Национальный исследовательский университет ИТМО
Факультет программной инженерии и компьютерной
Дисциплина «Вычислительная математика»

Отчет
по лабораторной работе №4
Вариант 25

Выполнил:

Туляков Е.В.

P32101

Преподаватель:

Рыбаков С. Д.

Санкт-Петербург
2023г

Цель лабораторной работы: найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов.

Метод Наименьших квадратов:

Мерой отклонения многочлена $\varphi(x)$ от заданной функции *f(x)* на множестве точек $((x_i, y_i)$ является величина S (критерий минимизации), равная сумме квадратов разности между значениями многочлена и функции для всех точек $x_0, x_1, \dots, x_n$:

$$S = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} [\varphi(x_i) - y_i]^2 \to min$$

Задача нахождения наилучших значений параметров $a_0, a_1, \dots, a_m$ сводятся к некоторой минимизации отклонений ε_i.

Параметры $a_0, a_1, \dots, a_m$ эмпирической формулы находятся из условия минимума функции $S = S(a_0, a_1, a_2, \dots, a_m)$,.

Так как здесь параметры выступают в роли независимых переменных функции S, то ее минимум найдем, приравнивая к нулю частные производные по этим переменным (m – степень многочлена, n - число точек в таблице) :

$$\frac{\partial S}{\partial a_0} = 2\sum_{i=1}^{n} a_0 + a_1x_i + \dots + a_{m-1}x_i^{m-1} + a_mx_i^m - y_i = 0$$

$$\frac{\partial S}{\partial a_1} = 2\sum_{i=1}^{n} (a_0 + a_1x_i + \dots + a_{m-1}x_i^{m-1} + a_mx_i^m - y_i)x_i = 0$$

$$\dots \dots \dots$$

$$\frac{\partial S}{\partial a_m} = 2\sum_{i=1}^{n} (a_0 + a_1x_i + \dots + a_{m-1}x_i^{m-1} + a_mx_i^m - y_i)x_i^m = 0$$

Вычислительная реализация:

$$y = \frac{5x}{x^4 + 11}$$

i	1	2	3	4	5	6	7	8	9	10	11
X	-2.0	-1.8	-1.6	-1.4	-1.2	-1.0	-0.8	-0.6	-0.4	-0.2	0.0
Y	-0.370	-0.419	-0.456	-0.472	-0.459	-0.417	-0.351	-0.270	-0.181	-0.091	0.000

$$SX = \sum_{i=1}^{n} x_i = -11 \qquad SXX = \sum_{i=1}^{n} x_i^2 = 15.4$$

$$SY = \sum_{i=1}^{n} y_i = -3.486 \qquad SXY = \sum_{i=1}^{n} x_iy_i = 3.988$$

$$SXXX = \sum_{i=1}^{n} x_i^3 = -24.2 \qquad SXXY = \sum_{i=1}^{n} x_i^2y_i = -6.062$$

$$SXXXX = \sum_{i=1}^{n} x_i^4 = 40.533$$

Линейная:

$$\begin{cases} aSXX + bSX = SXY \\ aSX + bn = SY \end{cases} \quad \begin{cases} 15.4a - 11b = 3.988 \\ -11x + 11b = -3.486 \end{cases} \quad \begin{cases} a = 0.114 \\ b = -0.203 \end{cases}$$

