

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА  
(национальный исследовательский университет)»**

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

**Лабораторная работа № 6**

«Сравнение скорости сходимости метода Якоби и Зейделя»

по дисциплине «Численные методы линейной алгебры»

Работу выполнил

студент группы ИУ9-72Б

Жук Дмитрий

**Цель работы**

Используя результаты предыдущих лабораторных работ реализовать и сравнить скорость сходимости методов Зейделя и Якоби.

**Реализация**

Был создан репозиторий на GitHub <https://github.com/ZhukDmitryOlegovich/num-methods>

Используя ранее реализованный функционал, а именно классы, представляющие из себя абстракцией квадратной матрицы и вектора. По аналогии с методом Гаусса, была создана похожая реализация метода.

Так как метод Зейделя и Якоби является крайне схожими по логике работы, то они были объединены в единую функцию, а чтобы выбрать «мод» (то какой метод запустить), используются дополнительные параметры функции.

**Вывод**

В ходе выполнения лабораторной работы были реализованы выше заявленные методы, а также был выявлен тот факт, что в подавляющем большинстве случаем метода Зейделя сходится быстрее чем метод Якоби.

Данную работу так же можно открыть и посмотреть по ссылке <https://zhukdmitryolegovich.github.io/num-methods/lab6/>.

**Приложение**

    jakobi(

        other: Vector<A>,

        option: { z?: boolean; eps?: number; maxCount?: number; },

    ) {

        const N = this.countColons();

        const { z, eps = 0, maxCount = Infinity } = option;

        const x = new Vector(fromLength(N, () => 0));

        let xBefore = x;

        let count = 0;

        do {

            xBefore = Vector.fromMatrix(x.clone());

            const xNext = z ? x : xBefore;

            for (let i = 0; i < N; i++) {

                x.matrix[0][i] = other.matrix[0][i]

                    - fromLength(

                        N,

                        (j) => (j === i ? 0 : this.matrix[j][i] \* xNext.matrix[0][j]),

                    )

                        .reduce((a, b) => a + b, 0);

            }

            count++;

        } while (

            count < maxCount

            && eps < Math.abs(Vector.fromMatrix(xBefore.add(x.mulN(-1))).norma())

        );

        return { count, result: x };

    }

Изображение выглядит как текст

Автоматически созданное описание