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 №1

**Метод Холецкого**

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](#_Toc513044797)

[Цель работы 4](#_Toc513044798)

[1. Постановка задачи 5](#_Toc513044799)

[2. Описание метода Холецкого 6](#_Toc513044800)

[2.1 Тривиальный алгоритм 6](#_Toc513044801)

[2.1.1 Последовательный алгоритм 6](#_Toc513044802)

[2.1.2 Параллельный алгоритм 8](#_Toc513044803)

[2.2 Блочный алгоритм 9](#_Toc513044804)

[2.2.1 Последовательный алгоритм 9](#_Toc513044805)

[2.2.2 Параллельный алгоритм 12](#_Toc513044806)

[3. Программная реализация разложения Холецкого с использованием технологии OpenMP 13](#_Toc513044807)

[4. Результаты экспериментов 16](#_Toc513044808)

[Заключение 18](#_Toc513044809)

[Литература 19](#_Toc513044810)

Введение

Метод Холецкого (метод квадратных корней) — алгоритм, предложенный французским офицером и математиком Андре-Луи Холецким в конце Первой Мировой войны незадолго до гибели в августе 1918 г. Впервые опубликован 1924 г. его сослуживцем.

Суть метода заключается в представлении симметричной положительно-определённой матрицы в виде , где — нижняя треугольная матрица со строго положительными элементами на диагонали. Такое разложение всегда существует и единственно для любой симметричной положительно-определённой матрицы.

Симметричность матрицы позволяет хранить и вычислять только чуть больше половины ее элементов, что почти вдвое экономит как необходимые для вычислений объемы памяти, так и количество операций в сравнении, например, с разложением по методу Гаусса. При этом альтернативное LU - разложение, использующее симметрию матрицы, все же несколько быстрее разложения Холецкого (нет операций извлечения квадратных корней), но требует больше памяти, ограниченность которой оказалась решающей для вычислителей времен 50-х гг. XX в.

Разложение Холецкого дает возможность заменить сложную задачу решения системы уравнений с полностью заполненной матрицей двумя простыми задачами – решение двух систем с треугольной матрицей.

Работа посвящена реализации данного метода. Основной акцент сделан на модификацию тривиального алгоритма с использованием параллельных вычислений.

Цель работы

Требуется:

1. Реализовать тривиальный алгоритм разложения Холецкого.
2. Реализовать блочный алгоритм разложения Холецкого (модификация).
3. Реализовать параллельную версию блочного алгоритма с использованием технологии OpenMP.
4. Провести эксперименты с использованием реализованных алгоритмов с последующим получением временных результатов исполнения и их анализом.
5. Постановка задачи

Исходные данные:

Положительно-определенная симметричная матрица (элементы ).

Матрицы с такими свойствами возникают, например, при использовании метода наименьших квадратов и численном решении дифференциальных уравнений.

Требуется:

Применить алгоритм разложения Холецкого к исходным данным с целью получить нижнюю треугольную матрицу (элементы ), такую, что

1. Описание метода Холецкого

В основе метода лежит алгоритм специального -разложения матрицы (), в результате чего она приводится к виду , где () – нижняя треугольная матрица со строго положительными элементами на диагонали.

Согласно этому утверждению при решении систем уравнений вида , матрицу можно заменить разложением . Тогда система примет вид .

Если обозначить , то можно решить задачу в 2 этапа:

1. Из определить ;
2. Из определить .

Обе эти системы с треугольными матрицами и, следовательно, легко решаются. То есть разложение Холецкого дает возможность заменить сложную задачу решения системы уравнений с полностью заполненной матрицей двумя простыми задачами – решение двух систем с треугольной матрицей.

2.1 Тривиальный алгоритм

2.1.1 Последовательный алгоритм

Элемент есть произведение *i*-й строки матрицы на *j-*й столбец матрицы :

Матрица симметричная, значит стоит рассмотреть лишь случай когда . Так как при , данные условия можно записать в виде:

При получаем:

Откуда:

Далее, при получаем:

Если предположить, что метод квадратного корня применяется к системе уравнений с действительной симметричной положительно-определенной матрицей, то элементы матрицы можно вычислить, начиная с ее левого угла, по следующим расчетным формулам:

Число операций, требующихся для выполнения разложения с помощью вышеуказанных формул:

Следует отметить, что в дополнение к указанным действиям для расчетов по данным формулам потребуется операций извлечения корня.

Если матрица факторизована в виде , то обратный ход метода квадратного корня состоит в последовательном решении двух систем уравнений:

Решения этих систем находятся по формулам, аналогичным формулам обратного хода метода Гаусса:

Вычисления по вышеуказанным формулам потребуют операций, что не оказывает существенного влияния на кубическую трудоемкость метода.

Общее время работы метода (где – время выполнения одной операции):

При больших время работы метода Холецкого будет примерно в два раза меньше, чем у метода Гаусса, это сокращение объясняется тем, что – симметричная матрица, и метод Холецкого учитывает данную особенность задачи.

