МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ТЭВН

Лабораторная работа №1

“Интерполяция и решение трансцендентных уравнений”

Вариант № 12

Факультет: ФЭН

Группа: Эн1-42

Студент: Тимофеев А.А.

Преподаватель: Емельянова Е.С.

Новосибирск 2024

Цель работы: изучить численные методы решения нелинейных алгебраических уравнений и интерполяции. Научиться программировать решение этих задач на языке Python, а также - решать их с использованием стандартных функций библиотек Python.

# **Задание**

1. В соответствии с вариантом в Приложении №1 разработать алгоритм и написать программу, выполняющую решение нелинейного алгебраического уравнения или задачи интерполяции заданным в варианте методом.
   1. При решении уравнений, отделение корней выполнить графическим способом. Привести в отчёте интервалы поиска всех корней, или начальные приближения (в зависимости от метода).
   2. Для тестирования алгоритма решения уравнений, построить график функции в левой части уравнения и отметить на нём найденные корни.
   3. Для тестирования алгоритма интерполяции построить график интерполяционного полинома и отложить на нём заданные точки и точку, вводимую пользователем (должны лечь на график). Алгоритм интерполяции должен быть применим к произвольному количеству точек данных.
   4. Сравнить результаты решения задачи при помощи написанной программы с результатом применения для решения задачи функций из библиотек python.
2. В соответствии с вариантом в Приложении №2 выполнить решение следующих задач (можно использовать стандартные функции библиотек python):
   1. Заданы параметры электрической цепи. Сопротивление нагрузки нелинейно и зависит от тока. Необходимо рассчитать ток в электрической цепи. Отделение корней выполнить графическим способом.
   2. Задана табличная зависимость некоторых физических величин. Необходимо построить график этой зависимости в виде гладкой функции.
3. Составить отчёт по проделанной работе. Отчет должен содержать:
   1. Титульный лист, цель работы, задание. Все страницы, кроме первой, должны быть пронумерованы в верхнем правом углу. В верхнем колонтитуле в одной строке с номером страницы слева должна быть размещена подпись ФИО студента и надпись “Лабораторная работа №1”.
   2. Постановку каждой задачи.
   3. Блок-схему решения задачи из п.1 (для задач из п.2 блок-схемы можно не приводить).
   4. Листинги программ в виде текста, оформленного в дополнении Code Blocks, язык python, тема github.
   5. Результаты решения задач.

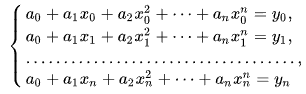
Задание 1

В этом задании требуют решить задачи интерполяции заданным в варианте методом. Исходные данные видны в таблице 1.

Таблица 1

| 1 | 1,6 | 2,3 | 3,0 |
| --- | --- | --- | --- |
| 2,71 | 4,95 | 9,97 | 20,08 |

Интерполяция Лагранжа - это задача поиска такого многочлена степени *n*, который строго проходит через все заданные n+1 точки данных , … . Собственно, такой многочлен называется многочленом Лагранжа и он существует в единственном экземпляре для заданного набора точек данных. Коэффициенты многочлена Лагранжа можно определить из решения системы линейных алгебраических уравнений (1):

 (1)

Число коэффициентов многочлена Лагранжа строго равно числу точек данных (старшая степень - на единицу меньше). В связи с этим нахождение коэффициентов из системы уравнений (1) при большом числе точек данных может представлять сложную задачу (большой объём вычислений). Поэтому для нахождения значения *y* интерполирующей функции Лагранжа в некоторой точке *x* можно использовать предложенное Лагранжем выражение (2):

|  | (2) |
| --- | --- |

Я создам список из данных мне чисел и создам массив из чисел пользователя. Далее создам функцию lagrange, которая принимает данные значения точек. Далее создаем несколько циклов и которые будут перебирать по индексу значения из массивов и дальше значения будут подставляться в формулу(2) и функция возвращать будет *y* который мы будем отображать на графике. Но начал я с создания блок схемы, она представлена на рисунке 1.

