МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа № 3**

Дисциплина: Системный анализ и обработка информации

Тема: аппроксимация функций по данным измерений методом наименьших квадратов с весовыми коэффициентами

Выполнил: студент группы ВТ-31

Ковалёв И. Д.

Проверил:

Полунин А. И.

Белгород 2020

**Цель работы**: изучить методы аппроксимации при наличии ошибок в измерениях функции.

**Содержание работы**

Производится измерение значений функции в некоторые заданные моменты времени. Вследствие погрешностей измерительной техники замер значения функции производится с ошибкой. Найти значения коэффициентов аппроксимирующего полинома методом наименьших квадратов и с помощью этого полинома найти интерполяционное значение функции для . Сравнить эти значения с точными значениями функции в этих точках.

Расчетные значения моментов измерений:

*,*

|  |  |  |  |
| --- | --- | --- | --- |
| Точное значение функции | Аппроксимирующий полином | Число функций К | измерения функции Y |
|  |  | 5 | 0,2 |

Найдем значения коэффициентов аппроксимирующего полинома с помощью МНК:

(1)

Где: F – матрица, элементами которой являются значения базисной функции в заданные моменты времени,

К – матрица весовых коэффициентов,

Y – вектор, содержащий значения базисной функции в заданный момент времени.

Вычислив (1), получим вектор коэффициентов : [0.015, 0.056, 0.855, -0.0136, 0.010].

Найдем значения интерполяционной функции для Х = 4,5 и Х = 9,5 и сравним с точными значениями :

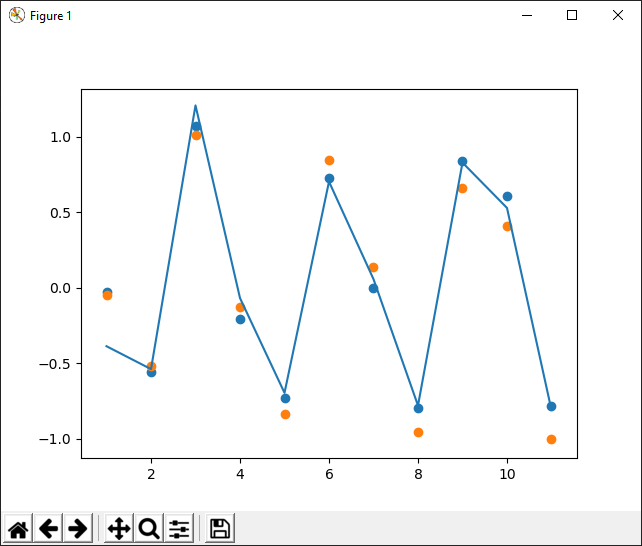
F(4.5) = -0.900

F(9.5) = 0.988

Z(4.5) = -0.923

Z(9.5) = 0.998

Результат работы программы:



**Вывод:** в процессе выполнения лабораторной работы был изучен метод аппроксимации при наличии ошибок в измерениях функции, а именно метод наименьших квадратов с весовыми коэффициентами.

Текст программы:

import random

import math

import numpy

import scipy.integrate as integrate

import scipy.optimize as optimize

import matplotlib.pyplot as pyplot

sz = 11

v = 5

sg = 0.2

k = 5

def fx(x):

    return math.exp(-x) + math.cos(2 \* x)

def gx(i, x):

    return math.cos(i \* x)

def zx(x, func):

    res = 0

    for i in range(0, k):

        res += a[i] \* func(i, x)

    return res

if \_\_name\_\_ == "\_\_main\_\_":

    y\_t = []

    v = []

    y = []

    K = []

    x = []

    Z = []

    for i in range (0, sz):

        y\_t.append(fx(i + 1))

        x.append(i + 1)

        K.append(numpy.zeros(sz))

        K[i][i] = sg \* (i + 1) \* 0.1

        if (random.randint(0, 1)):

            y.append(y\_t[i] + K[i][i])

        else:

            y.append(y\_t[i] - K[i][i])

    F = numpy.zeros((sz, k))

    for j in range(0, sz):

        for i in range(0, k):

            F[j][i] = gx(i, x[j])

    a = numpy.linalg.inv(F.transpose().dot(K).dot(F)).dot(F.transpose()).dot(K).dot(y)

    for i in x:

        Z.append(zx(i, gx))

    print(x, y)

    pyplot.scatter(x, y)

    pyplot.scatter(x, y\_t)

    pyplot.plot(x, Z)

    pyplot.show()