2.1.2 Параллельный алгоритм

Все вычисления данного алгоритма сводятся к однотипным вычислительным операциям над строками матрицы . В основу параллельной реализации разложения может быть положен принцип распараллеливания по данным. В качестве базовой подзадачи можно принять все вычисления, связанные с обработкой одной строки матрицы .

Рассмотрим общую схему параллельных вычислений и возникающие при этом информационные зависимости между базовыми подзадачами.

При выполнении прямого хода метода необходимо осуществить итерацию для получения нижней треугольной матрицы . Выполнение итерации , , включает ряд последовательных действий. Прежде всего, в самом начале итерации необходимо вычислить диагональный элемент . Зная диагональный элемент, подзадачи выполняют вычисление -го столбца матрицы .

При выполнении обратного хода метода Холецкого подзадачи выполняют необходимые вычисления для нахождения значений сначала вспомогательного вектора , а затем вектора неизвестных . Как только какая-либо подзадача , , определяет значение своей переменной (или ), это значение должно быть использовано всеми подзадачами с номерами , где при решении системы относительно , и при решении системы относительно ; подзадачи подставляют полученное значение новой неизвестной и выполняют корректировку значений для элементов вектора правой части.

При использовании систем с общей памятью размер матрицы, описывающей систему линейных уравнений, является большим, чем число потоков (т.е., ). Следовательно, базовые подзадачи нужно укрупнить, объединив в рамках одной подзадачи несколько строк матрицы. Здесь также можно использовать циклическую схему распределения данных, аналогично методу Гаусса.

2.2 Блочный алгоритм

2.2.1 Последовательный алгоритм

Опишем вычислительную процедуру модифицированного алгоритма, основанную на идее разбиения матрицы на блоки и ориентированную на эффективную работу с кэш-памятью. Разложение осуществляется путем переписывания исходной матрицы элементами искомого фактора Холецкого сверху вниз по блокам.

Первый шаг блочного алгоритма заключается в следующем. Пусть размер блока равен , тогда исходную матрицу можно представить в виде:

где

– подматрица матрицы размера ,

– размера ,

–размера .

Искомый фактор Холецкого также можно записать в блочном виде как:

где , , – соответствующего размера подматрицы фактора . Рассмотрим теперь связь между блоками фактора и исходной матрицы, запишем в виде:

откуда получаем:

Используя данные матричные равенства, можно найти блоки , из искомого разложения. Блок может быть получен с помощь тривиального алгоритма, изложенного ранее. Далее, используя известный блок и найденный блок можно найти блок . Для этого потребуется решить вспомогательных систем изуравнений с одинаковой матрицей и разными правыми частями. Но так как матрица является треугольной, то решение одной системы потребует лишь операций. Всего же для нахождения блока потребуется операций с плавающей точкой.

Следующий шаг алгоритма состоит в вычислении редуцированной матрицы , при котором используется ставший известным блок , блок и составленное ранее соотношение:

Как следует из данной формулы, фактор Холецкого для матрицы совпадает с искомым блоком и для его нахождения можно применить описанный алгоритм рекурсивно. По построению матрица является симметричной положительно определенной, и при реализации алгоритма нет необходимости вычислять ее полностью, достаточно вычислить ее нижний треугольник.

Процесс блочной факторизации представлен далее:



Известно, что данная вычислительная процедура (как и другие возможные реализации метода Холецкого) включает в себя следующее число операций:

При этом вклад матричных операций в общее число действий (аналогично блочному -разложению) аппроксимируется величиной:

где .

Отсюда следует, что матричные операции, в частности, будут составлять большую часть вычислений, и к реализации матричного умножения нужно подойти с особой тщательностью. В данном случае для проведения операции матричного умножения целесообразно воспользоваться блочной схемой умножения матриц, выбрав для этого свой размер блока, меньший .

2.2.2 Параллельный алгоритм

После внимательного анализа блочного алгоритма можно заключить, что распараллеливание возможно для следующих вычислительных процедур:

* вычисление блока (параллельная версия тривиального алгоритма);
* вычисление блока (параллельное решение набора систем линейных уравнений с треугольной матрицей);
* выполнение матричного умножения;
* выполнение обратного хода.

В силу большей трудоемкости прямого хода алгоритма (т.е. собственно разложения) эффективность параллельного блочного алгоритма будет определяться эффективностью распараллеливания матричных операций, составляющих основную долю операция прямого хода.

