

Стреловидная декомпозиция для блочно-трехдиагональной СЛАУ *

П.А. Белов, Е.Р. Нугуманов, С.Л. Яковлев

Санкт-Петербургский Государственный Университет

Представлен метод стреловидной декомпозиции обеспечивающий эффективное параллельное решение блочно-трехдиагональной системы линейных алгебраических уравнений (СЛАУ). Вычислительное ускорение по отношению к методу матричной прогонки оценено аналитически путем учета числа элементарных операций умножения для параллельных и последовательных частей метода декомпозиции. Показано, что максимальное ускорение достигается при конечном числе параллельных процессоров. Для заданного размера исходной СЛАУ получены параметры вычислительной системы, при которых достигается максимальное ускорение. Вычислительные эксперименты подтверждают аналитические оценки вычислительного ускорения.

1. Введение

Многие задачи прикладной математики и вычислительной физики после дискретизации сводятся к системе линейных алгебраических уравнений (СЛАУ) с блочно-трехдиагональной матрицей. В качестве простейшего примера можно привести конечно-разностную аппроксимацию граничной задачи для двумерного уравнения Пуассона на равномерных сетках по каждой координате. В этом случае блоки матрицы будут разреженными. В действительности, матрица СЛАУ будет пятидиагональной, но с блочной структурой. Ленточные или заполненные блоки в такой задаче можно получить, если совместить конечно-разностную дискретизацию по одной координате с разложением по сплайнам или иным функциям по другой координате.

Сравнительно быстрым алгоритмом для решения блочно-трехдиагональных систем является метод матричной прогонки [1]. Данный метод обладает свойством устойчивости для блоков с диагональным преобладанием. Недостатком метода является его последовательность и трудность параллелизации на современных суперкомпьютерных и кластерных системах. Единственной возможностью параллелизации является параллелизация на уровне матричных операций умножения и обращения блоков.

Параллельные методы решения трехдиагональных СЛАУ разрабатывались с конца шестидесятых годов двадцатого века. К данным методам можно отнести широко известный метод циклической редукции [2], а также метод Вонга [3], который успешно применяется в современных расчетах [4]. Данные методы были обобщены для использования с ленточными и блочно-трехдиагональными матрицами [5]. Помимо перечисленных методов, для решения трехдиагональных СЛАУ был разработан более эффективный алгоритм, основанный на инженерном методе декомпозиции области. Применительно к СЛАУ, он заключается в разделении полной матрицы на независимые блоки и один связующий блок [6]. Сходные идеи используются в современных работах, например [7], а для ленточных матриц — приводятся в техническом отчете ScaLAPACK [8].

В данной работе мы обобщили метод декомпозиции на случай блочно-трехдиагональной СЛАУ. Метод сводится к преобразованию исходной блочно-трехдиагональной СЛАУ к эквивалентному виду со стреловидной формой новой матрицы путем перестановки блочных строк и столбцов исходной системы. В результате, матрица системы логически приводится к стреловидной форме, включающей крупные супер-блоки на диагонали, допускающие па-

*Работа поддержана грантом РФФИ № 14-02-00326⁴⁴⁷ и грантом СПбГУ в рамках проета № 11.38.241.2015. Все вычисления были проведены на оборудовании ресурсного центра “Вычислительный центр СПбГУ”.

параллельное обращение, разреженные супер-блоки вдоль правой и нижней сторон матрицы и одного связующего супер-блока в правом нижнем углу матрицы. Мы показываем, что данная стреловидная система допускает эффективное параллельное решение [9]. Для обращения супер-блоков используется метод матричной прогонки. Мы аналитически получаем вычислительное ускорение метода относительно метода матричной прогонки. Оно вычисляется из оценки числа мультипликативных операций на каждом шаге метода и сравнении его с аналогичными характеристиками метода матричной прогонки. Реализованный метод стреловидной декомпозиции позволил нам провести вычислительные эксперименты и подтвердить аналитические оценки на практике. В качестве тестовой СЛАУ использовалась дискретизация граничной задачи для двумерного интегро-дифференциального уравнения Фаддеева [10].

