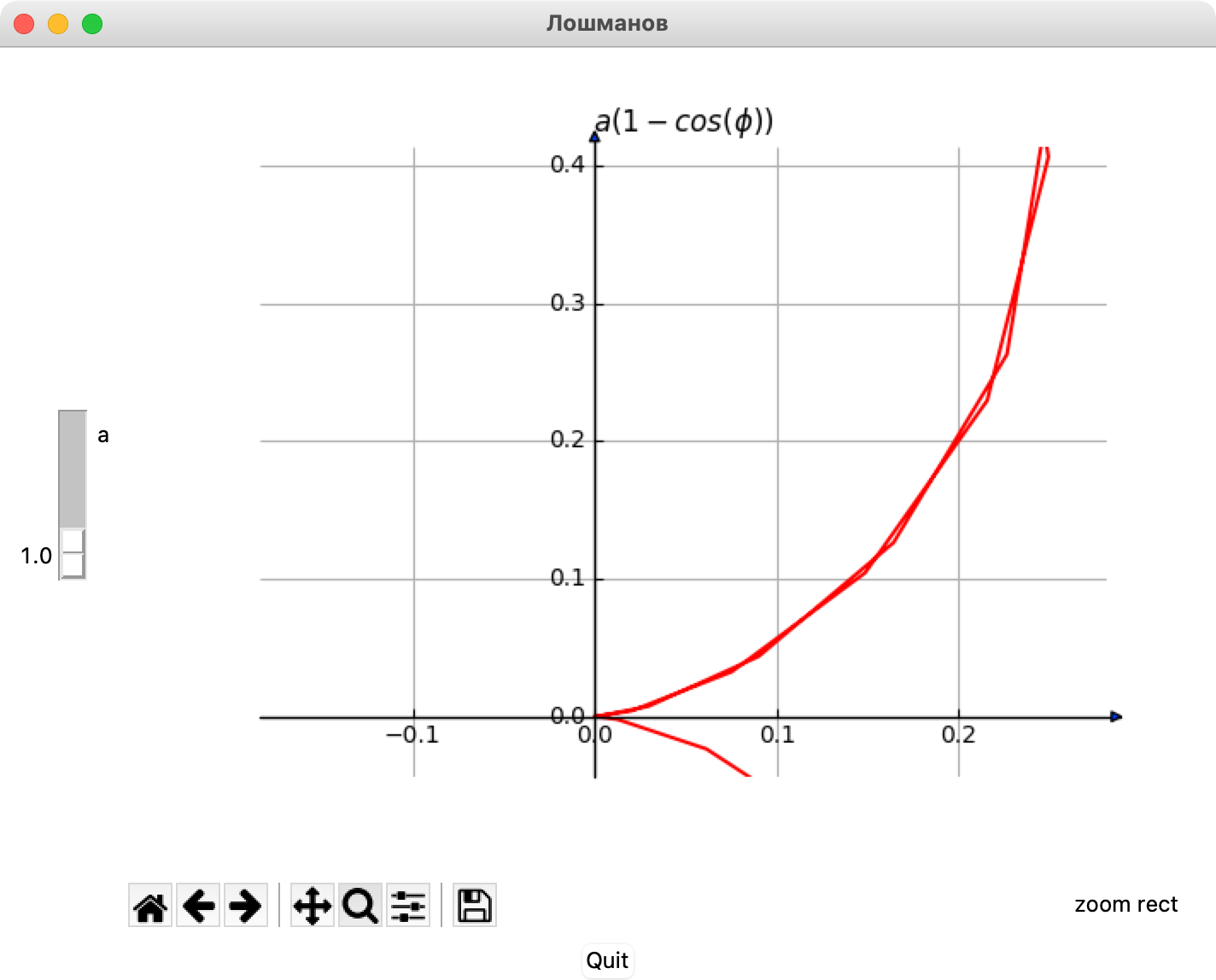
