Университет ИТМО

ФПИиКТ

# Лабораторная работа №4
# по Вычислительной математике

Выполнил: Балтабаев Дамир
Группа: P3210
Вариант: 3

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург
2022

# Цель лабораторной работы:

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

# Порядок выполнение работы:

2. **Методика проведения исследования:**
   a) Вычислить меру отклонения: $S = \sum_{i=1}^{n}[\varphi(x_i) - y_i]^2$ для всех исследуемых функций.
   b) Уточнить значения коэффициентов эмпирических функций, минимизируя функцию S.
   c) Сформировать массивы предполагаемых эмпирических зависимостей ($\varphi(x_i), \varepsilon_i$).
   d) Определить среднеквадратичное отклонение для каждой аппроксимирующей функции. Выбрать наименьшее значение и, следовательно, наилучшее приближение.
   e) Построить графики полученных эмпирических функций.

3. **Вычислительная реализация задачи:**
   a) Для заданной функции (см. таблицу 1) построить наилучшие линейное и квадратичное приближения по 11 точкам указанного интервала.
   b) Найти среднеквадратические отклонения. Ответы дать с тремя знаками после запятой.
   c) Построить графики линейного и квадратичного приближений и заданной функции.
   d) ***Привести в отчете подробные вычисления***.

4. **Программная реализация задачи:**
   a) Предусмотреть ввод исходных данных из файла/консоли (таблица *y=f(x)* должна содержать 10 - 12 точек).
   b) Реализовать метод наименьших квадратов, исследуя все функции п.1.
   c) Предусмотреть вывод результатов в файл/консоль.
   d) Для линейной зависимости вычислить коэффициент корреляции Пирсона.
   e) Программа должна отображать наилучшую аппроксимирующую функцию.
   f) Организовать вывод графиков функций, графики должны полностью отображать весь исследуемый интервал (с запасом).

# Рабочие формулы используемых методов:

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

$SX = \sum_{i=1}^{n} x_i$, $SXX = \sum_{i=1}^{n} x_i^2$, $SY = \sum_{i=1}^{n} y_i$, $SXY = \sum_{i=1}^{n} x_i y_i$

Получим систему уравнений для нахождения параметров *a* и *b*:

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

$$\Delta = SXX \cdot n - SX \cdot SX$$

$$\Delta_1 = SXY \cdot n - SX \cdot SY$$

$$\Delta_2 = SXX \cdot SY - SX \cdot SXY$$

$$a = \frac{\Delta_1}{\Delta}, \quad b = \frac{\Delta_2}{\Delta}$$

Коэффициент корреляции Пирсона позволяет определить наличие или отсутствие **линейной** связи между двумя переменными:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$S = S(a_0, a_1, a_2) = \sum_{i=1}^{n} (a_0 + a_1 x_i + a_2 x_i^2 - y_i)^2 \rightarrow min$$

