УДК 519.688 **DOI: 10.18413/2518-1092-2025-10-3-0-7**

Лихошерстный А.Ю.[1]
Великая Я.Г.[2]

СТРУННО-ВОЛНОВОЙ ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ СЛАУ ДЛЯ РАЗРЕЖЕННЫХ МАТРИЦ

[1)] Научно-технический центр «АПМ»,
Октябрьский б-р, 14, Московская область, г. Королев, 141070, Россия
[2)] Белгородский государственный национальный исследовательский университет,
ул. Победы, 85, г. Белгород, 308015, Россия

e-mail: ozzy.osbourne.man@gmail.com, velikaya@bsuedu.ru

Аннотация
В статье рассматривается параллельный алгоритм решения систем линейных алгебраических уравнений для симметричных разреженных матриц, который позволяет разбивать большую задачу на множество мелких подзадач, тем самым как повышая быстродействие, так и уменьшая потребление памяти. В его основе лежит способ одновременного вычисления промежуточных значений при разложении матрицы с сохранением балансировки нагрузки на процессоры таким образом, чтобы при получении окончательного результата левых частей разложения от них не зависели правые части разложения. Такой подход позволяет исходную матрицу жесткости представить в виде произведения большого количества простых матриц и решать систему линейных алгебраических уравнений в виде последовательности решений методом подстановки. Для уменьшения заполнения разреженных матриц разложения использовался приближенный метод минимальной степени, который помимо того, что является одним из самых эффективных и быстродействующих из существующих на сегодняшний момент времени, позволяет для разработанного алгоритма более равномерно распределить нагрузку вычислений. Разработанный метод реализован в программных продуктах НТЦ АПМ для систем с общей памятью, но также может быть реализован и для систем с распределенной памятью.
Ключевые слова: алгоритм минимальной степени; струнно-волновой алгоритм; метод конечных элементов; система линейных алгебраических уравнений; матричная факторизация
Для цитирования: Лихошерстный А.Ю., Великая Я.Г. Струнно-волновой параллельный алгоритм решения СЛАУ для разреженных матриц // Научный результат. Информационные технологии. – Т.10, №3, 2025. – С. 72-79. DOI: 10.18413/2518-1092-2025-10-3-0-7

Likhosherstnyy A.Yu.[1]
Velikaya Ya.G.[2]

STRING-WAVE DIRECT PARALLEL SOLVER FOR SPARSE SYSTEM OF LINEAR EQUATIONS

[1)] APM Ltd.,
14 Oktyabrsky Boulevard, Moscow region, Korolev, 141070, Russia
[2)] Belgorod State National Research University,
85 Pobedy St., Belgorod, 308015, Russia

e-mail: ozzy.osbourne.man@gmail.com, velikaya@bsuedu.ru

Abstract
The article discusses a parallel algorithm for solving systems of linear algebraic equations for symmetric sparse matrices, which allows you to split a large task into many small subtasks, thereby both increasing performance and reducing memory consumption. It is based on a method of simultaneous calculation of intermediate values during matrix decomposition while maintaining load balancing on processors so that when the final result of the left parts of the decomposition is

obtained, the right parts of the decomposition do not depend on them. This approach allows the initial stiffness matrix to be represented as a product of a large number of simple matrices and solve a system of linear algebraic equations in the form of a sequence of solutions by substitution. To reduce the filling of sparse decomposition matrices, an approximate minimum degree method was used, which, in addition to being one of the most efficient and fastest existing at the moment, allows the developed algorithm to distribute the load of calculations more evenly. The developed method is implemented in NTC APM software products for systems with shared memory, but it can also be implemented for systems with distributed memory.

