Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования

«Уральский федеральный университет  
имени первого Президента России Б.Н.Ельцина»

Институт фундаментального образования  
Кафедра интеллектуальных информационных технологий

**РАЗРАБОТКА ПАРАЛЛЕЛЬНОГО АЛГОРИТМА РЕШЕНИЯ СЛАУ**

КУРСОВАЯ РАБОТА

Пояснительная записка

|  |  |
| --- | --- |
| Руководитель | Мокрушин А.А. |
| Студенты гр. ФОМ-151001 | Ибакаева А.А., Шевченко Т.Р. |

Екатеринбург – 2015

Оглавление

[Введение 3](#_Toc440362941)

[Постановка задачи 4](#_Toc440362942)

[Последовательный алгоритм 5](#_Toc440362943)

[Описание алгоритма 5](#_Toc440362944)

[Анализ алгоритма 5](#_Toc440362945)

[Параллельный алгоритм 6](#_Toc440362946)

[Описание алгоритма 6](#_Toc440362947)

[Анализ алгоритма 6](#_Toc440362948)

[Численный эксперимент 7](#_Toc440362949)

[Выводы 8](#_Toc440362950)

# Введение

Под системой линейных алгебраических уравнений (СЛАУ) подразумевают систему (1).

(1)

Система уравнений (1) содержит *m* уравнений и *n* неизвестных *(x1, x2, x3, …, xn).* Прилагательное «линейных» означает, что все неизвестные (их еще называют переменными) входят только в первой степени.

Параметры называют коэффициентами, – свободными членами СЛАУ.

Если все свободные члены , то СЛАУ называют однородной. Если среди свободных членов есть хотя бы один, отличный от нуля, то СЛАУ называют неоднородной.

Решением СЛАУ (1) называют всякую упорядоченную совокупность чисел *(a1, a2, …, an)*, если элементы этой совокупности, подставленные в заданном порядке вместо неизвестных *x1, x2, x3, …, xn*, обращают каждое уравнение СЛАУ в тождество.

Любая однородная СЛАУ имеет хотя бы одно решение: нулевое (тривиальное), т.е. *x1* = *x2* = *x3* = … = *xn* = 0.

Если СЛАУ (1) имеет хотя бы одно решение, ее называют совместной, если же решений нет – несовместной. Если совместная СЛАУ имеет ровно одно решение, её именуют определённой, если бесконечное множество решений – неопределённой.

С каждой СЛАУ можно связать несколько матриц; более того – саму СЛАУ можно записать в виде матричного уравнения. Для СЛАУ (1) рассмотрим такие матрицы:

;

;

; .

Матрица *A* называется матрицей системы. Элементы данной матрицы представляют собой коэффициенты заданной СЛАУ.

Матрица называется расширенной матрицей системы. Её получают добавлением к матрице системы столбца, содержащего свободные члены *b1, b2, ..., bm*. Обычно этот столбец отделяют вертикальной чертой, – для наглядности.

Матрица-столбец *B* называется матрицей свободных членов, а матрица-столбец *X* – матрицей неизвестных.

Используя введённые выше обозначения, СЛАУ (1) можно записать в форме матричного уравнения: *A⋅X = B*.

Целью курсовой работы является разработка и сравнение работы последовательного и параллельного алгоритмов для решения СЛАУ.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Реализовать последовательный алгоритм;
2. Оценить сложность последовательного алгоритма.
3. Посчитать время работы последовательного алгоритма.
4. Разработать параллельный алгоритм.
5. Оценить сложность параллельного алгоритма.
6. Посчитать время работы параллельного алгоритма.
7. Сравнить результаты работы последовательного и параллельного алгоритмов.

# Постановка задачи

Необходимо реализовать последовательный алгоритм решения системы линейных алгебраических уравнений методом Гаусса, оценить его сложность и посчитать время работы.

Также требуется разработать параллельный алгоритм решения системы линейных алгебраических уравнений на основе метода Гаусса, оценить его сложность и посчитать время работы.

После реализации алгоритмов необходимо сравнить время их работы на матрицах больших размеров и объяснить полученные результаты.

Для реализации алгоритмов можно использовать язык программирования С++, параллельный алгоритм разрабатывать с использованием библиотеки MPI.

# Последовательный алгоритм

## Описание алгоритма

Метод Гаусса основывается на возможности выполнения преобразований линейных уравнений, которые не меняют при этом решение рассматриваемой системы (эквивалентные преобразования). К ним относятся:

1. Умножение любого из уравнений на ненулевую константу,
2. Перестановка уравнений,
3. Прибавление к уравнению любого другого уравнения системы.

Метод Гаусса состоит из последовательного выполнения двух этапов. Первый этап – прямой ход метода Гаусса – исходная система линейный уравнений приводится к верхнему треугольному виду *Ux = c,* где матрица коэффициентов получаемой системы имеет вид (2).

(2)

Прямой ход метода Гаусса состоит в последовательном исключении неизвестных в уравнениях решаемой системы линейных уравнений. На итерации , метода производится исключение неизвестной *i* для всех уравнений с номерами *k*, больших *i* . Для этого из этих уравнений осуществляется вычитание строки *i* умноженной на константу с тем, чтобы результирующий коэффициент при неизвестной *xi*в строках оказался нулевым – все необходимые вычисления могут быть определены при помощи соотношений:

где , , .