Определим меру отклонения: $S = \sum_{i=1}^{n} \varepsilon_i^2 = 5{,}191$

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

**Аппроксимирующая функция задана степенной функцией вида:**

$$\varphi(x) = ax^b$$

**Аппроксимирующая функция задана экспоненциальной функцией вида:**

$$\varphi(x) = ae^{bx}$$

**Аппроксимирующая функция задана логарифмической функцией вида:**

$$\varphi(x) = a\ln(x) + b$$

# Листинг программы

## Линейная функция:

```
@Override
  public void approximate() {
      double SX = 0;
      double SXX = 0;
      double SY = 0;
      double SXY = 0;
      double xAverage = 0;
      double yAverage = 0;

      for (int i = 0; i < getN(); i++) {
          SX += getX()[i];
          SXX += getX()[i] * getX()[i];
          SY += getY()[i];
          SXY += getX()[i] * getY()[i];
      }
      xAverage = SX / getN();
      yAverage = SY / getN();

      a = (SXY * getN() - SX * SY) / (SXX * getN() - SX * SX);
      b = (SXX * SY - SX * SXY) / (SXX * getN() - SX * SX);

      double numerator = 0;
      double firstPartDenominator = 0;
      double secondPartDenominator = 0;
      for (int i = 0; i < getN(); i++) {
          S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
          numerator += (getX()[i] - xAverage) * (getY()[i] - yAverage);
```

```
            firstPartDenominator += Math.pow((getX()[i] - xAverage), 2);
            secondPartDenominator += Math.pow((getY()[i] - yAverage), 2);
        }
        r = numerator / Math.sqrt(firstPartDenominator *
secondPartDenominator);
        sigma = Math.sqrt(S / getN());

    }

    @Override
    public double linearXArgument(double x) {
        return a * x + b;
    }
```

## Полиномиальная функция 2-й степени:

```
   @Override
    public void approximate() {
        double X4S = 0;
        double X3S = 0;
        double X2S = 0;
        double X2YS = 0;
        double XS = 0;
        double XYS = 0;
        double YS = 0;

        for (int i = 0; i < getN(); i++) {
            X4S += Math.pow(getX()[i], 4);
            X3S += Math.pow(getX()[i], 3);
            X2S += Math.pow(getX()[i], 2);
            X2YS += Math.pow(getX()[i], 2) * getY()[i];
            XS += getX()[i];
            XYS += getX()[i] * getY()[i];
            YS += getY()[i];
        }
        double[][] firstMatrix = {{getN(), XS, X2S},
                {XS, X2S, X3S},
                {X2S, X3S, X4S}};

        double[][] secondsMatrix = {{YS, XS, X2S},
                {XYS, X2S, X3S},
                {X2YS, X3S, X4S}};

        double[][] thirdMatrix = {{getN(), YS, X2S},
                {XS, XYS, X3S},
                {X2S, X2YS, X4S}};

        double[][] fourthMatrix = {{getN(), XS, YS},
                {XS, X2S, XYS},
                {X2S, X3S, X2YS}};
```

```
        double delta = det(firstMatrix);
        double delta1 = det(secondsMatrix);
        double delta2 = det(thirdMatrix);
        double delta3 = det(fourthMatrix);

        a = delta3 / delta;
        b = delta2 / delta;
        c = delta1 / delta;

        for (int i = 0; i < getN(); i++) {
            S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
        }

        sigma = Math.sqrt(S / getN());

    }

    @Override
    public double linearXArgument(double x) {
        return a * Math.pow(x, 2) + b * x + c;
    }
```

## Полиномиальная функция 3-й степени:

```
     @Override
    public void approximate() {
        double XS = 0;
        double X2S = 0;
        double X3S = 0;
        double X4S = 0;
        double X5S = 0;
        double X6S = 0;

        double YS = 0;
        double YXS = 0;
        double YX2S = 0;
        double YX3S = 0;
        for (int i = 0; i < getN(); i++) {
            XS += getX()[i];
            X2S += Math.pow(getX()[i], 2);
            X3S += Math.pow(getX()[i], 3);
            X4S += Math.pow(getX()[i], 4);
            X5S += Math.pow(getX()[i], 5);
            X6S += Math.pow(getX()[i], 6);

            YS += getY()[i];
            YXS += getY()[i] * getX()[i];
            YX2S += getY()[i] * Math.pow(getX()[i], 2);
            YX3S += getY()[i] * Math.pow(getX()[i], 3);
        }

        double[][] firstMatrix = {{getN(), XS, X2S, X3S},
                {XS, X2S, X3S, X4S},
                {X2S, X3S, X4S, X5S},
                {X3S, X4S, X5S, X6S}};

        double[][] secondsMatrix = {{YS, XS, X2S, X3S},
                {YXS, X2S, X3S, X4S},
                {YX2S, X3S, X4S, X5S},
```

```
                {YX3S, X4S, X5S, X6S}};

        double[][] thirdMatrix = {{getN(), YS, X2S, X3S},
                {XS, YXS, X3S, X4S},
                {X2S, YX2S, X4S, X5S},
                {X3S, YX3S, X5S, X6S}};

        double[][] fourthMatrix = {{getN(), XS, YS, X3S},
                {XS, X2S, YXS, X4S},
                {X2S, X3S, YX2S, X5S},
                {X3S, X4S, YX3S, X6S}};

        double[][] fifthMatrix = {{getN(), XS, X2S, YS},
                {XS, X2S, X3S, YXS},
                {X2S, X3S, X4S, YX2S},
                {X3S, X4S, X5S, YX3S}};


        double delta = det(firstMatrix);
        double delta1 = det(secondsMatrix);
        double delta2 = det(thirdMatrix);
        double delta3 = det(fourthMatrix);
        double delta4 = det(fifthMatrix);

        a = delta4 / delta;
        b = delta3 / delta;
        c = delta2 / delta;
        d = delta1 / delta;

        for (int i = 0; i < getN(); i++) {
            S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
        }

        sigma = Math.sqrt(S / getN());


    }

    @Override
    public double linearXArgument(double x) {
        return a * Math.pow(x, 3) + b * Math.pow(x, 2) + c * x + d;
    }
```