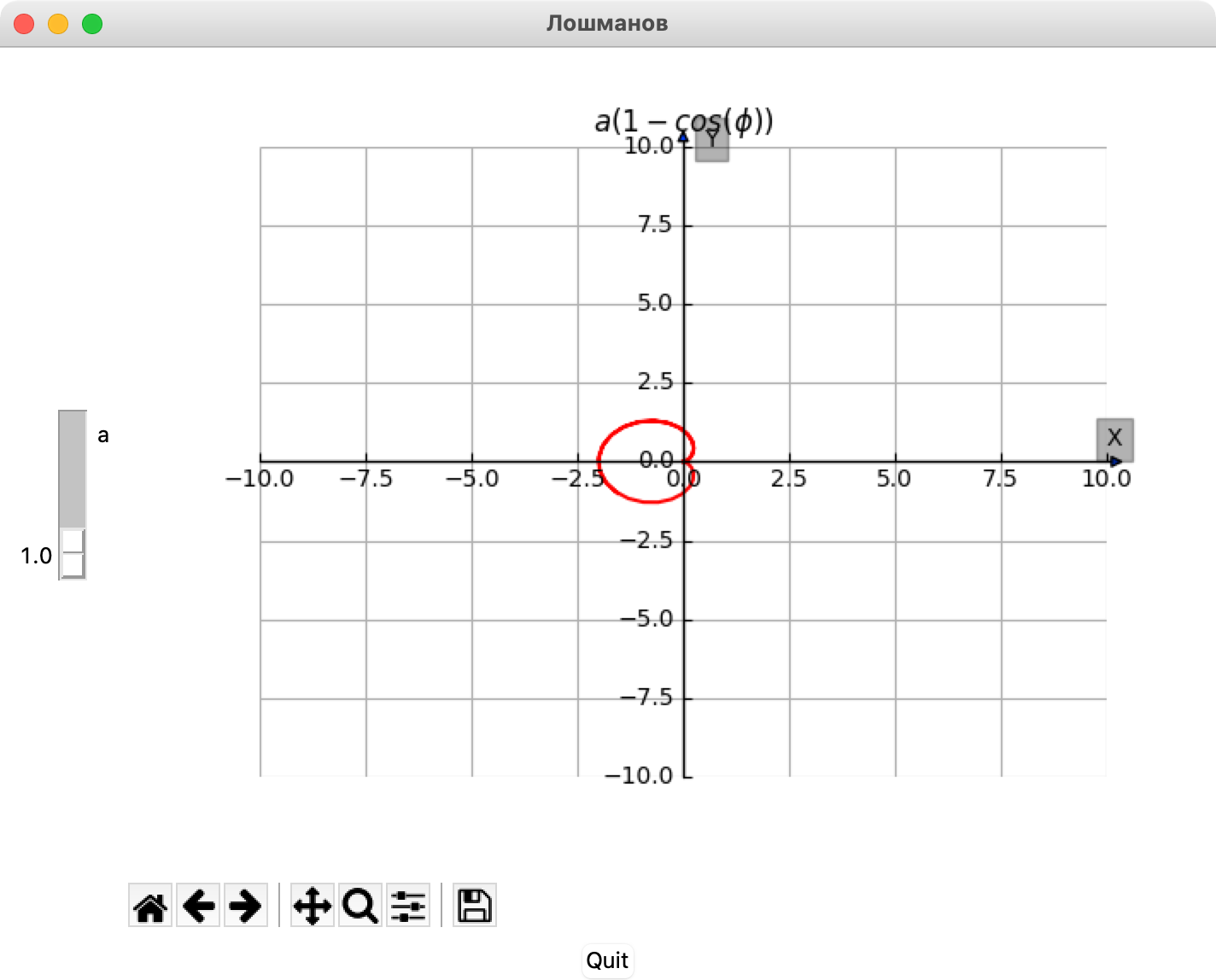
2.



