

УДК 519.612

doi 10.26089/NumMet.v16r457

**ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ
УРАВНЕНИЯ ГЕЛЬМГОЛЬЦА ПРЯМЫМ МЕТОДОМ С ИСПОЛЬЗОВАНИЕМ
МАЛОРАНГОВОЙ АППРОКСИМАЦИИ И HSS-ФОРМАТА**

Б. М. Глинский¹, В. И. Костин², Н. В. Кучин³, С. А. Соловьев⁴, В. А. Чеверда⁵

Предложен алгоритм решения систем линейных алгебраических уравнений (СЛАУ), основанный на методе исключения Гаусса и предназначенный для решения уравнения Гельмгольца в трехмерных неоднородных средах. Для решения СЛАУ, возникающих в геофизических приложениях, разработана параллельная версия алгоритма, направленная на использование гетерогенных высокопроизводительных вычислительных систем, содержащих узлы с MPP- и SMP-архитектурой. Малоранговая аппроксимация, HSS-формат и динамическое распределение промежуточных результатов среди кластерных узлов позволяют решать задачи в разы большие, чем при использовании традиционных прямых методов, сохраняющих блоки L -фактора в полном ранге (Full-Rank, FR). Использование предложенного алгоритма позволяет сократить время расчетов, что актуально для решения трехмерных задач геофизики. Численные эксперименты подтверждают упомянутые преимущества предложенного малорангового прямого метода (Low-Rank, LR) по сравнению с прямыми FR-методами. На модельных геофизических задачах показана “жизнеспособность” реализованного алгоритма.

Ключевые слова: уравнение Гельмгольца, алгоритмы решения разреженных линейных систем, прямой метод Гаусса, аппроксимация матрицами малого ранга, HSS-формат матриц, распределенные параллельные системы, гетерогенные высокопроизводительные вычислительные системы.

1. Алгоритм решения систем линейных алгебраических уравнений с разреженной матрицей коэффициентов. Предметом рассмотрения в настоящей статье является система линейных алгебраических уравнений (СЛАУ)

$$AU = F \tag{1}$$

с разреженной $N \times N$ -матрицей A и с K правыми частями, которые являются столбцами $N \times K$ -матрицы F . Столбцы соответствующих решений образуют $N \times K$ -матрицу U . Такая система, например, возникает в результате численной аппроксимации краевой задачи для уравнений в частных производных эллиптического типа. В частности, расчеты, приведенные ниже в этой статье, проводились для численного решения уравнения Гельмгольца

$$\Delta u + \frac{(2\pi\nu)^2}{V^2} u = \delta(\bar{r} - \bar{r}_s) f \tag{2}$$

в трехмерном пространстве с условиями излучения на бесконечности. Область, в которой отыскивается решение этого уравнения, представляет собой некоторый параллелепипед D , окруженный так называемым идеально поглощающим слоем (PML от английского Perfectly Matching Layer), на внешней границе ставятся однородные граничные условия Дирихле. Слой PML необходим для уменьшения влияния отражений от внешних границ [5]. В уравнении (2) ν — частота, V — скорость звука, r_s — координаты источника, f — источник.

¹ Сибирский суперкомпьютерный центр при Институте вычислительной математики и математической геофизики СО РАН (ССКЦ, ИВМиМГ СО РАН), просп. Лаврентьева, 6, 630090, г. Новосибирск; исполнительный директор Сибирского суперкомпьютерного центра, e-mail: gbm@sscc.ru

² Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (ИНГГ СО РАН), проспект Коптюга, 3, 630090, Новосибирск; ст. науч. сотр., e-mail: vctrkstn@mail.ru

³ Сибирский суперкомпьютерный центр при Институте вычислительной математики и математической геофизики СО РАН (ССКЦ, ИВМиМГ СО РАН), просп. Лаврентьева, 6, 630090, Новосибирск; главный специалист, e-mail: kuchin@sscc.ru

⁴ Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (ИНГГ СО РАН), проспект Коптюга, 3, 630090, Новосибирск; науч. сотр., e-mail: 511ssa@mail.ru

⁵ Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (ИНГГ СО РАН), проспект Коптюга, 3, 630090, Новосибирск; зав. лабораторией, e-mail: CheverdaVA@ipgg.sbras.ru

