Университет ИТМО

Факультет ФПИ и КТ

Лабораторная работа №4

“Аппроксимация функции методом наименьших квадратов”

По вычислительной математике

Вариант 8

Выполнил: Рогачев М. С.

Группа: P32082

Преподаватель: Машина Е. А.

Санкт-Петербург

2023

1) Задание

Задание:

1. Сформировать таблицу табулирования заданной функции на указанном интервале (см. табл. 1)
2. Построить линейное и квадратичное приближения по 11 точкам заданного интервала;
3. Найти среднеквадратические отклонения для каждой аппроксимирующей функции. Ответы дать с тремя знаками после запятой;
4. Выбрать наилучшее приближение;
5. Построить графики заданной функции, а также полученные линейное и квадратичное приближения;
6. Привести в отчете **подробные вычисления**.

2) Вычислительная реализация

Вычислительная часть лабораторной работы №4

$y = \frac{3x}{x^3+8}$, $x \in [-2, 0]$; $h = 0,2$

1. Таблица табулирования:

x	-2	-1,8	-1,6	-1,4	-1,2	-1	-0,8	-0,6	-0,4	-0,2	0
y	-0,25	-0,292	-0,328	-0,355	-0,357	-0(3)	-0,285	-0,221	-0,149	-0,075	0

