Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Лабораторная работа № 3

«Аппроксимация методом наименьших квадратов.
Двухпараметрические модели.»
по курсу «Численные методы»

Выполнила:
студент группы ИУ9-61Б
Окутин Денис
Проверила:
Домрачева А. Б.

Москва, 2024

1. Цель

Целью данной работы является изучение приближения заданной функции путем аппроксимации алгебраическими многочленами, применяя метод наименьших квадратов, используя априорные данные о приближаемой функции.

Постановка задачи

Дано:

$x_a = \frac{x_0 + x_n}{2}$, $x_g = \sqrt{x_n x_0}$, $x_h = \frac{2}{\frac{1}{x_0} + \frac{1}{x_n}}$

Одна из функций:

$$z_1(x) = a * x + b.$$

$$z_2(x) = a * x^b$$

$$z_3(x) = a * x^e$$

$$z_4(x) = a * \ln(x) + b$$

$$z_5(x) = \frac{a}{x} + b$$

$$z_6(x) = \frac{1}{a * x + b}$$

$$z_7(x) = \frac{x}{a * x + b}$$

$$z_8(x) = a * e^{\frac{b}{x}}$$

$$z_9(x) = \frac{1}{a * \ln(x) + b}$$

функция $y = f(x)$ задана конечным набором точек

$y_i = f(x_i),\ i = \overline{0, n}$ на отрезке $[a, b], a = x_0,\ b = x_n,\ x_i = a + ih, h = \frac{(b-a)}{n}$

x_i	x_0	x_1	…	x_{n-1}	x_n
y_i	y_0	y_1	…	y_{n-1}	y_n

Задание:

- Построить графики таблично заданной функции и функции z(x);

- Найти значения x_a , x_g, x_h, y_a, y_g, y_h, $z(x_a)$, $z(x_a)$, $z(x_a)$, δ_1 … δ_9, $\delta_k = \min\ \delta_i$;
- Составить систему уравнений для определения a и b и решить её;
- Найти среднеквадратичное отклонение (СКО) Δ и относительную ошибку δ;

Индивидуальный вариант:

$y = f(x)$ задана конечным набором точек:

x_i	1	1.5	2	2.5	3	3.5	4	4.5	5
y_i	1.24	1.74	1.61	2.16	3.06	2.88	4.53	5.40	7.07

2. Основные теоретические сведения и этапы работы

Значения приближаемой функции $y = f(x)$ заданы лишь в узлах $(x_i, y_i), i = 0, \dots, n$. Необходимо для начала решить задачу аппроксимации: найти гладкую аналитически заданную функцию $z(x)$, доставляющую наименьшее значение величине

$$\text{СКУ} = \sqrt{\sum_{k=0}^{n} (z(x_k) - y_k)^2} = \sqrt{\sum_{k=0}^{n} \varepsilon_k^2}.$$

Данную величину называют среднеквадратичным уклонением (СКУ) функции $z(x)$ от системы узлов $(x_i, y_i), i = 0, \dots, n$, а описанный подход к решению задачи приближения функции – методом наименьших квадратов (МНК)

Как правило, $z(x)$ отыскивают в виде линейной комбинации заранее заданных функций:

$$z(x) = \lambda_1 \varphi_1(x) + \cdots + \lambda_m \varphi_m(x).$$

Параметры λ_i, $i = 1, \dots, m$ являются решениями линейной системы наименьших квадратов

$$A\lambda = b,$$

где λ – столбец параметров λ_i, $A = \left(a_{ij}\right)$ – симметричная положительно определенная матрица с коэффициентами $a_{ij} = \sum_{k=0}^{n} \varphi_i(x_i)\varphi_j(x_k)$,
b – столбец правой части системы, $b_i = \sum_{k=0}^{n} \varphi_i(x_k) y_k$, $i, j = 1, \dots, m$.

Таким образом, система МНК имеет единственное решение $\lambda_1^*, \dots, \lambda_m^*$, дающее СКУ наименьшее значение (для всех функций данного вида). Система решается методом квадратного корня.

