МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ТЭВН

Лабораторная работа №4

“Отображение графической информации в Python ”

Вариант № 19

Факультет: ФЭН

Группа: ЭН2-31

Студент: Ситников Р.И.

Преподаватель: Петрищев А.В.

Новосибирск 2023

# 1.Цель работы

Знакомство с возможностями работы с массивами в среде Python, и простейшими способами визуализации информации.

# 2.Задание №1

Для своего варианта функции построить график.

# 3.Пояснения к заданию №1

Функция представлена на рисунке №1.

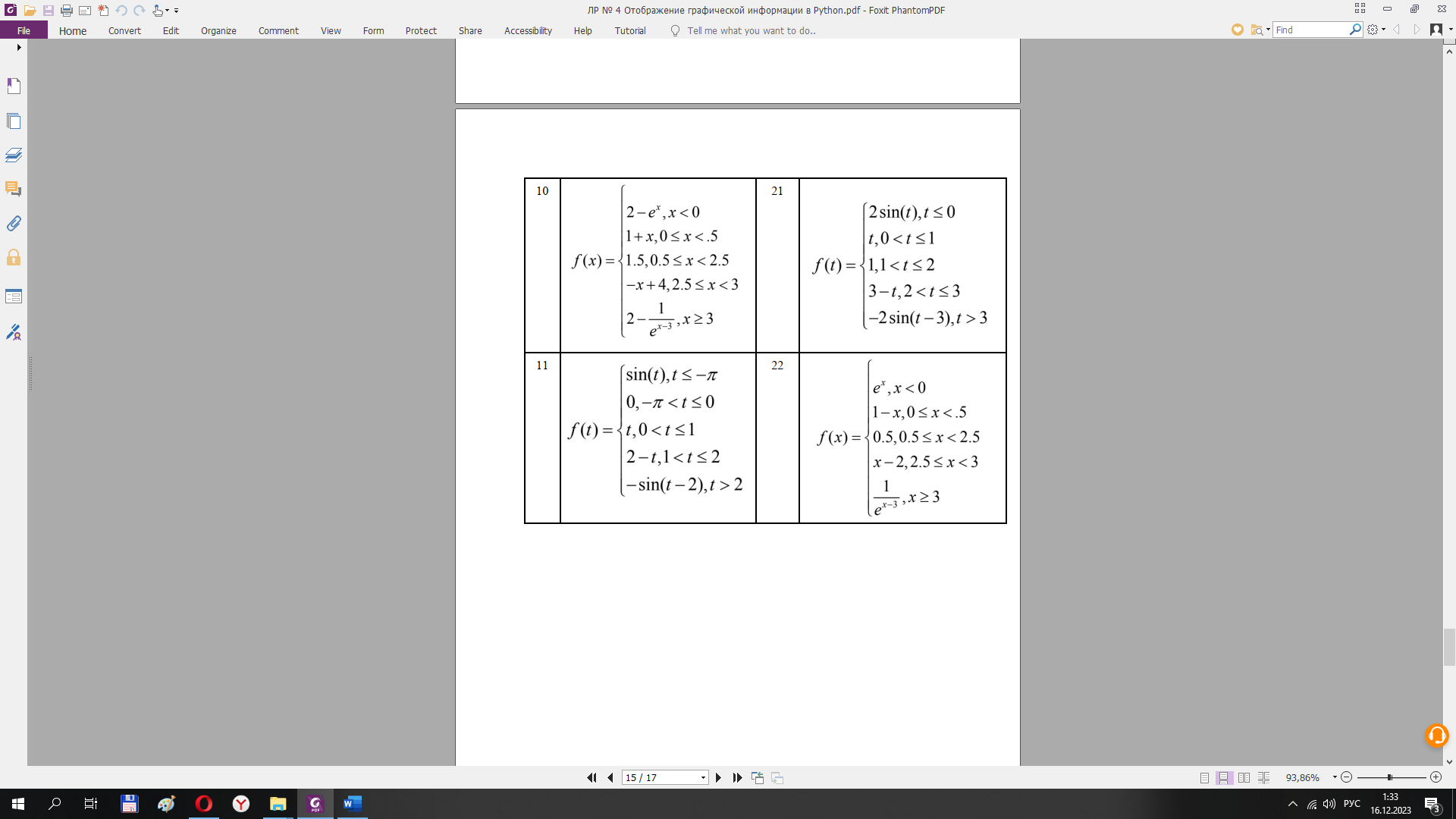


Рисунок №1 – Функция

Для построения сложной функции она была разбита на 5 простых с соответствующими ОДЗ:

Блок-схема программы представлена на рисунке №1.

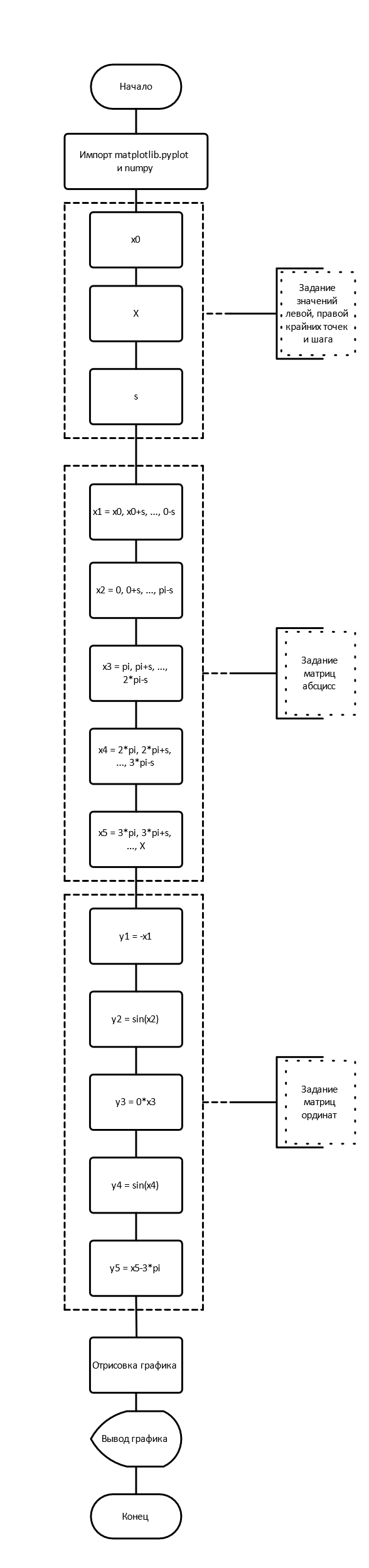


Рисунок №1 – Блок-схема программы (задание №1)

Листинг программы представлен далее.

import matplotlib.pyplot as plt

import numpy as np

x0 = -pi #левая крайняя точка

X = 4\*pi #правая крайняя точка

s = 0.001 #шаг

#задание матриц абсцисс

x1 = np.arange(x0, 0, s)

x2 = np.arange(0, pi, s)

x3 = np.arange(pi, 2\*pi, s)

x4 = np.arange(2\*pi, 3\*pi, s)

x5 = np.arange(3\*pi, X+s, s)

#задание матриц ординат

y1 = -x1

y2 = np.sin(x2)

y3 = 0\*x3

y4 = np.sin(x4)

y5 = x5 - 3\*pi

#отрисовка графика

plt.plot(x1,y1,color='blue')

plt.plot(x2,y2,color='blue')

plt.plot(x3,y3,color='blue')

plt.plot(x4,y4,color='blue')

plt.plot(x5,y5,color='blue')

plt.show()

------

import numpy as np

import matplotlib.pyplot as plt

def f(x):

if x < 0:

return np.exp(x)

elif 0 <= x < 5:

return 1 - x

elif 0.5 <= x < 2.5:

return 0.5

elif 2.5 < x < 3:

return x - 2

else:

return 1 / np.exp(x) - 3

x = np.linspace(-5, 5, 1000)

y = [f(i) for i in x]

plt.plot(x, y)

plt.xlabel('x')

plt.ylabel('f(x)')

plt.title('Graph of f(x)')

plt.grid(True)

plt.show()

В начале программа импортирует необходимые библиотеки, далее с помощью функции np.arange задаются матрицы абсцисс, далее задаются матрицы ординат (y3 задана с помощью умножения нуля на матрицу абсцисс x3, таким образом y3 тоже становится матрицой), далее отрисовываются графики с помощью полученных значений и окрашиваются единим синим цветом.