# Экспоненциальная функция:

```
   @Override
    public void approximate() {
        double SX = 0;
        double SXX = 0;
        double SY = 0;
        double SXY = 0;

        for (int i = 0; i < getN(); i++) {
            SX += getX()[i];
            SXX += getX()[i] * getX()[i];
            SY += Math.log(getY()[i]);
            SXY += getX()[i] * Math.log(getY()[i]);
        }
```

```
        a = (SXX * SY - SX * SXY) / (SXX * getN() - SX * SX);
        b = (SXY * getN() - SX * SY) / (SXX * getN() - SX * SX);

        a = Math.pow(Math.E, a);

        for (int i = 0; i < getN(); i++) {
            S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
        }
        sigma = Math.sqrt(S / getN());
    }
    @Override
    public double linearXArgument(double x) {
        return a * Math.pow(Math.E, b * x);
    }
```

## Логарифмическая функция:

```
   @Override
    public void approximate() {

        double SX = 0;
        double SXX = 0;
        double SY = 0;
        double SXY = 0;

        for (int i = 0; i < getN(); i++) {
            SX += Math.log(getX()[i]);
            SXX += Math.log(getX()[i]) * Math.log(getX()[i]);
            SY += getY()[i];
            SXY += Math.log(getX()[i]) * getY()[i];
        }

        a = (SXY * getN() - SX * SY) / (SXX * getN() - SX * SX);
        b = (SXX * SY - SX * SXY) / (SXX * getN() - SX * SX);
        for (int i = 0; i < getN(); i++) {
            S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
        }
        sigma = Math.sqrt(S / getN());
    }

    @Override
    public double linearXArgument(double x) {
        return a * Math.log(x) + b;
    }
```