2. Метод стреловидной декомпозиции

В работе изучается блочно-трехдиагональная СЛАУ, описываемая уравнением

$$A_i X_{i-1} + C_i X_i + B_i X_{i+1} = F_i, \quad A_1 = B_N = 0, \quad (1)$$

где $A_i, C_i, B_i, i = 1, \dots, N$ блоки размера $n \times n$ матрицы системы. Элементы F_i , размера $n \times l$, где $l \geq 1$, в правой части – это блоки супервектора \vec{F} . Искомый супервектор \vec{X} состоит из блоков X_i . Полный размер матрицы $(nN) \times (nN)$.

На рисунке 1 показан пример преобразования исходной матрицы СЛАУ для $N = 15$ к эквивалентному виду со стреловидной формой новой матрицы. Преобразование СЛАУ осуществляется перестановкой блочных-строк и столбцов. Перестановка строк приводит к перестановке блоков в правой части системы, а перестановка столбцов – к перестановке блоков искомого решения. Полученная стреловидная СЛАУ может быть кратко записана в виде блочной матрицы 2×2

$$\begin{pmatrix} \mathbf{S} & \mathbf{W}_R \\ \mathbf{W}_L & \mathbf{H} \end{pmatrix} \begin{pmatrix} s \\ h \end{pmatrix} = \begin{pmatrix} F_s \\ F_h \end{pmatrix}. \quad (2)$$

Здесь \mathbf{S} состоит из независимых супер-блоков на диагонали $S^k, k = 1, \dots, M$. Остальные элементы обозначены на рисунке 1. Решение системы (2) дается формулами

$$\begin{cases} s = \mathbf{S}^{-1} F_s - \mathbf{S}^{-1} \mathbf{W}_R h \\ h = (\mathbf{H} - \mathbf{W}_L \mathbf{S}^{-1} \mathbf{W}_R)^{-1} (F_h - \mathbf{W}_L \mathbf{S}^{-1} F_s) \end{cases}. \quad (3)$$

Уравнение (3) в правой части содержит матричные умножения и обращения, которые, при детальном рассмотрении, обладают большой степенью параллелизма. Независимость супер-блоков S^k позволяет вычислять произведения $\mathbf{S}^{-1} F_s, \mathbf{S}^{-1} \mathbf{W}_R$ параллельно. Более того, вместо вычислений обратных матриц, достаточно просто решить набор СЛАУ значительно меньшего размера. Ввиду разреженной структуры супер-блоков $\mathbf{W}_L, \mathbf{W}_R$ количество умножений также многократно сокращается. В системе (3) сначала вычисляется решение $h = (h^1, \dots, h^{M-1})^T$, а потом оставшая часть s полного решения $\vec{X} = (s, h)^T$ находится параллельно по формуле

$$s^k = z^k - Z^k h^{k-1} - Z^k h^k, \quad k = 1, \dots, M, \quad (4)$$

где $h^0 = h^M = 0$.

В нашей статье, для решения системы с супер-блоком S^k и для вычисления части решения h мы используем метод матричной прогонки, хотя для этих операций можно использовать метод стреловидной декомпозиции вложенно.

