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»

Выполнил:

студент группы ИУ9-61Б

Локшин Вячеслав

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.

2. 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:

$$z_1(x) = a \cdot x + b$$

$$z_2(x) = a \cdot x^b$$

$$z_3(x) = a \cdot x^e$$

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

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

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

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

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

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

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 $x_a, x_g,\ x_h,\ y_a,\ y_g,\ y_h,\ z(x_a),\ z(x_g)$, $z(x_h), \delta_1 \dots \delta_9, \delta_k = \min\limits_i \delta_i$
- Составить систему уравнений для определения a и b и решить её;
- Найти среднеквадратичное отклонение (СКО) Δ и относительну 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	2.61	1.62	1.17	0.75	0.30	0.75	1.03	0.87	0.57

3. Основные теоретические сведения и этапы работы

Необходимо построить гладкую монотонную функцию, близкую к точкам, затем вычислить значения $x_a,\ x_g,\ x_h,\ y_a,\ y_g,\ y_h,\ z(x_a),\ z(x_g)$, $z(x_h)$

Значения $\delta_1 \dots \delta_9$ вычисляются по формулам:

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

Выбрав функцию , $\delta_k = \min_i \delta_i$, найти гладкую аналитически заданную функцию z_k(x) (Рисунок 1, 2).

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 1 - Нахождение коэффициентов a и b в случае 1

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 2 - Нахождение коэффициентов a и b в случаях 2, 3, 4, 5, 6, 7, 8, 9

Значения приближаемой функции $y=f(x)$ заданы в узлах (x_i, y_i), $i=0,\dots,n$. Необходимо 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, $z(x)$ можно найти в виде линейной комбинации заранее заданных функций:

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

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

$$A\lambda = b,$$

где λ – столбец параметров $\lambda_i,\ A = (a_{ij})$ – 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:

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

4. Реализация

```
Вычисляем средние для x:

x0 = 1, xn = 5

xa = (x0+xn)/2 = 3.0

xg = sqrt(x0*xn) = 2.23606797749979

xh = 2 / (1/x0 + 1/xn) = 1.6666666666666667

Вычисляем средние для y:

y0 = 2.61, yn = 0.57

ya = (y0+yn)/2 = 1.59

yg = sqrt(y0*yn) = 1.2197130810153671 ~ 1.22

yh = 2 / (1/y0 + 1/yn) = 0.9356603773584905 ~ 0.94

Посчитаем значение функции в z(xa), z(xg), z(xh):

z(xa) = z(3.00) = 0.85    // 0.8456

z(xg) = z(2.24) = 1.12    // 1.118946793278549

z(xh) = z(0.94) = 2.61    // 2.6108539019963706

Посчитаем дельты:

d1 = |0.85 - 1.59| = 0.74

d2 = |1.12 - 1.22| = 0.10

d3 = |0.85 - 1.22| = 0.37

d4 = |1.12 - 1.59| = 0.47

d5 = |2.61 - 1.59| = 1.02

d6 = 0.09 -> минимум

d7 = |2.61 - 0.94| = 1.67

d8 = |2.61 - 1.22| = 1.39

d9 = |1.12 - 0.94| = 0.19
```

Листинг 1. Нахождение коэффицентов a, b, СКУ, СКО

```go
package main

import (
    "fmt"
    "log"
    "math"
```

```
)

func getSum(nums []float64) float64 {
    summer := 0.0
    for _, num := range nums {
        summer += num
    }
    return summer
}

func getInverseArr(nums []float64) []float64 {
    inverse := make([]float64, len(nums))
    for i, num := range nums {
        inverse[i] = 1 / num
    }
    return inverse
}

func getSquaresArr(nums []float64) []float64 {
    squares := make([]float64, len(nums))
    for i, num := range nums {
        squares[i] = num * num
    }
    return squares
}

func getDivideFirstArrBySecond(first, second []float64) []float64 {
    if len(first) != len(second) {
        log.Fatalf("len first != len second")
    }
    dividers := make([]float64, len(first))

    for i := 0; i < len(first); i++ {
        dividers[i] = first[i] / second[i]
    }
    return dividers
}

func getCoefficient(x, y []float64,
    f func(ySum, xInverseSum, xInverseSquaresSum, yDivideByXSum, N float64)
float64) float64 {

    if len(x) != len(y) {
        log.Fatalf("len x != len y")
    }

    N := float64(len(x))
    xInverse := getInverseArr(x)
    xInverseSquares := getSquaresArr(xInverse)
    yDivideByX := getDivideFirstArrBySecond(y, x)

    ySum := getSum(y)
    xInverseSum := getSum(xInverse)
    xInverseSquaresSum := getSum(xInverseSquares)
    yDivideByXSum := getSum(yDivideByX)

    return f(ySum, xInverseSum, xInverseSquaresSum, yDivideByXSum, N)
}

func getACoefficient(x, y []float64) float64 {
    return getCoefficient(
        x, y,
        func(ySum, xInverseSum, xInverseSquaresSum, yDivideByXSum, N
```

```
float64) float64 {
            return (ySum*xInverseSquaresSum - xInverseSum*yDivideByXSum) /
               (N*xInverseSquaresSum - xInverseSum*xInverseSum)
        })
}

func getBCoefficient(x, y []float64) float64 {
    return getCoefficient(
        x, y,
        func(ySum, xInverseSum, xInverseSquaresSum, yDivideByXSum, N
float64) float64 {
            return (N*yDivideByXSum - xInverseSum*ySum) /
               (N*xInverseSquaresSum - xInverseSum*xInverseSum)
        })
}

func getSCU(a, b float64, x, y []float64) float64 {
    scuElements := make([]float64, len(x))

    for i := 0; i < len(x); i++ {
        scuElements[i] = math.Abs(a + b/x[i] - y[i])
    }

    scuSquaresElements := getSquaresArr(scuElements)

    return getSum(scuSquaresElements)
}

func getSCO(scu float64, N float64) float64 {
    return scu / math.Sqrt(N)
}

func main() {

    x := []float64{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5}
    y := []float64{2.61, 1.62, 1.17, 0.75, 0.30, 0.75, 1.03, 0.87, 0.57}

    fmt.Println("x:", x)
    fmt.Println("y:", y)

    a := getACoefficient(x, y)
    b := getBCoefficient(x, y)
    fmt.Printf("Коэффициенты a: %f и b: %f\n", a, b)

    fmt.Printf("Итоговая функция: y = %f + %f / x\n", a, b)

    scu := getSCU(a, b, x, y)
    fmt.Printf("СКУ: %f\n", scu)

    sco := getSCO(scu, float64(len(x)))
    fmt.Printf("Средняя ошибка апроксимации: %f\n", sco)

}
```