Конечно-разностная аппроксимация уравнения (2) на 27-точечном шаблоне ($3 \times 3 \times 3$) параллелепипедальной сетки с равномерными шагами сетки (см. подробнее в [3]) обеспечивает второй порядок аппроксимации уравнения с приемлемой численной дисперсией и приводит к системе линейных уравнений вида (1), в которой правая часть уравнения (2) после дискретизации становится вектором правой части в (1), а сеточная функция конечно-разностной аппроксимации уравнения (2) — столбцом неизвестных в (1). Формально говоря, таким образом мы получим систему уравнений с одной правой частью. Однако если требуется решить исходную задачу для нескольких положений источника (в геофизических приложениях количество источников может достигать сотен и даже тысяч), соответствующие векторы правых частей можно собрать в одну матрицу F . Такой подход имеет определенное преимущество при использовании прямых методов решения СЛАУ — факторизация матрицы коэффициентов производится один раз для всех правых частей. Отметим, что из-за использования РМЛ матрица коэффициентов в (1) комплексная ($A \in C^{N \times N}$) и симметричная, но не эрмитова. Комплексность матрицы коэффициентов вдвое увеличивает требования к оперативной памяти по сравнению с вещественным случаем. Кроме того, возрастает цена арифметических операций.

Ограничиваясь случаем симметричной матрицы коэффициентов, кратко напомним метод решения СЛАУ (1), основанный на разложении матрицы на треугольные множители. Метод стоит в основном из двух шагов.

1. Матрица коэффициентов представляется в виде произведения трех сомножителей $A = LDL^t$, где матрица L — нижнетреугольная с единицами на главной диагонали, D — диагональная матрица, L^t — матрица, транспонированная к L . Этот шаг может занимать существенную долю времени решения полной задачи, особенно в случае сравнительно небольшого числа правых частей K .
2. Решение системы линейных уравнений с факторизованной матрицей — прямое исключение, обращение диагональной матрицы и обратное исключение:
 - а) прямое исключение $Z = L^{-1}F$,
 - б) $Y = D^{-1}Z$,
 - в) обратное исключение $U = L^{-t}Y$.

В случае значительного числа правых частей K время на выполнение этого шага может быть сравнимо со временем, затраченным на факторизацию.

Здесь и далее предполагается (и более не оговаривается специально), что соответствующие операции осуществимы. Иначе говоря, матрицы, для которых вычисляются обратные, невырождены. Например, в формулах выше матрица D — невырождена (нижнетреугольная матрица L имеет единицы на диагонали и всегда невырождена). Мы не рассматриваем вопросы численной устойчивости алгоритмов, излагаемых ниже.

В случае разреженной матрицы A число ненулевых элементов в треугольном факторе L обычно больше (иногда значительно больше), чем в исходной матрице A . Замечено, что если в исходной разреженной матрице переставить строки и соответствующим образом переставить столбцы, сохраняя ее симметрию, то число ненулевых элементов в треугольном факторе L может измениться (иногда существенно). Поэтому в практических реализациях LDL^t -разложения для разреженных матриц его обычно предваряют дополнительным шагом перестановки столбцов и строк матрицы, призванным уменьшить число ненулевых элементов в L [2]. Мы в дальнейшем предполагаем такую перестановку строк и столбцов в матрице коэффициентов сделанной и не будем больше на этом останавливаться. Отметим только, что для матрицы, полученной дискретизацией уравнения Гельмгольца, с этой целью мы применяли алгоритм вложенных сечений (Nested Dissections, ND).

2. Алгоритм получения LDL^t -разложения матрицы коэффициентов. Описание алгоритма разложения начнем с рассмотрения симметричной матрицы в блочном виде

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} L_{11}^t & L_{21}^t \\ 0 & L_{22}^t \end{pmatrix}.$$

Элементы этого разложения вычисляются за следующие три шага:

- 1) LDL^t -разложение блока A_{11} ($L_{11}D_1L_{11}^t = A_{11}$);
- 2) $L_{21} = A_{21}L_{11}^{-t}D_1^{-1}$;

3) $S_{22} = A_{22} - L_{21}D_1L_{21}^t$;

4) LDL^t -разложение блока S_{22} ($L_{22}D_2L_{22}^t = S_{22}$).

