Федеральное государственное автономное образовательное учреждение высшего образования

***Университет ИТМО***

Факультет программной инженерии и компьютерной техники

Дисциплина: **Вычислительная математика**

Лабораторная работа №4

*“* **Аппроксимация фунцкии***”*

**Вариант: 9**

Выполнил: Кузнецов Максим Александрович

Группа: Р3211

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург 2022 г

**Цель работы:**

**Цель лабораторной работы**: найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов

**Условия и задание:**

№ варианта определяется как номер в списке группы согласно ИСУ.

*Вычислительная реализация задачи:*

а) Для заданной функции (см. таблицу 1) построить наилучшие линейное и квадратичное приближения по 11 точкам указанного интервала.

b) Найти среднеквадратические отклонения. Ответы дать с тремя знаками после запятой.

c) Построить графики линейного и квадратичного приближений и заданной функции.

d) ***Привести в отчете подробные вычисления***.

*Программная реализация задачи:*

* 1. Предусмотреть ввод исходных данных из файла/консоли (таблица *y=f(x)* должна содержать 10–12 точек).
  2. Реализовать метод наименьших квадратов, исследуя все функции:

линейную функцию;

полиномиальную функцию 2-й степени;

полиномиальную функцию 3-й степени;

экспоненциальную функцию;

логарифмическую функцию;

степенную функцию.

* 1. Предусмотреть вывод результатов в файл/консоль.
  2. Для линейной зависимости вычислить коэффициент корреляции Пирсона.
  3. Программа должна отображать наилучшую аппроксимирующую функцию.
  4. Организовать вывод графиков функций, графики должны полностью отображать весь исследуемый интервал (с запасом).

**Вычислительная задача по варианту:**

|  |  |  |
| --- | --- | --- |
| 9 |  |  |

|  |  |
| --- | --- |
| X | Y |
| 0 | 0 |
| 0.2 | 0.159 |
| 0.4 | 0.318 |
| 0.6 | 0.469 |
| 0.8 | 0.592 |
| 1 | 0.667 |
| 1.2 | 0.679 |
| 1.4 | 0.633 |
| 1.6 | 0.554 |
| 1.8 | 0.465 |
| 2 | 0.381 |

Квадратичная аппроксимация:

Подставим в систему:

Изображение выглядит как текст, человек

Автоматически созданное описание

Решим, получим:

a0 = -0.277

a1 = 1.888

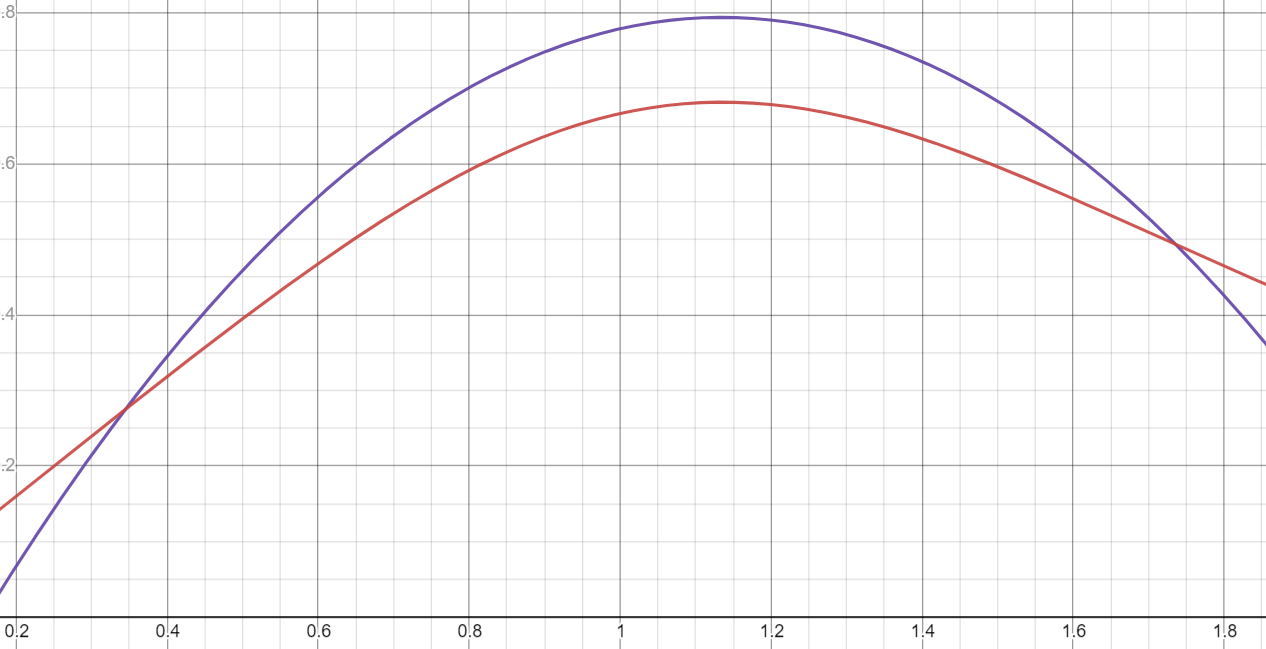
a2 = -0.832

**Полученная аппроксимация: -0.277+1.888x-0,832x^2**

|  |  |
| --- | --- |
| X | P(X) |
| 0 | 0.277 |
| 0.2 | 0.067 |
| 0.4 | 0.345 |
| 0.6 | 0.556 |
| 0.8 | 0.701 |
| 1 | 0.779 |
| 1.2 | 0.790 |
| 1.4 | 0.735 |
| 1.6 | 0.614 |
| 1.8 | 0.426 |
| 2 | 0.171 |

Среднеквадратичное отклонение: 0.101414

График:



Линейная аппроксимация:

Решим следующую систему

Изображение выглядит как текст

Автоматически созданное описание

Полученные результаты

a = 0,205

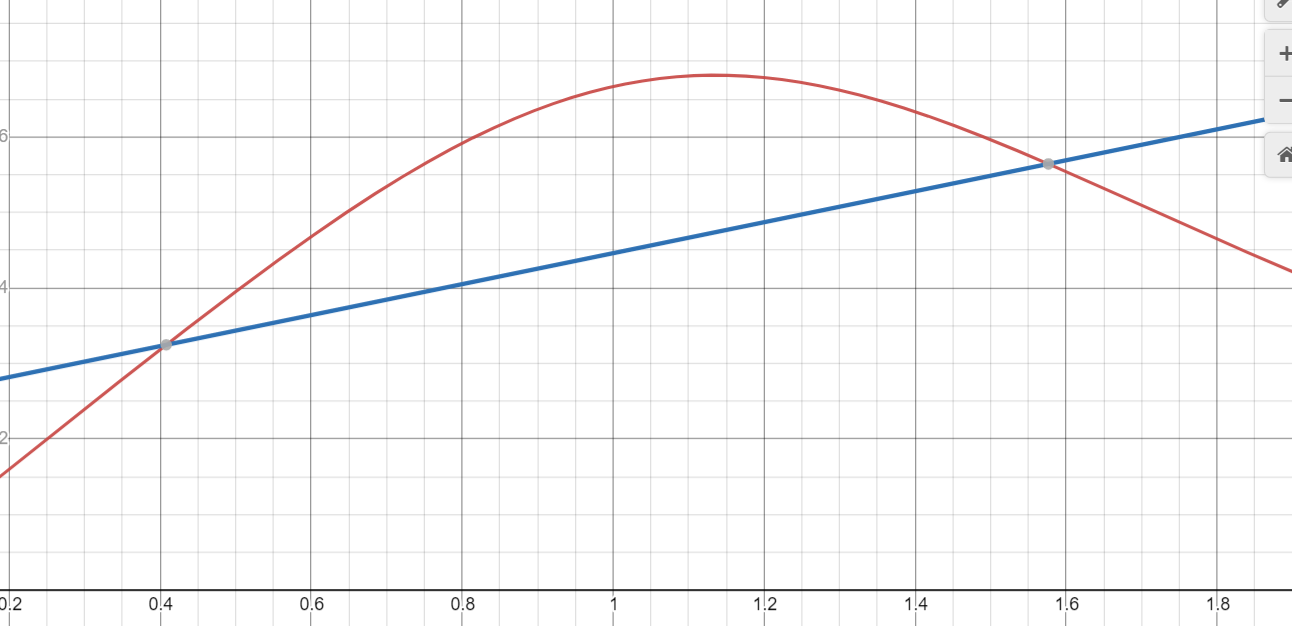
b = 0,241

**Полученная аппроксимация: 0,205x + 0,241**

|  |  |
| --- | --- |
| X | P(X) |
| 0 | 0.241 |
| 0.2 | 0.282 |
| 0.4 | 0.323 |
| 0.6 | 0.364 |
| 0.8 | 0.405 |
| 1 | 0.446 |
| 1.2 | 0.487 |
| 1.4 | 0.528 |
| 1.6 | 0.569 |
| 1.8 | 0.610 |
| 2 | 0.651 |

Среднеквадратичное отклонение: 0,1680

График



**Листинг программы:**

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

import matplotlib.lines as mlines

from numpy import genfromtxt

import sys

import math

import pandas as pd

from IPython.display import display, HTML

import matplotlib.lines as mlines

from matplotlib.pyplot import figure

from math import sqrt

x: int

y: int

content = []

def draw(f, a, b, c, d):

    plt.figure(figsize=(8, 4))

    plt.plot(x, y, 'ko')

    plt.plot(x, f)

    plt.grid()

    plt.show()

def cof\_cor(Xs, Ys):