$$y = 0.114x - 0.203$$

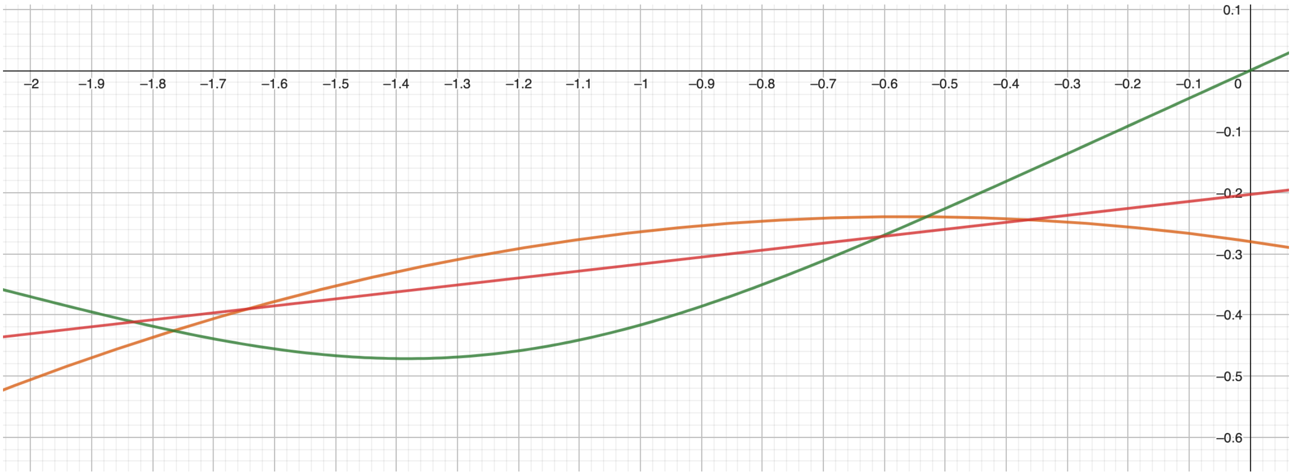
Квадратичная:

$$\begin{cases} cn + bSX + aSXX = SY \\ cSX + bSXX + aSXXX = SXY \\ cSXX + bSXXX + aSXXXX = SXXY \end{cases} \quad \begin{cases} 11c - 11b + 15.4a = -3.486 \\ -11c + 15.4b - 24.2a = 3.988 \\ 15.4c - 24.2b + 40.533a = -6.062 \end{cases}$$

$$\begin{cases} a = -0.129 \\ b = -0.145 \\ c = -0.280 \end{cases}$$

$y = -0.129x^2 - 0.145x - 0.28$

$\sigma_1 = 0.096$
$\sigma_2 = 0.141$
Лучшее приближение - линейное.



Код методов:

```
private static double linearApprox() {
    Dot[] dots = DotCollection.getDots();
    double[] coefficients = getLinearCoefficients(dots);
    double deviation = getDeviation(coefficients);
    save(coefficients, deviation);
    Printer.printP(coefficients, deviation);
    Printer.printR(getR(dots));
    return deviation;
}
```

```
private static double quadraticApprox() {
```

```java
    numberApprox++;
    Dot[] dots = DotCollection.getDots();
    double[] coefficients = getQuadraticCoefficients(dots);
    double deviation = getDeviation(coefficients);
    save(coefficients, deviation);
    Printer.printP(coefficients, deviation);
    return deviation;
}

private static double cubicApprox() {
    numberApprox++;
    Dot[] dots = DotCollection.getDots();
    double[] coefficients = getCubicCoefficients(dots);
    double deviation = getDeviation(coefficients);
    save(coefficients, deviation);
    Printer.printP(coefficients, deviation);
    return deviation;
}

private static double powerApprox() {
    numberApprox++;
    Dot[] dots = DotCollection.getDots();
    Dot[] cloneDots = cloneDots(dots);
    for (Dot dot : cloneDots) {
        double x = dot.getX();
        double y = dot.getY();
        dot.setX(Math.log(x));
        dot.setY(Math.log(y));
    }
    double[] coefficients = getLinearCoefficients(cloneDots);
    double[] ab = new double[2];
```

```java
    ab[0] = Math.pow(Math.E, coefficients[1]);
    ab[1] = coefficients[0];
    double deviation = getDeviation(ab);
    save(ab, deviation);
    Printer.printP(ab, deviation);
    return deviation;
}

private static double exponentialApprox() {
    numberApprox++;
    Dot[] dots = DotCollection.getDots();
    Dot[] cloneDots = cloneDots(dots);
    for (Dot dot : cloneDots) {
        double y = dot.getY();
        dot.setY(Math.log(y));
    }
    double[] coefficients = getLinearCoefficients(cloneDots);
    double[] ab = new double[2];
    ab[0] = Math.pow(Math.E, coefficients[1]);
    ab[1] = coefficients[0];
    double deviation = getDeviation(ab);
    save(ab, deviation);
    Printer.printP(ab, deviation);
    return deviation;
}

private static double logApprox() {
    numberApprox++;
    Dot[] dots = DotCollection.getDots();
    Dot[] cloneDots = cloneDots(dots);
    for (Dot dot : cloneDots) {
```

```
        double x = dot.getX();
        dot.setX(Math.log(x));
    }
    double[] coefficients = getLinearCoefficients(cloneDots);
    double deviation = getDeviation(coefficients);
    save(coefficients, deviation);
    Printer.printP(coefficients, deviation);
    return deviation;
}
```

