

Национальный исследовательский университет информационных технологий,  
механики и оптики

Факультет Программной Инженерии и Компьютерной Техники

Вариант №12

Лабораторная работа №4

«Аппроксимация функции методом наименьших  
квадратов»

По дисциплине:

«Вычислительная математика»

Работу выполнила:

Студентка группы Р3212

Никонова Наталья Игоревна

Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург

2022

## Цель

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

## Вычислительная реализация

$$y = \frac{4x}{x^4 + 2} \quad x \in [-2, 0] \quad h = 0,2$$

№	1	2	3	4	5	6	7	8	9	10	11	Σ
x	-2	-1.8	-1.6	-1.4	-1.2	-1	-0.8	-0.6	-0.4	-0.2	0	-11
y	-0.444	-0.575	-0.748	-0.959	-1.178	-1.333	-1.328	-1.127	-0.789	-0.399	0	-8.88
$(ax + b - y)^2$	0.219	0.100	0.015	0.012	0.122	0.276	0.293	0.131	0.002	0.105	0.493	1.769
$(a_0 + a_1 * x + a_2 * x^2 - y)^2$	0.035	0.003	0.028	0.023	0.002	0.008	0.022	0.010	0.000	0.004	0.002	0.137

$$n = 11 \quad \sum x_i = -11.0 \quad \sum x_i^2 = 15.4 \quad \sum x_i^3 = -24.2 \quad \sum x_i^4 = 40.533 \quad \sum y_i = -8.88$$

$$\sum x_i y_i = 9.343 \quad \sum x_i^2 y_i = -11.861$$

$$\Delta = n * \sum x_i^2 - \left( \sum x_i \right)^2 = 11 * 15.40 - (-11)^2 = 48.4$$

$$\Delta_1 = n * \sum x_i y_i - \sum x_i * \sum y_i = 11 * 9.343 - (-11) * (-8.88) = 5.093$$

$$\Delta_2 = \sum x_i^2 * \sum y_i - \sum x_i * \sum x_i y_i = 15.4 * (-8.88) - (-11) * 9.343 = -33.979$$

$$a = \frac{\Delta_1}{\Delta} = \frac{5.093}{48.4} = 0.105 \quad b = \frac{\Delta_2}{\Delta} = \frac{-33.979}{48.4} = -0.702$$

Линейная аппроксимация:  $-0.702 + 0.105 * x$ . СКО:  $\sqrt{\frac{1.769}{11}} = 0.401$

$$\begin{cases} a_0 n + a_1 \sum x_i + a_2 \sum x_i^2 = \sum y_i \\ a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 = \sum x_i y_i \\ a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 = \sum x_i^2 y_i \end{cases} \Rightarrow$$

11	-11	15.4	-8.88
-11	15.4	-24.2	9.343
15.4	-24.2	40.533	-11.861

Переставим строки 1 и 3

15.4	-24.2	40.533	-11.861
-11	15.4	-24.2	9.343

11	-11	15.4	-8.88
----	-----	------	-------

Выразим 1-ый элемент из 1-ой строки и подставим в 2 и 3. Получим матрицу:

15.4	-24.2	40.533	-11.861
0	-1.886	4.752	0.871
0	6.286	-13.552	-0.408

Переставим строки 2 и 3

15.4	-24.2	40.533	-11.861
0	6.286	-13.552	-0.408
0	-1.886	4.752	0.871

Выразим 2-ой элемент из 2-ой строки и подставим в 3. Получим матрицу:

15.4	-24.2	40.533	-11.861
0	6.286	-13.552	-0.408
0	0	0.686	0.749

$$a_2 = \frac{0.749}{0.686} = 1.091 \quad a_1 = \frac{-0.408 + 13.552 * 1.091}{6.286} = 2.287$$

$$a_0 = \frac{-11.861 - 40.533 * 1.091 + 24.2 * 2.287}{15.4} = -0.048$$

Квадратичная аппроксимация:  $-0.048 + 2.287 * x + 1.091 * x^2$ . СКО:  $\sqrt{\frac{0.137}{11}} = 0.111$



■  $\frac{4x}{x^4+2}$

■  $-0.702 + 0.105x$

■  $-0.048 + 2.287x + 1.091x^2$

## Код

Полностью туть [https://github.com/nanikon/computational\\_math/tree/lab4/approximation](https://github.com/nanikon/computational_math/tree/lab4/approximation)