Первый и четвертый шаги представляют собой вычисление LDL^t -разложений матриц A_{11} и S_{22} соответственно, второй шаг — вычисление поддиагонального блока L_{21} , а третий — построение так называемого дополнения Шура S_{22} . Важно отметить, что последовательность действий здесь определена списком — вычисление элементов разложения нижнего блока (L_{22} и D_2) не может начаться до того, как закончится вычисление L_{21} , которое, в свою очередь, может начаться лишь по завершении разложения блока A_{11} . Это замечание может быть представлено в виде графа (так называемого дерева исключений), в котором вершина с номером 1 соответствует вычислениям элементов L_{11} , L_{21} первого блочного столбца матрицы L (пунктам 1 и 2 приведенного выше списка), а вершина с номером 2 соответствует вычислению L_{22} (рис. 1, слева). Зависимость вершины 2 от вершины 1 по дереву исключений как раз отражает тот факт, что обработка второго блочного столбца происходит по завершении обработки первого столбца.

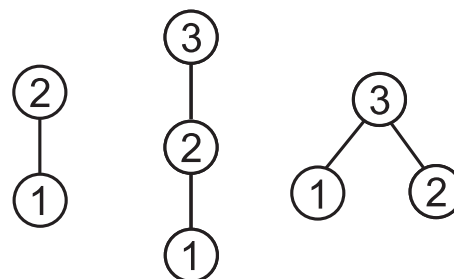


Рис. 1. Деревья исключений

Приведенные рассуждения могут быть легко обобщены на случай блочной 3×3 - матрицы

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{pmatrix} \begin{pmatrix} L_{11}^t & L_{21}^t & L_{31}^t \\ 0 & L_{22}^t & L_{32}^t \\ 0 & 0 & L_{33}^t \end{pmatrix}.$$

Сопоставляя вычислениям элементов первого столбца L вершину с номером 1, элементов второго столбца вершину с номером 2 и элементов третьего столбца вершину с номером 3, мы получим дерево исключений, определяющее порядок вычислений: после обработки вершины 1 следует обработать вершину 2, затем вершину 3 (рис. 1, посередине).

В частном случае, когда матрица имеет нулевые блоки A_{21} и A_{12} , ситуация меняется:

$$A = \begin{pmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{pmatrix} \begin{pmatrix} L_{11}^t & 0 & L_{31}^t \\ 0 & L_{22}^t & L_{32}^t \\ 0 & 0 & L_{33}^t \end{pmatrix}.$$

Легко видеть, что вычисление элементов L_{22} , L_{32} второго столбца матрицы L не требует знания элементов первого столбца и может быть произведено независимо от первого столбца. Дерево исключений выглядит уже иначе: в нем третий узел зависит от первого и второго, в то время как эти два узла поставлены на один уровень, что означает возможность их параллельной обработки (рис. 1, справа).

Структура L -фактора разреженной матрицы, соответствующей конечно-разностной аппроксимации краевой задачи для описанного выше уравнения Гельмгольца (перестановка столбцов по методу ND применена!), проиллюстрирована на рис. 2. На этом же рисунке также приведено соответствующее дерево исключений: каждому узлу дерева соответствует панель матрицы L , состоящая из диагонального блока и всех его поддиагональных. Заметим, что рис. 2 играет лишь иллюстративную роль — число блоков может варьироваться в зависимости от используемой разностной сетки. Однако он правильно отражает тот факт, что дерево исключений является бинарным.

Кроме того, на левой части рисунка отражен тот факт, что размеры диагональных блоков матрицы L возрастают по мере продвижения сверху вниз вдоль диагонали. Отметим также, что в этом направлении возрастает плотность заполнения блоков, изображенная на рисунке оттенками серого цвета на шаблоне матрицы L : чем темнее цвет, тем больше ненулевых элементов в соответствующем блоке. Белый цвет означает нулевой блок, черный цвет — плотную матрицу.

Дерево исключений полностью отражает зависимости по данным в процессе вычислений. Например, нахождение панели матрицы L , соответствующей узлу 13 на рис. 2, может начаться лишь после того, как закончатся вычисления в узлах 9 и 10. Эти узлы, в свою очередь, должны ожидать окончания вычислений в узлах 1–4. Узлу 13 для начала разложения нужны не только данные от своих непосредственных потомков 9 и 10, но и от потомков своих потомков 1–4. Следует отметить, однако, что узел 13 может получить данные от узлов 1–4 до окончания вычислений в узлах 9 и 10 и использовать их для частичного

вычисления дополнения Шура. Получив затем данные от узлов 9 и 10 и закончив дополнение Шура с этими данными, узел 13 может выполнить LDL^t -разложение соответствующего диагонального блока, а также найти всю 13-ю панель матрицы L . Иначе говоря, по окончании разложения в каком-то из узлов этот узел может разослать данные, необходимые для выполнения соответствующих дополнений Шура, вверх по дереву всем узлам, которые от него зависят. После приема таких данных узлы могут произвести частичные вычисления дополнения Шура с теми данными, которые они получили.

