МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики (ИЦТЭФ)

Кафедра вычислительной техники и электроники (ВТиЭ)

Отчет по лабораторной работе № 1

**Аппроксимация функции в базисе алгебраических многочленов по методу МНК**

(дисциплина «Вычислительная математика»)

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Осипенко

Проверил: к.ф-м.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иордан В.И.

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Краткая теория метода

Метод наименьших квадратов основывается на минимизации функционала:



где n – число точек измерения.

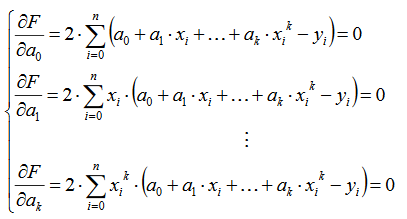
В данном методе аппроксимирующая функция f(x) представляется полиномом степени k:



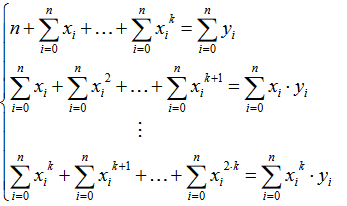
или с подстановкой



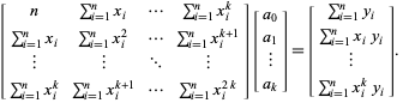
Для нахождения неизвестных коэффициентов находят производные от данного функционала по соответствующему коэффициенту, и приравнивают их к нулю:



После преобразований получаем систему уравнений:



или в матричной форме



или в матричном уравнении



Решая полученную систему уравнений, находят a0, a1, ..., ak.

# Описание алгоритма

1. Инициализация переменных num = 15, max\_ord = 5, l = -15, r = 15
2. X = np.linspace(l,r,num)
3. Y = 1/(1+np.exp(-1\*X))
4. Для k в промежутке от 1 до max\_ord+1 сделать:
   1. Создать матрицу matrx размером num на k
   2. Для i в промежутке от 0 до num:
      1. Для j в промежутке от 0 до k:
         1. matrx[i,j] = X[i]\*\*j
   3. посчитать коэффициенты a по формуле:

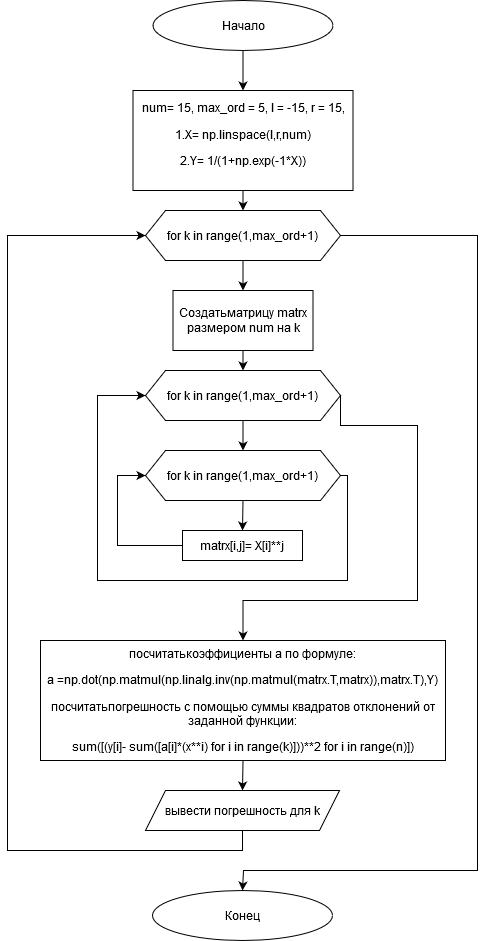
a = np.dot(np.matmul(np.linalg.inv(np.matmul(matrx.T,matrx)),matrx.T),Y)

* 1. посчитать погрешность с помощью суммы квадратов отклонений от заданной функции:

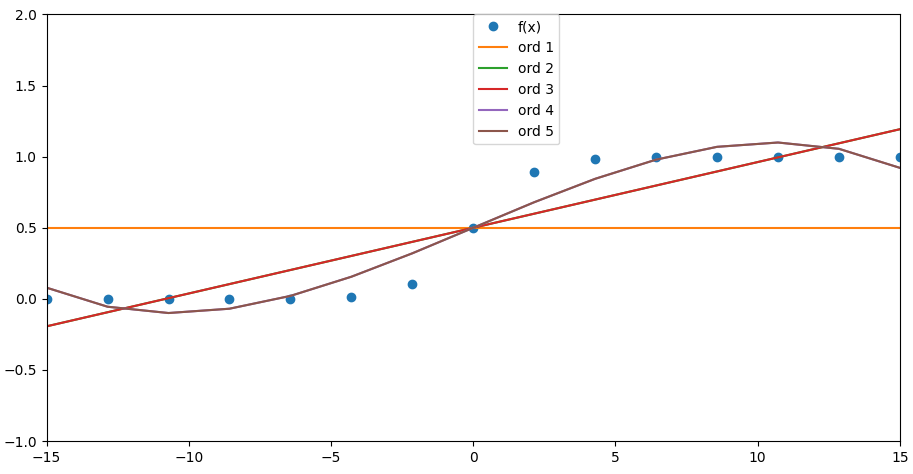
sum([(y[i] - sum([a[i]\*(x\*\*i) for i in range(k)]))\*\*2 for i in range(n)])

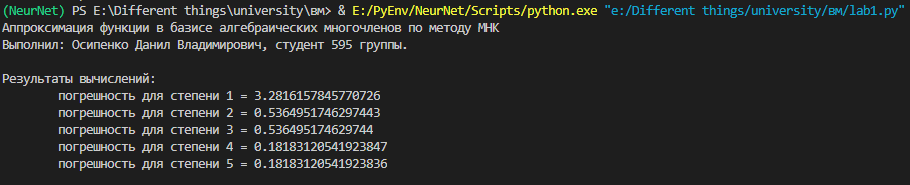
* 1. вывести погрешность для k

# Блок-схема алгоритма



# Тестовый пример

****

****

# Список литературы

1. Волков Е. А. Численные методы: учеб. пособие – Москва: Наука, 1987.
2. Weisstein, Eric W. "Least Squares Fitting--Polynomial. " From *MathWorld* -- A Wolfram Web Resource. URL: <https://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html>

# Приложение

**Текст программы:**

import numpy as np

import matplotlib.pyplot as plt

#--------------------------

print('Аппроксимация функции в базисе алгебраических многочленов по методу МНК \nВыполнил: Осипенко Данил Владимирович, студент 595 группы.\n')

#--------------------------

f = lambda x: 1/(1+np.exp(-1\*x))

p = lambda x,a,k: sum([a[i]\*(x\*\*i) for i in range(k)])

F = lambda x,y,a,k,n: sum([(y[i] - p(x[i],a,k))\*\*2 for i in range(n)])

#--------------------------

num = 15

max\_ord = 5

l,r = -15,15

X = np.linspace(l,r,num)

Y = f(X)

delta = []

#--------------------------

plt.xlim(l,r)

plt.ylim(-1,2)

plt.plot(X,Y,'o',label='f(x)')

#--------------------------

print('Результаты вычислений:')

for k in range(1,max\_ord+1,):

    matrx = np.ndarray((num,k))

    for i in range(num):

        for j in range(k):

            matrx[i,j] = X[i]\*\*j

    a = np.dot(np.matmul(np.linalg.inv(np.matmul(matrx.T,matrx)),matrx.T),Y)

    print(f'\tпогрешность для степени {k} = {F(X,Y,a,k,num)}')

    plt.plot(X,p(X,a,k),label=f'ord {k}')

#--------------------------

plt.legend(bbox\_to\_anchor=(0.5, 1), loc='upper left', borderaxespad=0.)

plt.show()