Преобразование точек в элементы матрицы и её расчёт:

```
protected fun convertToMatrixAndSolve(data: List<Pair<BigDecimal, BigDecimal>>, n: Int): List<BigDecimal> {
    val variables = listOf(BigDecimal(data.size)) +
        (1 ≤ .. ≤ 2 * n).map { i -> data.sumOf { coords -> coords.first.pow(i) } }
    val bColumn = (0 ≤ .. ≤ n).map { i -> data.sumOf { coords -> coords.second * coords.first.pow(i) } }
    println((variables + bColumn).joinToString(separator = " "))
    val matrix = Matrix(dim: n + 1).apply { this: Matrix
        (0 ≤ .. ≤ n).forEach { i ->
            setRow(i, (variables.subList(i, i + n + 1) + bColumn[i]) as MutableList<BigDecimal>)
        }
    }
    println("Получена матрица:")
    matrix.printMatrix()
    return SLaeCalculator.calculate(matrix)
        .also { println("Решение матрицы: " + it.joinToString(separator = " ")) }
}
```

## Расчёт СКО

```
protected fun calculateStandardDeviation(
    data: List<Pair<BigDecimal, BigDecimal>>,
    function: (BigDecimal) -> BigDecimal
) : BigDecimal =
    data.sumOf { coords -> (function(coords.first) - coords.second).pow(n: 2).also { print("$it ") } }
        .also { println("\nСумма квадратов отклонений: $it") }
        .divide(BigDecimal(data.size), MathContext.DECIMAL64) BigDecimal!
        .sqrt(MathContext.DECIMAL64)
        .also { println("СКО: $it") }
```

## Линейная аппроксимация

```
println("Линейная аппроксимация")
println("Точки не изменяются: " + points.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(y), sum(x*y)")

val coef = convertToMatrixAndSolve(points, n: 1)
val strFunction = "${coef[0]} ${toStr(coef[1])} * x"
val function = { x: BigDecimal -> coef[0] + coef[1] * x }
```

## Коэффициент Пирсона

```

private fun correlation(points: List<Pair<BigDecimal, BigDecimal>>) {
    val midX = points.sumOf { coords -> coords.first }.divide(BigDecimal(points.size), MathContext.DECIMAL64)
    val midY = points.sumOf { coords -> coords.second }.divide(BigDecimal(points.size), MathContext.DECIMAL64)
    val xMinusMidX = points.sumOf { coords -> (coords.first - midX).pow(n: 2) }
    val yMinusMidY = points.sumOf { coords -> (coords.second - midY).pow(n: 2) }
    val xy = points.sumOf { coords -> (coords.first - midX) * (coords.second - midY) }
    val coef = xy.divide(xMinusMidX.multiply(yMinusMidY).sqrt(MathContext.DECIMAL64), MathContext.DECIMAL64)
    val decision = when {
        coef > BigDecimal.ZERO -> "прямо-пропорциональная зависимость, "
        coef < BigDecimal.ZERO -> "обратно-пропорциональная зависимость, "
        else -> "зависимость отсутствует"
    } + when {
        coef == BigDecimal.ZERO -> ""
        coef.abs() > BigDecimal(val: 0.99) -> "связь строгая"
        coef.abs() > BigDecimal(val: 0.9) -> "связь весьма высокая"
        coef.abs() > BigDecimal(val: 0.7) -> "связь высокая"
        coef.abs() > BigDecimal(val: 0.5) -> "связь заметная"
        coef.abs() > BigDecimal(val: 0.3) -> "связь умеренная"
        else -> "связь слабая"
    }
    println("Коэффициент корреляции Пирсона: $coef - $decision")
}

```

## Квадратичная аппроксимация

```

println("Квадратичная аппроксимация")
println("Точки не изменяются: " + points.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(x^3), sum(x^4), sum(y), sum(x*y), sum(x^2*y)")

val coef = convertToMatrixAndSolve(points, n: 2)
val strFunction = "${coef[0]} ${toStr(coef[1])} * x ${toStr(coef[2])} * x^2"
val function = { x: BigDecimal -> coef[0] + coef[1] * x + coef[2] * x.pow(n: 2) }

```

## Кубическая аппроксимация

```

println("Кубическая аппроксимация")
println("Точки не изменяются: " + points.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(x^3), sum(x^4), sum(x^5), sum(x^6), sum(y), sum(x*y), sum(x^2*y), sum(x^3*y)")

val coef = convertToMatrixAndSolve(points, n: 3)
val strFunction = "${coef[0]} ${toStr(coef[1])} * x ${toStr(coef[2])} * x^2 ${toStr(coef[3])} * x^3"
val function = { x: BigDecimal -> coef[0] + coef[1] * x + coef[2] * x.pow(n: 2) + coef[3] * x.pow(n: 3) }

