МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальная научно-образовательная корпорация ИТМО»

ФАКУЛЬТЕТ ПииКТ

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине

«Вычислительная математика»

Вариант №3

Выполнил:

Студент группы P3219

Билобрам Денис Андреевич

Преподаватель:

Бострикова Дарья Константиновна

Санкт-Петербург, 2024

1. **Цель лабораторной работы:**

Найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов.

1. **Порядок выполнения:**
2. Вычислительная реализация задачи.
3. Программная реализация задачи.

**3. Вычислительная часть**

Функция и интервал:

****

1) Таблица табулирования функции:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | -2 | -1.8 | -1.6 | -1.4 | -1.2 | -1.0 | -0.8 | -0.6 | -0.4 | -0.2 | 0 |
| Y | -0.421 | -0.533 | -0.669 | -0.818 | -0.946 | -1 | -0.938 | -0.766 | -0.528 | -0.266 | 0 |

2) Линейное и квадратичные приближения:

Для нахождения линейной функции аппроксимации решим систему уравнений:

По методу Крамера:

Линейное функция аппроксимации –

Для нахождения квадратичной функции аппроксимации решим систему уравнений:

По методу Крамера:

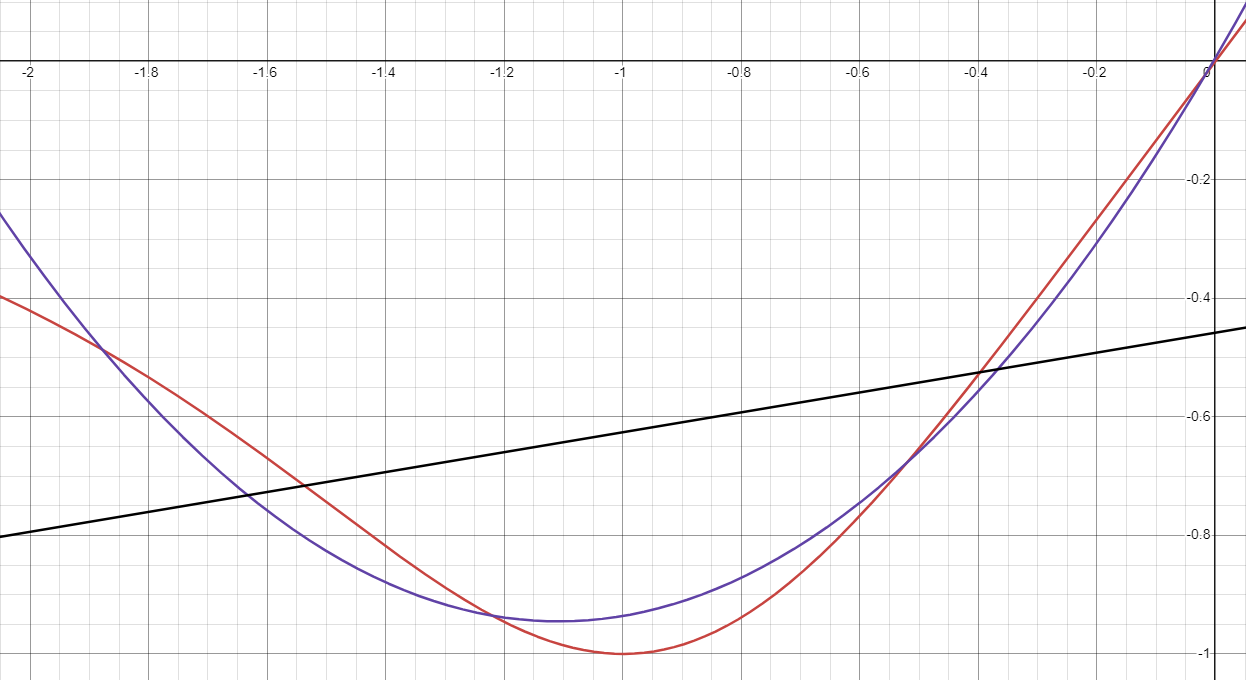
Квадратичная функция аппроксимации –

3) Среднеквадратичное отклонение аппроксимирующих функций:

;

4) Наилучшим приближением будет квадратичное так как среднеквадратичное отклонение минимальное.

5) Графики:



Красный график – исходная функция.

Чёрный график – линейное приближение.

Синий график – квадратичное приближение.

**4. Листинг программы**

**def** **lin\_approx**(x\_i, y\_i):

A = np.vstack([x\_i, np.ones(len(x\_i))]).T

m, c = np.linalg.lstsq(A, y\_i, rcond=None)[**0**]

**return** f"{round(m, 3)}\*x+{round(c, 3)}"

**def** **quad\_approx**(x\_i, y\_i):

A = np.vstack([x\_i\*\***2**, x\_i, np.ones(len(x\_i))]).T

a, b, c = np.linalg.lstsq(A, y\_i, rcond=None)[**0**]

**return** f"{round(a, 3)}\*x\*\*2+{round(b, 3)}\*x+{round(c, 3)}"

**def** **cub\_approx**(x\_i, y\_i):

A = np.vstack([x\_i\*\***3**, x\_i\*\***2**, x\_i, np.ones(len(x\_i))]).T

b = y\_i

a, b, c, d = np.linalg.lstsq(A, b, rcond=None)[**0**]