    X\_mean = np.mean(Xs)

    Y\_mean = np.mean(Ys)

    numerator = 0

    denominator\_x = 0

    denominator\_y = 0

    for x, y in zip(Xs, Ys):

        numerator += (x-X\_mean)\*(y-Y\_mean)

        denominator\_x += (x-X\_mean) \*\* 2

        denominator\_y += (y-Y\_mean) \*\* 2

    return numerator / sqrt(denominator\_x \* denominator\_y)

def lin\_approx(Xs, Ys):

    A = np.array([[sum(Xs \*\* 2), sum(Xs)],[sum(Xs), len(Xs)]])

    B = np.array([sum(Xs\*Ys), sum(Ys)])

    a, b = np.linalg.solve(A, B)

    return a, b

def linear(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    #Подсчет аппроксимации

    a, b = lin\_approx(Xs, Ys)

    print(f'Коэффициент корреляции = {cof\_cor(Xs, Ys)}')

    f = a \* Xs + b

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nЛинейная аппроксимация\nf = {a} \* x + {b}')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, 0, 0)

def squared(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    #Подсчет аппроксимации

    A = np.array([[len(Xs), sum(Xs), sum(Xs \*\* 2)],

                  [sum(Xs), sum(Xs \*\* 2), sum(Xs \*\* 3)],

                  [sum(Xs \*\* 2), sum(Xs \*\* 3), sum(Xs \*\* 4)]])

    B = np.array([sum(Ys), sum(Xs\*Ys), sum((Xs\*\*2)\*Ys)])

    c, b, a = np.linalg.solve(A, B)

    f = a \* Xs \*\* 2 + b \* Xs + c

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #Мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nКвадратичная аппроксимация\nf = {a} \* x^2 + {b} \* x +{c}')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, c, 0)

def triple(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    #Подсчет аппроксимации

    A = np.array([[len(Xs), sum(Xs), sum(Xs \*\* 2), sum(Xs \*\* 3)],

                  [sum(Xs), sum(Xs \*\* 2), sum(Xs \*\* 3), sum(Xs \*\* 4)],

                  [sum(Xs \*\* 2), sum(Xs \*\* 3), sum(Xs \*\* 4), sum(Xs \*\* 5)],

                  [sum(Xs \*\* 3), sum(Xs \*\* 4), sum(Xs \*\* 5), sum(Xs \*\* 6)]])

    B = np.array([sum(Ys), sum(Xs\*Ys), sum((Xs\*\*2)\*Ys), sum((Xs \*\*3 )\*Ys)])

    d, c, b, a = np.linalg.solve(A, B)

    f = a \* Xs \*\* 3 + b \* Xs \*\* 2 + c \* Xs + d

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #Мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nТретичная аппроксимация\nf = {a} \* x^3 + {b} \* x^2 + {c} \* x + {d}')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, c, d)

    return a, b, c, S, delta

def power(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    log\_Xs = np.log(Xs)

    log\_Ys = np.log(Ys)

    if True in np.isnan(log\_Xs) or True in np.isnan(log\_Ys):

        raise ValueError

    #Подсчет аппроксимации

    b, a = lin\_approx(log\_Xs, log\_Ys)

    a = np.exp(a)

    f = a \* (Xs \*\* b)

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #Мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nСтепенная аппроксимация\nf = {a} \* x \*\* {b}')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, 0, 0)

def exponential(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    log\_Ys = np.log(Ys)

    if True in np.isnan(log\_Ys):

        raise ValueError

    #Подсчет аппроксимации

    b, a = lin\_approx(Xs, log\_Ys)

    a = np.exp(a)

    f = a \* (np.exp(Xs \* b))

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #Мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nЭкспоненциальная аппроксимация\nf = {a} \* e\*\*(x \* {b})')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, 0, 0)

    return a, b, '-', S, delta

def logarithm(array):

    Xs = np.array(array[0])

    Ys = np.array(array[1])

    log\_Xs = np.log(Xs)

    if True in np.isnan(log\_Xs):

        raise ValueError

    #Подсчет аппроксимации

    a, b = lin\_approx(log\_Xs, Ys)

    f = a \* (log\_Xs) + b

    #СКО

    eps = Ys - f

    delta = (sum(eps \*\* 2) / len(Xs)) \*\* 0.5

    #Мера отклонения

    S = sum(eps \*\* 2)

    print(f'\n\nЛогарифмическая аппроксимация\nf = {a} \* ln(x) + {b}')

    print(f'Мера отклонения = {S}')

    print(f'СКО = {delta}\n\n')

    draw(f, a, b, 0, 0)

    return a, b, '-', S, delta

def run():

    again = True

    while again:

        again = False

        in\_type = input('Введите:\n\t\* k - если вводить с клавиатуры\n\t\* f - если хотите вводить из файла\n')

        if in\_type.strip() == 'k':

            line\_x = input()

            content.append([float(x) for x in line\_x.split(" ")])

            line\_y = input()

            content.append([float(x) for x in line\_y.split(" ")])

        elif in\_type.strip() == 'f':

            with open("input.txt") as f:

              for line in f:

                content.append([float(x) for x in line.split(" ")])

            f.close()

        else:

            print('Введено неверно, попробуйте снова.')