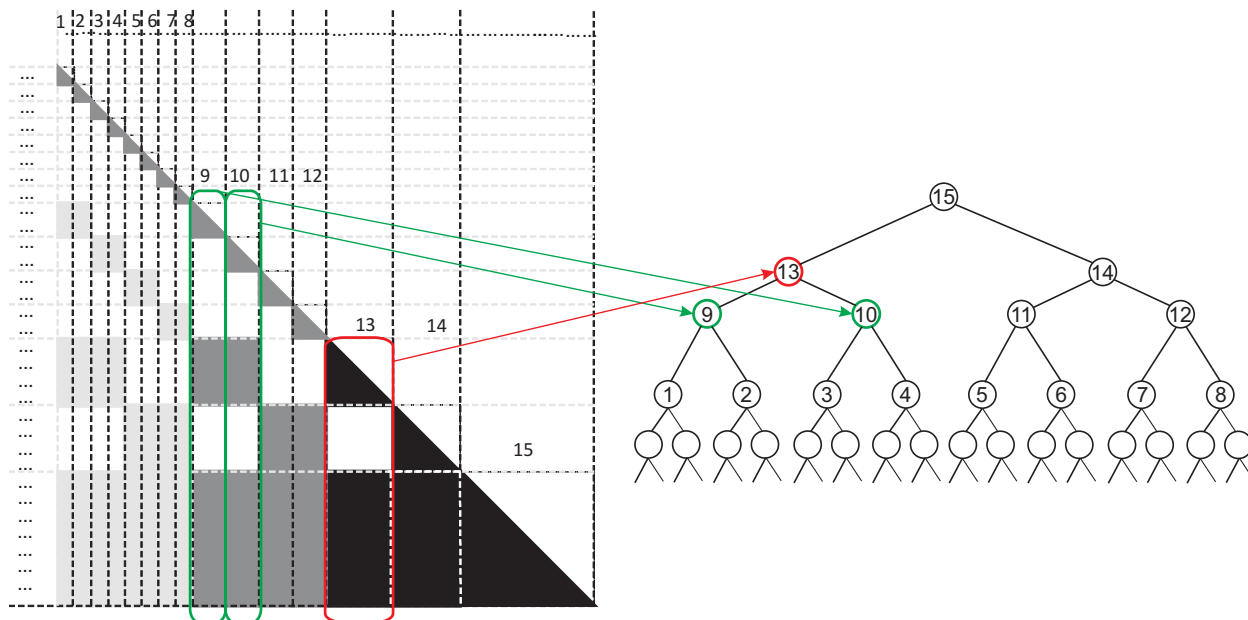


Рис. 2. Шаблон множителя L (слева) и соответствующее ему дерево исключений (справа)

Когда исходная матрица происходит из аппроксимации уравнений в частных производных (например, в случае уравнения Гельмгольца), поддиагональные блоки матрицы L обладают замечательным свойством — численные ранги этих матриц меньше их размеров [4]. Иначе говоря, ненулевой поддиагональный блок L_{ij} размера $m \times n$ может быть аппроксимирован произведением $m \times r$ -матрицы p на $r \times n$ -матрицу q :

$$\|L_{ij} - pq\| < \epsilon \|L_{ij}\|$$

с небольшим значением r . В случае $r \ll \min(m, n)$ это позволяет “сжать” информацию об L_{ij} — взамен хранения массива чисел с числом элементов mn хранить два массива чисел общего объема $r(m+n)$. Более того, такая аппроксимация позволяет также “экономить” на числе операций в вычислениях, в которых участвует L_{ij} . Необходимо отметить, однако, что за этот выигрыш приходится “заплатить” вычислением матриц p и q . Ясно, что для того чтобы накладные расходы не перекрыли получаемый выигрыш, описанный подход следует применять начиная с достаточно больших значений m и n . Для получения малоранговой аппроксимации можно использовать блочную модификацию алгоритма крестовой аппроксимации (см. [7]).

Применение малоранговой аппроксимации поддиагональных блоков матрицы L проиллюстрировано на рис. 3 справа, где схематически показано, сколько элементов поддиагональных блоков матрицы L нужно хранить.

