Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4 по "Вычислительной математике"

Аппроксимация функции методом наименьших квадратов

Вариант 4

Выполнил: Дьяконов Михаил Павлович
Группа: P3211
Преподаватель: Малышева Татьяна Алексеевна

Университет ИТМО, 2021

Цель работы

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

Рабочие формулы

Линейная функция:

$$\phi = ax + b$$

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

Квадратичная функция:

$$\phi = ax^2 + bx + c$$

$$\begin{cases} \text{cn} + \text{bSX} + \text{aSXX} = \text{SY} \\ \text{cSX} + \text{bSXX} + \text{aSXX} = \text{SXY} \\ \text{cSXX} + \text{bSXXX} + \text{aSXXX} = \text{SXXY} \end{cases}$$

Степенная функция:

$$\phi = ax^b$$

Замена $Y = ln(\phi(x)); A = ln(a); B = b; X = ln(x)$

Экспоненциальная функция:

$$\phi = ae^(bx)$$

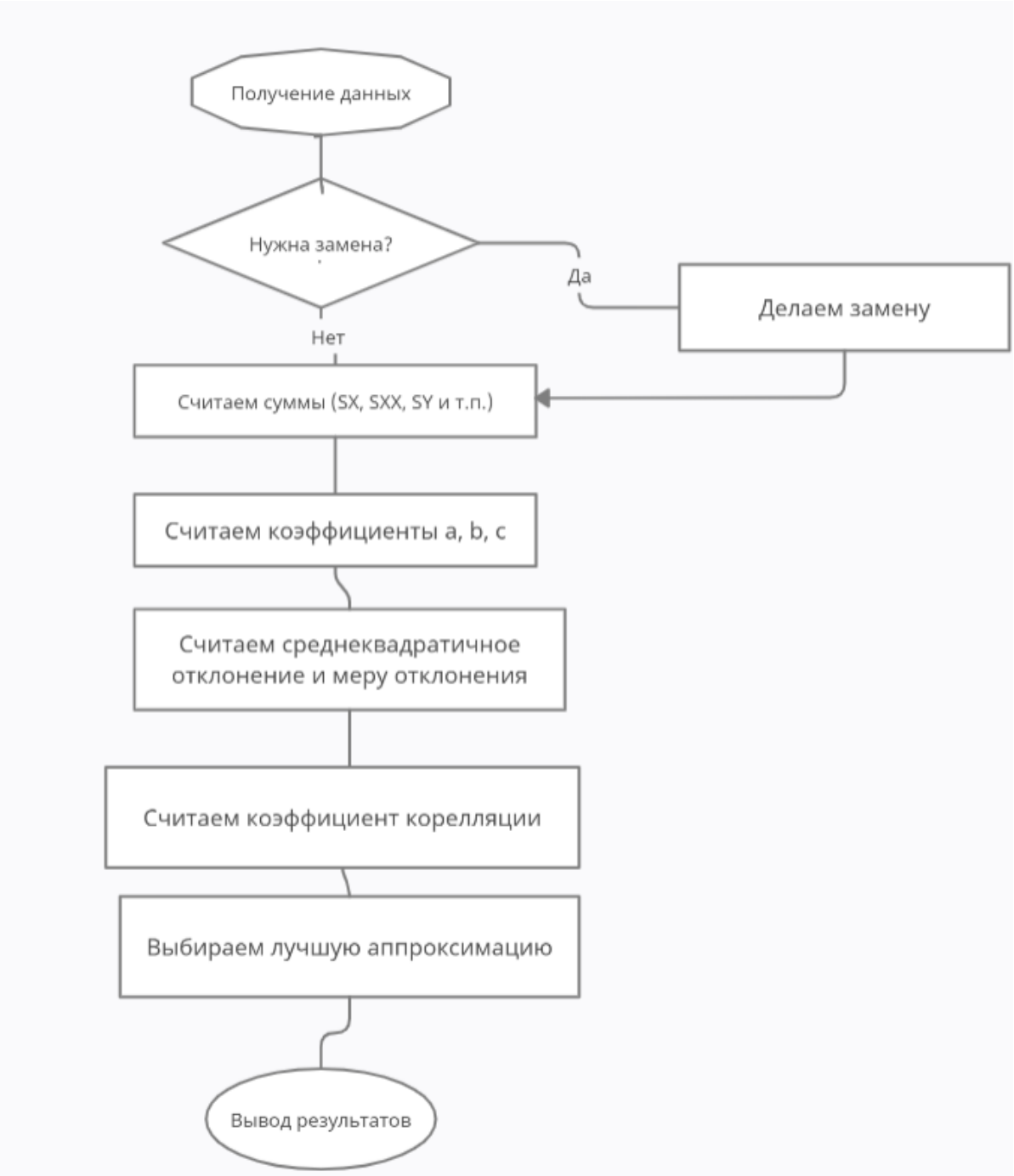
Замена $Y = ln(\phi(x)); A = ln(a); B = b$