# Степенная функция:

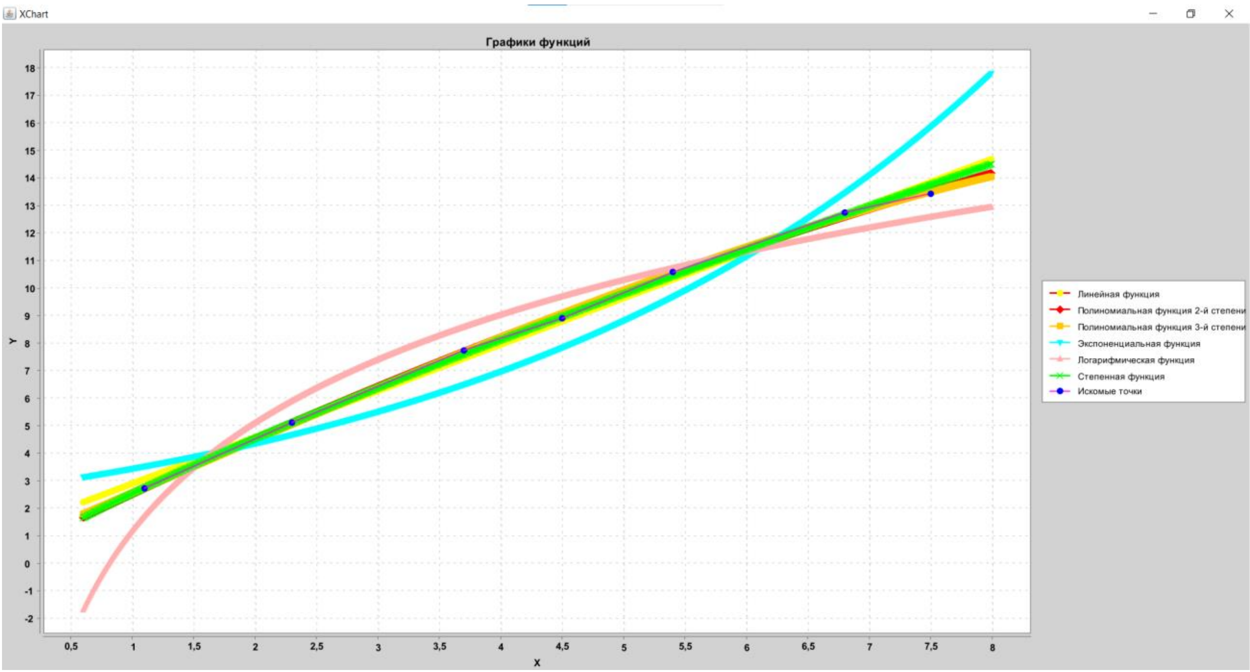
```java
@Override
public void approximate() {
    double SX = 0;
    double SXX = 0;
    double SY = 0;
    double SXY = 0;

    for (int i = 0; i < getN(); i++) {
        SX += Math.log(getX()[i]);
        SXX += Math.log(getX()[i]) * Math.log(getX()[i]);
        SY += Math.log(getY()[i]);
        SXY += Math.log(getX()[i]) * Math.log(getY()[i]);
    }
    a = (SXX * SY - SX * SXY) / (SXX * getN() - SX * SX);
    b = (SXY * getN() - SX * SY) / (SXX * getN() - SX * SX);

    a = Math.pow(Math.E, a);
    for (int i = 0; i < getN(); i++) {
        S += Math.pow((linearXArgument(getX()[i]) - getY()[i]), 2);
    }
    sigma = Math.sqrt(S / getN());
}

@Override
public double linearXArgument(double x) {
    return a * Math.pow(x, b);
}
```