На этом же рисунке показана дополнительная возможность сжатия информации об L — представление диагональных блоков в так называемом HSS-формате (Hierarchically Semiseparable Structure) [6]. Как и в случае поддиагональных блоков, HSS-формат применяется начиная с некоторого номера диагонального блока, так как перевод диагонального блока из обычного плотного формата в HSS-формат сопряжен с дополнительными вычислениями.

В результате факторизации, использующей малоранговую аппроксимацию и HSS-формат, получаем аппроксимацию \tilde{L} матрицы L . Решение \tilde{x} системы линейных уравнений $\tilde{L}\tilde{D}\tilde{L}^t\tilde{x} = b$ аппроксимирует решение исходной системы. Ошибка этой аппроксимации напрямую зависит от ошибки малоранговой аппроксимации внедиагональных блоков и ошибки перевода в HSS-формат. Корреляция этих ошибок, а также детали алгоритма подробно описаны в статье [1]. Заметим, что в случае большой ошибки в решении итерационное уточнение может помочь исправить ситуацию.

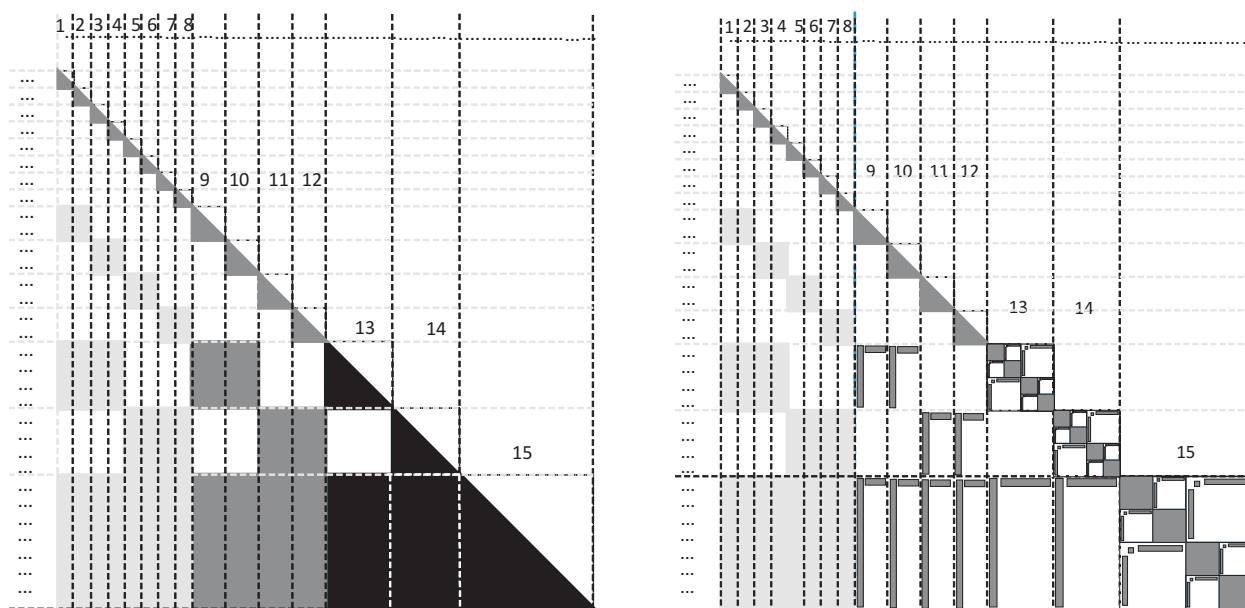


Рис. 3. Шаблон множителя L (слева) и его малоранговая аппроксимация (справа)

3. Организация параллельных вычислений. При попытках реализовать алгоритм LDL^t -факторизации, схематично описанный в предыдущем разделе статьи, на гетерогенном вычислительном кластере НКС-30Т возник ряд практических вопросов, например:

- достаточно ли оперативной памяти на узлах кластера для проведения вычислений;
- какие узлы кластера назначаются для обработки соответствующих узлов дерева исключений;
- каким образом оптимизировать обмены между узлами кластера.

Эти и многие другие вопросы требуют ответа.