```

## Степенная аппроксимация

```

val modifyPoints = runCatching { this: PowerApproximation
    points.map { coords ->
        Pair(
            ln(coords.first.toDouble()).toBigDecimal(MathContext.DECIMAL64),
            ln(coords.second.toDouble()).toBigDecimal(MathContext.DECIMAL64)
        )
    }
}.getOrElse { run { this: PowerApproximation
    println("Невозможно использовать степенную аппроксимацию, так как есть точки с x и y, меньше нуля")
    return null
}

println("Степенная аппроксимация")
println("x и y линеаризируются: " + modifyPoints.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(y), sum(x*y)")

val coef = convertToMatrixAndSolve(modifyPoints, n: 1)
val b = coef[1]
val a = exp(coef[0].toDouble()).toBigDecimal(MathContext.DECIMAL64)
val strFunction = "$a * x^$b"
val function = { x: BigDecimal -> a * x.toDouble().pow(b.toDouble()).toBigDecimal(MathContext.DECIMAL64) }

```

## Экспоненциальная аппроксимация

```

val modifyPoints = runCatching { this: ExponentialApproximation
    points.map { coords ->
        Pair(coords.first, ln(coords.second.toDouble()).toBigDecimal(MathContext.DECIMAL64))
    }
}.getOrElse { run { this: ExponentialApproximation
    println("Невозможно использовать экспоненциальную аппроксимацию, так как есть точки с y меньше нуля")
    return null
}

println("Экспоненциальная аппроксимация")
println("x не изменяется, y линеаризируется: " + modifyPoints.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(y), sum(x*y)")

val coef = convertToMatrixAndSolve(modifyPoints, n: 1)
val b = coef[1]
val a = exp(coef[0].toDouble()).toBigDecimal(MathContext.DECIMAL64)
val strFunction = "$a * e^($b * x)"
val function = { x: BigDecimal -> a * exp((x * b).toDouble()).toBigDecimal(MathContext.DECIMAL64) }

```

## Логарифмическая аппроксимация

```

val modifyPoints = runCatching { this: LogarithmicApproximation
    points.map { coords ->
        Pair(ln(coords.first.toDouble()).toBigDecimal(MathContext.DECIMAL64), coords.second)
    }
}.getOrElse { run { this: LogarithmicApproximation
    println("Невозможно использовать логарифмическую аппроксимацию, так как есть точки с x и y, меньше нуля")
    return null
}

println("Логарифмическая аппроксимация")
println("x линеаризуется, y не изменяется: " + modifyPoints.joinToString(separator = ", ") { "(" + it.first + ";" + it.second + ")" })
println("n, sum(x), sum(x^2), sum(y), sum(x*y)")

val coef = convertToMatrixAndSolve(modifyPoints, n: 1)
val strFunction = "${coef[0]} ${toStr(coef[1])} * ln(x)"
val function = { x: BigDecimal -> coef[0] + coef[1] * ln(x.toDouble()).toBigDecimal(MathContext.DECIMAL64) }

```

## Вывод программы

Для ввода точек с клавиатуры введите 1, для ввода с файла - 2

2

Введите имя файла, в котором на первой строке указано кол-во точек, а в каждой следующей - координаты одной точки по x и по y через пробел

*input3*

Линейная аппроксимация

Точки не изменяются: (1.1;2.73), (2.3;5.12), (3.7;7.74), (4.5;8.91), (5.4;10.59), (6.8;12.75), (7.5;13.43), (8.8;13.61), (9.1;15.20), (10.0;17.28)

n, sum(x), sum(x^2), sum(y), sum(x\*y)

10 59.2 432.34 107.36 759.011

Получена матрица:

10,00000 59,20000 107,36000

59,20000 432,34000 759,01100

Перестановка строк 1 и 2. Получена матрица:

59,20000 432,34000 759,01100

10,00000 59,20000 107,36000

Из матрицы выражен элемент 1. Получена матрица:

59,20000 432,34000 759,01100

0,00000 -13,83041 -20,85132

Из матрицы выражен элемент 2. Получена матрица:

59,20000 432,34000 759,01100

0,00000 -13,83041 -20,85132

Решение матрицы: 1.810751868679443 1.507643265425769

Получена функция: 1.810751868679443 + 1.507643265425769 \* x

0.54635670826513018745561845896321 0.02506882562629972567405800371689 0.12317857159098934494207537781689 0.09913268673063672925544282781225 0.40701146012566292669044054833936 0.4

