федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
Вариант 7

Студент: Кулагин Вячеслав Дмитриевич
Преподаватель: Наумова Надежда Александровна

Группа: P3209

Санкт-Петербург
2025

Содержание

1 Цели работы **2**

2 Вычислительная реализация задачи **2**
2.1 Линейное приближение . . . 2
2.2 Квадратичное приближение . . . 3

3 Программная реализация задачи **5**
3.1 Примеры работы . . . 5
3.2 Линейная . . . 6
3.3 Экспонента . . . 9
3.4 Логарифмическая . . . 9
3.5 Степенная . . . 10
3.6 Функция второй степени . . . 11
3.7 Функция третьей степени . . . 14

4 Вывод **17**

1 Цели работы

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

2 Вычислительная реализация задачи

Исходные данные:

$$y = \frac{23x}{x^4 + 7}; x \in [-2, 0]; h = 0,2$$

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
x_i	-2.0	-1.8	-1.6	-1.4	-1.2	-1.0	-0.8	-0.6	-0.4	-0.2	0.0
y_i	-2.00	-2.37	-2.72	-2.97	-3.04	-2.88	-2.48	-1.94	-1.31	-0.66	0.00

2.1 Линейное приближение

Представим функцию в виде: $\varphi(x) = ax + b$. Рассмотрим систему:

$$\begin{cases} aSXX + bSX = SXY \\ aSX + bn = SY \end{cases}$$

Найдем суммы:

$$SX = \sum_{i=1}^{n} x_i = -11$$

$$SXX = \sum_{i=1}^{n} x_i^2 = 15.4$$

$$SY = \sum_{i=1}^{n} y_i = -22.35$$

$$SXY = \sum_{i=1}^{n} x_i y_i = 27.09$$

Получаем:

$$\begin{cases} 15.4a - 11b = 27.09 \\ -11a + 11b = -22.35 \end{cases}$$

Решив систему, получаем коэффициенты: $a = 1.0773$, $b = -0.9545$. Таким образом, понимаем, что исходное уравнение $\varphi(x) = 1.0773x - 0.9545$.
Найдём среднеквадратичное отклонение ($\gamma = (\varphi(x_i) - y_i)^2$):

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
x_i	-2	-1.8	-1.6	-1.4	-1.2	-1	-0.8	-0.6	-0.4	-0.2	0
y_i	-2.00	-2.37	-2.72	-2.97	-3.04	-2.88	-2.48	-1.94	-1.31	-0.66	0.00
$\varphi(x_i)$	-3.1091	-2.8936	-2.6782	-2.4627	-2.2473	-2.0318	-1.8163	-1.6009	-1.3854	-1.1700	-0.9545
γ	1.2301	0.2784	0.0014	0.2574	0.6313	0.7110	0.4448	0.1120	0.0058	0.2631	0.9111

Получаем коэффициент:

$$\delta = \sqrt{\frac{\sum_{i=1}^{n}(\varphi(x_i) - y_i)^2}{n}} = 0.664$$

2.2 Квадратичное приближение

Представим функцию в виде: $\varphi(x) = a_0 + a_1 x + a_2 x^2$. Рассмотрим систему:

$$\begin{cases} a_0 n + a_1 \sum_{i=1}^n x_i + a_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i \\ a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 + a_2 \sum_{i=1}^n x_i^3 = \sum_{i=1}^n x_i y_i \\ a_0 \sum_{i=1}^n x_i^2 + a_1 \sum_{i=1}^n x_i^3 + a_2 \sum_{i=1}^n x_i^4 = \sum_{i=1}^n x_i^2 y_i \end{cases}$$

Найдем суммы:

$$\sum_{i=1}^{n} x_i = -11$$

$$\sum_{i=1}^{n} x_i^2 = 15.4$$

$$\sum_{i=1}^{n} x_i^3 = -24.2$$

$$\sum_{i=1}^{n} x_i^4 = 40.53$$

$$\sum_{i=1}^{n} y_i = -22.35$$

$$\sum_{i=1}^{n} x_i y_i = 27.09$$

$$\sum_{i=1}^{n} x_i^2 y_i = -38.22$$

Получаем систему:

$$\begin{cases} a_0 \cdot 11 + a_1 \cdot (-11) + a_2 \cdot 15.4 = -22.35 \\ a_0 \cdot (-11) + a_1 \cdot 15.4 + a_2 \cdot (-24.2) = 27.09 \\ a_0 \cdot 15.4 + a_1 \cdot (-24.2) + a_2 \cdot 40.53 = -38.22 \end{cases}$$