При описании алгоритма в предыдущем разделе было замечено, что в процессе продвижения сверху вниз вдоль диагонали матрицы L размеры блоков возрастают и они становятся более плотными. Поэтому возможна ситуация, когда оперативной памяти одного узла станет недостаточно для обработки какого-то узла дерева исключений. Простой прием, проиллюстрированный на рис. 4, может помочь справиться с такой ситуацией. Он состоит в разбиении блока на несколько подблоков и последовательной обработке их (см. рис. 4 слева). Соответствующим образом изменяется дерево исключений — оно становится полубинарным, где каждый узел имеет одного или двух потомков.

Пусть перед началом вычислений исходная матрица в разреженном формате с надлежащим образом переставленными строками и столбцами хранится на одном кластерном узле, скажем в узле с номером 0. Это предположение оправдывается тем, что число ненулевых элементов в исходной матрице значительно меньше числа ненулевых элементов в треугольной матрице L и хранение полной разреженной матрицы A на одном узле возможно. От такого предположения можно избавиться, но это приведет к усложнению описания алгоритма, что выходит за рамки настоящей статьи. Таким образом, одна из функций узла с номером 0 кластера — хранение исходной матрицы. Вычислительные узлы кластера получают блоки исходной матрицы по мере необходимости. Кроме того, узел с номером 0 кластера хранит результаты так называемой символьной факторизации — специального шага, предворяющего численную факторизацию матрицы и имеющего целью получение информации о структуре матрицы L .

После символьной факторизации позиции всех ненулевых элементов матрицы L становятся известными и, в частности, известен объем требуемой памяти для хранения ненулевых элементов матрицы L . Описанный в предыдущем разделе прием малоранговой аппроксимации поддиагональных блоков и HSS-формат для хранения диагональных блоков матрицы L позволяют уменьшить требования к необходимой памяти, однако, в отличие от результата символьной факторизации, это уменьшение памяти не допускает априорной оценки. Результаты становятся известными после аппроксимации.

Переходим к описанию непосредственно алгоритма вычислений на кластере. Для этого нам достаточно отобразить дерево исключений на список вычислительных узлов кластера. Вычислительный узел получает соответствующие данные для проведения вычислений от узла с номером 0 (группу столбцов

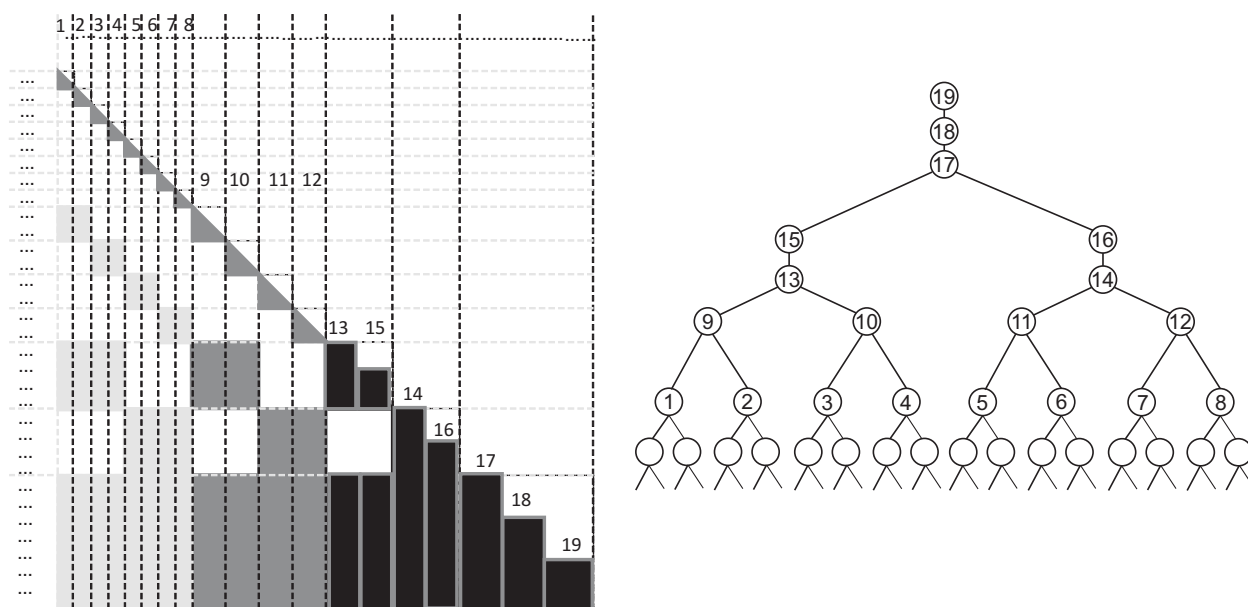


Рис. 4. Модифицированный шаблон фактора L (слева), показанный для простоты в полноранговом представлении, и соответствующее ему полубинарное дерево исключений (справа)

исходной матрицы, соответствующую узлу дерева исключений), а также результаты вычислений своих потомков по дереву исключений. Назначение вычислительного узла кластера для обработки узла дерева исключений происходит динамически, и мы опишем, как это происходит.