1. Программная реализация разложения Холецкого с использованием технологии OpenMP

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

1. *gen\_square\_symmetric\_positive\_definite\_matrix*

Алгоритм генерации симметричной положительно-определенной матрицы . Основан на свойстве . Матрица заполняется случайными числами из диапазона [-range, range], где range - параметр данной функции. Затем для вычисляется транспонированная матрица . Симметричная положительно определенная матрица – результат произведения

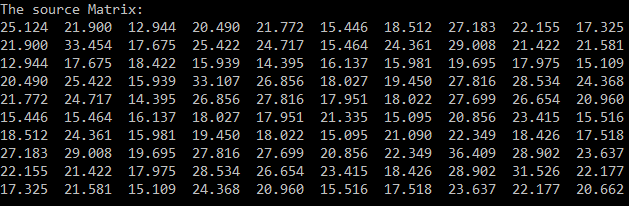


Рис. 1 Пример генерации симметричной положительно-определенной матрицы

1. *Сholesky\_Decomposition*

Алгоритм нахождения нижней треугольной матрицы . Использует следующий

набор вспомогательных функций:

* + *Cholesky\_Decomposition\_line* – функция, реализующая последовательный алгоритм. Используется также для вычисления блока ;
  + *Cholesky\_Decomposition\_Second\_Iteration–* функция*,* вычисляющая блок для блочного алгоритма;
  + *Cholesky\_Decomposition\_Third\_Iteration* – функция, вычисляющая редуцированную матрицу, где исходная и результирующая матрицы являются блоками;
  + *Cholesky\_Decomposition\_Recursive* – функция, реализующая параллельный алгоритм;
  + *matrix\_transposition\_block –* функция, вычисляющая транспонированную матрицу для прямоугольного блока матрицы;
  + *inverse\_matrix –* функция, вычисляющая обратную матрицу;
  + *determinant\_matrix –* функция, вычисляющая детерминант матрицы;
  + *adjoint\_matrix –* функция, вычисляющая алгебраические дополнения;
  + *get\_cofactor –* вспомогательная функция для вычисления алгебраических дополнений;
  + *matrix\_multiplication\_block\_res* – функция, вычисляющая произведение матриц, для которых результирующая матрица и левая матрица являются блоками;
  + *matrix\_subtraction\_block* – функция, вычисляющая разницу прямоугольных матриц.

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 sections* указывает, что блоки, помеченные директивами *#pragma omp section* могут быть выполнены параллельно.

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

1. *check\_result\_cholesky\_decomposition*

Проверка корректности работы алгоритма.

А именно, проверка равенства

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 Холецкого, ускорение () и эффективность ().

, , где

– количество *OpenMP* потоков,

– время исполнения параллельного алгоритма с использованием потоков,

– время исполнения последовательного алгоритма с использованием одного

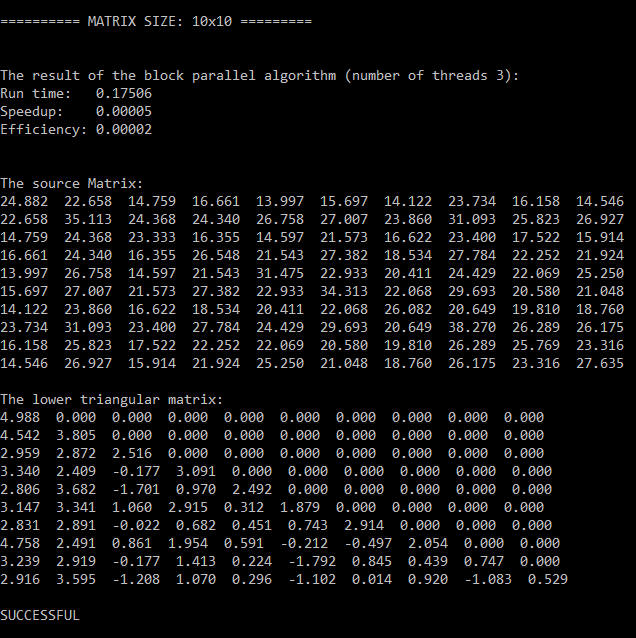


Рис. 2 Пример работы программы

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}. Оптимальный размер блока установлен экспериментальным путем.

Рис. 3 Зависимость ускорения от числа потоков

Рис. 4 Зависимость эффективности от числа потоков

Заключение

В данной лабораторной работе был рассмотрен метод разложения Холецкого, который используется для задач нахождения численного решения систем линейных алгебраических уравнений с симметричной положительно определенной матрицей. В рамках данной работы были реализованы последовательная и параллельная версия тривиального и блочного алгоритма разложения симметричной положительно-определенной матрицы методом Холецкого.

Для подтверждения эффективности параллельного блочного алгоритма над его последовательной версией была проведена серия экспериментов. Из полученных результатов можно заметить, что ускорение происходит на матрице с размером 2500x2500 и более, при чем наибольшее значение достигается при использовании всех четырех потоков.

При увеличении количества потоков эффективность использования *OpenMP* снижается из-за преобладания количества последовательного кода над параллельным.

Литература

1. Баркалов К.А. Образовательный комплекс «Параллельные численные методы». Лекционные материалы. – Н.Новгород, Изд-во ННГУ, 2011.
2. Воеводин В.В. Математические основы параллельных вычислений – М.: Изд. Моск. ун-та, 1991. 345 с.
3. Гергель В. П. Теория и практика параллельных вычислений. – М.: БИНОМ, 2007.
4. Разложение Холецкого - https://ru.wikipedia.org/wiki/Разложение\_Холецкого, 23 марта 2018.