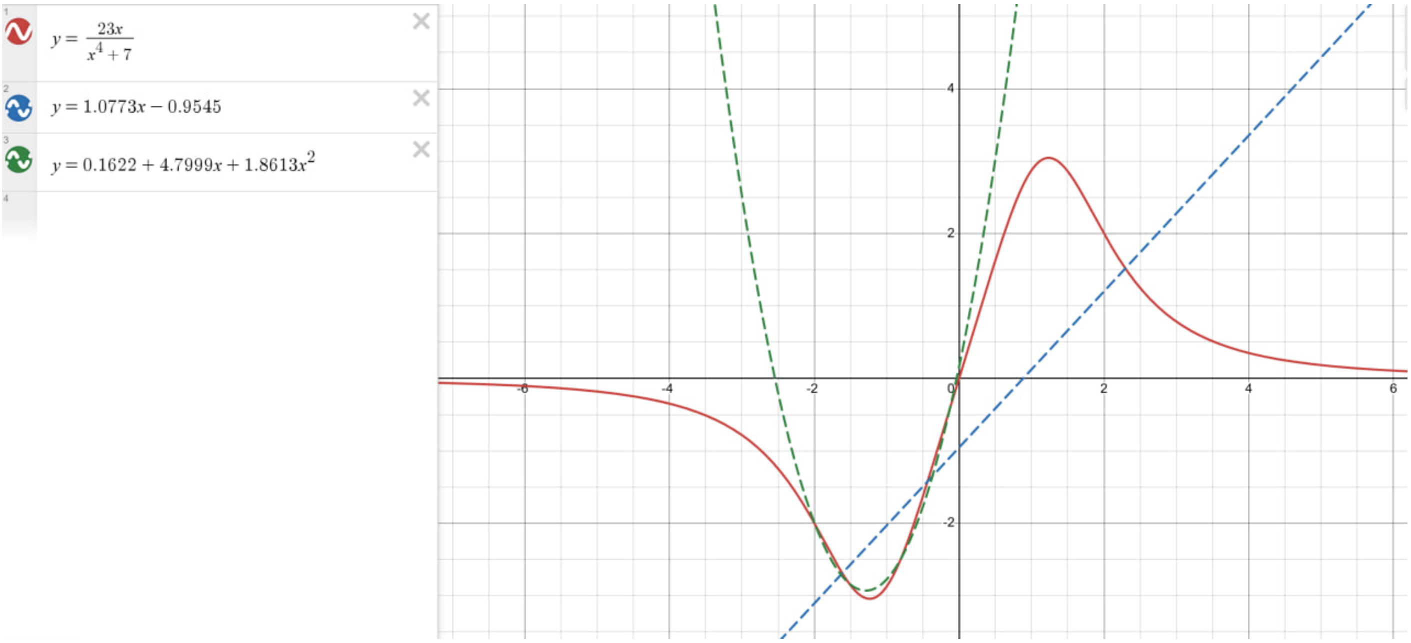
Решив систему, получаем коэффициенты: $a_0 = 0.1622$, $a_1 = 4.7999$, $a_2 = 1.8613$. Таким образом, получаем уравнение: $\varphi(x) = 0.1622 + 4.7999x + 1.8613x^2$.
Найдём среднеквадратичное отклонение ($\gamma = (\varphi(x_i) - y_i)^2$):

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
x_i	-2	-1.8	-1.6	-1.4	-1.2	-1	-0.8	-0.6	-0.4	-0.2	0
y_i	-2.00	-2.37	-2.72	-2.97	-3.04	-2.88	-2.48	-1.94	-1.31	-0.66	0.00
$\varphi(x_i)$	-1.9924	-2.4470	-2.7527	-2.9095	-2.9174	-2.7764	-2.4865	-2.0477	-1.4600	-0.7233	0.1622
γ	0.0001	0.0066	0.0014	0.0037	0.0155	0.0097	0.0000	0.0126	0.0226	0.0044	0.0263

Получаем коэффициент:

$$\delta = \sqrt{\frac{\sum_{i=1}^{n} (\varphi(x_i) - y_i)^2}{n}} = 0.097$$

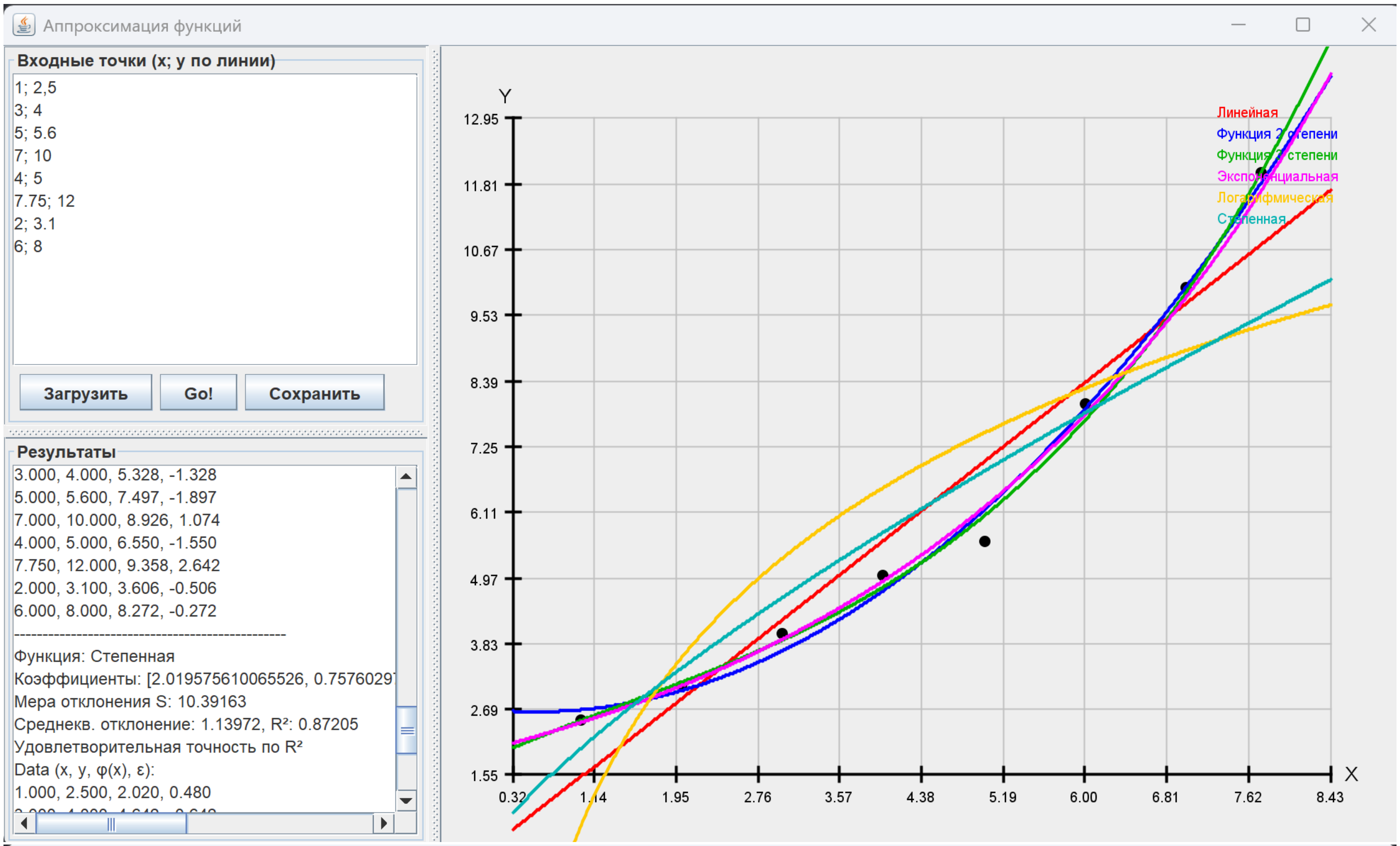
Сравнивая полученные значения, понимаем, что $0.097 < 0.664$. Таким образом, квадратичное приближение больше подходит к данному уравнению. Это также видно и графически:

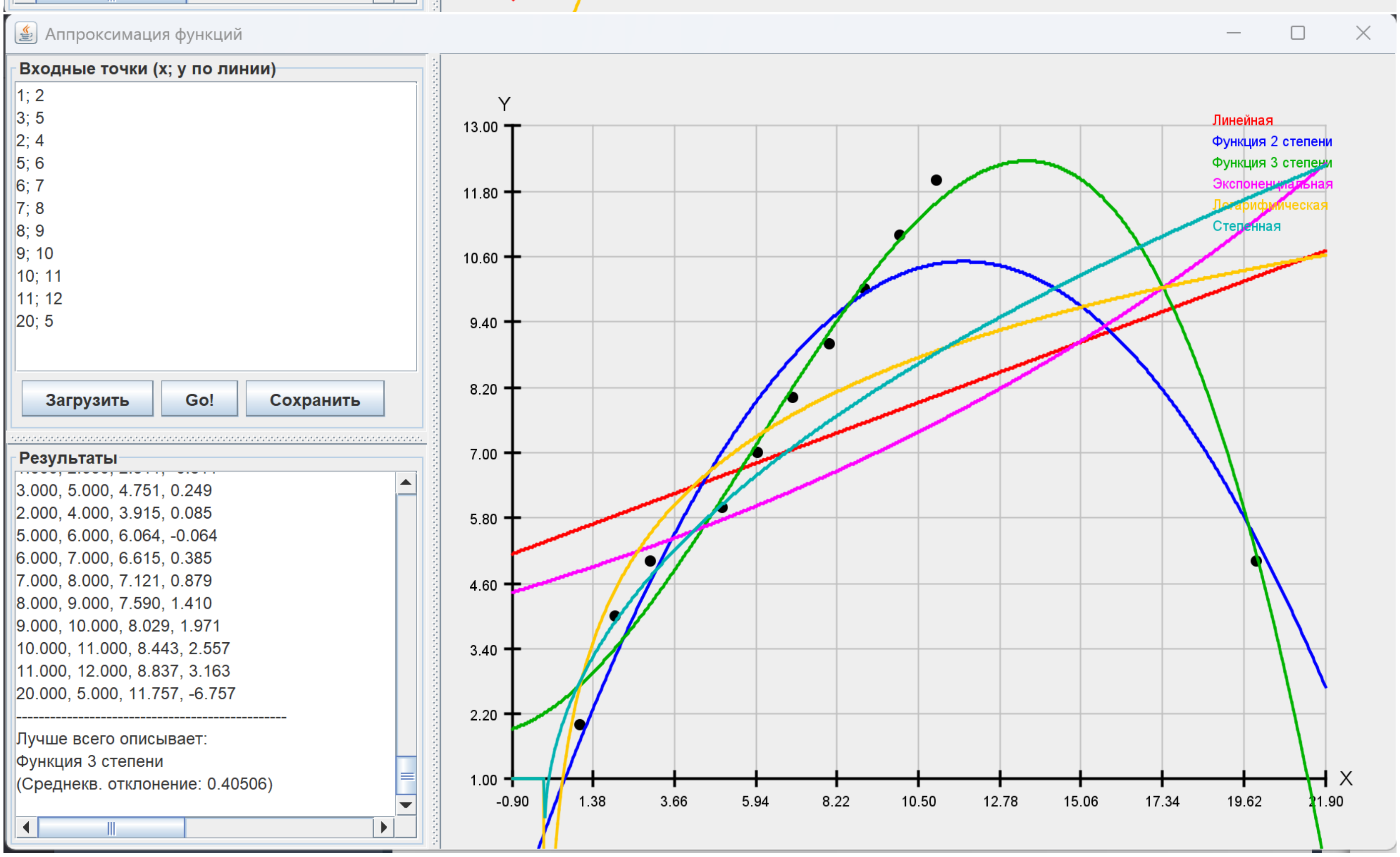


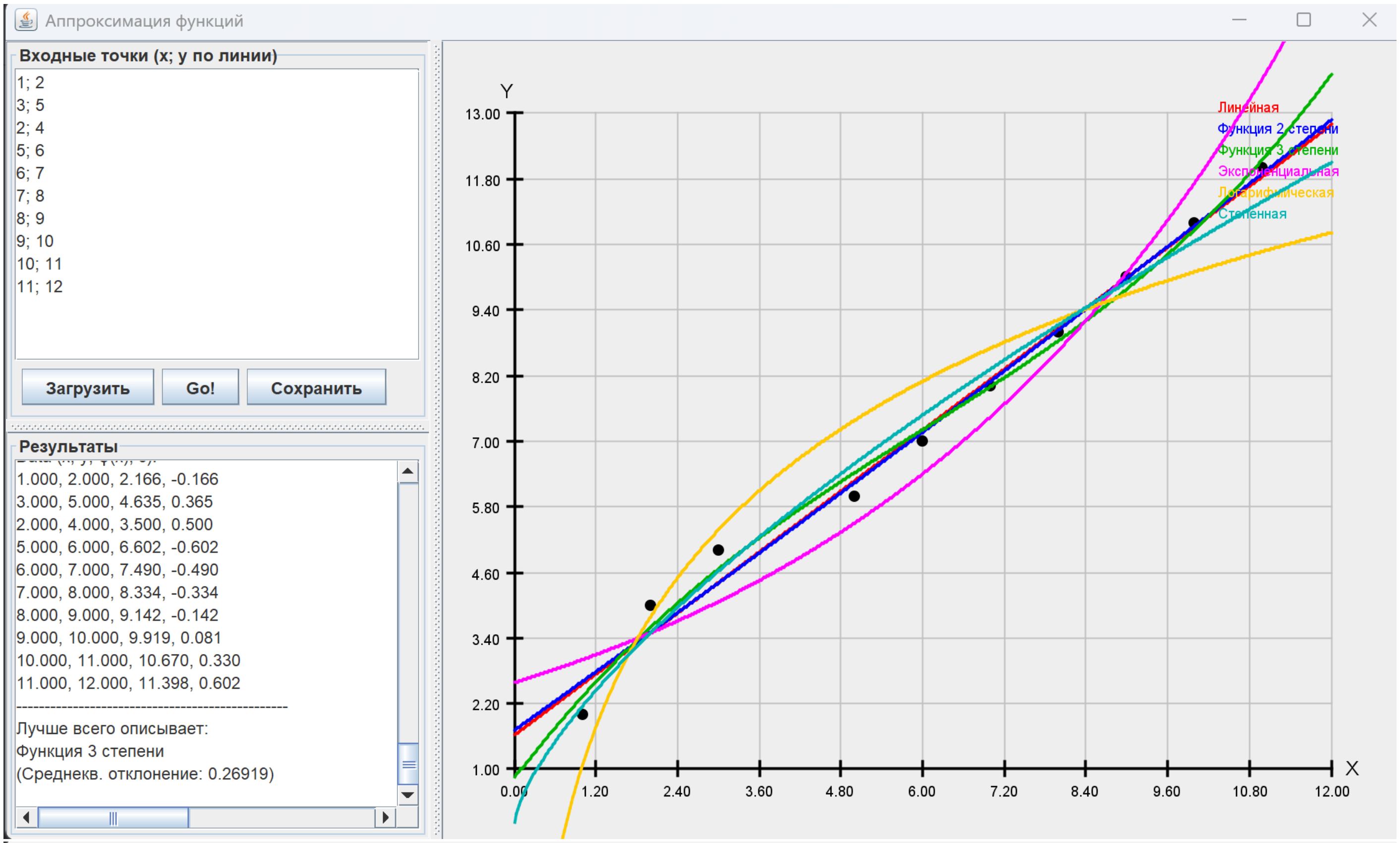
$y = \frac{23x}{x^4+7}$
$y = 1.0773x - 0.9545$
$y = 0.1622 + 4.7999x + 1.8613x^2$

