**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

по дисциплине «Методы численного анализа»

на тему: «Интерполяция и аппроксимация экспериментальных данных»

Выполнл: студент гр. ИП-22

Коваленко А.И.

Принял: преподаватель

Процкая М.А.

Гомель 2022

**Цель работы**: Получить навыки применения таких приемов обработки экспериментальных данных, как аппроксимация и интерполяция, научиться выполнять графическую интерпретацию полученных результатов*.*

**Задание 1**

В результате эксперимента были получены числовые значения функции в табличном виде (таблица 1).

Задать координаты точек исходной табличной функции, построить ее график.

Выполнить аппроксимацию исходной функции по заданной аналитической зависимости, получить численные значения коэффициентов аппроксимирующей функции.

Сделать графическую интерпретацию результатов, построив на одном поле графики исходной и аппроксимирующей функций.

|  |  |
| --- | --- |
| Вариант  ант | З а д а н и е |
| 7 | В таблице приведены данные о зависимости теплопроводности легких бетонов (*у*, Вт/(м∙Со) от плотности (*х*, кг/м3).   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | *хi* | 800 | 900 | 1000 | 1100 | 1200 | 1300 | 1400 | 1500 | 1600 | 1700 | | *у*i | 0,2 | 0,22 | 0,24 | 0,28 | 0,33 | 0,38 | 0,4 | 0,42 | 0,44 | 0,47 |   Предполагая линейную зависимость *у* от *х*, определить параметры линейной регрессии , используя метод наименьших квадратов. Получить прогноз теплопроводности при плотности 1800 кг/м3. |

Листинг программы:

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** numpy.linalg **import**\*  
  
*#написать через linalg***def** amount(mass, exp):  
 sum = 0  
 **for** i **in** range(0, len(mass)):  
 sum += pow(mass[i], exp)  
 **return** sum  
  
  
**def** multiply(mass1, mass2):  
 mult = 0  
 **for** i **in** range(0, len(mass1)):  
 mult += mass1[i] \* mass2[i]  
 **return** mult  
  
  
**def** f(a, b, x):  
 **return** a \* x + b  
  
  
**def** main():  
 x = [800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700]  
 y = [0.2, 0.22, 0.24, 0.28, 0.33, 0.38, 0.4, 0.42, 0.44, 0.47]  
 a = (int(len(x)) \* int(multiply(x, y)) - int(amount(x, 1)) \* int(amount(y, 1))) / \  
 (int(len(x)) \* int(amount(x, 2)) - int(pow(amount(x, 1), 2)))  
 b = (int(amount(y, 1)) - a \* int(amount(x, 1))) / int(len(x))  
 x1800 = a \* 1800 + b  
 print(**"Коэфф. А: "**, a,**"\nКоэфф. В: "**, b)  
 t = np.linspace(800,1800, 2000)  
 plt.plot(t, f(a, b, t), **'y'**)  
 **for** i **in** range(0, len(x)):  
 plt.plot(x[i], y[i], **'ro'**, markersize=3)  
 plt.grid()  
 plt.plot(1800, x1800, **'ro'**, markersize=3, color=**'blue'**)  
 plt.show()  
 print(**"при x = 1800\ny = "**, x1800)  
main()

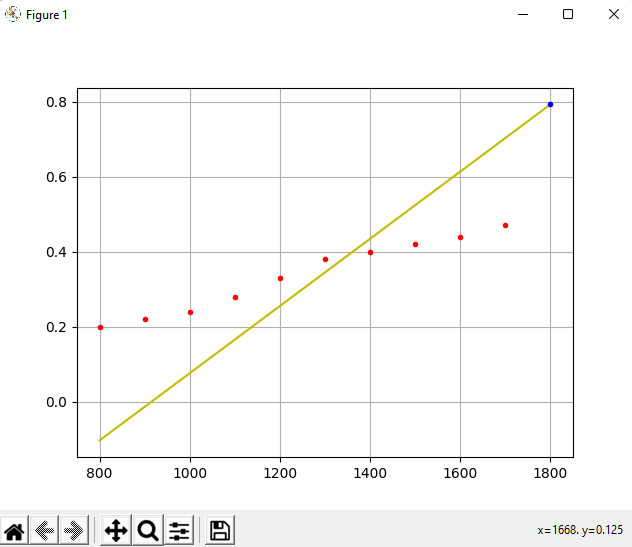
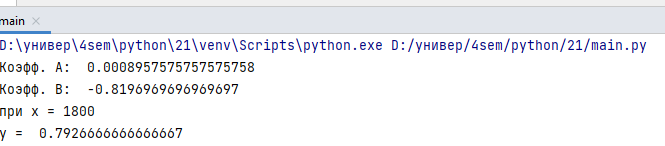


Рисунок 1 – График решения



В результате вычислений отрицательным, что в корне не верно. Это связано с неточностью метода вычисления.