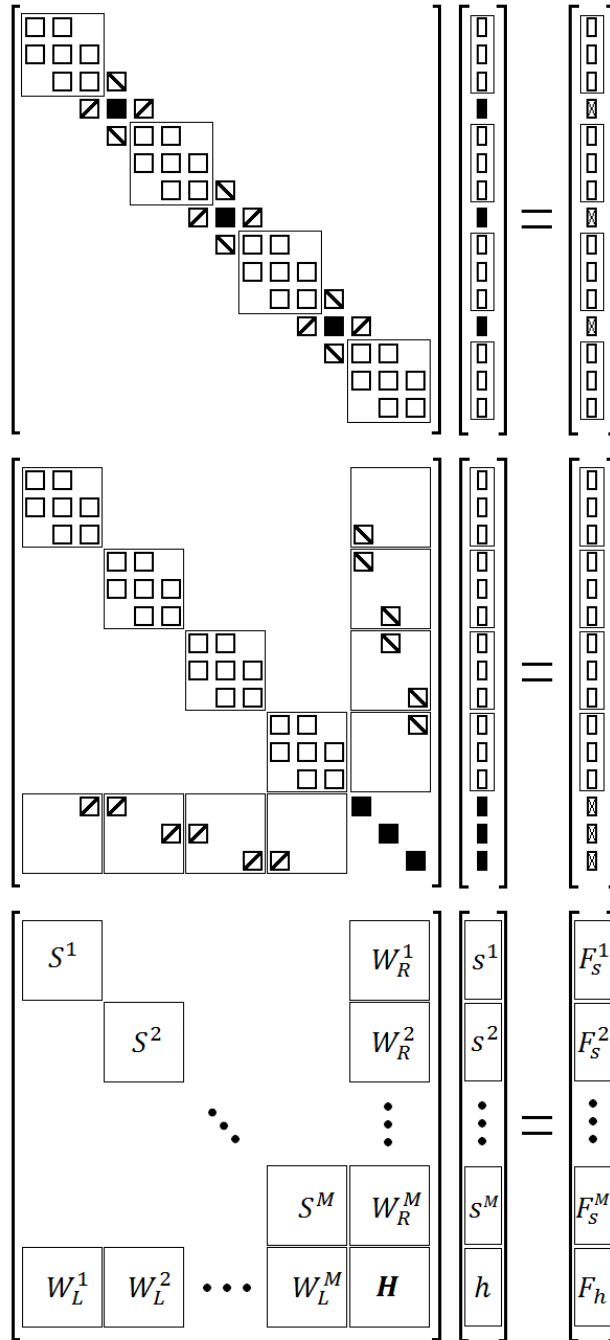


Рис. 1. Пример преобразования СЛАУ в методе стреловидной декомпозиции. Верхний рисунок: исходная СЛАУ с выделением блоков, которые будут переставлены. Средний рисунок: преобразованная СЛАУ с матрицей стреловидной формы. Нижний рисунок: обозначения элементов преобразованной СЛАУ с матрицей стреловидной формы, принятые в данной статье.

3. Вычислительное ускорение

Вычислительное ускорение метода стреловидной декомпозиции по отношению к последовательному методу матричной прогонки можно оценить как отношение времени вычисления решения СЛАУ методом прогонки к соответствующему времени метода декомпозиции. Время вычисления непосредственно связано с числом последовательных операций каждого метода. Мы сосчитали количество последовательных мультипликативных операций метода стреловидной декомпозиции и, как следствие, оценили вычислительное ускорение. Аддитивные операции не учитывались, считая, что они требуют значительно меньшего компьютерного времени.

В соответствии с [11], мы считаем, что как для умножения, так и для обращения блока размера $n \times n$ требуется ровно n^3 мультипликативных операций. Данная оценка согласуется с оценкой количества операций в общепринятой реализации данных алгоритмов [12], а именно LAPACK. Тогда количество мультипликативных операций для последовательного алгоритма матричной прогонки

$$O_{MS} = (3N - 2)(n^3 + n^2l).$$

Для метода декомпозиции полное количество операций умножения дается формулой

$$\begin{aligned} & \left((4N_1 - 1)n^3 + (4N_1 - 1)n^2l \right) + \left((6N_M - 3)n^3 + (4N_M - 1)n^2l \right) + \\ & + \sum_{k=2}^{M-1} \left(7N_k n^3 + 5N_k n^2l \right) + (3M - 5)(n^3 + n^2l), \end{aligned} \quad (5)$$

где $N_k, k = 1, \dots, M$ обозначает количество блоков на диагонали супер-блока S^k . В общем случае N_k могут быть различными, но удовлетворяют соотношению

$$\sum_{k=1}^M N_k = N - (M - 1).$$

Максимальное ускорение достигается, когда время работы каждого процессора по решению системы с супер-блоком S^k одинаково. В этом случае N_k равны друг другу при $k = 2, \dots, M - 1$, а N_1 и N_M несколько больше остальных. Выигрыш в производительности по сравнению со случаем, когда все $N_k, k = 1, \dots, M$ равны друг другу достаточно мал. Поэтому, для простоты, мы рассмотрим случай равенства всех N_k друг другу. В этом случае, если количество параллельных процессоров равно числу блоков M на диагонали, то максимальное число последовательных операций метода стреловидной декомпозиции будет

$$O_{AD} = ((N - M + 1) / M) (7n^3 + 5n^2l) + (3M - 5)(n^3 + n^2l)$$

Тогда вычислительное ускорение, как отношение $S = O_{MS} / O_{AD}$, легко получается:

$$S = \frac{3N - 2}{3M - 5 + \left(5 + \frac{2}{1+l/n} \right) \left(\frac{N-M+1}{M} \right)}. \quad (6)$$