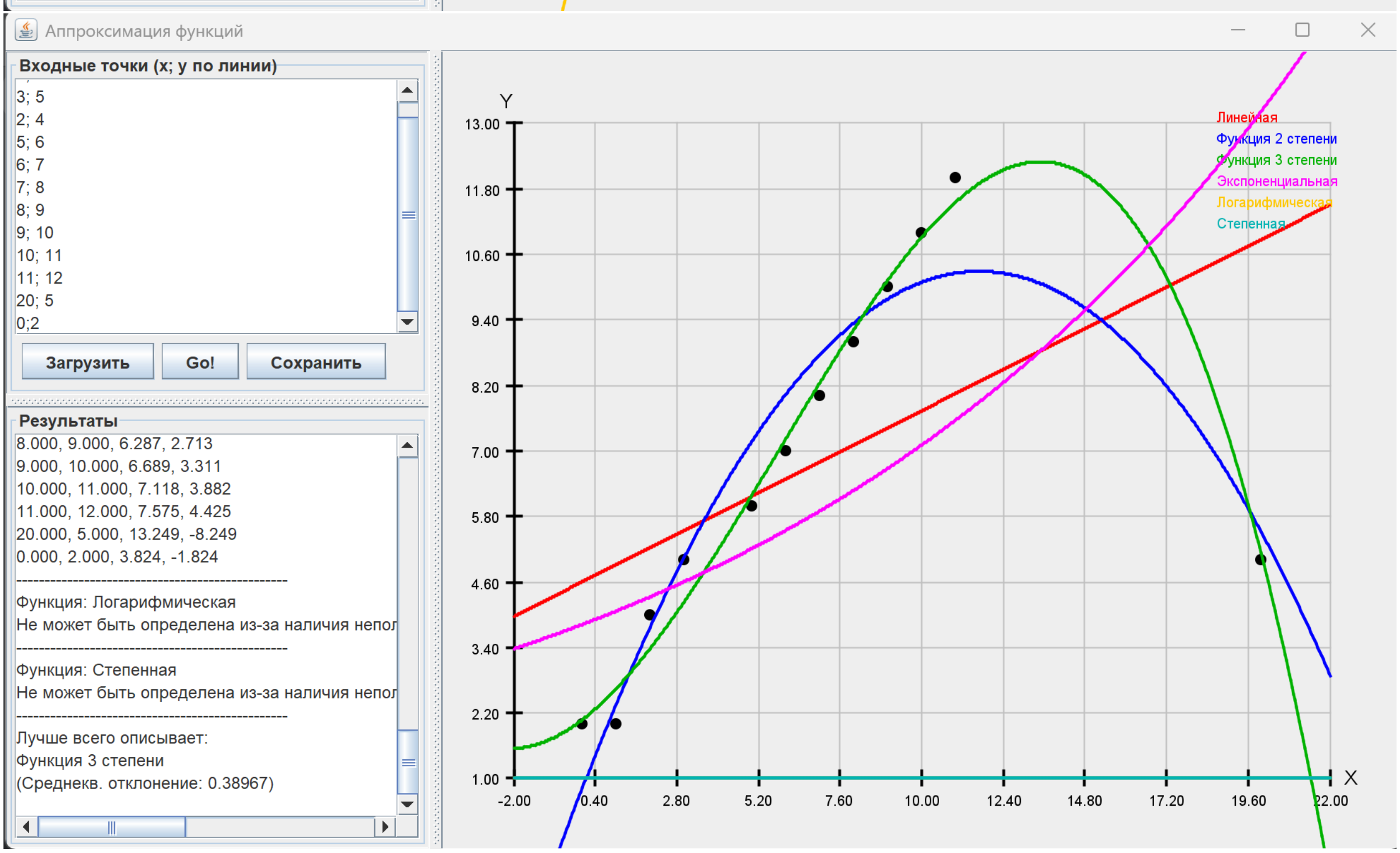
3 Программная реализация задачи

3.1 Примеры работы







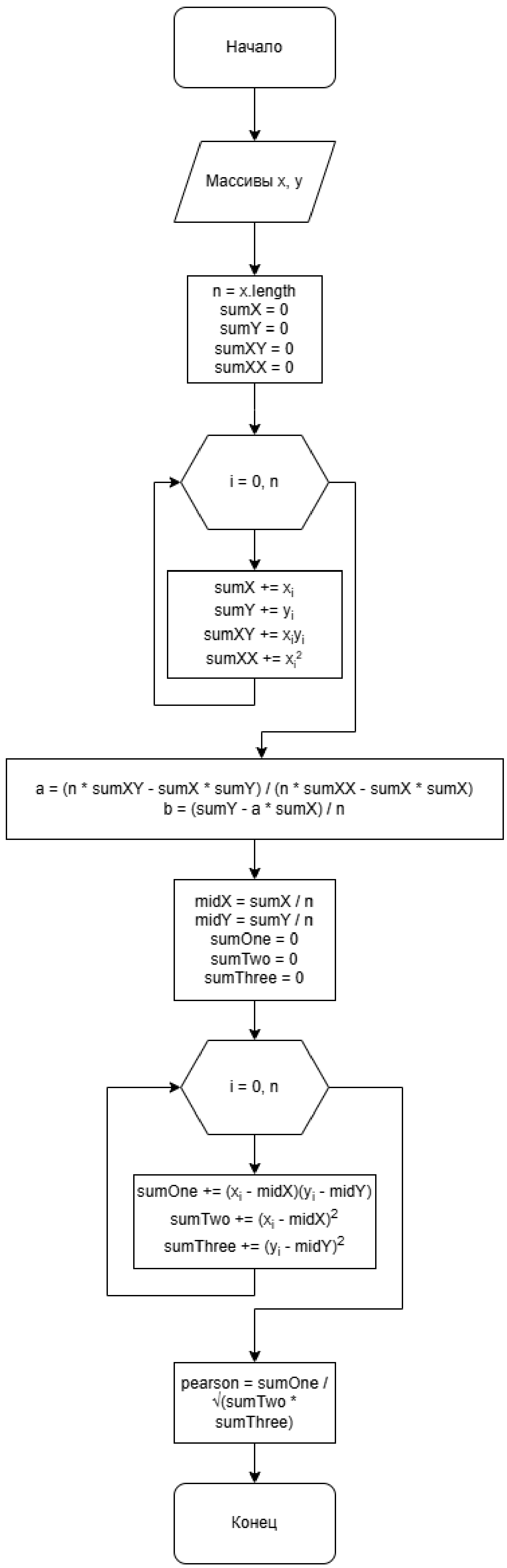


3.2 Линейная

```
public void fit(double[] x, double[] y) {
    int n = x.length;
```

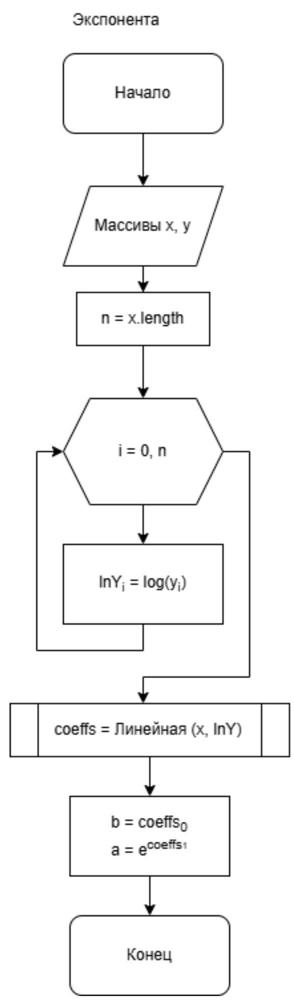
```java
    double sumX = 0, sumY = 0, sumXY = 0, sumXX = 0;
    for (int i = 0; i < n; i++) {
        sumX += x[i];
        sumY += y[i];
        sumXY += x[i] * y[i];
        sumXX += x[i] * x[i];
    }
    a = (n * sumXY - sumX * sumY) / (n * sumXX - sumX * sumX);
    b = (sumY - a * sumX) / n;

    double midX = sumX / n;
    double midY = sumY / n;

    double sumOne = 0, sumTwo = 0, sumThree = 0;

    for (int i = 0; i < n; i++) {
        sumOne += (x[i] - midX) * (y[i] - midY);
        sumTwo += (x[i] - midX) * (x[i] - midX);
        sumThree += (y[i] - midY) * (y[i] - midY);
    }
    pearson = sumOne / Math.sqrt(sumTwo * sumThree);
}
```

Линейная


Начало
Массивы x, y
n = x.length
sumX = 0
sumY = 0
sumXY = 0
sumXX = 0
i = 0, n
sumX += x_i
sumY += y_i
sumXY += x_iy_i
sumXX += x_i^2
a = (n * sumXY - sumX * sumY) / (n * sumXX - sumX * sumX)
b = (sumY - a * sumX) / n
midX = sumX / n
midY = sumY / n
sumOne = 0
sumTwo = 0
sumThree = 0
i = 0, n
sumOne += (x_i - midX)(y_i - midY)
sumTwo += (x_i - midX)^2
sumThree += (y_i - midY)^2
pearson = sumOne / √(sumTwo * sumThree)
Конец