Сумма квадратов отклонений: 4.18885684449655576725780448587754

СКО: 0.6472137857382641

Коэффициент корреляции Пирсона: 0.9889323636278941 - прямо-пропорциональная зависимость, связь весьма высокая

Квадратичная аппроксимация

Точки не изменяются: (1.1;2.73), (2.3;5.12), (3.7;7.74), (4.5;8.91), (5.4;10.59), (6.8;12.75), (7.5;13.43), (8.8;13.61), (9.1;15.20), (10.0;17.28)

n, sum(x), sum(x^2), sum(x^3), sum(x^4), sum(y), sum(x\*y), sum(x^2\*y)

10 59.2 432.34 3484.090 29633.8822 107.36 759.011 6011.2485

Получена матрица:

10,00000 59,20000 432,34000 107,36000

59,20000 432,34000 3484,09000 759,01100

432,34000 3484,09000 29633,88220 6011,24850

Перестановка строк 1 и 3. Получена матрица:

432,34000 3484,09000 29633,88220 6011,24850

59,20000 432,34000 3484,09000 759,01100

10,00000 59,20000 432,34000 107,36000

Из матрицы выражен элемент 1. Получена матрица:

432,34000 3484,09000 29633,88220 6011,24850

0,00000 -44,73390 -573,65582 -64,10486

0,00000 -21,38681 -253,09004 -31,67984

Из матрицы выражен элемент 2. Получена матрица:

432,34000 3484,09000 29633,88220 6011,24850

0,00000 -44,73390 -573,65582 -64,10486

0,00000 0,00000 21,16880 -1,03198

Из матрицы выражен элемент 3. Получена матрица:

432,34000 3484,09000 29633,88220 6011,24850

0,00000 -44,73390 -573,65582 -64,10486

0,00000 0,00000 21,16880 -1,03198

Решение матрицы: 0.6592085525787528 2.058184057477724 -0.04875001556291067

Получена функция: 0.6592085525787528 + 2.058184057477724 \* x - 0.04875001556291067 \* x^2

0.01801594713969511059267550678036589449 0.00022934989668861202450582101911680249 0.01766191769130614064656272847093206729 0.00056877461400333247055848233011

Сумма квадратов отклонений: 3.00048807244358950021491636163888631958

СКО: 0.5477671104076613

#### Кубическая аппроксимация

Точки не изменяются: (1.1;2.73), (2.3;5.12), (3.7;7.74), (4.5;8.91), (5.4;10.59), (6.8;12.75), (7.5;13.43), (8.8;13.61), (9.1;15.20), (10.0;17.28)

n, sum(x), sum(x^2), sum(x^3), sum(x^4), sum(x^5), sum(x^6), sum(y), sum(x\*y), sum(x^2\*y), sum(x^3\*y)

10 59.2 432.34 3484.090 29633.8822 260642.55562 2344933.547854 107.36 759.011 6011.2485 50621.35277

Получена матрица:

10,00000 59,20000 432,34000 3484,09000 107,36000

59,20000 432,34000 3484,09000 29633,88220 759,01100

432,34000 3484,09000 29633,88220 260642,55562 6011,24850

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

Перестановка строк 1 и 4. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

59,20000 432,34000 3484,09000 29633,88220 759,01100

432,34000 3484,09000 29633,88220 260642,55562 6011,24850

10,00000 59,20000 432,34000 3484,09000 107,36000

Из матрицы выражен элемент 1. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

0,00000 -71,18483 -944,62433 -10210,11323 -101,12295

0,00000 -193,17225 -2709,19806 -30339,75829 -270,34459

0,00000 -25,85487 -315,75364 -3246,31463 -37,93290

Перестановка строк 2 и 3. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

0,00000 -193,17225 -2709,19806 -30339,75829 -270,34459

0,00000 -71,18483 -944,62433 -10210,11323 -101,12295

0,00000 -25,85487 -315,75364 -3246,31463 -37,93290

Из матрицы выражен элемент 2. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

0,00000 -193,17225 -2709,19806 -30339,75829 -270,34459

0,00000 0,00000 53,72717 970,22222 -1,49977

0,00000 0,00000 46,85520 814,46795 -1,74900

Из матрицы выражен элемент 3. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

0,00000 -193,17225 -2709,19806 -30339,75829 -270,34459

0,00000 0,00000 53,72717 970,22222 -1,49977

0,00000 0,00000 0,00000 -31,65796 -0,44106

Из матрицы выражен элемент 4. Получена матрица:

3484,09000 29633,88220 260642,55562 2344933,54785 50621,35277

0,00000 -193,17225 -2709,19806 -30339,75829 -270,34459

0,00000 0,00000 53,72717 970,22222 -1,49977

0,00000 0,00000 0,00000 -31,65796 -0,44106

Решение матрицы: -0.5713061874273741 3.131309759783080 -0.2795050676108028 0.01393211571054992

Получена функция: -0.5713061874273741 + 3.131309759783080 \* x - 0.2795050676108028 \* x^2 + 0.01393211571054992 \* x^3

0.0311603474510306564283164003657588584704 0.0406572798513318268639348735090879703296 0.0236602863805961695372635173979395563776 0.0480373109220185866878772421992356000000 0.043

Сумма квадратов отклонений: 2.4577544011181561422798449945028125600256

СКО: 0.4957574408032779

#### Степенная аппроксимация

x и y линеаризируются: (0.09531017980432493;1.004301609196868), (0.8329091229351039;1.633154439051416), (1.308332819650179;2.046401687601636), (1.504077396776274;2.187174241482)

n, sum(x), sum(x^2), sum(y), sum(x\*y)

10 16.04446533346144783 30.1630921797941457511920684087717149 22.555614426977050 39.76462233046035263518525903433664

Получена матрица:

10,00000 16,04447 22,55561

16,04447 30,16309 39,76462

Перестановка строк 1 и 2. Получена матрица:

16,04447 30,16309 39,76462

10,00000 16,04447 22,55561

Из матрицы выражен элемент 1. Получена матрица:

16,04447 30,16309 39,76462

0,00000 -2,75522 -2,22840

Из матрицы выражен элемент 2. Получена матрица:

16,04447 30,16309 39,76462

0,00000 -2,75522 -2,22840

Решение матрицы: 0.9578998544095475 0.8087907956532690

Получена функция: 2.606217286150233 \* x^0.8087907956532690

0.007236293046917743712027051209513509166099557603440401440836 0.000067450252662738489035911220581370962050134294745033053641 0.053482155888575356888539537947487599695193932844

Сумма квадратов отклонений: 3.152577558347475812673532901957654617748029061127735481304491