Легко видеть, что зависимость вычислительного ускорения от отношения l/n достаточно слабая. Как функция M , ускорение ведет себя линейно $S = 3M/7$ при $M \ll N$. Графики вычислительного ускорения при $l = 1$ и различных N и n приведены на рисунке 2. При дальнейшем увеличении M , ускорение достигает своего максимума и потом убывает. Число параллельных процессоров, при котором вычислительное ускорение достигает своего максимума дается формулой

$$M^* = \sqrt{\frac{(N + 1)450}{3} \left(5 + \frac{2}{1 + l/n} \right)}. \quad (7)$$

Таким образом, при использовании метода стреловидной декомпозиции все супер-блоки на диагонали можно выбирать одинакового размера, т.к. уменьшение ускорения в этом случае незначительно. Оптимальное количество параллельных процессоров M^* для фиксированного количества блоков N на диагонали СЛАУ дается формулой (7).

4. Практические результаты

Аналитические оценки, приведенные в предыдущей главе, были проверены в практических расчетах. Метод стреловидной декомпозиции был реализован на языке C/C++ с использованием вызовов подпрограмм для работы с матрицами библиотеки LAPACK 3.5.0. Параллелизм был реализован с помощью технологии OpenMP. Численные эксперименты проводились на вычислительной системе с общей памятью (SMP) и 64 процессорами Intel Xeon.

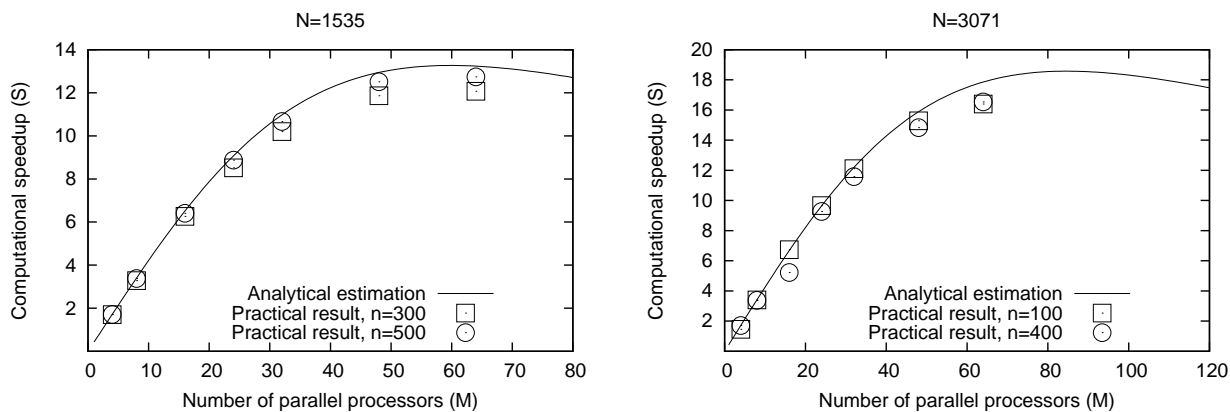


Рис. 2. Аналитическая оценка вычислительного ускорения метода стреловидной декомпозиции по отношению к методу матричной прогонки (сплошная кривая) и практические результаты измерения ускорения (квадратики и кружочки) в зависимости от числа параллельных процессоров. Размеры блоков $n = 100 - 500$, число блоков на диагонали $N = 1535$ (левый график), $N = 3071$ (правый график), используется один вектор в правой части.

В качестве тестовой задачи решалась СЛАУ, полученная из дискретизации граничной задачи для интегро-дифференциального уравнения Фаддеева [13]. Для чистоты эксперимента, нулевые элементы блоков заменялись относительно малыми случайными числами. Правая часть состояла из одного вектора, т.е. $l = 1$.

Полученные результаты для $N = 1535, 3071$ и $n = 100 - 500$ показаны на рисунке 2. Видно, что практические результаты хорошо согласуются с аналитическими оценками. При $M \ll N$ наблюдается линейное ускорение и общий тренд полученного вычислительного ускорения также соответствует аналитическим результатам. Небольшие отклонения могут быть вызваны неидеальностью работы с памятью, а также системными процессами, выполняющимися параллельно. Последнее больше всего проявляется при $M = 64$. Тем не менее, аналитические оценки для данного случая в целом верны и слабо зависят от n .