3.3 Экспонента

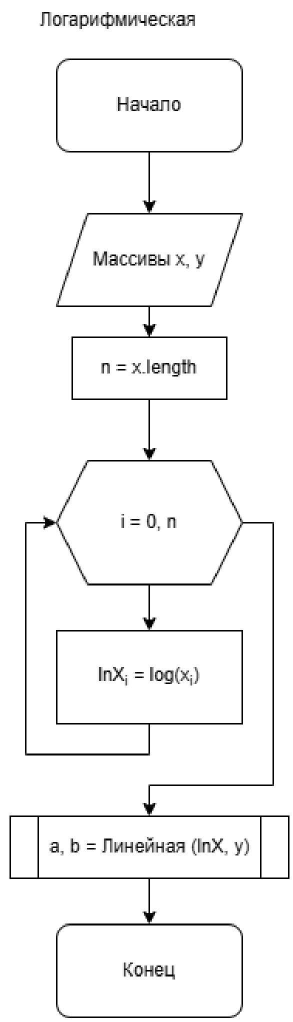
```
public void fit(double[] x, double[] y) {
    int n = x.length;
    double[] lnY = new double[n];
    for (int i = 0; i < n; i++) {
        lnY[i] = Math.log(y[i]);
    }

    LinearFunction linear = new LinearFunction();
    linear.fit(x, lnY);
    double[] coeffs = linear.getCoefficients();
    b = coeffs[0];
    a = Math.exp(coeffs[1]);
}
```



3.4 Логарифмическая

```
public void fit(double[] x, double[] y) {
    int n = x.length;
    double[] lnX = new double[n];
```

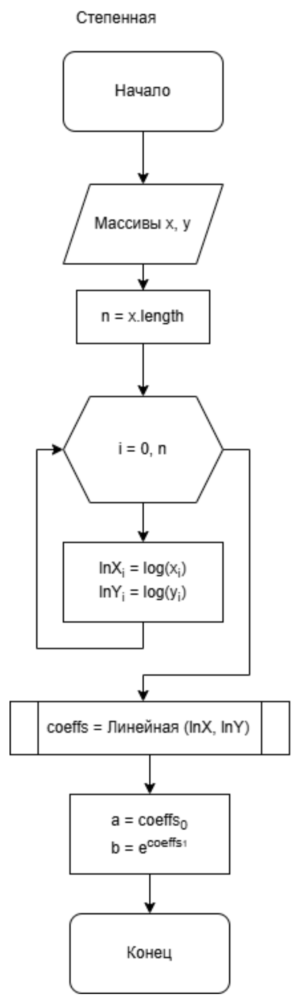
```
        for (int i = 0; i < n; i++) {
            lnX[i] = Math.log(x[i]);
        }

        LinearFunction linear = new LinearFunction();
        linear.fit(lnX, y);
        double[] coeffs = linear.getCoefficients();

        b = coeffs[0];
        a = coeffs[1];
    }
```



3.5 Степенная

```
    public void fit(double[] x, double[] y) {
        int n = x.length;
        double[] lnX = new double[n];
        double[] lnY = new double[n];
        for (int i = 0; i < n; i++) {
            lnX[i] = Math.log(x[i]);
```

```
            lnY[i] = Math.log(y[i]);
        }

        LinearFunction linear = new LinearFunction();
        linear.fit(lnX, lnY);
        double[] coeffs = linear.getCoefficients();
        b = coeffs[0];
        a = Math.exp(coeffs[1]);
    }
```

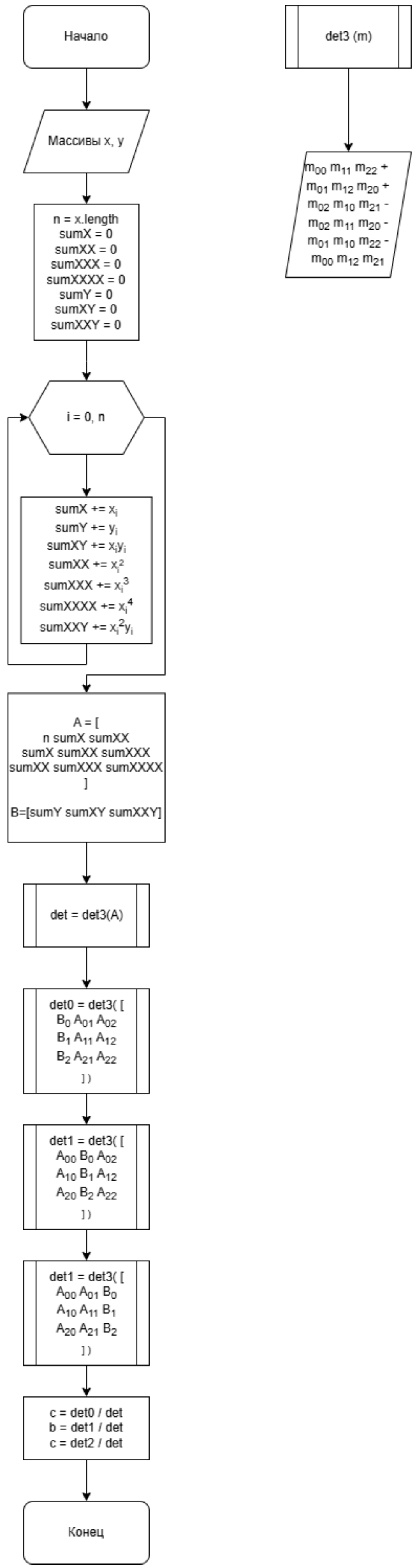


3.6 Функция второй степени

```
    public void fit(double[] x, double[] y) {
        int n = x.length;
        double sumX = 0, sumXX = 0, sumXXX = 0, sumXXXX = 0, sumY = 0, sumXY = 0, su
        for (int i = 0; i < n; i++) {
            sumX += x[i];
            sumXX += x[i] * x[i];
            sumXXX += x[i] * x[i] * x[i];
            sumXXXX += x[i] * x[i] * x[i] * x[i];
            sumY += y[i];
```

```
            sumXY += x[i] * y[i];
            sumXXY += x[i] * x[i] * y[i];
        }

        double[][] A = {
                {n, sumX, sumXX},
                {sumX, sumXX, sumXXX},
                {sumXX, sumXXX, sumXXXX}
        };
        double[] B = {sumY, sumXY, sumXXY};

        double det = det3(A);
        double det0 = det3(new double[][]{{B[0], A[0][1], A[0][2]},
                {B[1], A[1][1], A[1][2]},
                {B[2], A[2][1], A[2][2]}});
        double det1 = det3(new double[][]{{A[0][0], B[0], A[0][2]},
                {A[1][0], B[1], A[1][2]},
                {A[2][0], B[2], A[2][2]}});
        double det2 = det3(new double[][]{{A[0][0], A[0][1], B[0]},
                {A[1][0], A[1][1], B[1]},
                {A[2][0], A[2][1], B[2]}});

        c = det0 / det;
        b = det1 / det;
        a = det2 / det;
    }

    private static double det3(double[][] m) {
        return m[0][0] * m[1][1] * m[2][2]
                + m[0][1] * m[1][2] * m[2][0]
                + m[0][2] * m[1][0] * m[2][1]
                - m[0][2] * m[1][1] * m[2][0]
                - m[0][1] * m[1][0] * m[2][2]
                - m[0][0] * m[1][2] * m[2][1];
    }
```