Полученный график представлен на рисунке №2.

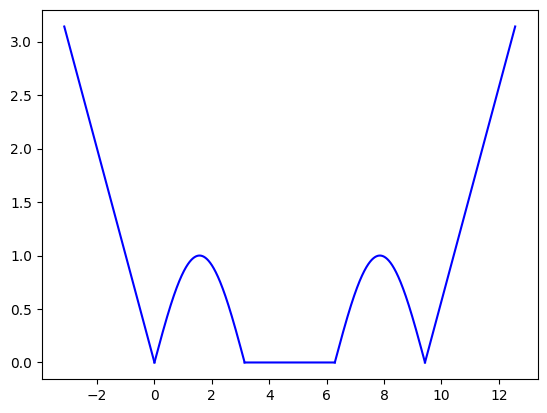


Рисунок №2 – График (задание №1)

# 4.Задание №2

Для своего варианта функции напряжения от времени построить график в определенном диапазоне значений времени. Определить значения напряжения в заданных точках временной шкалы. Отметить на графике максимальное значение напряжения.

# 5.Пояснения к заданию №2

19 варианту соответствует функция:

Диапазон значений времени принадлежит от 0 до 3π.

Значения точек, в которых нужно определить значение напряжения, равняются 3, 6, 9.

Блок-схема программы представлена на рисунке №3.

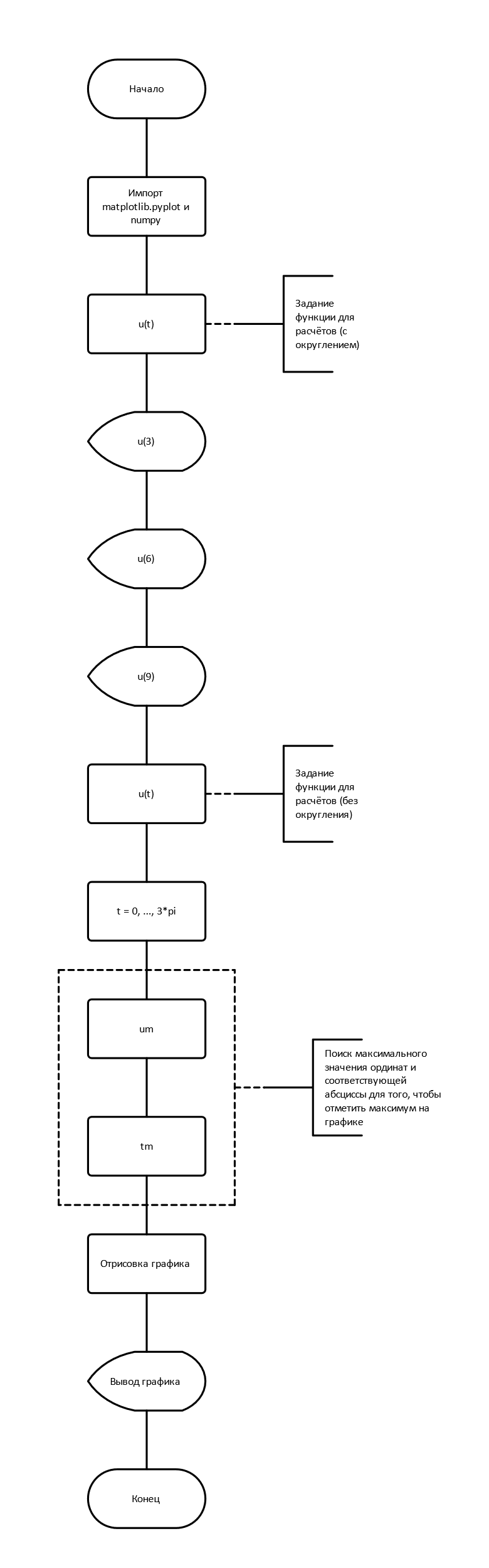


Рисунок №3 – Блок-схема программы(задание №2)