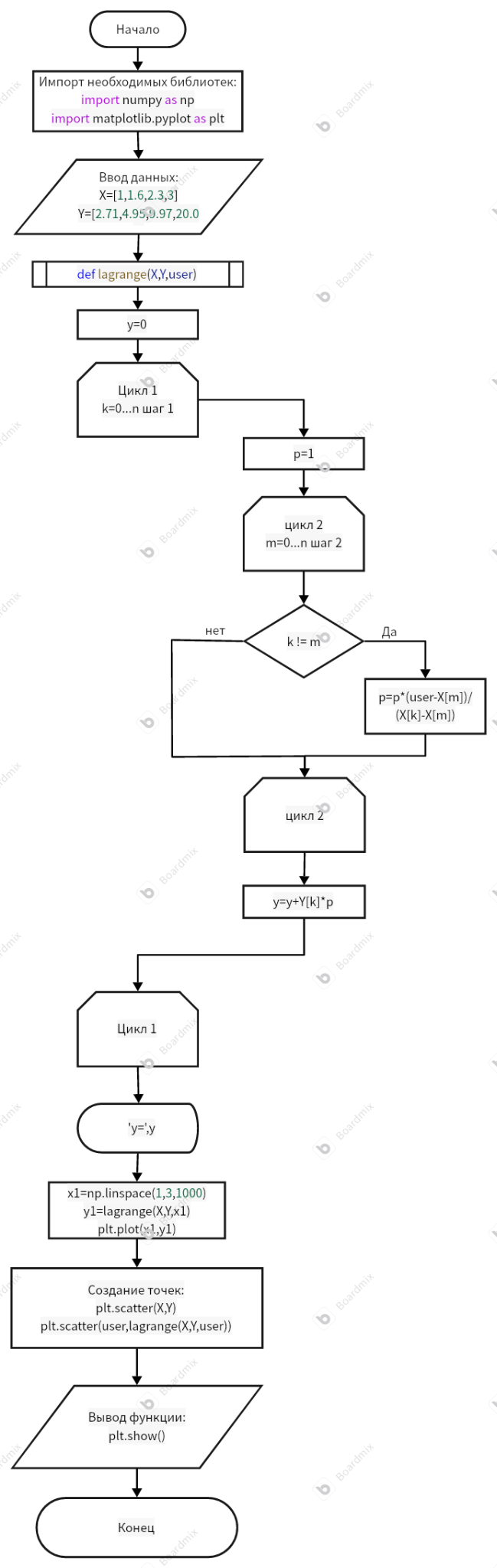


Рисунок 1 - Блок схема

Далее опираясь на блок схему, я написал программу, листинг которой представлен ниже.

| **import** numpy **as** np **import** matplotlib.pyplot **as** plt  X=[1,1.6,2.3,3] Y=[2.71,4.95,9.97,20.08] user=np.array([2,2.5])  **def** **lagrange**(X,Y,x):  y=0  **for** k **in** range(len(X)):  p=1  **for** m **in** range(len(X)):  **if** m!=k:p=p\*(x-X[m])/(X[k]-X[m])  y=y+Y[k]\*p  **return** y x1=np.linspace(1,3,1000) y1=lagrange(X,Y,x1) plt.plot(x1,y1) plt.scatter(X,Y) plt.scatter(user,lagrange(X,Y,user)) plt.grid() plt.legend(['Интерполяционный полином',"Исходные точки","Искомые точки"]) |
| --- |

После запуска данного программного кода появится следующий график - рисунок 2

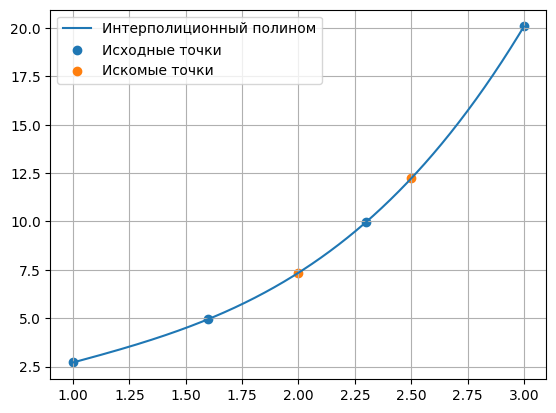


Рисунок 2 - график функции

После этого написал код при помощи уже готовой функции interp1d из библиотеки scipy.interpolate. Код ниже.

| **from** scipy.interpolate **import** interp1d f=interp1d(X,Y,'cubic') x2=np.arange(1,3,0.01) y2=f(x2) plt.figure(2) plt.plot(x2,y2) plt.plot(X,Y,'r\*') plt.scatter(user,f(user)) plt.grid() |
| --- |

После запуска программы будет вывод графика - рисунок 3.