Логарифмическая функция:

$$\phi = aln(x) + b$$

Замена $X = ln(x); A = ln(a); B = b$

Блок-схема


Получение данных
Нужна замена?
Да
Делаем замену
Нет
Считаем суммы (SX, SXX, SY и т.п.)
Считаем коэффициенты a, b, c
Считаем среднеквадратичное отклонение и меру отклонения
Считаем коэффициент корелляции
Выбираем лучшую аппроксимацию
Вывод результатов

Листинг программы

FunctionResearcher.ts

```
    export class FunctionResearcher {

    private static readonly FUNCTION_COLORS: string[] = ['green', 'red', 'blue', 'pink', 'yellow'];
    private static readonly MANAGERS: ApproximationManager[] = [new LinearApproximationManager(),
        new QuadraticApproximationManager(), new SedateApproximationManager(),
        new ExponentialApproximationManager(), new LogarithmicApproximationManager()];
    private static readonly FUNCTION_VIEWS: string[] = ['ax+b', 'ax^2+bx+c', 'ax^b', 'ae^(bx)',
         'a*ln(x)+b'];

    research(points: Point[]): FinalResult {
        const functions: ApproximatingFunction[] =
            FunctionResearcher.MANAGERS.map((m) => m.solve(points));

        const resultFunctions: ResearchResult[] = functions.map((f, i) => {
            const s: number = points.map((point) => Math.pow(f.fnc(point.x) - point.y, 2))
                .reduce((a, b) => a + b);
            return {
                color: FunctionResearcher.FUNCTION_COLORS[i],
                view: FunctionResearcher.FUNCTION_VIEWS[i],
                fnc: f.fnc,
                a: MathUtils.roundToFixed(f.a),
                b: MathUtils.roundToFixed(f.b),
                c: f.c === undefined ? f.c : MathUtils.roundToFixed(f.c),
                deviationMeasure: MathUtils.roundToFixed(s),
                standardDeviation: MathUtils.roundToFixed(Math.sqrt(s / points.length))
            };
        });
        const correlationCoeff: number =
             MathUtils.roundToFixed(CorrelationCalculator.calculate(points));

        return {functions: resultFunctions, correlationCoeff: correlationCoeff};
    }
}
```

CorrelationCalculator.ts:

```
export class CorrelationCalculator {

    static calculate(points: Point[]): number {
        const xMiddle = points.map(point => point.x).reduce((a, b) => a + b) / points.length;
        const yMiddle = points.map(point => point.y).reduce((a, b) => a + b) / points.length;

        return points.map(point => (point.x - xMiddle) * (point.y - yMiddle)).reduce((a, b) => a+b) /
            Math.sqrt(points.map(point => Math.pow(point.x - xMiddle, 2)).reduce((a , b) => a+b)
                * points.map(point => Math.pow(point.y - yMiddle, 2)).reduce((a , b) => a+b));
    }
}
```