Keywords: approximate minimum degree; string-wave algorithm; finite element method; system of linear equations; matrix factorization

For citation: Likhosherstnyy A.Yu., Velikaya Ya.G. String-Wave Direct Parallel Solver for Sparse System of Linear Equations // Research result. Information technologies. – T.10, №3, 2025. – P. 72-79. DOI: 10.18413/2518-1092-2025-10-3-0-7

ВВЕДЕНИЕ

В настоящее время для инженерных расчетов в области строительства и машиностроения активно используется метод конечных элементов (МКЭ) [1]. Применение МКЭ предполагает решение систем линейных алгебраических уравнений (СЛАУ) большой размерности. Матрицы таких систем, называемые матрицами жёсткости, обладают рядом свойств, зависящих от структуры и количества узлов конечно-элементной сетки, а также характера задачи. Эти матрицы являются симметричными и в значительной степени разреженными.

С помощью СЛАУ решаются такие задачи как: статический расчет, динамический расчет, расчет собственных частот, расчет устойчивости, нелинейные расчеты и др. В общем случае СЛАУ [2] имеет следующий вид:

$$K_{n,n}X_{n,m} = F_{n,m}, \tag{1}$$

где

K – матрица жесткости;

X – векторы перемещений;

F – векторы нагрузки для системы конечных элементов;

n – размерность матрицы жесткости;

m – количество загружений.

Ввиду того, что матрица жесткости является разреженной, она представлена в строчном разреженном формате CSR (Compressed Sparse Rows) [3, 4], где структура хранения использует три одномерных массива:

1) массив ненулевых элементов матрицы жесткости построчно;

2) массив номеров столбцов ненулевых элементов построчно;

3) массив местоположения первого ненулевого элемента в каждой строке.

Такой формат позволяет существенно сократить объем исходных данных и уменьшить количество выполняемых над ними операций, т.е. при расчетах сокращается расход памяти и увеличивается быстродействие.

ОСНОВНАЯ ЧАСТЬ

На сегодня используют два различных подхода решения СЛАУ: методы прямого решения и итерационные методы. Каждый подход обладает определенными преимуществами, но чаще всего пользуются прямыми методами. Главной проблемой итерационных методов является частая плохая обусловленность матриц жесткости большой размерности, когда ее определитель равен малой величине или стремится к нулю. И даже процесс предобуславливания не дает абсолютной гарантии сходимости решения.

Для упрощения решения СЛАУ с симметричной матрицей жесткости применяют следующее ее разложение [5]:

$$K_{n,n} = L_{n,n} D_{n,n} L^T_{n,n}, \tag{2}$$

где

L – нижняя треугольная матрица (хранится в разреженном формате), у которой все диагональные элементы равны единице;

D – диагональная матрица (хранится в виде одномерного массива).

Решение СЛАУ $LDL^T X = F$ в этом случае сводится к последовательному решению трех простейших задач методом подстановки:

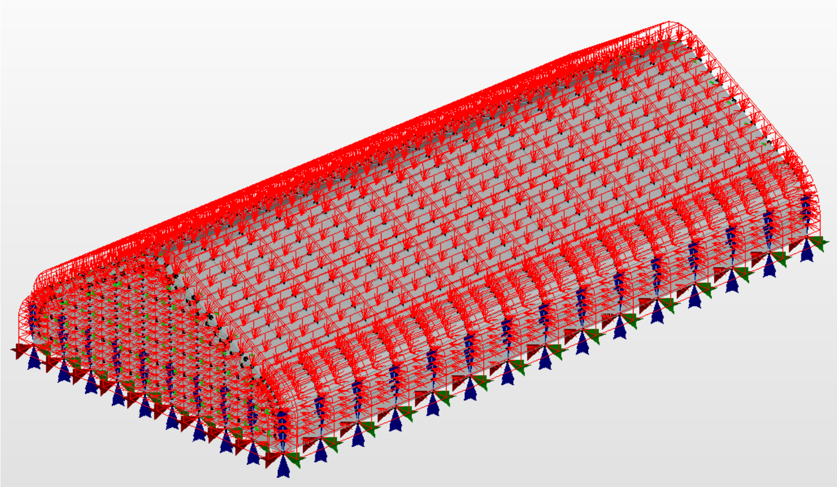
$$\begin{cases} L_{n,n} Y_{n,m} = F_{n,m}; \\ D_{n,n} Z_{n,m} = Y_{n,m}; \\ L^T_{n,n} X_{n,m} = Z_{n,m}. \end{cases} \tag{3}$$