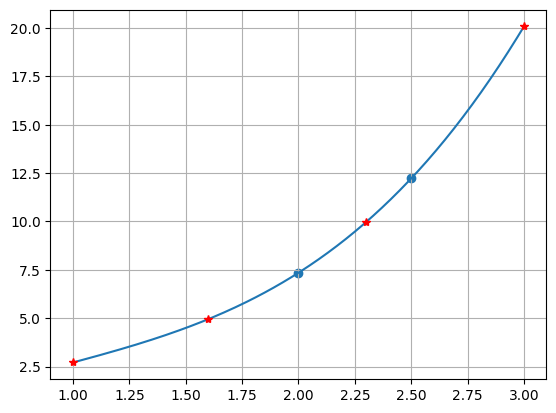


Рисунок 3 - график функции

Если сравнить графики, то видно что похожи и можно сделать вывод, что у меня получилось написать функцию, похожую по функционалу с interp1d.

Задание 2

1) Нам заданы параметры электрической цепи. Сопротивление нагрузки нелинейно и зависит от тока. Необходимо рассчитать ток в электрической цепи. Отделение корней выполнить графическим способом. Нам дано: В, Ом и также функция(2)

|  | (2) |
| --- | --- |

Далее я написал код для решения данной задачи. Я использовал готовую функцию bisect из библиотеки scipy.optimize. представлен ниже

| **import** numpy **as** np **import** matplotlib.pyplot **as** plt  **def** **func**(i):  E= 50  r=10  **return** i-E/(r+100\*np.exp(-0.025\*i\*\*2))  i=np.arange(1,5,0.01) y=func(i) plt.figure(1) plt.plot(i,y) plt.grid() **from** scipy.optimize **import** bisect I = bisect(func,0,50) print('Расчёт тока схемы I = {:.2f}'.format(I)) |
| --- |

Если запустить, то будет следующий ответ и график - рисунок 4.

| Расчёт тока схемы I = 0.46 |
| --- |

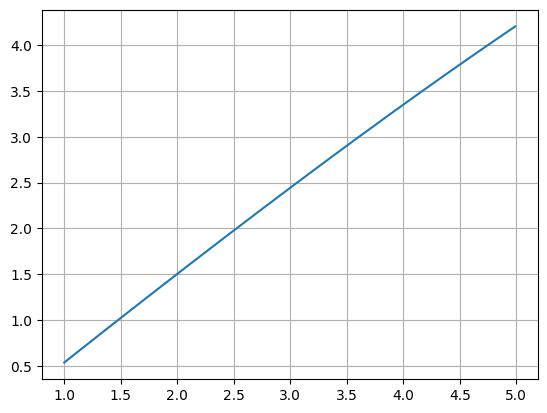


Рисунок 4 - график отделения корней

Из этого можно сказать что код работает и применение метода численного решения уравнения в энергетике очень полезно.

2) Задана табличная зависимость некоторых физических величин. Необходимо построить график этой зависимости в виде гладкой функции. У нас есть данные, представленные в таблице 2.

Таблица 2

| U, В | 0 | 0,9 | 1,1 | 1,3 |
| --- | --- | --- | --- | --- |
| I, А | 0 | 1,2 | 2,7 | 5,6 |

Я написал программу, которая показана ниже

| **import** numpy **as** np **import** matplotlib.pyplot **as** plt **from** scipy.interpolate **import** interp1d  x=[0,0.9,1.1,1.3] y=[0,1.2,2.7,5.6]  f=interp1d(x,y,'cubic') x1=np.arange(0,1.3,0.001) y1=f(x1) plt.plot(x1,y1) plt.scatter(x,y) plt.plot() plt.grid() plt.legend(['Интерполяционный полином',"Исходные точки","Искомые точки"]) |
| --- |

Если запустить данный программный код, то появится график функции - рисунок 5

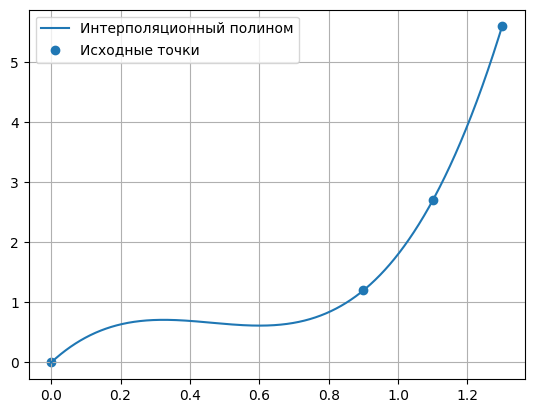


Рисунок 5 - график функции

Вывод: на лабораторной работе номер 1 я узнал что такое интерполяция и трансцендентные уравнения, как их решать. Узнал как это применяется в электроэнергетике, как при помощи языка Python можно выводить интерполирующий полином и методы решения трансцендентных уравнений.