Если приближаемая функция достаточно гладкая, хотя вид ее и неизвестен, аппроксимирующую функцию нередко ищут в виде алгебраического многочлена

$$z(x) = \lambda_1 + \lambda_2 x + \dots + \lambda_m x^{m-1}.$$

Тогда $\varphi_i = x^{i-1}$ и элементы матрицы A получают по формулам:

$$a_{ij} = \sum_{k=0}^{n} x_k^{i+j-1},$$

а свободные члены –

$$b_i = \sum_{k=0}^{n} y_k x_k^{i+j-1}, \qquad i, j = 1, \dots, m.$$

Абсолютной погрешностью аппроксимации выступает СКО:

$$\Delta = \frac{\text{СКУ}}{\sqrt{n}} = \frac{1}{\sqrt{n}} \sqrt{\sum_{k=0}^{n} \left(y_k - \lambda_1 - \lambda_2 x_k - \dots - \lambda_m x_k^{m-1}\right)^2},$$

относительная ошибка:

$$\delta = \frac{\Delta}{||y||} = \frac{\Delta}{\sqrt{\sum_{k=0}^{n} y_k^2}}$$

Затем необходимо вычислить значения величин, представленных на рис1.

$$\delta_1=|z(x_a)-y_a|,\quad \delta_2=|z(x_g)-y_g|,\quad \delta_3=|z(x_a)-y_g|,$$
$$\delta_4=|z(x_g)-y_a|,\quad \delta_5=|z(x_h)-y_a|,\quad \delta_6=|z(x_a)-y_h|,$$
$$\delta_7=|z(x_h)-y_h|,\quad \delta_8=|z(x_h)-y_g|,\quad \delta_9=|z(x_g)-y_h|$$

Рисунок 1. Формулы для нахождения наименьшего значения k.

Выбрав, функцию, дающая наименьшее значение, найти гладкую аналитически заданную функцию $z\_new(x)$, доставляющую наименьшее значение соответствующей величине (см рис 2,3).

Определение коэффициентов a и b для выбранной функции.
Предположим, что наименьшим из δ_i оказалось δ_1. В этом случае $z(x)=z_1(x)=ax+b$. Тогда СКУ равно $\sum_{i=0}^{n}(ax_i+b-y_i)^2$.
Коэффициенты a и b находим по МНК (см. работу 8):

$$a\sum_{i=0}^{n} x_i^2+b\sum_{i=0}^{n} x_i=\sum_{i=0}^{n} x_i y_i,$$
$$a\sum_{i=0}^{n} x_i+b(n+1)=\sum_{i=0}^{n} y_i.$$

Коэффициенты a и b в остальных случаях находим после предварительных преобразований.

Рисунок 2. Нахождение коэффициентов a и b в тривиальном случае

I. Функции $z_2(x)$, $z_3(x)$ и $z_8(x)$ следует предварительно прологарифмировать (например, $\ln(z_2(x))=\ln a + b\ln x$). После этого нужно минимизировать величину

$$\sum_{i=0}^{n}(\ln a + b\ln x_i - \ln y_i)^2$$

и решать соответствующую систему относительно $\ln a$ и b. Элементами матрицы этой системы будут следующие:

$$A = \sum_{i=0}^{n}(\ln x_i)^2,$$

$$B = \sum_{i=0}^{n}\ln x_i,$$

$$D_1 = \sum_{i=0}^{n}(\ln x_i \cdot \ln y_i),$$

$$D_2 = \sum_{i=0}^{n}\ln y_i.$$

После решения системы по величине $\ln a$ определяем a.

2. В случае $z_6(x)$, $z_7(x)$ и $z_9(x)$ следует предварительно перейти к обратным величинам, например $\frac{1}{z_7(x)} = \frac{ax+b}{x} = a + \frac{b}{x}$. Минимизируется величина $\sum_{i=0}^{n}\left(a + \frac{b}{x_i} - \frac{1}{y_i}\right)^2$, и элементы матрицы соответствующей системы уравнений будут состоять из сумм обратных величин $1/x_i$ и $1/y_i$.