2 порядка



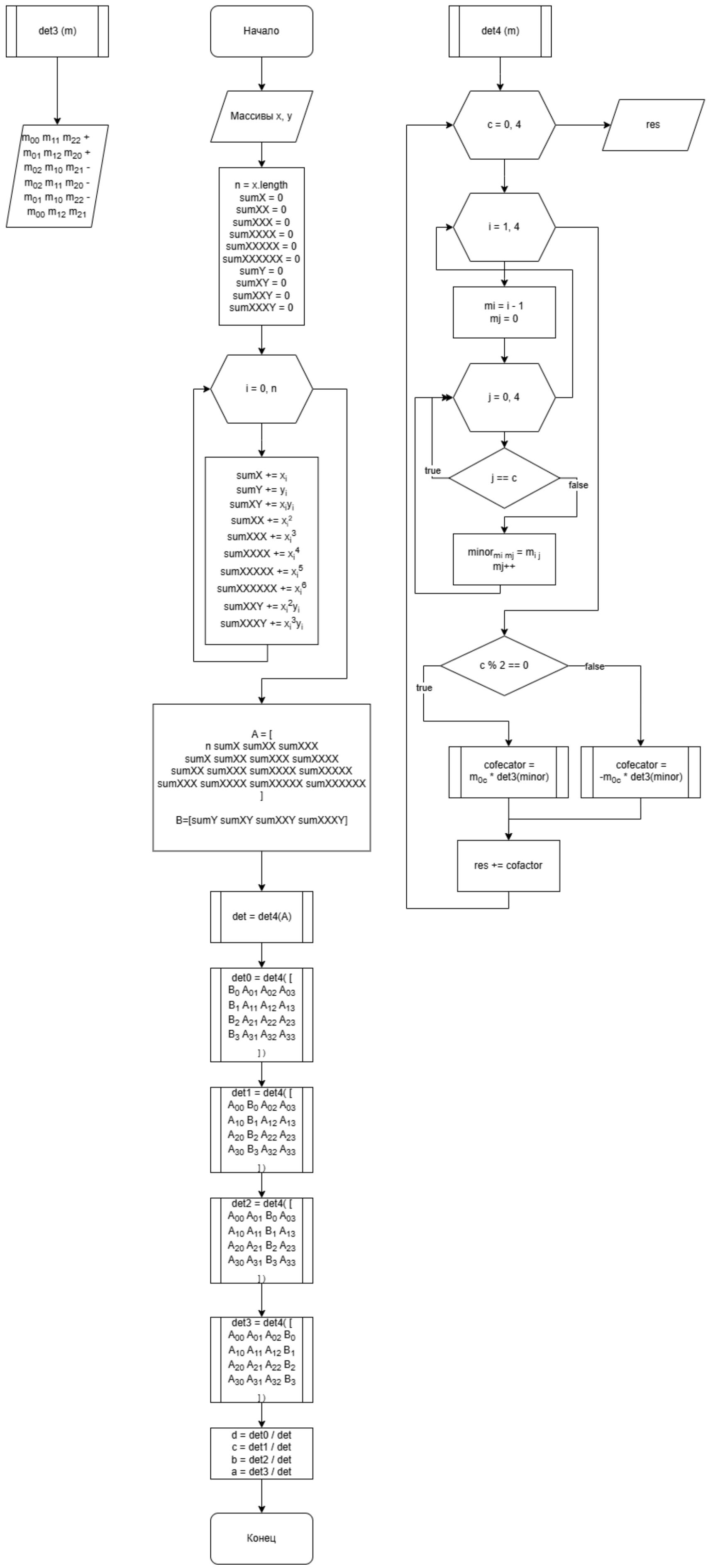
Начало
Массивы x, y
n = x.length
sumX = 0
sumXX = 0
sumXXX = 0
sumXXXX = 0
sumY = 0
sumXY = 0
sumXXY = 0
i = 0, n
sumX += x_i
sumY += y_i
sumXY += $x_i y_i$
sumXX += x_i^2
sumXXX += x_i^3
sumXXXX += x_i^4
sumXXY += $x_i^2 y_i$
A = [
n sumX sumXX
sumX sumXX sumXXX
sumXX sumXXX sumXXXX
]
B=[sumY sumXY sumXXY]
det = det3(A)
det0 = det3([
B_0 A_{01} A_{02}
B_1 A_{11} A_{12}
B_2 A_{21} A_{22}
])
det1 = det3([
A_{00} B_0 A_{02}
A_{10} B_1 A_{12}
A_{20} B_2 A_{22}
])
det1 = det3([
A_{00} A_{01} B_0
A_{10} A_{11} B_1
A_{20} A_{21} B_2
])
c = det0 / det
b = det1 / det
c = det2 / det
Конец
det3 (m)
m_{00} m_{11} m_{22} +
m_{01} m_{12} m_{20} +
m_{02} m_{10} m_{21} -
m_{02} m_{11} m_{20} -
m_{01} m_{10} m_{22} -
m_{00} m_{12} m_{21}

