МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

Направление подготовки: Фундаментальная информатика и информационные технологии

Магистерская программа: Инженерия программного обеспечения

**ОТЧЕТ**

по лабораторной работе №2

**Метод бисопряженных градиентов**

Допущена к защите  **Выполнил:**

студент группы 381706-1м

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шептунов В. О.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись подпись

**Проверил:**

к.ф-м.н., доцент кафедры МОСТ Баркалов К. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Рецензент:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ученая степень, ученое звание

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород  
2018

**Содержание**

[Введение 3](#_Toc513554305)

[Цель работы 4](#_Toc513554306)

[1. Постановка задачи 5](#_Toc513554307)

[2. Описание метода бисопряженных градиентов 6](#_Toc513554308)

[2.1 Последовательная версия метода 6](#_Toc513554309)

[2.2 Параллельная версия метода 7](#_Toc513554310)

[3. Программная реализация метода бисопряженных градиентов с использованием технологии OpenMP 8](#_Toc513554311)

[4. Результаты экспериментов 11](#_Toc513554312)

[Заключение 13](#_Toc513554313)

[Литература 14](#_Toc513554314)

Введение

Решение систем линейных алгебраических уравнений (СЛАУ) вида - одна из классических задач линейной алгебры. СЛАУ называется разряженной, а также совместной, если матрица состоит преимущественно из нулевых элементов и ее определитель отличен от нуля. Методы решения таких систем разделяют на прямые(точные) и итерационные. Оба подхода имеют свои достоинства и недостатки. Например, при использовании прямых методов возникает проблема заполнения матрицы системы при выполнении разложения, что может приводить к неудовлетворительным затратам памяти. При использовании итерационных методов, не приводящих к заполнению матрицы, можно в ряде случаев наблюдать достаточно медленную сходимость к решению. На практике выбор метода во многом зависит от решаемой задачи.

Данной лабораторная работа посвящена итерационному методу решения СЛАУ, суть которого заключается в построении последовательности приближений к точному решению . А именно методу бисопряженных градиентов, являющимся обобщением метода сопряженных градиентов на случай СЛАУ с произвольной разряженной, невырожденной матрицей .

Основной акцент работы сделан на реализацию, анализ и сравнение последовательной и параллельной версий метода. Применение технологии OpenMP в задаче параллельной реализации метода на языке программирования С++.

Цель работы

Требуется:

1. Изучить метод бисопряженных градиентов.
2. Реализовать последовательный метод бисопряженных градиентов для разряженных матриц.
3. Реализовать параллельный метод бисопряженных градиентов для разряженных матриц с использованием технологии OpenMP.
4. Провести эксперименты с использованием реализованных методов с последующим получением временных результатов исполнения и их анализом. Сделать необходимые выводы.
5. Постановка задачи

Исходные данные:

Разряженная, невырожденная матрица (элементы ). плотный вектор . В отличие от метода сопряженных градиентов, матрица может быть несимметричной.

Результат:

Плотный вектор . Количество выполненных итераций -

Требуется:

Методом бисопряженных градиентов решить СЛАУ вида:

(1)

1. Описание метода бисопряженных градиентов

Метод бисопряжённых градиентов для СЛАУ вида (1) генерирует последовательность векторов x(s)∈Rm, s=0,1,2,…, где x(s) − приближенное решение системы.

Известно, что матрица – симметричная и положительно определенная. Следовательно, можно перейти к решению новой системы

2.1 Последовательная версия метода

Подготовка перед итерационным процессом:

1. Выбрать начальное приближение , откуда начнется итерационный процесс.

Далее последовательно, итеративно находится приближение к искомому решению. На -ой итерации:

Критерием остановки итерационного процесса могут служить количество итераций *,* невязка или разница приближений.

2.2 Параллельная версия метода

Основной прирост производительности параллельной версии алгоритма дает использование параллельных операций вычисления скалярного произведения, суммы векторов, а также произведения матрицы на вектор. Каждая из операций вычисления скалярного произведения и суммы векторов, выполняемая на  процессах, за счет сдваивания может ускориться до *.* Ускорение же умножения матрицы на вектор может достигать  раз.

1. Программная реализация метода бисопряженных градиентов с использованием технологии OpenMP

Для достижения поставленных целей работы было реализовано консольное приложение на языке С++. Код программы содержит следующий набор функций:

1. *generate\_crs\_matrix*

Алгоритм генерации разряженной невырожденной матрицы в формате СRS (Column Row Storage):

typedef struct{

int n; // Число строк в матрице

int m; // Число столбцов в матрице

int nz; // Число ненулевых элементов в разреженной матрице

std::vector<double> val; // Массив значений матрицы по строкам

std::vector<int> colIndex; // Массив номеров столбцов

std::vector<int> rowPtr; // Массив индексов начала строк

}CRSMatrix;

Не нулевые элементы в количестве nz матрицы заполняется случайными числами из диапазона [-range, range], где range, nz, - параметры данной функции. В данной реализации n = m. Т.е. матрица квадратная.