3. Функции $z_4(x)$ и $z_5(x)$ остаются без изменения, однако элементы матрицы в этих случаях будут состоять из сумм значений $\ln x_i$ и $1/x_i$ соответственно.

Рисунок 3. Нахождение коэффициентов a и b в остальных случаях

3. Реализация

Листинг 1. Программное решение задачи.

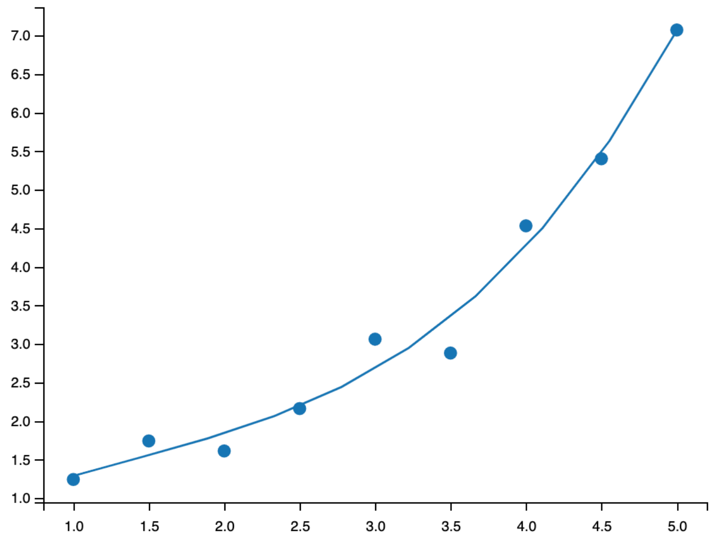
```
import math

import numpy as np
from matplotlib import pyplot as plt


def find_lambd(A, b):
    return (np.linalg.inv(A).dot(b.T)).T


def find_A(xs, n, m):
    return np.array([[sum(xs[k] ** (i + j) for k in range(0, n + 1)) for j in range(0, m)] for i in range(0, m)])
```

```
def find_b(xs, ys, n, m):
    return np.array([sum(ys[k] * (xs[k] ** i) for k in range(0, n + 1)) for i in
range(0, m)])


def z(lambd, x, m):
    return sum([lambd[i] * x ** i for i in range(m)])


def count_a_g_h(arr):
    a = (arr[0] + arr[len(arr) - 1]) / 2
    g = math.sqrt(arr[0] * arr[len(arr) - 1])
    h = 2 / ((1 / arr[0]) + (1 / arr[len(arr) - 1]))
    return a, g, h


def find_z_num(z):
    nums = [
        [1, 3, 6],
        [4, 2, 9],
        [5, 8, 7]
    ]

    x_a, x_g, x_h = count_a_g_h(x)
    y_a, y_g, y_h = count_a_g_h(y)

    x_ = [x_a, x_g, x_h]
    y_ = [y_a, y_g, y_h]
    min_ = 10000
    min_idx = (0, 0)
    for i in range(len(x_)):
        for j in range(len(y_)):
            if abs(z(x_[i]) - y_[j]) < min_:
                min_idx = (i, j)
                min_ = abs(z(x_[i]) - y_[j])
    return nums[min_idx[0]][min_idx[1]]


def show_z(z, X, Y):
    plt.scatter(X, Y)
    x_ = np.linspace(1, 5, 10)
    y_ = list(map(z, x_))
    plt.plot(x_, y_)


def new_z(x, y, n, m):
    x_ln = list(map(np.log, x))
    y_swap = list(map(lambda x_var: 1 / x_var, y))

    A = find_A(x_ln, n, m)
    b = find_b(x_ln, y_swap, n, m)

    lambd = find_lambd(A, b)
    return lambda x_var: 1 / (lambd[0] + lambd[1] * math.log(x_var, math.e))


if __name__ == "__main__":
    x = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]
    y = [1.24, 1.74, 1.61, 2.16, 3.06, 2.88, 4.53, 5.40, 7.07]

    m = 4
    n = len(x) - 1

    A = find_A(x, n, m)
```

```
b = find_b(x, y, n, m)
lambd = find_lambd(A, b)
z_ = lambda p: z(lambd, p, m)

D = sum([(y[k] - z_(x[k])) for k in range(m + 1)]) ** 2
D = np.sqrt(D) / np.sqrt(n)
print("\nCKO z:", D)

show_z(z_, x, y)
z_num = find_z_num(z_)
z_new = new_z(x, y, n, m)

D = sum([(y[k] - z_new(x[k])) for k in range(m + 1)]) ** 2
D = np.sqrt(D) / np.sqrt(n)
print("\nCKO z_new:", D)

show_z(z_new, x, y)
plt.show()
```