**return** f"{round(a, 3)}\*x\*\*3+{round(b, 3)}\*x\*\*2+{round(c, 3)}\*x+{round(d, 3)}"

**def** **exp\_approx**(x\_i, y\_i):

**if** np.any(y\_i <= **0**):

**raise** **ValueError**("y\_i must be positive for exponential approximation")

A = np.vstack([x\_i, np.ones(len(x\_i))]).T

b = np.log(y\_i)

b, ln\_a = np.linalg.lstsq(A, b, rcond=None)[**0**]

a = np.exp(ln\_a)

**return** f"{round(a, 3)}\*ezp({round(b, 6)}\*x)"

**def** **log\_approx**(x\_i, y\_i):

**if** np.any(x\_i <= **0**):

**raise** **ValueError**("x\_i must be positive for logarithmic approximation")

A = np.vstack([np.log(x\_i), np.ones(len(x\_i))]).T

b, a = np.linalg.lstsq(A, y\_i, rcond=None)[**0**]

**return** f"{round(a, 3)}+{round(b, 3)}\*log(x)"

**def** **pow\_approx**(x\_i, y\_i):

**if** np.any(x\_i <= **0**) **or** np.any(y\_i <= **0**):

**raise** **ValueError**("x\_i and y\_i must be positive for power approximation")

A = np.vstack([np.log(x\_i), np.ones(len(x\_i))]).T

b, ln\_a = np.linalg.lstsq(A, np.log(y\_i), rcond=None)[**0**]

a = np.exp(ln\_a)

**return** f"{round(a, 3)}\*x\*\*{round(b, 3)}"

**5. Результаты выполнения программы**

Линейная аппроксимация: 4.039\*x+-3.302, среднеквадратичное отклонение: 3.63482681

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [-1.2825, -0.8786, -0.0708, 0.737, 2.7565, 4.776, 8.815, 12.854, 16.893, 37.088]

epsi\_i: [1.7825, 1.5786, 1.0708, 0.763, 0.2435, -0.776, -1.815, -2.854, -2.893, 2.912]

Квадратичная аппроксимация: 0.232\*x\*\*2+1.722\*x+-0.391, среднеквадратичное отклонение: 0.01694701247999997

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [0.528, 0.7257, 1.1351, 1.563, 2.714, 3.981, 6.863, 10.209, 14.019, 40.029]

epsi\_i: [-0.028, -0.0257, -0.1351, -0.063, 0.286, 0.019, 0.137, -0.209, -0.019, -0.029]

Кубическая аппроксимация: 0.005\*x\*\*3+0.155\*x\*\*2+1.988\*x+-0.571, среднеквадратичное отклонение: 0.016893231925000064

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [0.4624, 0.6787, 1.1212, 1.577, 2.7766, 4.065, 6.923, 10.181, 13.869, 39.809]

epsi\_i: [0.0376, 0.0213, -0.1212, -0.077, 0.2234, -0.065, 0.077, -0.181, 0.131, 0.191]

Экспоненциальная аппроксимация: 0.996\*exp(0.438484\*x), среднеквадратичное отклонение: 165.1821133466685

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [1.2402, 1.2957, 1.4145, 1.5442, 1.9227, 2.394, 3.7115, 5.7542, 8.921, 79.9045]

epsi\_i: [-0.7402, -0.5957, -0.4145, -0.0442, 1.0773, 1.606, 3.2885, 4.2458, 5.079, -39.9045]

Логарифмическая аппроксимация: 1.968+10.22\*log(x), среднеквадратичное отклонение: 39.009080678770935

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [-5.116, -3.2526, -0.3125, 1.968, 6.1119, 9.052, 13.1958, 16.1359, 18.4165, 25.5004]

epsi\_i: [5.616, 3.9526, 1.3125, -0.468, -3.1119, -5.052, -6.1958, -6.1359, -4.4165, 14.4996]

Степенная аппроксимация: 1.448\*x\*\*1.437, среднеквадратичное отклонение: 0.11065812385595575

x\_i: [0.5, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 10.0]

y\_i: [0.5, 0.7, 1.0, 1.5, 3.0, 4.0, 7.0, 10.0, 14.0, 40.0]

fi(x\_i): [0.5348, 0.695, 1.0508, 1.448, 2.5931, 3.9206, 7.0209, 10.6152, 14.6281, 39.6067]

epsi\_i: [-0.0348, 0.005, -0.0508, 0.052, 0.4069, 0.0794, -0.0209, -0.6152, -0.6281, 0.3933]

Наилучшая аппроксимация: 0.005\*x\*\*3+0.155\*x\*\*2+1.988\*x+-0.571

Коэффициент корреляции Пирсона для линейной зависимости: 0.9859913494726831

Коэффициент детерминации: 0.9998706986682367

Высокая точность аппроксимации.

**6. Выводы**

В ходе лабораторной работы был выполнен поиск оптимальной функции для аппроксимации заданной функции с использованием метода наименьших квадратов (МНК). Были рассмотрены различные варианты функций, и на основе сравнительного анализа ошибок аппроксимации была выбрана наиболее подходящая функция. Это позволило достичь высокой точности приближения исходной функции, что подтверждает эффективность использования МНК для задач аппроксимации.