MatrixUtils.ts:

```
export class MatrixUtils {

    static solveSLAU(coeffs: number[][], freeMembers: number[]): number[] {
        if (coeffs === undefined ||
            freeMembers === undefined
            || coeffs.length !== freeMembers.length) {
            return undefined;
        }
```

```
        const det: number = math.det(coeffs);
        const dets: number[] = [];

        for (let i = 0; i < freeMembers.length; i++) {
            const matrix = coeffs.map(arr => arr.slice());
            for (let j = 0; j < freeMembers.length; j++) {
                matrix[j][i] = freeMembers[j];
            }
            dets.push(math.det(matrix));
        }
        return Array(freeMembers.length).fill(0).map((v, i) => dets[i] / det);
    }
}
```

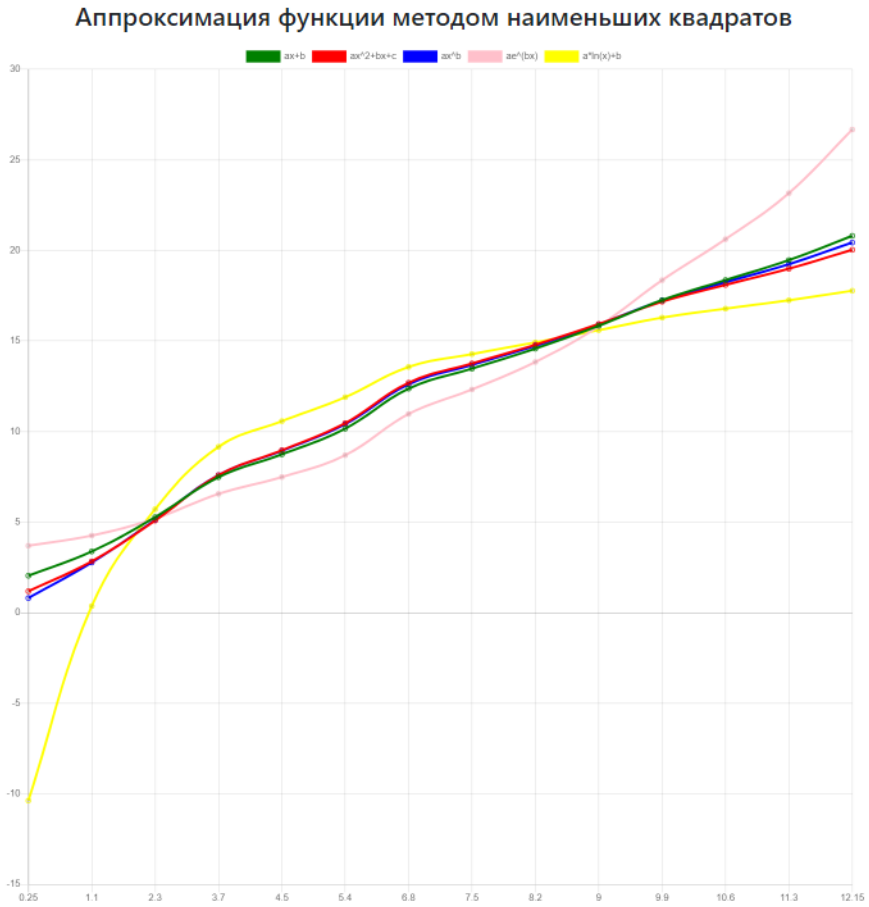
PointsUtils.kt:

```
export class PointUtils {

    static calculatePointsCharacteristics(points: Point[]): PointsCharacteristics {
        return {
            n: points.length,
            sx: this.calculateSumFromPointMapper(points, (point) => point.x),
            sxx: this.calculateSumFromPointMapper(points, (point) => point.x * point.x),
            sxxx: this.calculateSumFromPointMapper(points, (point) => point.x * point.x * point.x),
            sxxxx: this.calculateSumFromPointMapper(points, (point) => Math.pow(point.x, 4)),
            sy: this.calculateSumFromPointMapper(points, (point) => point.y),
            sxy: this.calculateSumFromPointMapper(points, (point) => point.x * point.y),
            sxxy: this.calculateSumFromPointMapper(points, (point) => point.x * point.x * point.y)
        };
    }

    static calculateSumFromPointMapper(points: Point[], mapper: (Point) => number) {
        return points.map(mapper).reduce((a, b) => a + b);
    }
}
```

Примеры



Введите точки для аппроксимации функции, кол-во точек должно быть **от 12 до 20**