Основной проблемой при таком разложении является заполнение разреженной треугольной матрицы L большим количеством ненулевых элементов, что влечет за собой значительный расход памяти и увеличение количества вычислительных операций над элементами этой матрицы. Поэтому перед процедурой разложения производят переупорядочивание строк и столбцов исходной матрицы жесткости. Для этого находят специальную матрицу переупорядочивания P такую, что:

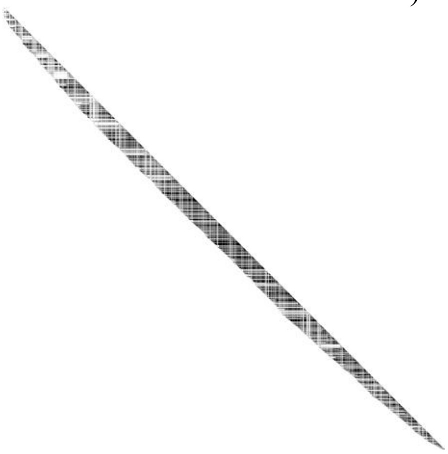
$$\left(P_{n,n} K_{n,n} P^T_{n,n}\right)\left(P_{n,n} X_{n,m}\right) = P_{n,n} F_{n,m}. \tag{4}$$

Матрица переупорядочивания P представляет собой матрицу, у которой в каждой строке и каждом столбце содержится только один элемент равный единице, а все остальные элементы равны нулю. Хранится эта матрица в виде массива номеров столбцов с единичными элементами.

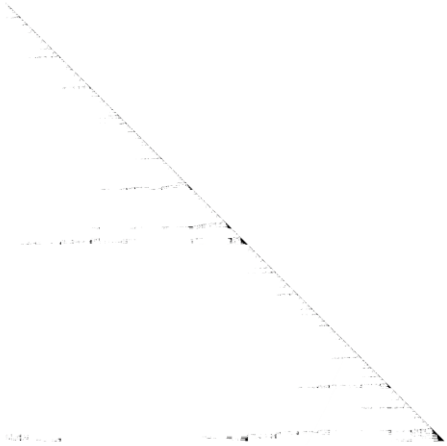
Для нахождения матрицы переупорядочивания часто применяют алгоритм Катхилла-Макки ввиду простоты его реализации, но такой алгоритм не является оптимальным с точки зрения уменьшения заполнения ненулевыми элементами матриц разложения. Более эффективными являются алгоритмы на основе применения метода минимальной степени. На практике данный метод в его исходном виде не используется в силу его значительной трудоемкости. Широко распространены две его модификации: множественный метод минимальной степени (Multiple Minimum Degree, MMD) и приближенный метод минимальной степени (Approximate Minimum Degree, AMD) [6]. Последний используется в программных продуктах компании НТЦ АПМ. На рисунке 1 показан результат вычисления нижней треугольно матрицы L с переупорядочиванием с помощью алгоритма Катхилла-Макки и приближенным методом минимальной степени.



а) Исходная модель



б) Cuthill–McKee



в) AMD

Рис. 1. Результат разложения после переупорядочивания матрицы
Fig. 1. The result of decomposition after matrix reordering

Здесь черными точками обозначены ненулевые элементы полученной матрицы разложения L. В таблице 1 показано количество полученных ненулевых элементов данной матрицы.