**Задание 2**

В результате эксперимента были получены числовые значения функции в табличном виде (таблица 2).

Задать координаты точек исходной табличной функции, построить ее график.

Выполнить аппроксимацию исходной функции по заданной аналитической зависимости, получить численные значения коэффициентов аппроксимирующей функции.

Сделать графическую интерпретацию результатов, построив на одном поле графики исходной и аппроксимирующей функций.

Листинг программы:

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
  
  
**def** amount(mass, exp):  
 sum = 0  
 **for** i **in** range(0, len(mass)):  
 sum += pow(mass[i], exp)  
 **return** sum  
  
  
**def** multiply(mass1, mass2):  
 mult = 0  
 **for** i **in** range(0, len(mass1)):  
 mult += mass1[i] \* mass2[i]  
 **return** mult  
  
**def** approximateFunctionQuadro(T, a, b, c):  
 value = (-b+np.sqrt(abs(4\*a\*(c-T))))/(2\*a)  
  
 **return** value  
  
  
**def** main():  
 x\_array = [1, 2, 3, 3.2, 3.6, 4, 5, 5.9, 6, 7.3]  
 y\_array = [550, 640, 704, 719, 735, 756, 810, 855, 865, 924]  
 n = len(x\_array); *# Кол=во опытных измерений.* sumx = amount(x\_array, 1)  
 sumx2 = amount(x\_array, 2)  
 sumx3 = amount(x\_array, 3)  
 sumx4 = amount(x\_array, 4)  
 sumy = amount(y\_array, 1)  
 sumxy = multiply(x\_array, y\_array)  
 sumx2y = 0;  
 plt.grid()  
  
 **for** i **in** range(0, n): *# подсчёт* sumx2y += x\_array[i] \* x\_array[i] \* y\_array[i]  
  
 M2 = np.array([[float(sumx4), float(sumx3), float(sumx2)],  
 [float(sumx3), float(sumx2), float(sumx)],  
 [float(sumx2), float(sumx), float(len(x\_array))]])  
 v2 = np.array([float(sumx2y), float(sumxy), float(sumy)])  
 result = np.linalg.solve(M2, v2)  
 a = result[0];  
 b = result[1];  
 c = result[2];  
 **for** i **in** range(0, n): *# подсчёт* plt.plot(x\_array[i], y\_array[i], marker=**'o'**, color=**"red"**, )  
 x\_array.insert(1, 1500)  
 y\_array.insert(1, int(approximateFunctionQuadro(1500, a, b, c)))  
  
 print(**"\nВремя алгоритма на 1500K "** + **'%4.2f '** % approximateFunctionQuadro(500, a, b, c) + **"\n"**)  
 *#print('%4.2f; %4.2f; %4.2f;'%a, b, c)* x = np.linspace(0, 20, 50)  
 y = approximateFunctionQuadro(x, a, b, c)  
 plt.plot(x, y, color=**"blue"**)  
 plt.show()  
 print(a, b, c)  
  
  
main()

Рисунок 2 – График решения

***Задание 3***

В результате эксперимента были получены числовые значения функции Y (таблица 3).

Значения аргумента функции X нужно создать с использованием формулы, приведенной ниже (2N+1 – количество элементов в векторе Y)



Построить график полученной функции Y(X).

Выполнить тригонометрическую интерполяцию исходной функции, получить численные значения коэффициентов тригонометрического полинома.

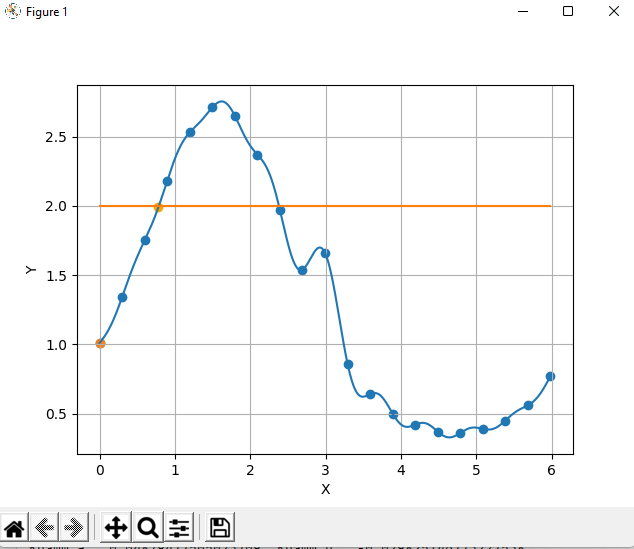
Сделать графическую интерпретацию результатов, построив на одном поле графики исходной и интерполирующей функций.