Все коэффициенты при неизвестных, расположенные ниже главной диагонали и левее столбца *i*, уже являются нулевыми. На *i*-ой итерации прямого хода метода Гаусса осуществляется обнуление коэффициентов столбца *i*, расположенных ниже главной диагонали, путем вычитания строки *i*, умноженной на нужную ненулевую константу. После проведения *(n-1)* подобной итерации матрица, определяющая систему линейных уравнений, становится приведенной к верхнему треугольному виду.

При выполнении прямого хода метода Гаусса строка, которая используется для исключения неизвестных, носит наименование ведущей, а диагональный элемент ведущей строки называется ведущим элементом. Выполнение вычислений является возможным только, если ведущий элемент имеет ненулевое значение.

Обратный ход. После приведения матрицы коэффициентов к верхнему треугольному виду становится возможным определение значений неизвестных. Из последнего уравнения преобразованной системы может быть вычислено значение переменной *xm-1*, после этого из предпоследнего уравнения становится возможным определение переменной *xm-2* и т.д. В общем виде, выполняемые вычисления при обратном ходе метода Гаусса могут быть представлены при помощи соотношений:

где

## Анализ алгоритма

Пусть n – количество строк в матрице, m – количество столбцов.

Вычислительная сложность прямого хода алгоритма Гаусса с выбором ведущей строки имеет порядок *O(n2\*m).*

Вычислительная сложность обратного хода алгоритма Гаусса составляет *O(n\*m).*

Общая сложность алгоритма получается равной *O(n2\*m).*

# Параллельный алгоритм

## Описание алгоритма

Пусть есть система линейных уравнений с матрицей коэффициентов *A* и вектором свободных переменных *B*.

.

Присоединим к матрице *А* вектор свободных переменных *B* и получим матрицу *A’*.

.

Матрицу *A’* разделим на полосы в нулевом процессе. Количество строк в полосе считается равным отношению количества строк в матрице к числу процессов. Если количество строк не делится нацело на количество процессов, то последние процессы получают на 1 строку в полосе больше. Нулевой процесс рассылает остальным процессам соответствующие им полосы матрицы *A’.* Все ненулевые процессы принимают полосы матрицы.

В каждом из процессов будем выполнять прямой ход Гаусса, начиная определение активной строки в нулевом процессе.

В нулевом процессе определяется текущая активная строка и рассылается остальным процессам. Нулевой процесс пересчитывает свою полосу для текущей активной строки. Затем нулевой процесс считает следующую активную строку, рассылает ее остальным потокам и пересчитывает свою полосу и т.д. пока в нулевом процессе не закончатся строки в полосе. Активной строкой последовательно становятся все строки в полосе.

В это время каждый ненулевой процесс принимает активные строки и пересчитывает свою полосу для текущей активной строки. Далее процесс определяет свою активную строку и рассылает ее всем процессам, ранг которых больше текущего процесса. После передачи активной строки данный процесс пересчитывает свою полосу для текущей активной строки. Затем процесс вычисляет следующую активную строку, отправляет ее другим процессам и пересчитывает свою полосу и т.д. пока не закончатся строки в полосе. В конце каждый процесс отправляет нулевому процессу свою готовую полосу матрицы.

Нулевой процесс принимает от остальных процессов полосы матрицы и собирает их в конечную матрицу.

Далее выполняем в нулевом процессе обратный ход Гаусса на основе полученной треугольной матрицы и получаем вектор решения системы линейных алгебраических уравнений.

## Анализ алгоритма

Анализ

# Численный эксперимент

Для 1 процесса:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | Среднее |
| 100 | 0.004 |  |  |  |  |  |  |
| 500 |  |  |  |  |  |  |  |
| 1000 | 5.208 |  |  |  |  |  |  |
| 2000 | 40.421 | 41.91 | 41.356 | 41.559 | 40.841 | 41.311 |  |
| 3000 | 136.27 |  |  |  |  |  |  |

Для 2 процессов:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | Среднее |
| 100 | 0.0044 | 0.01492 | 0.004694 | 0.00506 | 0.00524485 | 0.0046 | 0.00649 |
| 500 | 0.5112 | 0.50449 | 0.495783 | 0.50587 | 0.505992 | 0.5043 | 0.5046 |
| 1000 | 27.645 | 22.6307 | 8.1134 | 13.1221 | 23.5476 | 13.378 | 18.0728 |
| 2000 | 40.421 |  |  |  |  |  |  |
| 3000 |  |  |  |  |  |  |  |

Для 3 процессов:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | Среднее |
| 100 | 0.0039 |  |  |  |  |  |  |
| 500 |  |  |  |  |  |  |  |
| 1000 | 7.10117 | 6.99107 | 4.12922 | 3.94199 | 3.98016 |  |  |
| 2000 | 40.421 |  |  |  |  |  |  |
| 3000 |  |  |  |  |  |  |  |

Для 4 процессов:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | Среднее |
| 100 | 0.0032187 | 0.0031853 |  |  |  |  |  |
| 500 |  |  |  |  |  |  |  |
| 1000 | 3.75584 | 3.66515 | 3.68017 |  |  |  |  |
| 2000 | 40.421 |  |  |  |  |  |  |
| 3000 |  |  |  |  |  |  |  |

Для 5 процессов:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | Среднее |
| 100 |  |  |  |  |  |  |  |
| 500 |  |  |  |  |  |  |  |
| 1000 |  |  |  |  |  |  |  |
| 2000 | 40.421 |  |  |  |  |  |  |
| 3000 |  |  |  |  |  |  |  |

# Выводы

Заключение