# Результаты выполнения программы

```
Выберите способ ВЫВОДА значений (введите цифру):
1: Вывод в КОНСОЛЬ
2: Вывод в ФАЙЛ
1
Вывод данных в КОНСОЛЬ
Выберите способ ВВОДА значений (введите цифру):
1: С КЛАВИАТУРЫ
2: С ФАЙЛА
2
Ввод данных осуществляется с ФАЙЛА
Введите расположение вашего файла:
C:\Users\User\Desktop\Лабораторные работы\2 курс\2 семестр\Вычислительная математика\lab_4\src\main\resources\input
Введите количество точек:
7
Введите значения X (через пробел):
1.1
2.3
3.7
4.5
5.4
6.8
7.5
Введите значения Y (через пробел):
2.73
5.12
7.74
8.91
10.59
12.75
13.43
```

```
Линейная функция
Функция: 1.685382768738332*x+1.2167884769271677
Коэффициент a: 1.685382768738332
Коэффициент b: 1.2167884769271677
Мера отклонения S = 0.4730197919445184
Коэффициент корреляции Пирсона r = 0.9974189309974396
Среднеквадратическое отклонение δ = 0.2599504875780656

Полиномиальная функция 2-й степени
Функция: -0.05885292462866904*x^2+2.197385944878366*x+0.3742599606650451
Коэффициент a: -0.05885292462866904
Коэффициент b: 2.197385944878366
Коэффициент c: 0.3742599606650451
Мера отклонения S = 0.06900821293943504
Среднеквадратическое отклонение δ = 0.09928905344601188

Полиномиальная функция 3-й степени
Функция: -0.006041946721065101*x^3+0.019107039841418642*x^2+1.9118771220546966*x+0.6397716678961455
Коэффициент a: -0.006041946721065101
Коэффициент b: 0.019107039841418642
Коэффициент c: 1.9118771220546966
Коэффициент d: 0.6397716678961455
Мера отклонения S = 0.05939871933907776
Среднеквадратическое отклонение δ = 0.09211694379512363

Экспоненциальная функция
Функция: 2.7309451573402166*e^(0.23455048222905678*x)
Коэффициент a: 2.7309451573402166
Коэффициент b: 0.23455048222905678
Мера отклонения S = 10.707089854491679
Среднеквадратическое отклонение δ = 1.2367636253251397
```

```
Логарифмическая функция
Функция: 5.650037003535796*ln(x)+1.1988754276365305
Коэффициент a: 5.650037003535796
Коэффициент b: 1.1988754276365305
Мера отклонения S = 4.19977795240265
Среднеквадратическое отклонение δ = 0.774576193098306

Степенная функция
Функция: 2.5420901787906582*x^0.8380361310314202
Коэффициент a: 2.5420901787906582
Коэффициент b: 0.8380361310314202
Мера отклонения S = 0.1543956454706618
Среднеквадратическое отклонение δ = 0.14851437903961728

Наилучшая аппроксимирующая функция: Полиномиальная функция 3-й степени
Функция: -0.006041946721065101*x^3+0.019107039841418642*x^2+1.9118771220546966*x+0.6397716678961455
Ее среднеквадратическое отклонение δ = 0.09211694379512363
```


XChart
Графики функций
Линейная функция
Полиномиальная функция 2-й степени
Полиномиальная функция 3-й степени
Экспоненциальная функция
Логарифмическая функция
Степенная функция
Искомые точки
X
Y

# Вычислительная реализация задачи

Вариант 3

$y = \frac{4x}{x^4+1}$, $x \in [-2; 0]$; $h = 0,2$

Линейное и квадратичное приближение - ?

| X | -2 | -1,8 | -1,6 | -1,4 | -1,2 | -1 | -0,8 | -0,6 | -0,4 | -0,2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Y | -0,471 | -0,626 | -0,847 | -1,157 | -1,562 | -2 | -2,270 | -2,125 | -1,56 | -0,799 | 0 |

Линейная аппроксимация

$SX = \sum_{i=1}^{n} x_i$; $SXX = \sum_{i=1}^{n} x_i^2$; $SY = \sum_{i=1}^{n} y_i$; $SXY = \sum_{i=1}^{n} x_i y_i$;

$$\begin{cases} a\,SXX + b\,SX = SXY \\ a\,SX + bn = SY \end{cases}$$

$\Delta = SXX \cdot n - SX \cdot SX$

$\Delta_1 = SXY \cdot n - SX \cdot SY$

$\Delta_2 = SXX \cdot SY - SX \cdot SXY$

$a = \frac{\Delta 1}{\Delta}$; $b = \frac{\Delta 2}{\Delta}$

$SX = -11$; $SXX = 15,400$; $SY = -13,417$; $SXY = 12,793$

$\Delta = 15,4 \cdot 11 - (-11)^2 = 48,400$

$\Delta_1 = 12,793 \cdot 11 - (-11 \cdot (-13,417)) = -6,864$

$\Delta_2 = 15,4 \cdot (-13,417) - (-11) \cdot 12,793 = -65,899$

$a = \frac{\Delta 1}{\Delta} = -0,142$; $b = \frac{\Delta 2}{\Delta} = -1,362$

$P_1(x) = -0,142x + (-1,362)$

| $P_1(x) = ax+b$ | $-1{,}078$ | $-1{,}106$ | $-1{,}135$ | $-1{,}163$ | $-1{,}191$ | $-1{,}22$ | $-1{,}25$ | $-1{,}276$ | $-1{,}305$ | $-1{,}333$ | $-1{,}362$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\varepsilon_i$ | $-0{,}607$ | $-0{,}480$ | $-0{,}288$ | $-0{,}006$ | $0{,}371$ | $0{,}770$ | $1{,}022$ | $0{,}849$ | $0{,}255$ | $-0{,}534$ | $-1{,}362$ |