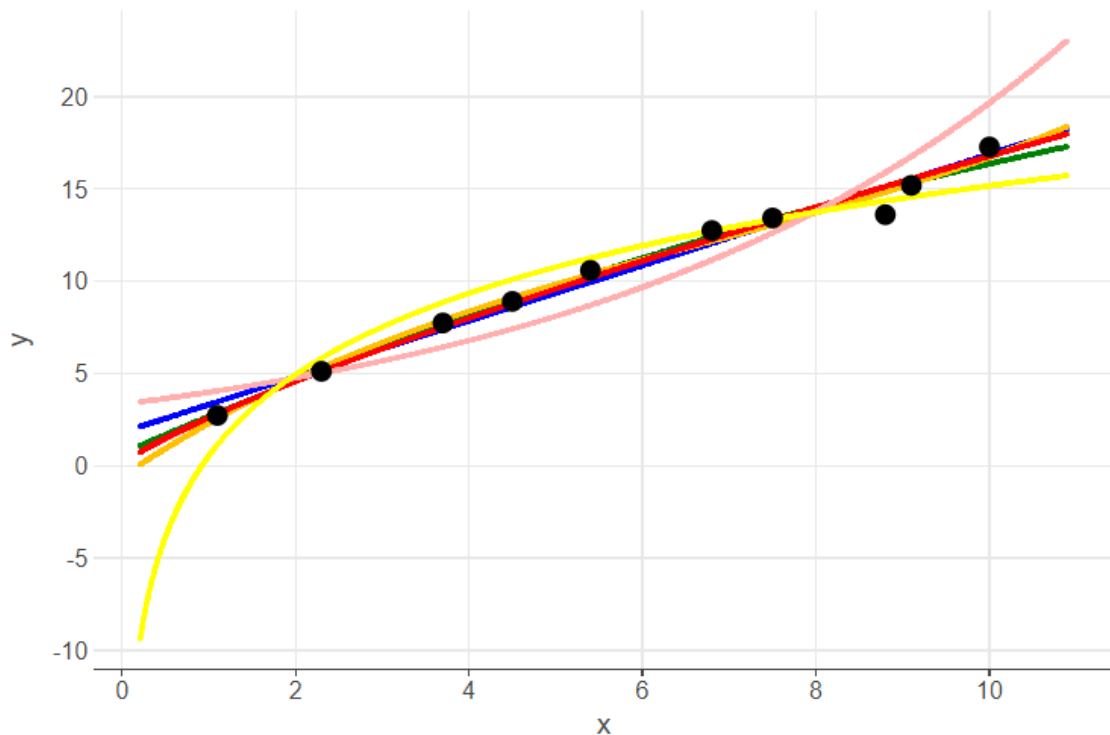
СКО: 0.561478188209255



```
Экспоненциальная аппроксимация
x не изменяется, y linearизируется: (1.1;1.004301609196868), (2.3;1.633154439051416), (3.7;2.046401687601636), (4.5;2.187174241482718), (5.4;2.359910159613315), (6.8;2.54553127:
n, sum(x), sum(x^2), sum(y), sum(x*y)
10 59.2 432.34 22.555614426977050 148.0436358124208519
Получена матрица:
10,00000 59,20000 22,55561
59,20000 432,34000 148,04364
Перестановка строк 1 и 2. Получена матрица:
59,20000 432,34000 148,04364
10,00000 59,20000 22,55561
Из матрицы выражен элемент 1. Получена матрица:
59,20000 432,34000 148,04364
0,00000 -13,83041 -2,45176
Из матрицы выражен элемент 2. Получена матрица:
59,20000 432,34000 148,04364
0,00000 -13,83041 -2,45176
Решение матрицы: 1.206105697962703 0.1772729298538855
Получена функция: 3.34045056624145 * e^(0.1772729298538855 * x)
1.7680679903715044065443893152982131699231656012108092792025 0.0095989059706070735790134060029734939571986623636889648900 1.6985984065190630549723638512161495453001134685748232:
Сумма квадратов отклонений: 25.8438826241734292958842603305297563640637810315905015361825
СКО: 1.607603266486276

Логарифмическая аппроксимация
x linearизируется, y не изменяется: (0.09531017980432493;2.73), (0.8329091229351039;5.12), (1.308332819650179;7.74), (1.504077396776274;8.91), (1.686398953570229;10.59), (1.916922:
n, sum(x), sum(x^2), sum(y), sum(x*y)
10 16.04446533346144783 30.1630921797941457511920684887717149 107.36 200.3652053410583145269
Получена матрица:
10,00000 16,04447 107,36000
16,04447 30,16309 200,36521
Перестановка строк 1 и 2. Получена матрица:
16,04447 30,16309 200,36521
10,00000 16,04447 107,36000
Из матрицы выражен элемент 1. Получена матрица:
16,04447 30,16309 200,36521
0,00000 -2,75522 -17,52120
Из матрицы выражен элемент 2. Получена матрица:
16,04447 30,16309 200,36521
0,00000 -2,75522 -17,52120
Решение матрицы: 0.5328904795897391 6.359270507525869
Получена функция: 0.5328904795897391 + 6.359270507525869 * ln(x)
2.5313010622749215542901524091877598812431205592658959875642447889 0.50351073104623898899158336130285073960017211532608849136907881 1.2386194031132241123720597050743787500271907:
Сумма квадратов отклонений: 11.5213370514886541049260001249200739362307887009021095404834005989
СКО: 1.073374913601424

Наилучшая аппроксимирующая функция: Кубическая аппроксимация: -0.5713061874273741 + 3.131309759783080 * x - 0.2795050676108028 * x^2 + 0.01393211571054992 * x^3, её СКО - 0.49575
Линейная аппроксимация: 1.810751868679443 + 1.507643265425769 * x, цвет - blue
Квадратичная аппроксимация: 0.6592085525787528 + 2.058184057477724 * x - 0.04875001556291067 * x^2, цвет - dark-green
Кубическая аппроксимация: -0.5713061874273741 + 3.131309759783080 * x - 0.2795050676108028 * x^2 + 0.01393211571054992 * x^3, цвет - orange
Степенная аппроксимация: 2.606217286150233 * x^0.80879079956532690, цвет - red
Экспоненциальная аппроксимация: 3.34045056624145 * e^(0.1772729298538855 * x), цвет - pink
Логарифмическая аппроксимация: 0.5328904795897391 + 6.359270507525869 * ln(x), цвет - yellow
```



## Выводы

В ходе выполнения лабораторной работы я познакомилась с применением метода наименьших квадратов для нахождения наилучших коэффициентов аппроксимирующей функции и повторила методы решения систем линейных алгебраических уравнений.