            again = True

    global x

    x = np.array(content[0])

    global y

    y = np.array(content[1])

    linear(content)

    squared(content)

    triple(content)

    power(content)

    exponential(content)

    logarithm(content)

run()

**Пример работы:**

Введите:

\* k - если вводить с клавиатуры

\* f - если хотите вводить из файла

f

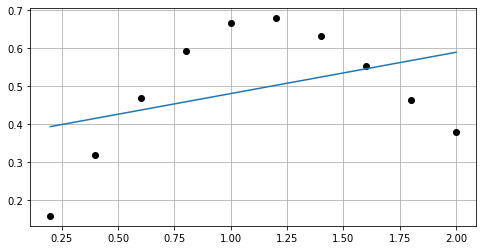
Коэффициент корреляции = 0.3917459046255321

Линейная аппроксимация

f = 0.1086969696969697 \* x + 0.37213333333333337

Мера отклонения = 0.215072496969697

СКО = 0.1466535021640114

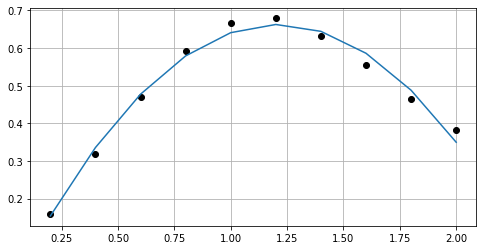


Квадратичная аппроксимация

f = -0.4996212121211979 \* x^2 + 1.2078636363636035 \* x +-0.06753333333331886

Мера отклонения = 0.004192375757575746

СКО = 0.020475291835712003

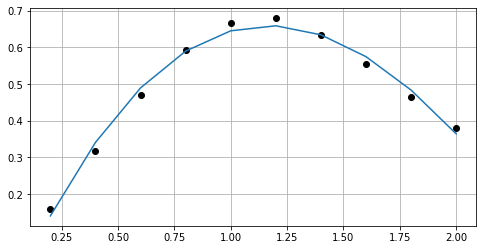


Третичная аппроксимация

f = 0.06983294483287979 \* x^3 + -0.7300699300697105 \* x^2 + 1.4204351204349117 \* x + -0.11546666666661744

Мера отклонения = 0.003228345920745921

СКО = 0.017967598394738014

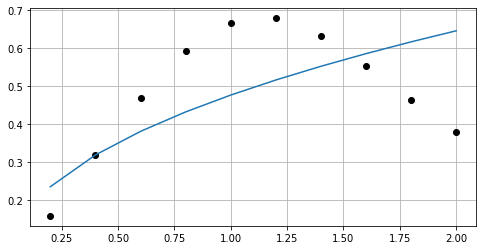


Степенная аппроксимация

f = 0.4769530595277907 \* x \*\* 0.43762360608042794

Мера отклонения = 0.20224322072624246

СКО = 0.14221224304758098

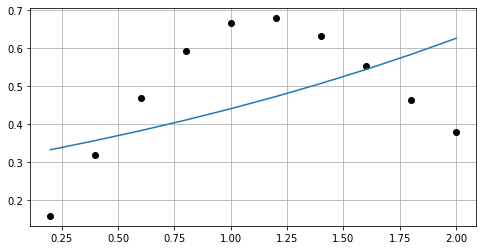


Экспоненциальная аппроксимация

f = 0.31050865647035136 \* e\*\*(x \* 0.3508027500223356)

Мера отклонения = 0.255583656520769

СКО = 0.15986983971993246

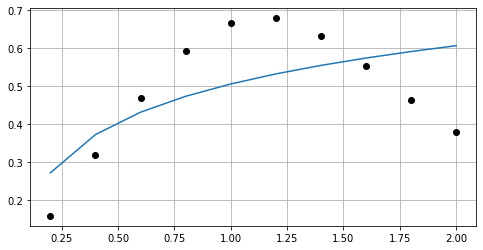


Логарифмическая аппроксимация

f = 0.14524289441897606 \* ln(x) + 0.5060785607283772

Мера отклонения = 0.1520460441937742

СКО = 0.12330695203181945



**Вывод:**

В результате выполнения лабораторной работы я:

* Познакомился с видами аппроксимации функции
* На практике в коде и на бумаге посчитал и реализовал различные виды аппроксимаций
* Вспомнил о методе МНК