Таблица 1

Результаты работы алгоритмов переупорядочивания матрицы

Table 1

Results of matrix reordering algorithms

	Исходная матрица жесткости	Cuthill–McKee	AMD
Количество ненулевых элементов	279966	4510257	469879

После процедуры переупорядочивания производится непосредственно разложение матрицы жесткости. В классическом варианте разложение LDL^T вычисляется следующим образом:

$$\begin{cases} D_j = K_{jj} - \sum_{k=1}^{j-1} L_{jk} D_k L_{jk}^T\,; \\ L_{ij} = \dfrac{K_{ij} - \sum_{k=1}^{j-1} L_{ik} D_k L_{jk}^T}{D_j}; \\ i = 1, \dots, n; \quad j = 1, \dots, n. \end{cases} \tag{5}$$

Выражение (5) плохо поддается распараллеливанию ввиду большого количества информационных зависимостей. Кроме того, вычисления не сбалансированы, что значительно снижает их эффективность. Для устранения этих проблем был разработан алгоритм, адаптированный для параллельных вычислений [7, 8]. Треугольную матрицу разложения L можно визуально представить как музыкальный инструмент в виде арфы (рисунок 2). Каждая струна арфы соответствует столбцу матрицы L. Игра на струнах осуществляется поочередно. Игра на j-ой струне означает нахождение окончательных значений элементов j-ого столбца L. При этом, каждая струна распространяет незатухающую звуковую волну в сторону струн, расположенных от нее правее. Т.е., каждый j-й столбец матрицы L частично изменяет значения элементов всех столбцов с номером выше j. Таким образом, можно параллельно вычислять влияние текущего столбца на все столбцы, расположенные правее.

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{41} & l_{42} & l_{43} & 1 & 0 & 0 & 0 & 0 & 0 \\ l_{51} & l_{52} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ l_{61} & l_{62} & 0 & l_{64} & 0 & 1 & 0 & 0 & 0 \\ l_{71} & l_{72} & l_{73} & 0 & 0 & 0 & 1 & 0 & 0 \\ l_{81} & l_{82} & 0 & l_{84} & 0 & l_{86} & 0 & 1 & 0 \\ l_{91} & l_{92} & 0 & 0 & l_{95} & l_{96} & l_{97} & l_{98} & 1 \end{pmatrix}$$

Рис. 2. Визуальное представление матрицы разложения в виде арфы
Fig. 2. Visual representation of the decomposition matrix in the form of a harp

Математически струнно-волновой алгоритм можно описать следующим образом. Сначала матрицам D и L присваиваются соответствующие значения матрицы жесткости:

$$\begin{cases} D_{ii} = K_{ii}; \\ L_{ij} = K_{ij}; \\ i = 1, \dots, n; \quad j = 1, \dots, i-1. \end{cases} \tag{6}$$

Далее элементы этих матриц изменяются согласно вышеописанному алгоритму:

$$\begin{cases} L_{ij} = \frac{L_{ij}}{D_{jj}}; \\ L_{ik} = L_{ik} - L_{kj} D_{jj} L_{ij}; \\ D_{ii} = D_{ii} - L_{ij} D_{jj} L_{ij}; \\ j = 1, \dots, n;\ i = j+1, \dots, n; k = 2, \dots, i-1. \end{cases} \tag{7}$$

Т.к. матрица разложения L является разреженной, то не на все правые столбцы оказывает влияние текущий столбец. В ходе проведенных исследований было выявлено, что изменяются только те столбцы, номер которых соответствует номерам строк ненулевых элементов текущего столбца. Данное обстоятельство позволяет устранить лишние итерации цикла параллельного алгоритма.

При решении СЛАУ больших размерностей прямым методом часто не хватает памяти. Струнно-волновой алгоритм позволяет уменьшить размер задачи, например, применив следующее разложение:

$$K = L_1 B L_1^T, \tag{8}$$

где

L_1 – первые столбцы треугольной матрицы разложения L;

B – полученная влиянием L_1 подматрица меньшей размерности (рисунок 3).

