МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

**Практикум по курсу**

**"Суперкомпьютеры и параллельная обработка данных"**

**Разработка параллельной версии программы**

**ОТЧЕТ**

**о выполненном задании**

студентки 321 учебной группы факультета ВМК МГУ

Бурдюговой Марии Витальевны

Москва, 2023 год

Оглавление

[Постановка задачи 2](#__RefHeading___Toc2261_2610638938)

[Описание алгоритма Гаусса - Зейделя решения системы линейных уравнений 2](#__RefHeading___Toc2263_2610638938)

[Результаты замеров времени выполнения и построение диаграммы 3](#__RefHeading___Toc2265_2610638938)

[Анализ результатов 4](#__RefHeading___Toc2271_2610638938)

[Выводы 4](#__RefHeading___Toc2273_2610638938)

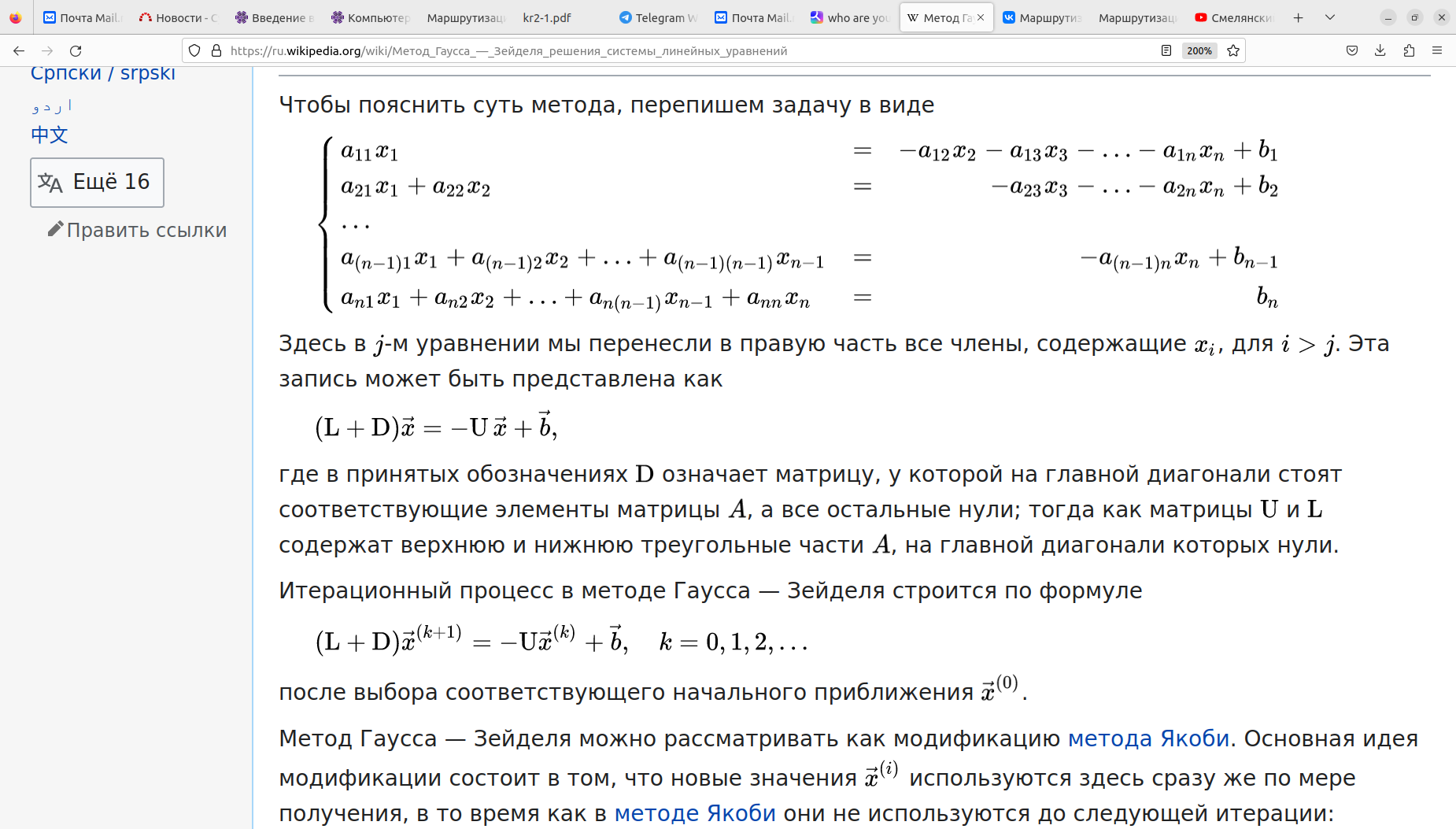
### Постановка задачи

Для предложенного алгоритма решения численной задачи с использованием метода релаксации Гаусса-Зейделя реализовать две версии параллельных программ с использованием технологии OpenMP:  
a) Вариант параллельной программы с распределением витков циклов при помощи директивы for.  
б) Вариант параллельной программы с использованием механизма задач. Для программ с регулярной зависимостью по данным допускается реализация и сравнение различных версий конвейерного выполнения циклов, параллелизм по гиперплоскостям и др.