$$S = \sum_{i=1}^{n} \varepsilon_i^2 = 5{,}397$$

$$\delta = \sqrt{\frac{S}{n}} = \sqrt{\frac{5{,}397}{11}} \approx 0{,}700$$

Квадратичное приближение

$$X4S = \sum_{i=1}^{n} x_i^4;\quad X3S = \sum_{i=1}^{n} x_i^3;\quad X2S = \sum_{i=1}^{n} x_i^2;$$

$$X2YS = \sum_{i=1}^{n} x_i^2 y_i;\quad XS = \sum_{i=1}^{n} x_i;\quad XYS = \sum_{i=1}^{n} x_i y_i;$$

$$YS = \sum_{i=1}^{n} y_i;$$

~~1) ( YS XS~~

1) $$\begin{pmatrix} 11 & XS & X2S \\ XS & X2S & X3S \\ X2S & X3S & X4S \end{pmatrix} = \begin{pmatrix} 11 & -11 & 15{,}4 \\ -11 & 15{,}4 & -24{,}2 \\ 15{,}4 & -24{,}2 & 40{,}533 \end{pmatrix}$$

$$\Delta = 11 \cdot (15{,}4 \cdot 40{,}533 - (-24{,}2)\cdot(-24{,}2)) - (-11)\cdot((-11)\cdot 40{,}533 - (-24{,}2)\cdot 15{,}4) + 15{,}4 \cdot ((-11)\cdot(-24{,}2) - 15{,}4 \cdot 15{,}4) = 66{,}4532$$

2) $\begin{pmatrix} YS & XS & X2S \\ XYS & X2S & X3S \\ X2YS & X3S & X4S \end{pmatrix} = \begin{pmatrix} -13,417 & -11 & 15,4 \\ 12,793 & 15,4 & -24,2 \\ -15,097 & -24,2 & 40,533 \end{pmatrix}$

$\Delta_1 = (-13,417)\cdot(15,4\cdot 40,533-(-24,2)\cdot(-24,2)) - 12,793\cdot$
$\cdot((-11)\cdot 40,533-(-24,2)\cdot 15,4) + (-15,097)\cdot((-11)\cdot$
$\cdot(-24,2) - 15,4\cdot 15,4) = -19,656$

3) $\begin{pmatrix} n & YS & X2S \\ XS & XYS & X3S \\ X2S & X2YS & X4S \end{pmatrix} = \begin{pmatrix} 11 & -13,417 & 15,4 \\ -11 & 12,793 & -24,2 \\ 15,4 & -15,097 & 40,533 \end{pmatrix} =$

$\Delta_2 = 11\cdot(12,793\cdot 40,533-(-15,097)\cdot(-24,2)) - (-11)\cdot((-13,417)\cdot$
$\cdot 40,533-(-15,097\cdot 15,4) + 15,4\cdot((-13,417)\cdot(-24,2) -$
$- 12,793\cdot 15,4) = 226,652$

4) $\begin{pmatrix} N & XS & YS \\ XS & X2S & XYS \\ X2S & X3S & X2YS \end{pmatrix} = \begin{pmatrix} 11 & -11 & -13,417 \\ -11 & 15,4 & 12,793 \\ 15,4 & -24,2 & -15,097 \end{pmatrix};$

$\Delta_3 = 11\cdot(15,4\cdot(-15,097)-(-24,2)\cdot 12,793) - (-11)\cdot$
$\cdot((-11)\cdot(-15,097)-(-24,2)\cdot(-13,417)) + 15,4\cdot((-11)\cdot 12,793 -$
$- 15,4\cdot(-13,417)) = 118,038$