Листинг программы представлен далее.

import matplotlib.pyplot as plt

import numpy as np

e = np.e

pi = np.pi

#задание функции для расчётов

def u(t):

return round(5\*e\*\*(-0.1\*t)\*abs(np.sin(t))+11\*e\*\*(-0.35\*t)\*abs(np.cos(t)),3)

#определение значений напряжения в заданных точек

print('u(3) =', u(3))

print('u(6) =', u(6))

print('u(9) =', u(9))

#задание функции для создания матрицы

def u(t):

return 5\*e\*\*(-0.1\*t)\*abs(np.sin(t))+11\*e\*\*(-0.35\*t)\*abs(np.cos(t))

#задание матрицы абсцисс

s = 0.001

t = np.arange(0, 3\*pi+s, s)

#поиск максимального значения ординат

um = np.max(u(t))

#поиск абсциссы, соответствующей Um

tm = t[int(np.where(u(t) == Um)[0])]

#отрисовка графика

plt.plot(t,u(t))

plt.annotate("Максимум", xy=(tm, um), xytext=(tm-1, um+2), fontsize=12, arrowprops = dict(arrowstyle = '->'))#отрисовка подписи максимума

#вывод графика

plt.show()

В начале импортируются библиотеки, для удобства уточняются константы, далее задаётся функция для расчётов с округлением, далее выводятся требуемые значения, далее задаётся функция для создания матрицы без округления. Округление не используется по двум причинам:

1. к матрице его нельзя применить;
2. иначе график будет «ступенчатым».

Далее задаются матрицы абсцисс и ординат, далее происходит поиск максимального значения ординат, далее происходит поиск соответствующей абсциссы, и, наконец, отрисовывается и выводится график.

Программа выдаёт данные значения напряжения в заданных точках:

Полученный график представлен на рисунке №4.

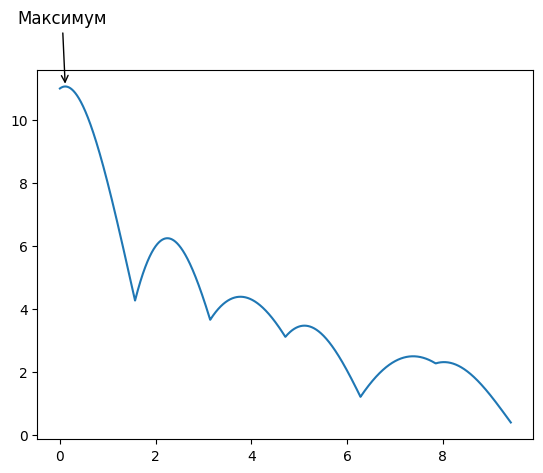


Рисунок №4 – График (задание №2)

# 6.Задание №3

Оформление графиков функций в пунктах 1 и 2 должно включать подписи по осям, легенду, заголовок, линии координатной сетки. Дополнительно установить толщину линии графика равной 2 пт.

# 7.Пояснения к заданию №3

Блок-схема программы представлена на рисунке №5.