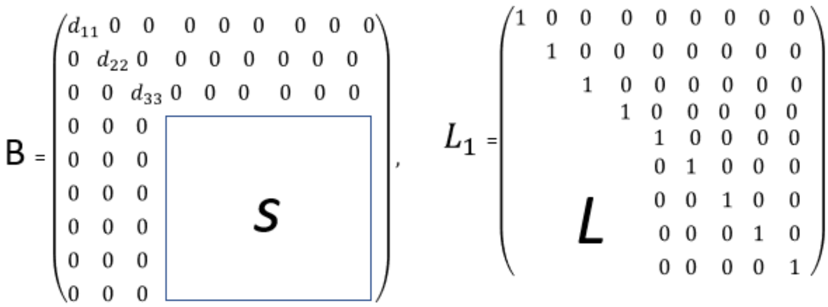


Рис. 3. Уменьшение размерности задачи

Fig. 3. Reducing the dimensionality of the problem

В свою очередь, для матрицы B можно применить аналогичное разложение. В итоге, струнно-волновой алгоритм позволяет получить следующее выражение:

$$K = L_1 L_2 \dots L_{\mathrm{n}-1} L_{\mathrm{n}} D L_{\mathrm{n}}^T L_{\mathrm{n}-1}^T \dots L_2^T L_1^T \tag{9}$$

Данное разложение матрицы жесткости позволяет разбить большую задачу на множество мелких подзадач, где решение СЛАУ также находится методом подстановки:

$$\begin{cases} L_1 X_n = F; \\ L_2 X_{n-1} = X_n; \\ \dots \\ L_n X_{i-1} = X_{i-2}; \\ D X_i = X_{i-1}; \\ L_{\mathrm{n}}^T X_{i+1} = X_i; \\ \dots \\ L_2^T X_2 = X_3; \\ L_1^T X_1 = X_2. \end{cases} \tag{10}$$

Для оценки эффективности параллельных вычислений применялись параметры ускорения и эффективности [9]. Под ускорением понимается отношение времени выполнения последовательного алгоритма ко времени выполнения параллельного алгоритма на p процессорах:

$$S_p = \frac{T_1}{T_p} \tag{11}$$

Эффективность использования параллельным алгоритмом процессоров при решении задачи определяется как отношение ускорения к количеству процессоров:

$$E_p = \frac{S_p}{p} \tag{12}$$

Результаты вычислительных экспериментов оценки эффективности параллельного алгоритма представлены в таблице 2. Алгоритм реализован с помощью технологии OpenMP [10]. Вычисления производились на шестиядерном процессоре Intel Core i7-8700K с 32 ГБ оперативной памяти DDR4. Для проведения вычислительных экспериментов использовалась программная система APMStructure3D, в которой загружались модели различных размерностей и при замере времени выполнения параллельного алгоритма учитывалось время на формирование матрицы жесткости.

Таблица 2

Оценка эффективности параллельного алгоритма

Table 2

Efficiency evaluation of the parallel algorithm

Размерность матрицы жесткости, n	Ускорение, S_p	Эффективность, E_p
94692	3,54	0,59
174267	3,75	0,63
244620	4,86	0,81
527919	4,72	0,79
755181	4,78	0,80

Таким образом, струнно-волновой алгоритм эффективно работает в связке с алгоритмом переупорядочивания AMD, разбивая большую задачу на множество мелких подзадач с наименьшими информационными зависимостями, что позволяет при его параллельной реализации достичь оптимальных значений параметров ускорения и эффективности. Разработанный алгоритм успешно применяется в программных продуктах компании НТЦ АПМ и постоянно модернизируется с целью повышения своего быстродействия.

ЗАКЛЮЧЕНИЕ

В статье представлен новый алгоритм решения систем линейных алгебраических уравнений для симметричных разреженных матриц. Данный алгоритм включает в себя метод переупорядочивания разреженных матриц AMD и струнно-волновой метод решения СЛАУ собственной разработки. Он успешно применяется в программных продуктах компании НТЦ АПМ. Разработанный решатель используется в таких программных модулях как статический и динамический расчет, расчет собственных частот и устойчивости при решении обобщенной задачи на собственные значения, расчет гармонических колебаний и многих других.

