УДК 004.4’422

АНАЛИЗ УСКОРЕНИЯ РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ С ТРЕУГОЛЬНОЙ ЛЕНТОЧНОЙ МАТРИЦЕЙ

Б.Я. Штейнберг
Южный федеральный университет
Россия, 344004, Ростов-на-Дону, Б. Садовая ул., 105
E-mail: borsteinb@mail.ru

В.А. Жилин
Южный федеральный университет
Россия, 344004, Ростов-на-Дону, Б. Садовая ул., 105
E-mail: vova.zhilin.03@bk.ru

Ключевые слова: параллельные вычисления, треугольная матрица, блочные вычисления, система уравнений, быстрые алгоритмы.

Аннотация: В данной работе описан алгоритм решения системы линейных уравнений с треугольной ленточной матрицей. К решению систем уравнений с такой матрицей сводятся многие задачи линейной алгебры. Алгоритм использует технологии параллельного программирования OpenMP на этапе преобразования исходной матрицы системы. Проведены численные эксперименты на компьютере с 8-ядерным процессором i7. Предлагаемый алгоритм демонстрирует ускорение для таких размерностей, при которых матрица не помещается в кэш-памяти процессора. Ускорение возрастает до 1.5 раза с ростом размерности матрицы. Представленный алгоритм может быть использован для разработки решателей пакетов прикладных программ.

1. Введение

Многие вычислительные задачи сводятся к решению систем линейных уравнений с матрицами специального вида. Часто в приложениях (например, при использовании метода конечных разностей) возникают ленточные матрицы, то есть матрицы, имеющие лишь несколько ненулевых диагоналей. Но при применении прямых методов решения систем уравнений с такими матрицами многие нулевые диагонали могут стать ненулевыми. Так происходит, например, при LU разложении или QR разложении матриц, а также при разложении симметричной положительно определенной матрицы методом Холецкого. В результате этих алгоритмов получаются треугольные матрицы, а если исходная матрица была ленточной, то получаются треугольные матрицы с ненулевой полосой возле главной диагонали. Разложение матриц в произведение треугольных особенно удобно при многократном решении систем линейных уравнений с одной и той же матрицей и различными правыми частями – такая задача возникает при итерационных методах решения систем линейных уравнений.

В данной работе исследуются алгоритмы решения систем уравнений с нижне-треугольной матрицей с ненулевой полосой возле главной диагонали. Верхне-треугольные матрицы могут быть рассмотрены аналогично.

Не умаляя общности, можем считать, что на главной диагонали стоят единицы.

В данной работе представлен быстрый алгоритм решения системы уравнений большой размерности с треугольной матицей, в которой ненулевые элементы образуют полосу (ленту) возле главной диагонали. Матрица хранится в памяти двумерным массивом, строки которого являются диагоналями полосы ненулевых элементов. Исходная система уравнений преобразуется к новой системе с блочно-двухдиагональной матрицей, у которой размер блока равен ширине полосы ненулевых элементов. Это преобразование выполняется параллельно. Представление матрицы в таком виде позволяет более быстро читать данные кэш-линейками. Для матриц большой размерности получено ускорение метода Гаусса в 1.5 раза. Ускорение возникает, когда матрица не помещается в кэш-памяти процессора. Результаты работы могут найти применение в пакетах прикладных программ.

2. Описание исходных данных

Будем рассматривать систему линейных уравнений с нижне-треугольной ленточной матрицей M размера n*n и вектором правой части b.

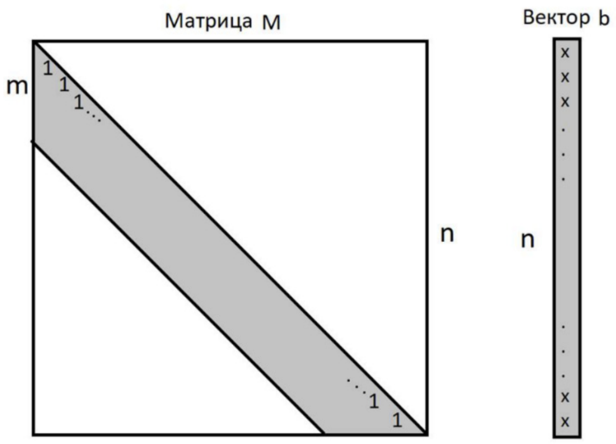

Рис. 1. Матрица исходной системы уравнений.

Все ненулевые элементы матрицы сосредоточены в m диагоналях расположенных рядом с главной. Будем полагать, что число n делится нацело на m. По главной диагонали единицы. В оперативной памяти матрица хранится двумерным массивом размерности m*n, по строкам которого расположены ненулевые диагонали (единицы главной диагонали в памяти не хранятся).