$a = \frac{\Delta_3}{\Delta}; \; b = \frac{\Delta_2}{\Delta}; \; c = \frac{\Delta_1}{\Delta}$

$a = 1,777; \; b = 3,411; \; c = -0,296$

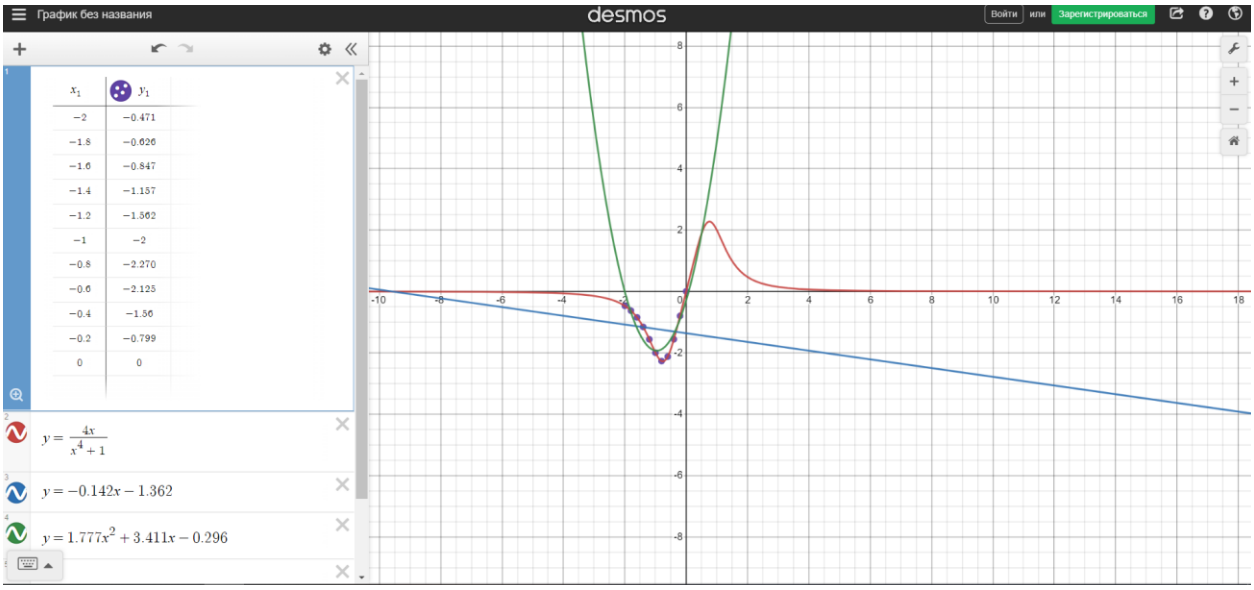
$1,777x^2 + 3,411x - 0,296$

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_2(x) = ax^2+bx+c$ | -0,012 | -0,680 | -1,206 | -1,589 | -1,831 | -1,930 | -1,888 | -1,703 | -1,376 | -0,907 | -0,296 |
| $\varepsilon$ | 0,459 | -0,054 | -0,359 | -0,432 | -0,269 | 0,070 | 0,382 | 0,422 | 0,184 | -0,108 | -0,286 |

$$S = \sum_{i=1}^{n} \varepsilon_i^2 = 1,064$$

$$\delta = \sqrt{\frac{S}{n}} = \sqrt{\frac{1,064}{11}} \approx 0,311$$

т.к. $0,311 < 0,7 \Rightarrow$ полином 2-й степени является наилучшей функцией для аппроксимации.




# Выводы

В ходе выполнения данной лабораторной работы я узнал что такое аппроксимация функций и поработал с аппроксимацией функций методом наименьших квадратов. В ходе исследования использовал такие функции, как: линейную функцию, полиномиальную функцию 2-й степени, полиномиальную функцию 3-й степени, экспоненциальную функцию, логарифмическую функцию, степенную функцию. Вычислил меру отклонения, среднеквадратическое отклонение, коэффициент Пирсона и научился определять наилучшую аппроксимирующую функцию.