3.



4.



1. Листинг программы

# Лошманов Юрий Андреевич

import numpy as np

import tkinter as tk

import matplotlib.pyplot as plt

from matplotlib.backends.\_backend\_tk import NavigationToolbar2Tk

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from mpl\_toolkits.axisartist import SubplotZero

import random

class App(tk.Tk):

def \_\_init\_\_(self):

tk.Tk.\_\_init\_\_(self)

self.title('Лошманов')

button = tk.Button(self, text='Quit', command=self.on\_click)

button.pack(side=tk.BOTTOM)

self.a = tk.DoubleVar()

self.a.set(1.0)

slider\_a = tk.Scale(self, from\_=10, to=1, resolution=0.1,

label='a', variable=self.a,

command=self.on\_change)

slider\_a.pack(side=tk.LEFT, padx = 0, fill=None, expand=False)

x, y = self.data(self.a.get())

self.fig, self.ax = self.configure\_plot()

xBorder, yBorder = self.calculate\_borders(x, y)

xlim = (-xBorder, xBorder)

ylim = (-yBorder, yBorder)

self.change\_limits(xlim, ylim)

self.draw\_text(xBorder, yBorder)

self.line1, = self.ax.plot(x, y, color='red')

self.canvas = FigureCanvasTkAgg(self.fig, master=self)

self.canvas.get\_tk\_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=1)

toolbar = NavigationToolbar2Tk(self.canvas, self)

toolbar.update()

self.canvas.draw()

def on\_click(self):

self.quit()

def on\_change(self, value):

x, y = self.data(self.a.get())

xBorder, yBorder = self.calculate\_borders(x, y)

xlim = (-xBorder, xBorder)

ylim = (-yBorder, yBorder)

self.change\_limits(xlim, ylim)

self.line1.set\_data(x, y)

for i in range(len(self.ax.texts)):

self.ax.texts[i].set\_visible(False)

self.draw\_text(xBorder, yBorder)

self.canvas.draw()

@staticmethod

def change\_limits(xlim, ylim):

plt.xlim(xlim)

plt.ylim(ylim)

@staticmethod

def calculate\_borders(x, y):

xmin = -10

xmax = 10

ymin = 0

ymax = 10

xBorder = max(abs(min(min(x), xmin)), abs(max(max(x), xmax)))

yBorder = max(abs(min(min(y), ymin)), abs(max(max(y), ymax)))

return xBorder, yBorder

def draw\_text(self, xBorder, yBorder):

self.ax.text(xBorder, yBorder / 20, 'X', size=10, bbox=dict(facecolor='black', alpha=0.3))

self.ax.text(xBorder / 20, yBorder, 'Y', size=10, bbox=dict(facecolor='black', alpha=0.3))

@staticmethod

def configure\_plot():

fig = plt.figure(1)

plt.style.use('seaborn-bright')

ax = SubplotZero(fig, 111)

fig.add\_subplot(ax)

for direction in ['xzero', 'yzero']:

ax.axis[direction].set\_axisline\_style('-|>')

ax.axis[direction].set\_visible(True)

for direction in ['left', 'right', 'bottom', 'top']:

ax.axis[direction].set\_visible(False)

plt.grid()

plt.title('$a(1-cos(\phi))$')

return fig, ax

@staticmethod

def data(a):

Phi = np.linspace(0, int(a) \* 10, int(a) \* 50)

r = a \* (1 - np.cos(Phi))

x = r \* np.cos(Phi)

y = r \* np.sin(Phi)

return x, y

if \_\_name\_\_ == '\_\_main\_\_':

app = App()

app.mainloop()

1. Выводы

Выполнив данную лабораторную работу, я научился пользоваться библиотекой Tkinter, Matplotlib и Numpy, запускать графический интерфейс, перерисовывать его в зависимости от изменения окна и отрисовывать график по массиву точек.

ЛИТЕРАТУРА

1. Документация Tkinter[Электронный ресурс]URL: <https://docs.python.org/3/library/tkinter.html> (Дата обращения: 21.09.2021).
2. Документация Matplotlib[Электронный ресурс]URL: <https://matplotlib.org> (Дата обращения: 21.09.2021).