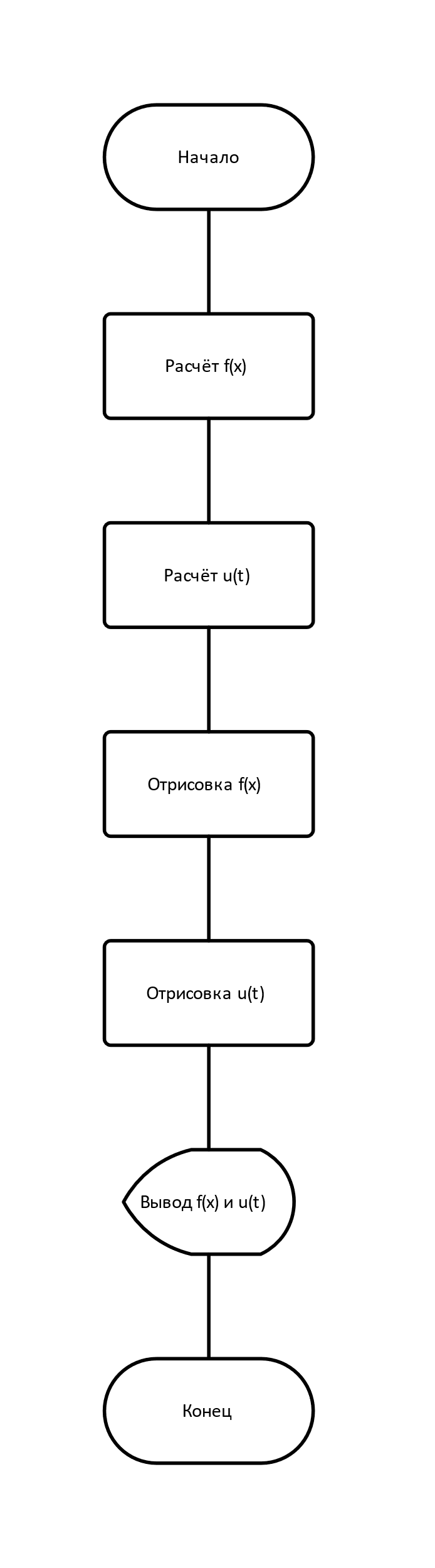


Рисунок №5 – Блок-схема программы (задание №3)

Листинг программы представлен далее.

import matplotlib.pyplot as plt

import numpy as np

e = np.e

pi = np.pi

s = 0.001

#расчёт значений для графика f(x)

x0 = -1

X = 11

x1 = np.arange(x0, 0, s)

x2 = np.arange(0, pi, s)

x3 = np.arange(pi, 2\*pi, s)

x4 = np.arange(2\*pi, 3\*pi, s)

x5 = np.arange(3\*pi, X+s, s)

y1 = -x1

y2 = np.sin(x2)

y3 = 0\*x3

y4 = np.sin(x4)

y5 = x5 - 3\*pi

#расчёт значений для графика u(t)

def u(t):

return 5\*e\*\*(-0.1\*t)\*abs(np.sin(t))+11\*e\*\*(-0.35\*t)\*abs(np.cos(t))

s = 0.001

t = np.arange(0, 3\*pi+s, s)

um = np.max(u(t))

tm = t[int(np.where(u(t) == um)[0])]

#отрисовка графика f(x)

plt.figure()

plt.plot(x1,y1,color='blue',linewidth = 2,label='f(x)')

plt.plot(x2,y2,color='blue',linewidth = 2)

plt.plot(x3,y3,color='blue',linewidth = 2)

plt.plot(x4,y4,color='blue',linewidth = 2)

plt.plot(x5,y5,color='blue',linewidth = 2)

plt.grid()

plt.legend()

plt.xlabel('x',fontsize=14)

plt.ylabel('y',fontsize=14)

plt.title('График функции f(x)')

#отрисовка графика u(t)

plt.figure()

plt.plot(t,u(t),label='u(t)')

plt.annotate("Максимум", xy=(tm, um), xytext=(tm-1, um+2), fontsize=12, arrowprops = dict(arrowstyle = '->'))

plt.grid()

plt.legend()

plt.xlabel('t',fontsize=14)

plt.ylabel('u',fontsize=14)

plt.title('График функции u(x)')

#вывод графиков

plt.show()

В начале рассчитываются значения x, y, t, u(t), далее отрисовываются графики согласно заданным условиям и выводятся.

Графики представлены на рисунках №6 и №7.