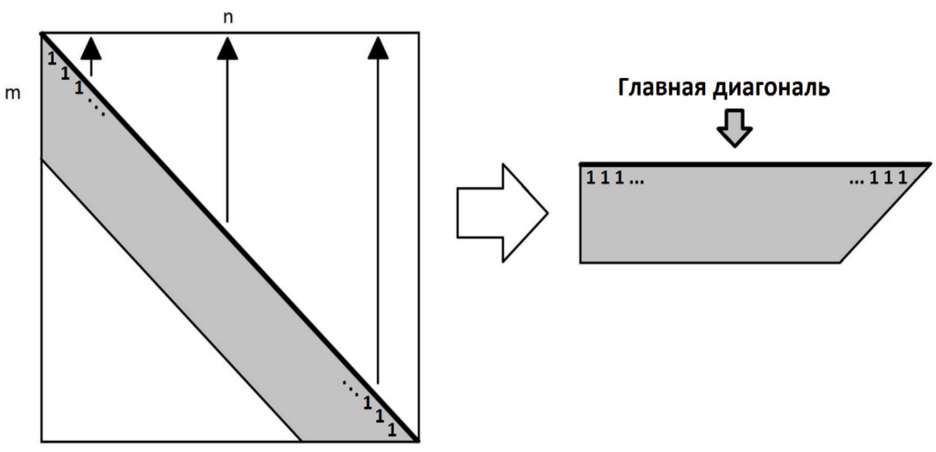

Рис. 2. Хранение ненулевых элементов матрицы двумерным массивом размерности m*n.

Исходную матрицу разобьем на блоки размера m*m. Блоки будут треугольными матрицами размера m*m, причем по главной диагонали блоки являются обратимыми нижне-треугольными матрицами с единицами по главной диагонали, а другие блоки – верхне-треугольные.

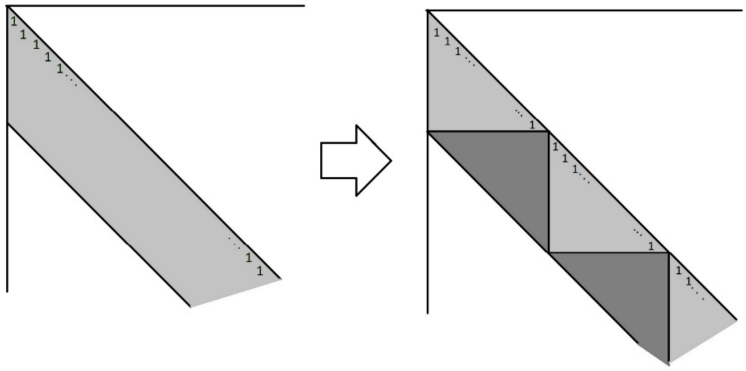

Рис. 3. Разбиение исходной матрицы на блоки размера m*m.

3. Преобразование исходной матрицы

Преобразуем исходную матрицу к блочно-двухдиагональному виду с единичными блоками по главной диагонали. Для этого, каждую блочную строку исходной системы уравнений умножим на блок, обратный к диагональному. По главной диагонали получатся единичные блоки, а на нулевой блочной диагонали будут квадратные (уже не треугольные) блоки размера m*m. Умножения блоков можно выполнять параллельно с использованием технологии OpenMP.

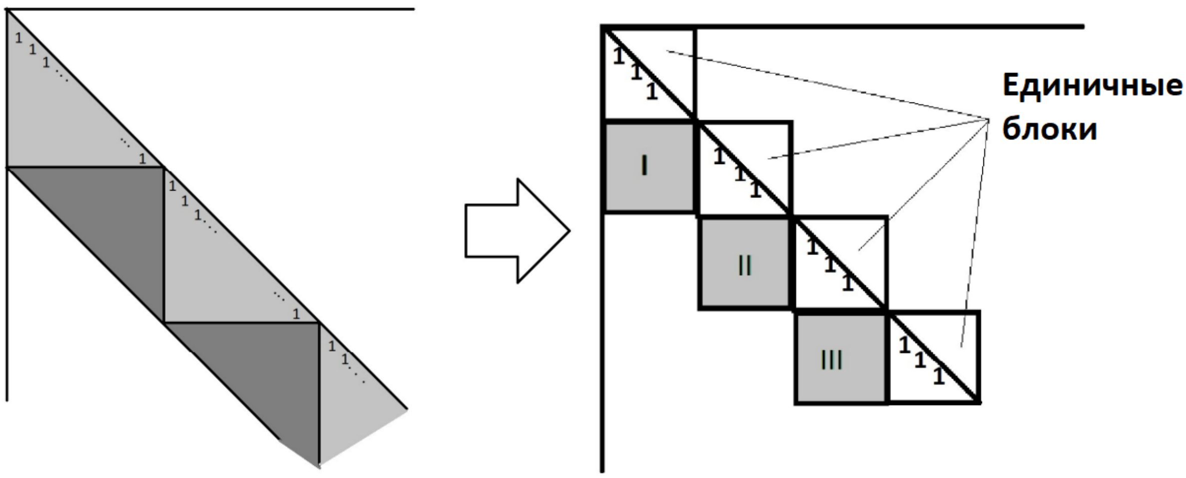

Рис. 4. Преобразование матрицы к блочной двухдиагональной с единичными блоками по главной диагонали.