2. Линейное приближение

В ходе вычислений получаем линейное приближение

$y = 0,1364x - 0,1043$

3. Квадратичное приближение

В ходе вычислений получаем квадратичное приближение

$y = 0,2067x^2 + 0,5498x + 0,0198$

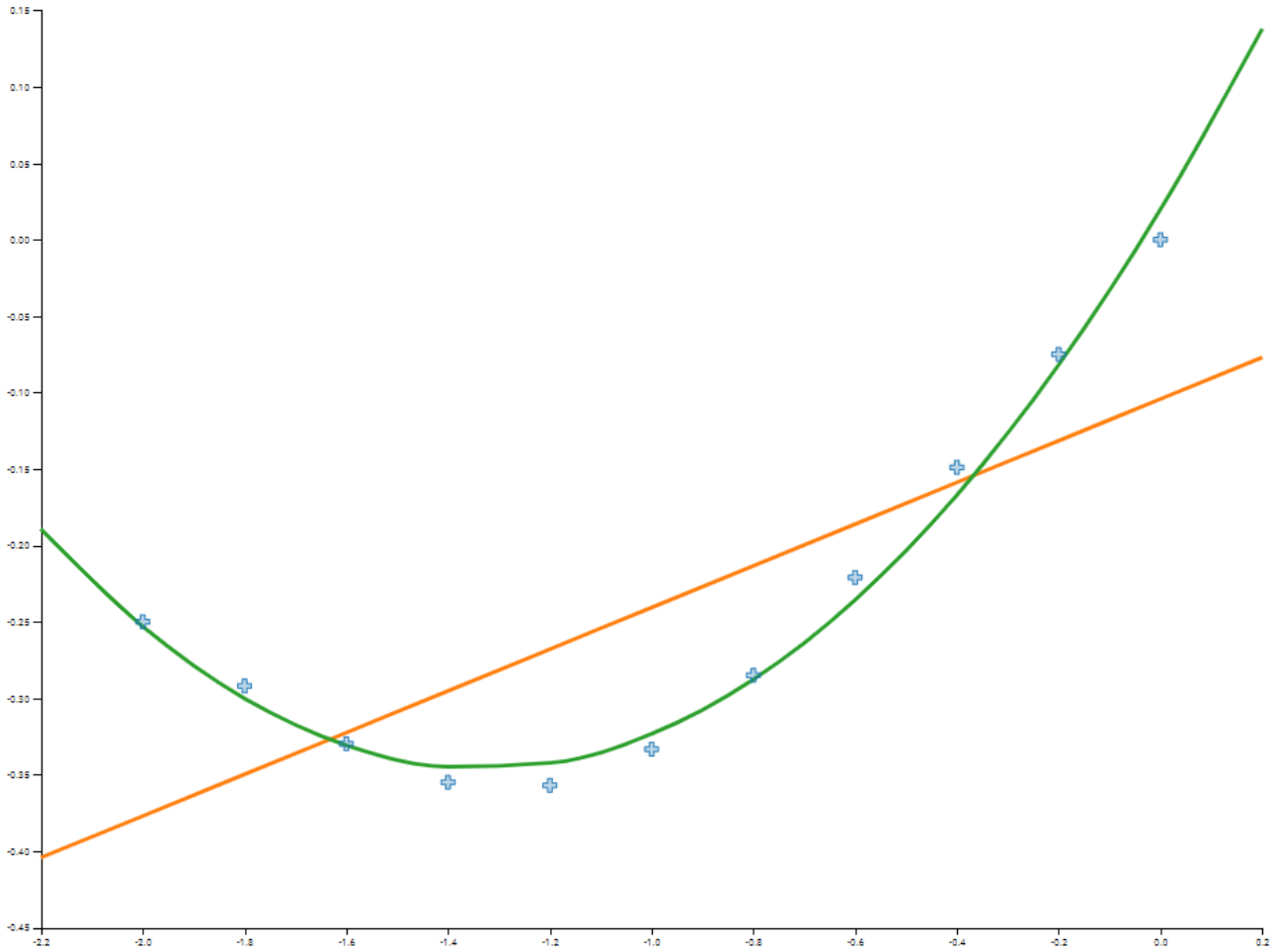
4. Среднее квадратичное отклонение

При линейном приближении: 0,2408

При квадратичном приближении: 0,0052

В этом случае наиболее лучшим будет квадратичное приближение.

График:



3) Листинг программы

```
import java.util.ArrayList;
import static java.lang.Math.*;
public class Methods {
    public static void solve(ArrayList<Point> in) {
        double[][] ans = new double[6][];

        double[][] f1 = {{in.size(), SX(in)},
                {SX(in), SX2(in)}};
        double[] c1 = {SY(in), SYX(in)};
        ans[0] = Function.iteration(f1, c1, 0.001);

        double[][] f2 = {{in.size(), SX(in), SX2(in)},
                {SX(in), SX2(in), SX3(in)},
                {SX2(in), SX3(in), SX4(in)}};
        double[] c2 = {SY(in), SYX(in), SYX2(in)};
        ans[1] = Function.iteration(f2, c2, 0.001);

        double[][] f3 = {{in.size(), SX(in), SX2(in), SX3(in)},
                {SX(in), SX2(in), SX3(in), SX4(in)},
                {SX2(in), SX3(in), SX4(in), SX5(in)},
                {SX3(in), SX4(in), SX5(in), SX6(in)}};
        double[] c3 = {SY(in), SYX(in), SYX2(in), SYX3(in)};
        ans[2] = Function.iteration(f3, c3, 0.001);
```

```
        double[][] f4 = {{in.size(), SX(in)},
                {SX(in), SX2(in)}};
        double[] c4 = {SLnY(in), SLnYX(in)};
        ans[3] = Function.iteration(f4, c4, 0.001);
        ans[3][0] = pow(Math.E, ans[3][0]);

        double[][] f5 = {{in.size(), SLnX(in)},
                {SLnX(in), SLnX2(in)}};
        double[] c5 = {SY(in), SYLnX(in)};
        ans[4] = Function.iteration(f5, c5, 0.001);

        double[][] f6 = {{in.size(), SLnX(in)},
                {SLnX(in), SLnX2(in)}};
        double[] c6 = {SLnY(in), SLnYLnX(in)};
        ans[5] = Function.iteration(f6, c6, 0.001);
        ans[5][0] = pow(Math.E, ans[5][0]);

        double min = Double.MAX_VALUE;
        int id = 0;
        Algo[] val = Algo.values();

        System.out.printf("\nКоэффициент корреляции Пирсона: %.5f\n\n",
Piers(in));
        System.out.println("Среднеквадратическое отклонения функций:");
        for (int i = 0; i < ans.length; i++) {
            double tmp = MO(in, val[i], ans[i]);

            if (min > tmp) {
                min = tmp;
                id = i;
            }

            System.out.println(val[i] + " : " + tmp);
        }

        System.out.println("\nЛучший выбор это " + val[id] + "\n");

        System.out.println("Функции:");
        for (int i = 0; i < ans.length; i++) {
            System.out.print(val[i] + " : ");
            Function.drawFunction(val[i], ans[i]);
        }

        System.out.println("");

        Output.write(in, ans);
        Chart.draw(in, ans, id);

    }

    static double MO(ArrayList<Point> in, Algo num, double[] c) {
        double ans = 0;

        for (Point point : in) {
            ans += pow(Function.getFunction(point.getX(), num, c) -
```

```
point.getY(), 2);
        }

        ans /= in.size();
        ans = sqrt(ans);

        return ans;
    }

    static double Piers(ArrayList<Point> in) {
        double ans = 0, a = 0, b = 0;
        double sx = SX(in) / in.size(), sy = SY(in) / in.size();

        for (Point point: in) {
            ans += (point.getX() - sx) * (point.getY() - sy);
        }

        for (Point point: in) {
            a += pow(point.getX() - sx, 2);
        }

        for (Point point: in) {
            b += pow(point.getY() - sy, 2);
        }

        return ans / sqrt(a * b);
    }

    static double SX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += q.getX();
        }

        return ans;
    }

    static double SX2(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 2);
        }

        return ans;
    }

    static double SX3(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 3);
        }

        return ans;
```

```
    }

    static double SX4(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 4);
        }

        return ans;
    }

    static double SX5(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 5);
        }

        return ans;
    }

    static double SX6(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 6);
        }

        return ans;
    }

    static double SY(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += q.getY();
        }

        return ans;
    }

    static double SYX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += q.getX() * q.getY();
        }

        return ans;
    }

    static double SYX2(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
```

```
            ans += pow(q.getX(), 2) * q.getY();
        }

        return ans;
    }

    static double SYX3(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(q.getX(), 3) * q.getY();
        }

        return ans;
    }

    static double SLnX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += log(q.getX());
        }

        return ans;
    }

    static double SLnX2(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += pow(log(q.getX()), 2);
        }

        return ans;
    }

    static double SLnY(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += log(q.getY());
        }

        return ans;
    }

    static double SLnYLnX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += log(q.getY()) * log(q.getX());
        }

        return ans;
    }
```

```
    static double SLnYX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += log(q.getY()) * q.getX();
        }

        return ans;
    }

    static double SYLnX(ArrayList<Point> in) {
        double ans = 0;

        for (var q: in) {
            ans += q.getY() * log(q.getX());
        }

        return ans;
    }

}
```