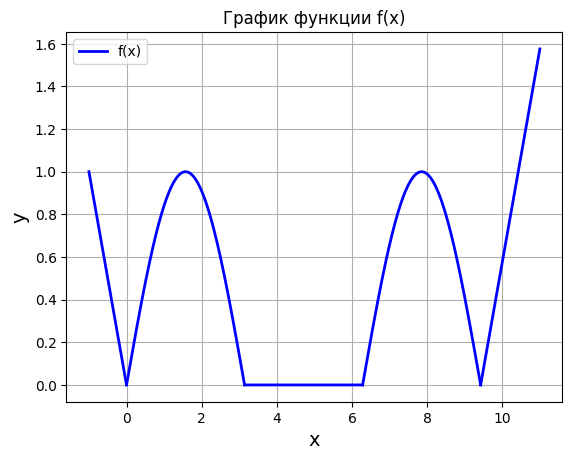


Рисунок №6 – График функции f(x)

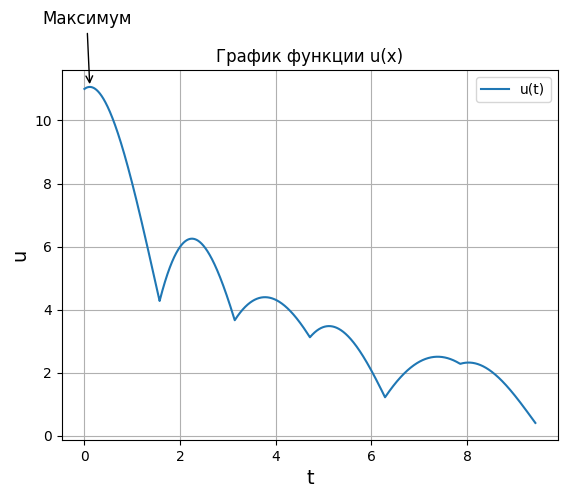


Рисунок №7 – График функции u(t)

# 8.Задание №4

Провести тестирование программы.

# 9.Пояснения к заданию №4

Для того, чтобы протестировать программы, был построен аналогичный график в Excel (рисунок №8).

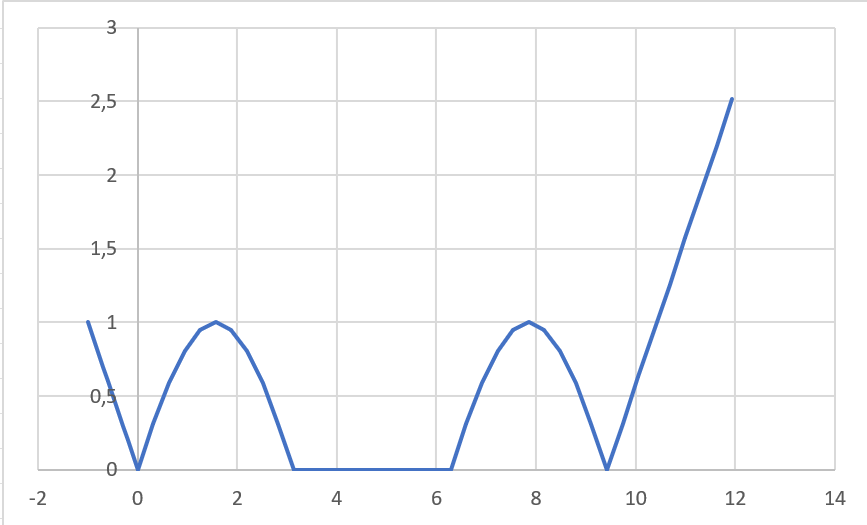


Рисунок №8 – График, построенный в Excel

Данный график аналогичен тому, который был построен программой, значит она работает корректно.

# 10.Результат работы

Доступен по ссылке.

<https://colab.research.google.com/drive/1XJD_K8Y755RaD9qEnMuY2a8zoJV0tTUh#scrollTo=J7F9Vlwer4Zp>

# 11.Выводы

В ходе выполнения работы выполнено ознакомление с возможностями работы с массивами в среде Python и простейшими способами визуализации информации.