4. Результаты

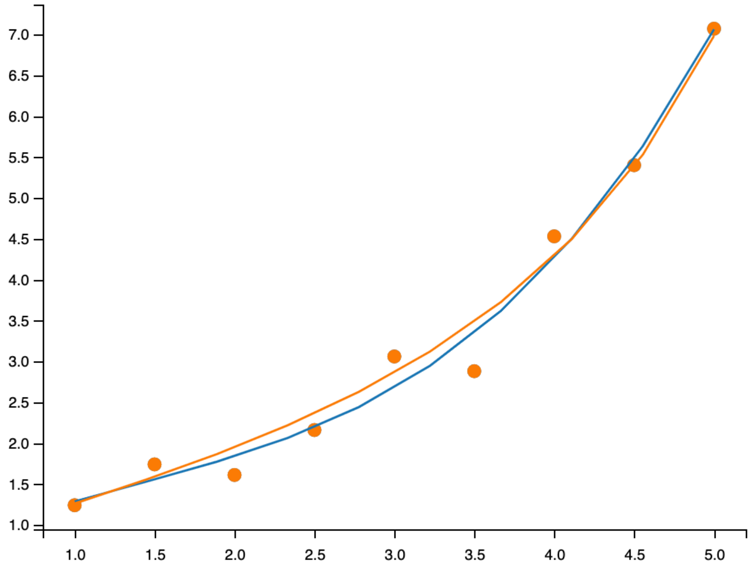
a) Был построен график функции z(x) – некоторая монотонная функция в виде многочлена с коэффициентами и f(x)



b) Были найдены значения x_a , x_g, x_h, y_a, y_g, y_h, $z(x_a)$, $z(x_a)$, $z(x_a)$, δ_1 … δ_9, $\delta_k = \min\ \delta_i$

```
СКО z: 0.09299525546328605
x_a = 3.0, x_g = 2.23606797749979, x_h = 1.6666666666666667
y_a = 4.155, y_g = 2.9608782480878877, y_h = 2.109939831528279
z_x_a = 2.6736363636370584, z_x_g = 1.9928228608369642, z_x_h = 1.6462414266130379
k = 9
```

c) Получен результат апроксимации (оранжевая линия)



d) Результирующее СКО

```
СКО z_new: 0.0754889650810652
```

5. Вывод

В ходе выполнения лабораторной работы был изучен и реализован метод приближения функции с помощью аппроксимации алгебраическими многочленами с применением метода наименьших квадратов, был найден аппроксимирующий многочлен и найдены его значения в серединах отрезков между заданными узловыми точками. В результате тестирования была получена гладкая аппроксимирующая функция, доставляющая наименьшее значение среднеквадратичному уклонению. Далее эта функция была использована для того, чтобы выбрать необходимую функцию z из условия лабораторной работы, для этого были вычислены 9 значений δ_i и выбрано минимальное из них. В моем случае это было δ_9. Т.е функция, для которой я искал параметры a и b выглядела следующим образом:

$$z_9(x) = \frac{1}{a * \ln(x) + b}$$

Для нее были вычислены неизвестные коэффициенты с помощью МНК и получено значение СКО, которое показало, что аппроксимация стала более точной.