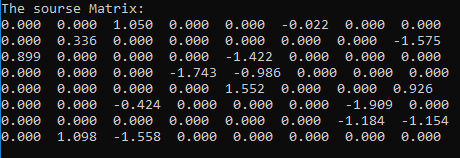


Рис. 1 Пример генерации невырожденной разряженной матрицы

1. *SLE\_Solver\_CRS\_BICG*

Непосредственно метод бисопряженных градиентов. Использует следующий набор вспомогательных функций:

* + *transpose\_CRSMatrix* – функция, реализующая транспонирование матрицы хранящаяся в CRS формате;
  + *mul\_vector\_on\_vector –* функция*,* вычисляющая скалярное произведение двух векторов;
  + *mul\_CRSMatrix\_on\_vector* – функция, вычисляющая произведение вектора на хранящуюся в CRS формате матрицу ;

Для параллельной реализации алгоритма, в вышеуказанных функциях были

использованы следующие директивы *OpenMP:*

* + *#pragma omp parallel for* - указывает что цикл, следующий за данной директивой, следует разделить по итерациям между потоками.
  + *#pragma omp parallel for reduction(+:sum)* указывает что цикл, следующий за данной директивой, следует разделить по итерациям между потоками и выполнить редукцию (сложение) относительно переменной *sum*.
  + *#pragma omp parallel for firstprivate* указывает о необходимости установить значение индивидуальной переменной, равное значению глобальной переменной.

А также условие *if* (PARALLEL), где макрос PARALLEL – указывает в каком случае применять директивы *#pragma omp*

1. *check\_result*

Проверка корректности работы алгоритма.

А именно, проверка равенства с заданным в качестве параметра .

1. *print\_result\_experiment\_to\_file*

Вывод результатов экспериментов в csv файл

При запуске приложения на экран выводятся временные результаты последовательной и параллельной версии метода бисопряженных градиентов, ускорение () и эффективность ().

, , где

– количество *OpenMP* потоков,

– время исполнения параллельного алгоритма с использованием потоков,

– время исполнения последовательного алгоритма с использованием одного

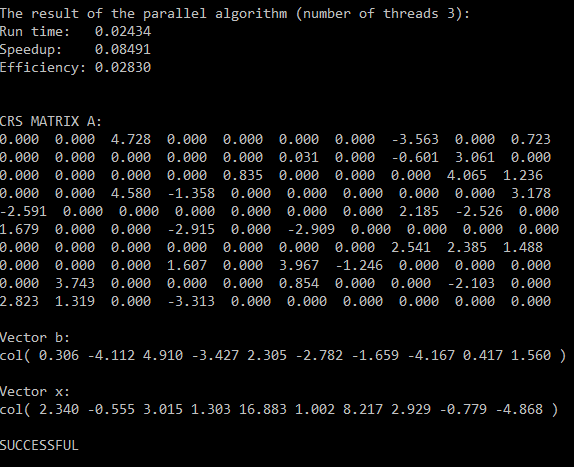


Рис. 2 Пример работы программы

1. Результаты экспериментов

В данной работе была проведена серия экспериментов для выявления зависимости ускорения и эффективности параллельного алгоритма от количества потоков.

Эксперименты проводились на: 4х ядерном процессоре Intel® Core™ i5-5200U CPU @2.2GHz с уровнями кэша: на матрицах {250 x 250, 500 x 500, 1000 x 1000, 1500 x 1500, 2000 x 2000, 2500 x 2500, 3000 x 3000}.

Рис. 3 Зависимость ускорения от числа потоков

Рис. 4 Зависимость эффективности от числа потоков

Заключение

В данной лабораторной работе был рассмотрен и реализован метод бисопряженных градиентов, который используется для задач нахождения численного решения СЛАУ.

Для подтверждения эффективности параллельного алгоритма над его последовательной версией была проведена серия экспериментов. Из полученных результатов можно заметить, что при небольших размерах матрицы параллельная версия метода бисопряженных градиентов уступает последовательной. Параллельная версия становится более эффективной с увеличением размера матрицы. Также при увеличении размера наблюдается тенденция к ускорению, на тестовой системе программа показала хорошее ускорение на большом объеме данных, почти в четыре раза на четырех *OpenMP* потоках, что говорит об эффективной параллельной реализации. На локальной машине результаты работы параллельной версии алгоритма не сильно отличаются от последовательной, однако наблюдается увеличение ускорения при увеличении числа потоков.

Литература

1. Баркалов К.А. Образовательный комплекс «Параллельные численные методы». Лекционные материалы. – Н.Новгород, Изд-во ННГУ, 2011.
2. Белов С.А., Золотых Н.Ю. Численные методы линейной алгебры. – Н.Новгород, Изд-во ННГУ, 2005.
3. Гергель В. П. Теория и практика параллельных вычислений. – М.: БИНОМ, 2007.