

**Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

Вычислительная математика

Лабораторная работа №1

Аппроксимация по методу наименьших модулей

Выполнила:

Гафурова Фарангиз Фуркатовна

Группа P3220

Преподаватель:

Перл Ольга Вячеславовна

г. Санкт-Петербург

2025

**Задача**

Title: Аппроксимация по методу наименьших модулей Description: [ДИСКЛЕЙМЕР: Это не всё, что вам нужно сделать для сдачи лабораторной работы по курсу, но этого достаточно, чтобы получить вариант на следующую работу.] Дан набор точек, по которым необходимо построить линейную аппроксимацию по методу наименьших модулей. Необходимо найти значение модуля наибольшего отклонения среди заданных точек относительно полученной линейной аппроксимации. [Обратите внимание, что при подготовке отчёта по лабораторной работе, желательно иметь несколько аппроксимирующих функций, кроме линейной.] Формат входных данных: x1 x2 x3 ... y1 y2 y3 ... где x1...xn - список значений аргумента для узлов интерполяции, y1...yn - список значений функции для соответствующего значения аргумента для узлов интерполяции. В тестах также вначале задаётся количество задаваемых точек, однако, в функцию этот параметр не передаётся. Формат выходных значений: вещественное число, являющееся значением модуля наибольшего отклонения исходных данных от полученной линейной аппроксимации.

**Описание Метода**

Метод наименьших модулей предназначен для построения линейной аппроксимации набора точек . В отличие от метода наименьших квадратов, он минимизирует наибольшее отклонение , обеспечивая большую устойчивость к выбросам.

Процедура метода:

1. Упорядочить точки по значениям x.
2. Найти медианные значения x и y.
3. Вычислить медианный наклон k = median(yi - yj / xi - xj) i != j.
4. Определить свободный коэффициент b = ym - kxm.
5. Найти максимальное отклонение:

### 

### 

**Блок-схема**

### 

### Код численного метода (Python)

#

# Complete the 'approximate\_linear\_least\_modules' function below.

#

# The function is expected to return a DOUBLE.

# The function accepts following parameters:

# 1. DOUBLE\_ARRAY x\_axis

# 2. DOUBLE\_ARRAY y\_axis

#

def med(ar):

return ar[len(ar) // 2]

def koefNak(x\_axis, y\_axis, n\_size):

return [(y\_axis[i] - y\_axis[j]) / (x\_axis[i] - x\_axis[j])

for i in range(n\_size) for j in range(i) if x\_axis[i] != x\_axis[j]]

def approximate\_linear\_least\_modules(x\_axis, y\_axis):

n\_size = len(x\_axis)

points = sorted(zip(x\_axis, y\_axis))

x\_axis, y\_axis = zip(\*points)

med\_x = med(x\_axis)

med\_y = med(y\_axis)

slopes = koefNak(x\_axis, y\_axis, n\_size)

slopes.sort()

med\_k = slopes[len(slopes) // 2] # Медианный наклон

med\_b = med\_y - med\_k \* med\_x # Смещение

# Вычисляем отклонения

res = [abs(y\_axis[i] - (med\_k \* x\_axis[i] + med\_b)) for i in range(n\_size)]

return max(res)

### 

**Примеры работы программы**

|  |  |
| --- | --- |
| y | res |
| 1 2 3 4 5 | 0 |
| -1 -0.5 0 0.5 1 | 0 |
| 3 3.1 2.9 3.2 3.0 | 0.2 |
| 2 4 6 8 10 | 0 |
| 2 3 4 5 6 | - |
| 1 1 1 1 1 10 1 1 1 1 | 9 |
| 10 -10 10 -10 10 | 10 |
| 2 5 -1 6 3 | 3 |

### Корректность сделанных выводов

* Метод позволяет строить линейную аппроксимацию, минимизируя максимальное отклонение.
* В сравнении с методом наименьших квадратов, он менее чувствителен к выбросам.
* Метод применим к любому набору точек, если значения различны.
* Алгоритмическая сложность: из-за вычисления всех парных наклонов.
* Численные ошибки минимальны, так как используется медианная оценка вместо средних значений.

**Вывод**

В ходе лабораторной работы мной был исследован метод наименьших модулей для линейной аппроксимации данных. Данный подход проявил свою надежность в условиях наличия выбросов, что делает его особенно привлекательным для анализа реалистичных данных. Однако стоит учитывать, что выполнение этого метода требует квадратичного времени из-за необходимости сортировки и вычисления медианных значений. Это может ограничить его применение при работе с большими объемами данных. В связи с этим, важно учитывать баланс между устойчивостью метода к аномальным значениям и его вычислительной сложностью.