5. Результаты

Была построена гладкая монотонная функция.

Были найдены значения $x_a, x_g,\ x_h,\ y_a,\ y_g,\ y_h,\ z(x_a),\ z(x_g),$ $z(x_h),\ \delta_1 \dots\ \delta_9$ и выбран тип уравнения для аппроксимирующей функции

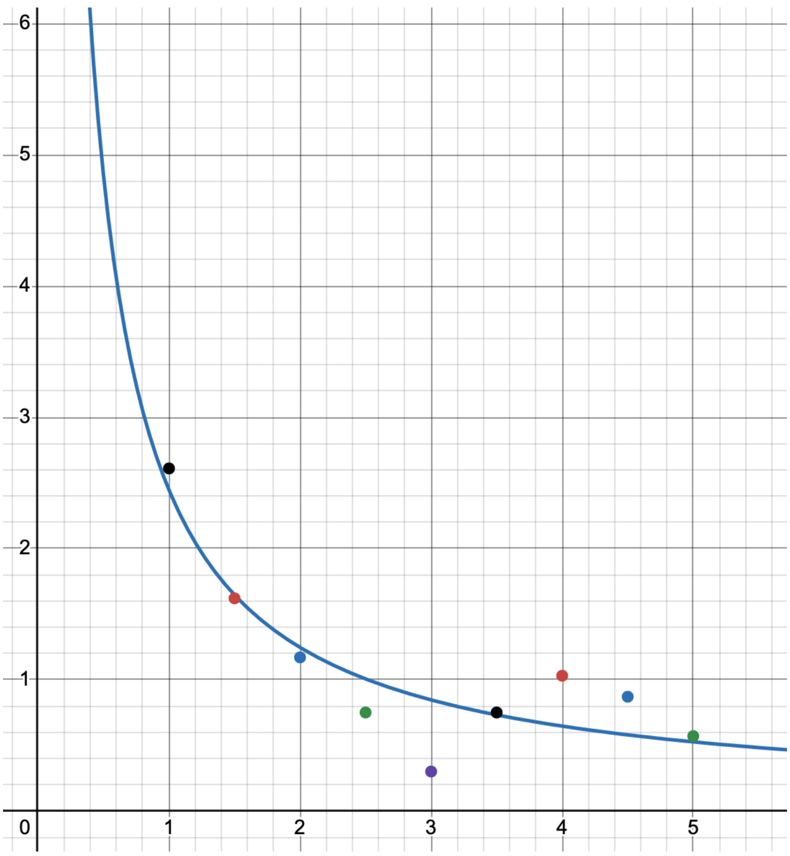
$$z_6(x) = \frac{1}{a \cdot x + b}$$

Результат программы:

Рисунок 1 – Пример вывода программы

```
/Users/slavaruswarrior/Library/Caches/JetBrains/GoLand2023.3/tmp/GoLar
x: [1 1.5 2 2.5 3 3.5 4 4.5 5]
y: [2.61 1.62 1.17 0.75 0.3 0.75 1.03 0.87 0.57]
Коэффициенты a: 0.045512 и b: 2.400349
Итоговая функция: y = 0.045512 + 2.400349 / x
СКУ: 0.631212
Средняя ошибка апроксимации: 0.210404
```

Рисунок 2 – Построенная функция



6. 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. В результате тестирования была получена гладкая аппроксимирующая функция, удовлетворяющая уравнению

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

Была написана программа, вычисляющая коэффициенты a и b этого случая, а также подсчет СКУ и СКО. Был построенный уточненный график.