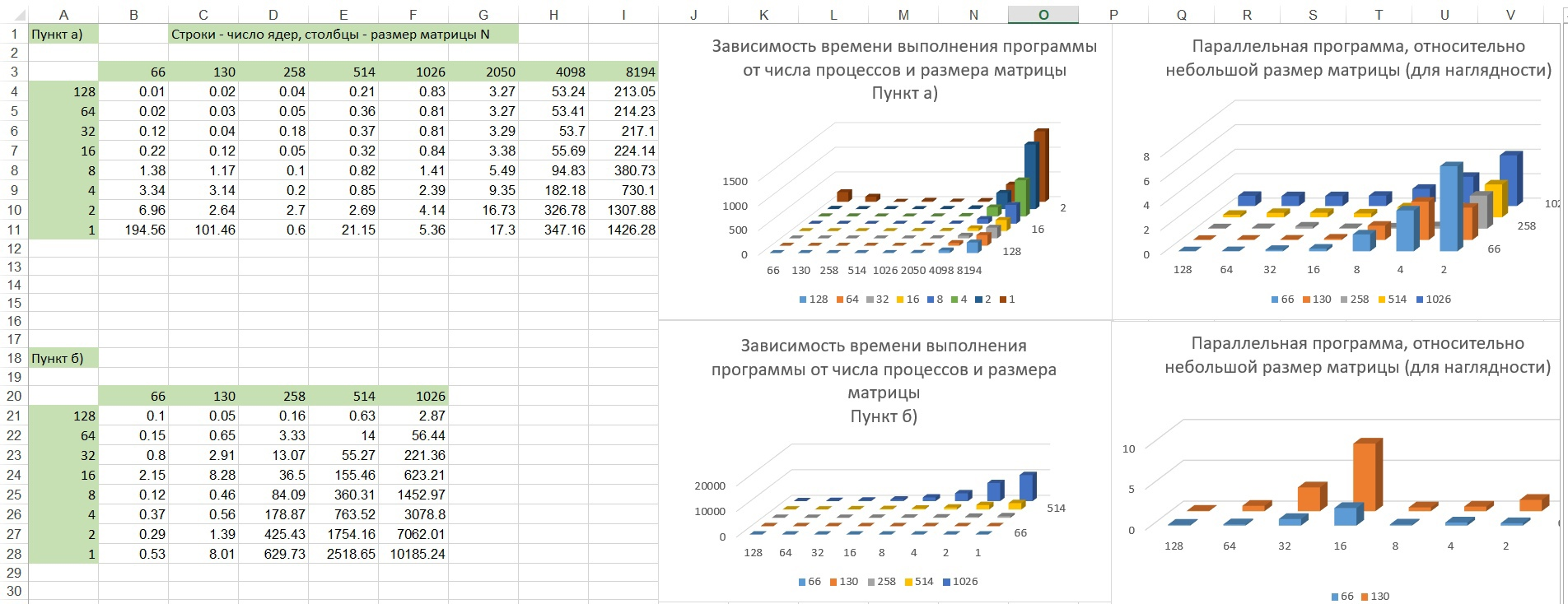
Также необходимо исследовать масштабируемость полученной программы, построить графики зависимости времени её выполнения от числа используемых нитей и объёма входных данных.

# Описание алгоритма Гаусса - Зейделя решения системы линейных уравнений





# Результаты замеров времени выполнения и построение диаграммы



# 

# Полный код можно найти по ссылке: https://github.com/murlinmurlo/skipod

# Анализ результатов

При увеличении количества потоков, особенно на больших размерах матриц, скорость выполнения алгоритма заметно увеличивается. Однако лучший результат получается распараллеливанием не на максимальное число потоков, а на 16. Это происходит из-за того, что при слишком большом количестве потоков тратятся ресурсы на управление ими, что приводит к дополнительным затратам времени.

# Выводы

OpenMP позволяет значительно увеличить производительность вычислений на многопоточных системах, однако необходимо выбирать оптимальное количество потоков для достижения наилучших результатов.