Разработанный алгоритм является параллельным и реализован с помощью технологии OpenMP. Вычислительные эксперименты показывают, что с увеличением количества процессоров и объема входных данных ускорение увеличивается без падения эффективности. Т.е. разработанный параллельный алгоритм является масштабируемым.

Дальнейшие исследования будут сосредоточены на повышении его быстродействия и реализации с помощью технологии CUDA для графических ускорителей NVIDIA и технологии MPI для систем с распределенной памятью.

Список литературы

1. Бате К., Вилсон Е. Численные методы анализа и метод конечных элементов. – Москва: Стройиздат, 1982. – 448 с.
2. Джеммель Дж. Вычислительная линейная алгебра. Теория и приложения. – Москва: Мир, 2001. – 430 с.
3. Писсанецки С. Технология разреженных матриц. – Лондон: Academic Press Inc, 1984. – 312 с.

4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – Москва: Мир, 1979. – 542 с.
5. Андреев Б. Численные методы. – Москва: издательство ВМК МГУ им Ломоносова, 2013. – 336 с.
6. Heggernes P., Eisenstat S.C., Kumfert G. The Computational Complexity of the Minimum Degree Algorithm // Четырнадцатая норвежская конференция компьютерных наук, Осло, Норвегия, 26-28 ноября, 2001. URL: https://digital.library.unt.edu/ark:/67531/metadc1410671/ (дата обращения: 09.07.2025).
7. Воеводин В., Воеводин Вл. Параллельные вычисления. – Спб: BHV, 2015. – 608 с.
8. Воеводин В.В. Математические основы параллельных вычислений. – Москва: МГУ, 1991. – 345 с.
9. Гергель В.П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. – Москва: Издательство Московского Университета, 2010. – 544 с.
10. Гергель В.П. Современные языки и технологии параллельного программирования. – Москва: Издательство Московского Университета, 2012. – 408 с.

References

1. Bathe K.J., Wilson E.L. Numerical methods in finite element analysis. Moscow: Stroyizdat, 1982. 448 p.
2. Demmel J. Computational linear algebra. Theory and applications. Moscow: Word, 2001. 430 p.
3. Pissanetsky S. Sparse Matrix Technology. London: Academic Press Inc, 1984. 312 p.
4. Aho A.V., Hopcroft J.E., Ulman J.D.: The design and Analisys of Computer Algorithms. Addison – Wesley: Reading, MA. Third printing, 1976. 542 p.
5. Andreev B. Numerical methods. Moscow: Publishing department of the Faculty of Computational Mathematics and Mathematics of Lomonosov Moscow State University, 2013. 336 p.
6. Heggernes P., Eisenstat S.C., Kumfert G. The Computational Complexity of the Minimum Degree Algorithm // 14-th Norwegian Computer Science Conference, Oslo, Norway, November 26-28, 2001. https://doi.org/10.2172/15002765
7. Voevodin V., Voevodin V. Parallel Computing. Saint Petersburg: BHV, 2015. 608 p.
8. Voevodin V.V. Mathematical foundations of parallel computing. World Scientific Publishing Co., Series in computer science. 1992. V.33. 343 p.
9. Gergel V.P. High-performance computing for multiprocessor, multi-core systems. Moscow: Moscow University Publishing House, 2010. 544 p.
10. Gergel V.P. Modern languages and parallel programming technologies. Moscow: Moscow University Publishing House, 2012. 408 p.

Лихошерстный Алексей Юрьевич, кандидат технических наук, математик-программист, научно-технический центр «АПМ», г. Королёв, Россия
Великая Яна Геннадьевна, кандидат технических наук, доцент кафедры математического и программного обеспечения информационных систем, Белгородский государственный национальный исследовательский университет, г. Белгород, Россия

Likhosherstnyy Alexey Yuryevich, Candidate of Technical Sciences, Mathematician and Programmer, APM Ltd, Korolev, Russia
Velikaya Yana Gennadievna, Candidate of Technical Sciences, Associate Professor of the Department of Mathematical and Software Support for Information Systems, Belgorod State National Research University, Belgorod, Russia