3.7 Функция третьей степени

```
public void fit(double[] x, double[] y) {
    int n = x.length;
    double sumX = 0, sumXX = 0, sumXXX = 0, sumXXXX = 0;
    double sumXXXXX = 0, sumXXXXXX = 0;
    double sumY = 0, sumXY = 0, sumXXY = 0, sumXXXY = 0;

    for (int i = 0; i < n; i++) {
        double xi = x[i];
        double yi = y[i];
        double xi2 = xi * xi;
        double xi3 = xi2 * xi;
        double xi4 = xi3 * xi;
        double xi5 = xi4 * xi;
        double xi6 = xi5 * xi;

        sumX += xi;
        sumXX += xi2;
        sumXXX += xi3;
        sumXXXX += xi4;
        sumXXXXX += xi5;
        sumXXXXXX += xi6;

        sumY += yi;
        sumXY += xi * yi;
        sumXXY += xi2 * yi;
        sumXXXY += xi3 * yi;
    }

    double[][] A = {
            {n, sumX, sumXX, sumXXX},
            {sumX, sumXX, sumXXX, sumXXXX},
            {sumXX, sumXXX, sumXXXX, sumXXXXX},
            {sumXXX, sumXXXX, sumXXXXX, sumXXXXXX}
    };
    double[] B = {sumY, sumXY, sumXXY, sumXXXY};

    double det = det4(A);
    double det0 = det4(new double[][]{
            {B[0], A[0][1], A[0][2], A[0][3]},
            {B[1], A[1][1], A[1][2], A[1][3]},
            {B[2], A[2][1], A[2][2], A[2][3]},
            {B[3], A[3][1], A[3][2], A[3][3]}
    });
    double det1 = det4(new double[][]{
            {A[0][0], B[0], A[0][2], A[0][3]},
            {A[1][0], B[1], A[1][2], A[1][3]},
            {A[2][0], B[2], A[2][2], A[2][3]},
```

```java
            {A[3][0], B[3], A[3][2], A[3][3]}
    });
    double det2 = det4(new double[][]{
            {A[0][0], A[0][1], B[0], A[0][3]},
            {A[1][0], A[1][1], B[1], A[1][3]},
            {A[2][0], A[2][1], B[2], A[2][3]},
            {A[3][0], A[3][1], B[3], A[3][3]}
    });
    double det3 = det4(new double[][]{
            {A[0][0], A[0][1], A[0][2], B[0]},
            {A[1][0], A[1][1], A[1][2], B[1]},
            {A[2][0], A[2][1], A[2][2], B[2]},
            {A[3][0], A[3][1], A[3][2], B[3]}
    });

    d = det0 / det;
    c = det1 / det;
    b = det2 / det;
    a = det3 / det;
}

private static double det3(double[][] m) {
    return m[0][0] * m[1][1] * m[2][2]
            + m[0][1] * m[1][2] * m[2][0]
            + m[0][2] * m[1][0] * m[2][1]
            - m[0][2] * m[1][1] * m[2][0]
            - m[0][1] * m[1][0] * m[2][2]
            - m[0][0] * m[1][2] * m[2][1];
}

private static double det4(double[][] m) {
    double res = 0;
    for (int c = 0; c < 4; c++) {
        double[][] minor = new double[3][3];
        for (int i = 1; i < 4; i++) {
            int mi = i - 1;
            int mj = 0;
            for (int j = 0; j < 4; j++) {
                if (j == c) continue;
                minor[mi][mj++] = m[i][j];
            }
        }
        double cofactor = ((c % 2) == 0 ? 1 : -1) * m[0][c] * det3(minor);
        res += cofactor;
    }
    return res;
}
```

3 порядка


det3 (m)
$m_{00} m_{11} m_{22}$ + $m_{01} m_{12} m_{20}$ + $m_{02} m_{10} m_{21}$ - $m_{02} m_{11} m_{20}$ - $m_{01} m_{10} m_{22}$ - $m_{00} m_{12} m_{21}$
Начало
Массивы x, y
n = x.length
sumX = 0
sumXX = 0
sumXXX = 0
sumXXXX = 0
sumXXXXX = 0
sumXXXXXX = 0
sumY = 0
sumXY = 0
sumXXY = 0
sumXXXY = 0
i = 0, n
sumX += x_i
sumY += y_i
sumXY += $x_i y_i$
sumXX += x_i^2
sumXXX += x_i^3
sumXXXX += x_i^4
sumXXXXX += x_i^5
sumXXXXXX += x_i^6
sumXXY += $x_i^2 y_i$
sumXXXY += $x_i^3 y_i$
A = [
n sumX sumXX sumXXX
sumX sumXX sumXXX sumXXXX
sumXX sumXXX sumXXXX sumXXXXX
sumXXX sumXXXX sumXXXXX sumXXXXXX
]
B=[sumY sumXY sumXXY sumXXXY]
det = det4(A)
det0 = det4([B_0 A_{01} A_{02} A_{03} B_1 A_{11} A_{12} A_{13} B_2 A_{21} A_{22} A_{23} B_3 A_{31} A_{32} A_{33}])
det1 = det4([A_{00} B_0 A_{02} A_{03} A_{10} B_1 A_{12} A_{13} A_{20} B_2 A_{22} A_{23} A_{30} B_3 A_{32} A_{33}])
det2 = det4([A_{00} A_{01} B_0 A_{03} A_{10} A_{11} B_1 A_{13} A_{20} A_{21} B_2 A_{23} A_{30} A_{31} B_3 A_{33}])
det3 = det4([A_{00} A_{01} A_{02} B_0 A_{10} A_{11} A_{12} B_1 A_{20} A_{21} A_{22} B_2 A_{30} A_{31} A_{32} B_3])
d = det0 / det
c = det1 / det
b = det2 / det
a = det3 / det
Конец
det4 (m)
c = 0, 4
res
i = 1, 4
mi = i - 1
mj = 0
j = 0, 4
true
j == c
false
$minor_{mi\ mj} = m_{i\ j}$
mj++
c % 2 == 0
true
false
cofecator = m_{0c} * det3(minor)
cofecator = $-m_{0c}$ * det3(minor)
res += cofactor

4 Вывод

Проведя эту работу, я научился аппроксимировать функции, а также узнал и реализовал условия, по которым можно однозначно утверждать, насколько та или иная аппроксимирующая функция точна в каждом конкретном случае.