Метод стреловидной декомпозиции использовался для численного решения блочно-трехдиагональной СЛАУ, полученной из граничной задачи в полярных координатах $\{\rho, \theta\}$ для интегро-дифференциального уравнения Фаддеева. Блочно-трехдиагональная матрица возникает из конечно-разностной аппроксимации второй производной по переменной ρ и разложения искомого решения граничной задачи по базису Эрмитовых кубических сплайнов по переменной θ [13]. Из решения граничной задачи определялись амплитуды рассеяния, соответствующие задаче нейтрон-дейтронного рассеяния. Применение данного метода позволило уменьшить общее время решения, включающее формирование СЛАУ, решение

системы, получение амплитуд рассеяния, в 7-10 раз.

5. Заключение

В работе представлен метод стреловидной декомпозиции для блочно-трехдиагональной СЛАУ. Показано, что данный метод допускает эффективное параллельное решение СЛАУ данного типа. Аналитически получено вычислительное ускорение метода по отношению к методу матричной прогонки. Реализованный метод стреловидной декомпозиции позволил нам провести вычислительные эксперименты и подтвердить аналитические оценки на практике. Использование метода стреловидной декомпозиции позволило уменьшить время решения СЛАУ в несколько десятков раз, что привело к сокращению общего времени вычисления искомых величин прикладной задачи в семь-десять раз.

Литература

1. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений, М.: Наука, 1978.
2. Hockney R.W. The potential calculation and some applications // Meth. Comput. Phys. 1970. Vol. 9. P. 136–211.
3. Wang H.H. A parallel method for tridiagonal equations // ACM Trans. Math. Software. 1981. Vol. 7. P. 170–183.
4. Волохова А.В., Земляная Е.В., Рихвицкий В.С. Параллельная оптимизация метода решения системы уравнений полярона с использованием алгоритма разбиений // Вычислительные методы и программирование. 2015. Т. 16. С. 281–289.
5. Johnsson S.L. Solving narrow banded systems on ensemble architectures // ACM Trans. Math. Software. 1985. Vol. 11. P. 271–288.
6. Ortega J.M. Introduction to parallel and vector solution of linear systems, New York: Springer, 1988.
7. Акимова Е.Н., Белоусов Д.В., Мисиллов В.Е. Алгоритмы решения обратных геофизических задач на многопроцессорных вычислительных системах // Сиб. журн. вычисл. матем. 2013. Т. 16. С. 107–121.
8. Cleary A., Dongarra J.J. Implementation in ScaLAPACK of divide-and-conquer algorithms for banded and tridiagonal linear systems // Report No. UT-CS-97-358 (LAWN No. 125), University of Tennessee, 1997.
9. Belov P.A., Nugumanov E.R., Yakovlev S.L. Computing binary scattering and breakup in three-body system // Nuclear Theory in the Supercomputing Era, 2012, Khabarovsk, Russia. Proceedings. Pacific National University. 2013. P. 121–134
10. Faddeev L.D., Merkuriev S.P. Quantum scattering theory for several particle systems, Dordrecht: Springer, 1993.
11. Самарский А.А., Гулин А.В. Численные методы, М.: Наука, 1989.
12. Dongarra J.J., et al. LINPACK Users' Guide, Philadelphia:SIAM, 1979.
13. Belov P.A., Yakovlev S.L. Asymptotic method for determining the amplitude for three-particle breakup: Neutron-deuteron scattering // Phys. Atom. Nucl. 2013. Vol. 76. P. 126–138.

Arrowhead decomposition for a block-tridiagonal system of linear equations

Pavel Belov, Eduard Nugumanov and Sergey Yakovlev

Keywords: arrowhead decomposition method, block-tridiagonal matrix, system of linear equations, parallel solution, computational speedup, matrix sweeping algorithm

The arrowhead decomposition method which allows efficient parallel solution of a block-tridiagonal system of linear equations is presented. The computational speedup with respect to the matrix sweeping algorithm is analytically estimated by taking into account the number of elementary operations of multiplication for the parallel and serial parts of the decomposition method. It is shown that the maximal speedup is achieved for the finite number of parallel processors. For a given size of the initial system of linear equations, the parameters of the computational system which give the maximal speedup are obtained. Computational experiments confirm the analytical estimations of the computational speedup.