Найти значение Х, при котором интерполирующая функция пересекает пороговое значение, указанное в таблице 3, нанести эту точку на график.

|  |  |
| --- | --- |
| 7 | 1.01; 1.34; 1.75; 2.18; 2.53; 2.71; 2.65; 2.37; 1.97; 1.54; 1.66; 0.86; 0.64; 0.5; 0.42; 0.37; 0.36; 0.39; 0.45; 0.56; 0.77  Пороговое значение равно 2 |

Листинг программы:

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
  
**def** PrintAB(x):  
 S = 0  
 **for** k **in** range(1, n + 1):  
 Sak = 0  
 Sbk = 0  
 **for** j **in** range(N):  
 Sak += Y[j] \* np.cos(k \* X[j])  
 Sbk += Y[j] \* np.sin(k \* X[j])  
 ak = 2 / N \* Sak *# коэффицент a* bk = 2 / N \* Sbk *# коэффицент b* print(**"Коэфф а: "**, ak, **" Коэфф b: "**, bk, **"\n"**)  
  
**def** f(x):  
 S = 0  
 **for** k **in** range(1,n+1):  
 Sak=0  
 Sbk=0  
 **for** j **in** range(N):  
 Sak += Y[j]\*np.cos(k\*X[j])  
 Sbk += Y[j]\*np.sin(k\*X[j])  
 ak = 2/N \* Sak *# коэффицент a* bk = 2/N \* Sbk *# коэффицент b* S += ak \* np.cos(k\*x) + bk \* np.sin(k\*x) *# подсчет ряда* G = a0 + S *# ряд Фурье* **return** G  
  
  
trashold\_value = 2  
Y = np.array(  
 [1.01, 1.34, 1.75, 2.18, 2.53, 2.71, 2.65, 2.37, 1.97, 1.54, 1.66, 0.86, 0.64, 0.5, 0.42, 0.37, 0.36, 0.39, 0.45, 0.56, 0.77]) *# начальны значение Yj*N = len(Y) *# 2n+1, j принадлежит диапазону от 1 до 2n+1*n = int((N - 1) / 2)  
X = [] *# 1) нахождение Хj***for** i **in** range(1, N + 1):  
 X.append(2 \* np.pi \* (i - 1) / N)  
X = np.array(X)  
print(X) *# вывод в консоль Xj*a0 = 1 / N \* sum(Y) *# 2) коэффицен а0*plt.scatter(X, Y)  
x = np.linspace(X.min(), X.max(), 1000)  
t = np.arange(0, 6, 0.01)  
plt.plot(x, f(x))  
  
PrintAB(x)  
  
x0 = 0  
**while** f(x0) > trashold\_value:  
 x0 += 0.01  
xp = x0  
plt.scatter(xp, f(xp))  
  
x1 = xp + 0.01  
**while** f(x1) < trashold\_value:  
 x1 += 0.01  
xp2 = x1 - 0.01  
print(**"xp2: "**,xp2, **" f(xp2): "**,f(xp2))  
plt.scatter(xp2, f(xp2), color=**'orange'**)  
  
plt.grid(**True**)  
plt.xlabel(**"X"**)  
plt.ylabel(**"Y"**)  
x0, y1 = [X.min(), X.max()], [trashold\_value, trashold\_value]  
plt.plot(x0, y1)  
plt.show()



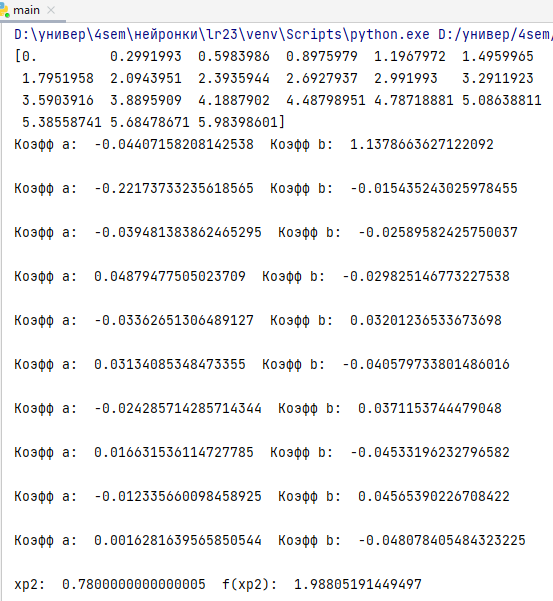


Рисунок 3 – График решения

**Вывод:** Получили навыки применения таких приемов обработки экспериментальных данных, как аппроксимация и интерполяция, научились выполнять графическую интерпретацию полученных результатов.