4) Примеры работы программы

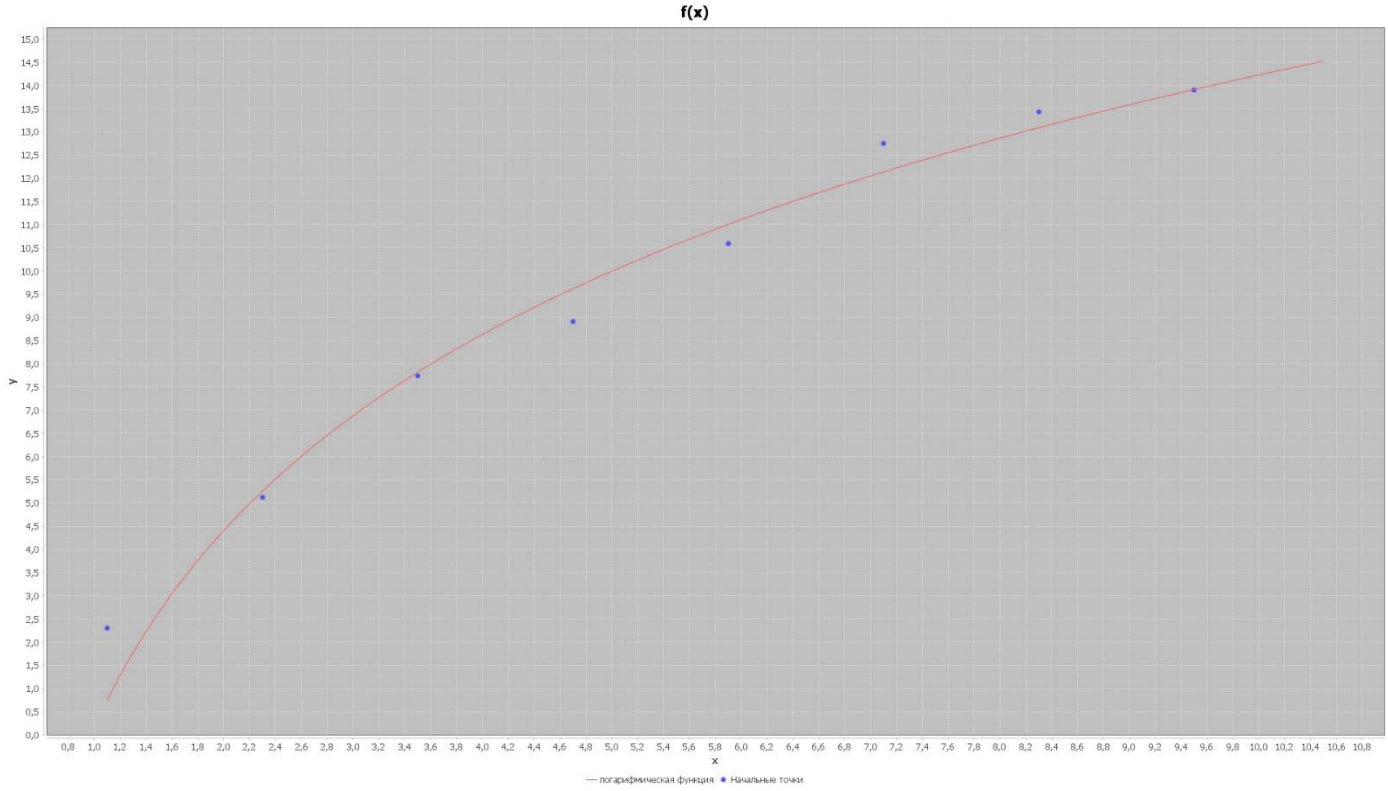
```
Это четвертая лабораторная работа по Вычислительной математике
Введите:
0 - для выхода
1 - для ввода данных из консоли
2 - для ввода данных из файла
2
Введите имя файла:
in1.txt
Point{x=1.1, y=2.3}
Point{x=2.3, y=5.12}
Point{x=3.7, y=7.74}
Point{x=4.5, y=8.91}
Point{x=5.4, y=10.59}
Point{x=6.8, y=12.75}
Point{x=7.5, y=13.43}
Point{x=8.0, y=13.9}

Коэффициент корреляции Пирсона: 0,99425

Среднеквадратическое отклонения функций:
линейная функция : 0.5801594911762069
полиномиальная функция 2-й степени : 5.637698540035298
полиномиальная функция 3-й степени : 11.173858053090285
экспоненциальная функция : 3.791996990947828
логарифмическая функция : 0.7466624259569722
степенная функция : 1.6988330605489341

Лучший выбор это линейная функция

Функции:
линейная функция : 1,82370 * x + 0,21102
полиномиальная функция 2-й степени : 0,22920 * x ^ 2 + 1,37680 * x + -7,54886
полиномиальная функция 3-й степени : 0,02930 * x ^ 3 + 0,19460 * x ^ 2 + 0,91060 * x + -14,24615
экспоненциальная функция : 1,19236 * e ^ (0,36326 * x)
логарифмическая функция : 6,37869 * ln(x) + 0,12114
степенная функция : 1,14410 * x ^ 1,29897
```



5) Вывод

в ходе лабораторной работы я научился реализовывать в программе методы для аппроксимации функций по точкам, такие как линейное приближение, квадратичное приближение и логарифмическое приближение и другие.