Результат работы:

```
1 4
2 6.079
3 7.2958
4 8.1589
5 8.8283
6 9.3753
7 9.8378
8 10.238
----------------------------------LINEAR----------------------------------
X     1,00     2,00     3,00     4,00     5,00     6,00     7,00     8,00
Y     5,09     5,91     6,74     7,56     8,39     9,22     10,04    10,87
P(x)  5,0863   5,9121   6,7379   7,5637   8,3895   9,2153   10,0412  10,8670
ξ     1,0863   -0,1669  -0,5579  -0,5952  -0,4388  -0,1600  0,2034   0,6290
P1(x) = (0,8258)x + (4,2605) σ = 0,5622
R = 0,9586

---------------------------------QUADRATIC---------------------------------
X     1,00     2,00     3,00     4,00     5,00     6,00     7,00     8,00
Y     4,28     5,80     7,08     8,14     8,97     9,56     9,93     10,06
P(x)  4,2794   5,7968   7,0838   8,1401   8,9659   9,5612   9,9259   10,0600
ξ     0,2794   -0,2822  -0,2120  -0,0188  0,1376   0,1859   0,0881   -0,1780
P2(x) = (-0,1153)x^2 + (1,8633)x + (2,5313) σ = 0,1923

-----------------------------------CUBIC-----------------------------------
X     1,00     2,00     3,00     4,00     5,00     6,00     7,00     8,00
Y     4,06     5,95     7,30     8,23     8,87     9,34     9,77     10,28
P(x)  4,0591   5,9542   7,3040   8,2345   8,8715   9,3409   9,7685   10,2803
ξ     0,0591   -0,1248  0,0082   0,0756   0,0432   -0,0344  -0,0693  0,0423
P3(x) = (0,0210)x^3 + (-0,3985)x^2 + (2,9437)x + (1,4929) σ = 0,0657

-----------------------------------POWER-----------------------------------
X     1,00     2,00     3,00     4,00     5,00     6,00     7,00     8,00
Y     4,27     5,80     6,94     7,89     8,70     9,43     10,10    10,71
P(x)  4,2732   5,8050   6,9443   7,8859   8,7033   9,4337   10,0988  10,7128
ξ     0,2732   -0,2740  -0,3515  -0,2730  -0,1250  0,0584   0,2610   0,4748
P4(x) = (4,2732)x^(0,4420) σ = 0,2873
```

--------------------------------EXPONENTIAL--------------------------------

X	1,00	2,00	3,00	4,00	5,00	6,00	7,00	8,00
Y	5,10	5,73	6,44	7,24	8,14	9,15	10,28	11,56
P(x)	5,0998	5,7320	6,4426	7,2413	8,1391	9,1481	10,2821	11,5568
ξ	1,0998	-0,3470	-0,8532	-0,9176	-0,6892	-0,2272	0,4443	1,3188

P5(x) = (4,5373)e^(0,1169x) σ = 0,8188

--------------------------------LOGARITHMIC--------------------------------

X	1,00	2,00	3,00	4,00	5,00	6,00	7,00	8,00
Y	4,00	6,08	7,30	8,16	8,83	9,38	9,84	10,24
P(x)	3,9999	6,0793	7,2957	8,1588	8,8282	9,3752	9,8377	10,2383
ξ	-0,0001	0,0003	-0,0001	-0,0001	-0,0001	-0,0001	-0,0001	0,0003

P6(x) = (3,0000)ln(x) + (3,9999) σ = 0,0002

---------------- RESULT : P6(x) is the best approximation ----------------

Вывод:

Во время выполнения лабораторной работы я изучил работу метода наименьших квадратов для построения аппроксимирующей функции, заданной в табличном виде. Программа реализует вычисление аппроксимации для элементарных функций. Выбор наилучшего приближения можно сделать, посчитав среднеквадратичное отклонение.