Блоки полученной блочно-диагональной матрицы могут храниться в том же двумерном массиве m*n, в котором хранилась исходная матрица.

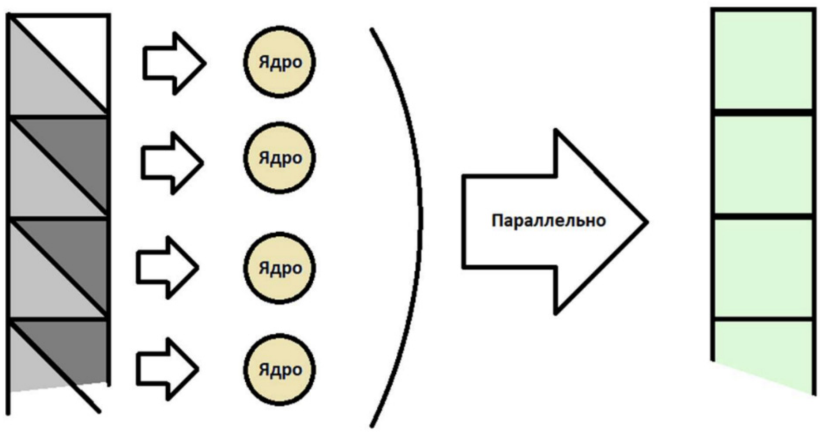

Рис. 5. Параллельное умножение блоков.

Решение системы уравнений с блочной двухдиагональной матрицей полученного вида можно записать рекуррентным алгоритмом, который при малых размерностях допускает ускорение при распараллеливании [2]. Ускорение в работе [2] достигается за счет того, что много вычислений выполняется в регистрах без обращения к оперативной памяти. Предполагается, что для рассматриваемой задачи этот алгоритм будет давать ускорение на процессорах типа «суперкомпьютер-на-кристалле» [3, 4], которые могут удерживать в локальной памяти каждого ядра относительно большое количество данных (блок матрицы).

4. Результаты численных экспериментов

Результаты численных экспериментов проводились на персональном компьютере с процессором: AMD Ryzen 7 5800HS, CPU Cores: 8, Threads: 16, Base Clock: 2.8GHz;

L1 Cache 32 KB per Core, L2 Cache: 512 KB per Core, L3 Cache: 16MB.

Компилятор Microsoft Visual C++ Compiler.

Для представленного в данной работе алгоритма время измерялось отдельно для каждого из двух этапов:

1) преобразование (параллельное) матрицы системы к блочной двухдиагональной,
2) решение системы уравнений с блочной двухдиагональной матрицей.

Таблица 1. Результаты численных экспериментов сравнения последовательного и параллельного алгоритмов при размерности блока 32 с элементами типа double.

Размерность матрицы	**Последовательный алгоритм для треугольной матрицы с ненулевой полосой, сек**	**Создание двухдиагональной матрицы, сек**	**Последовательный алгоритм для двухдиагональной матрицы, сек**	**Ускорение**
2 097 152	0,5336	0,5448	0.6067	0,464
4 194 304	1,0785	1.1156	1.1701	0,471
8 388 608	5,9287	2.0078	2.3212	1,370
16 777 216	41,6024	11.562	14.843	1,576

При увеличении используемого параллелизма (ка компьютере с большим количеством ядер) возможно ускорение первого этапа представленного алгоритма.

5. Заключение

Результаты численных экспериментов показывают, что представленный алгоритм дает ускорение на 8-ядерном процессоре Интел, когда матрица не помещается в L3-кэш. При этом, с ростом размерности матрицы растет и ускорение. Предполагается, что модификация данного алгоритма сможет давать ускорение и для меньших размерностей на процессорах типа «суперкомпьютер-на-кристалле».

Список литературы

1. Абрамян М.Э., Штейнберг Б.Я., Штейнберг О.Б. Технологии распараллеливания программ // Ростов-на-Дону: Изд-во Южного федерального университета, 2014. 148 с.
2. Штейнберг О.Б. Распараллеливание рекуррентных циклов с нерегулярным вычислением суперпозиций // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. 2020. Т. 13, № 3. С. 59-67.
3. Процессор НТЦ «Модуль». https://www.cnews.ru/news/top/2019-03-06_svet_uvidel_moshchnejshij_rossijskij_nejroprotsessor (дата обр. 26.03.2022).
4. SoC Esperanto. URL: https://www.esperanto.ai/technology/.