Выберите формат ввода и вывода

○ Из файла

◉ Из формы

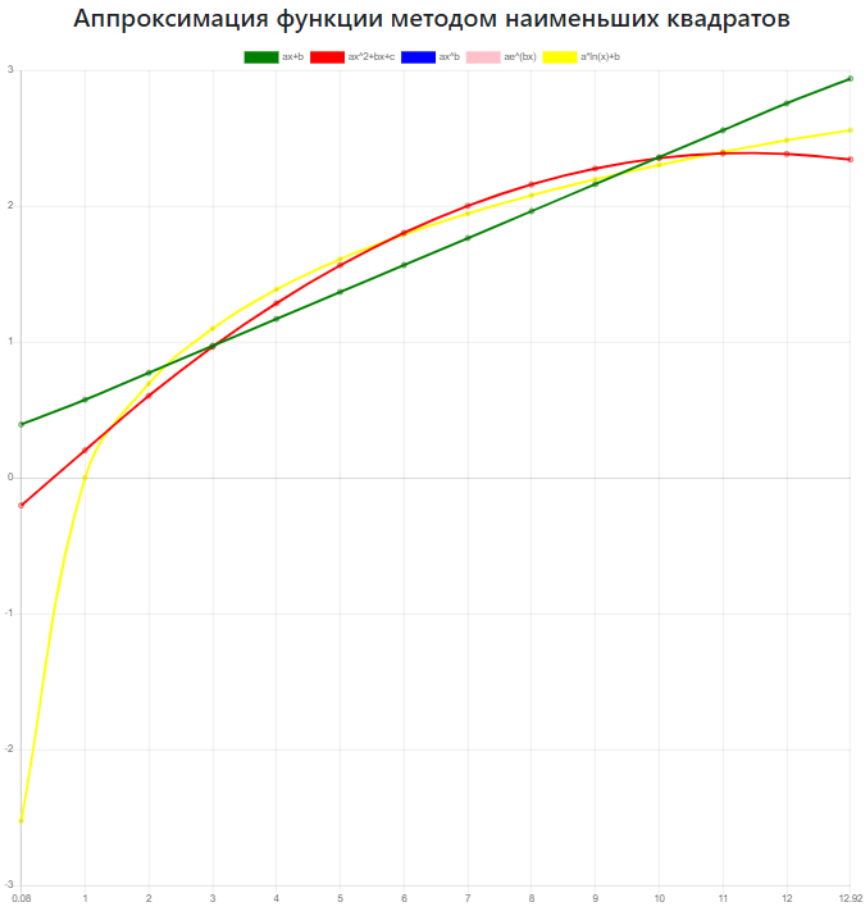
x	1,1	2,3	3,7	4,5	5,4	6,8	7,5	8,2	9,0	9,9	10,6	11,3
f(x)	2,73	5,12	7,74	8,91	10,59	12,75	13,43	15,01	15,77	17,02	18,32	18,99

Добавить точку | Удалить точку

Провести исследование

Цвет	Вид функции	a	b	c	Мера отклонения S	Среднеквадратичное отклонение δ
green	ϕ(x) = ax+b	1.5769	1.6461	-	1.3491	0.3353
red	ϕ(x) = ax^2+bx+c	-0.0324	1.9859	0.6871	0.3113	0.1611
blue	ϕ(x) = ax^b	2.5573	0.8323	-	0.3524	0.1714
pink	ϕ(x) = ae^(bx)	3.5468	0.1661	-	39.5932	1.8164
yellow	ϕ(x) = a*ln(x)+b	7.2479	-0.3276	-	19.7758	1.2837

Коэффициент корреляции 0.9978



Введите точки для аппроксимации функции, кол-во точек должно быть **от 12 до 20**

Выберите формат ввода и вывода

- (•) Из файла
- ( ) Из формы

Выберите файл со входными данными

Выберите файл log.json

Провести исследование

Результат

Вывод

В результате выполнения данной лабораторной работы я узнал, как работает аппроксимация методом наименьших квадратов. Также научился выбирать лучшую аппроксимацию по среднеквадратичному отклонению.