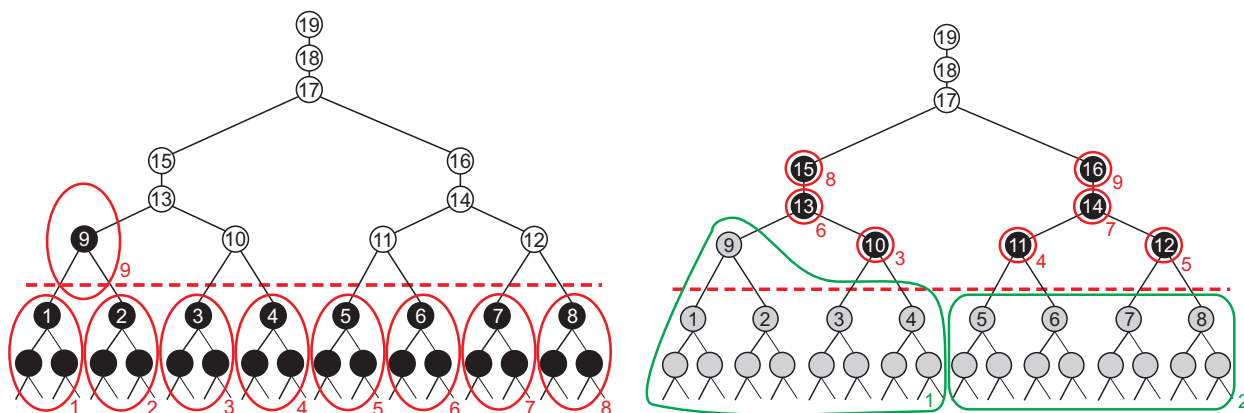


Рис. 5. Распределение данных по вычислительным узлам кластера на первом (слева) и втором (справа) шагах этапа факторизации

На первом шаге каждый вычислительный узел кластера имеет в своей памяти столбцы исходной матрицы, соответствующие некоторому поддереву ниже определенного уровня, обозначенного красной пунктирной линией на рис. 5. Если кластерных узлов больше числа поддеревьев, то каждый оставшийся узел хранит столбцы исходной матрицы, соответствующие некоторому узлу дерева, расположенному выше красной линии. На примере, изображенном на рис. 5, у нас имеется 9 вычислительных узлов кластера, обозначенных овалами красного цвета, между которыми перед первым шагом распределены столбцы исходной матрицы, соответствующие поддеревьям дерева исключений с 1-го по 8-й (рис. 5 слева). Отметим то, что 9-й кластерный узел хранит соответствующий ему 9-й узел дерева исключений. В этом распределении предполагается, что оперативной памяти на каждом из вычислительных узлов достаточно для хранения как соответствующих столбцов исходной матрицы, так и временных массивов и результатов факторизации столбцов до применения малоранговой аппроксимации.

После выполнения обработки узлов дерева с 1-го по 9-й и применения малоранговой аппроксимации результаты находятся в памяти с 1-го по 9-й вычислительных узлов. Следующим шагом алгоритма является сборка результатов в память меньшего количества вычислительных узлов. На рис. 5 справа

схематически показана такая сборка: результаты работы вычислительных узлов 1, 2, 3, 4, 9 собраны в памяти узла с номером 1, результаты работы узлов 5–8 собраны в памяти узла с номером 2. В дальнейшем вычислительные узлы с номерами 1 и 2 исключаются из списка вычислительных узлов и вычисления производятся на оставшихся узлах с номерами 3–9. Им назначаются следующие группы столбцов исходной матрицы в соответствии с порядком, определенным деревом исключений (см. рис. 5 справа). Функции узлов с номерами 1 и 2 ограничиваются хранением собранных результатов, которые они предоставляют по запросу других вычислительных узлов в дальнейших вычислениях (например, для вычисления дополнения Шура последующих столбцов исходной матрицы).

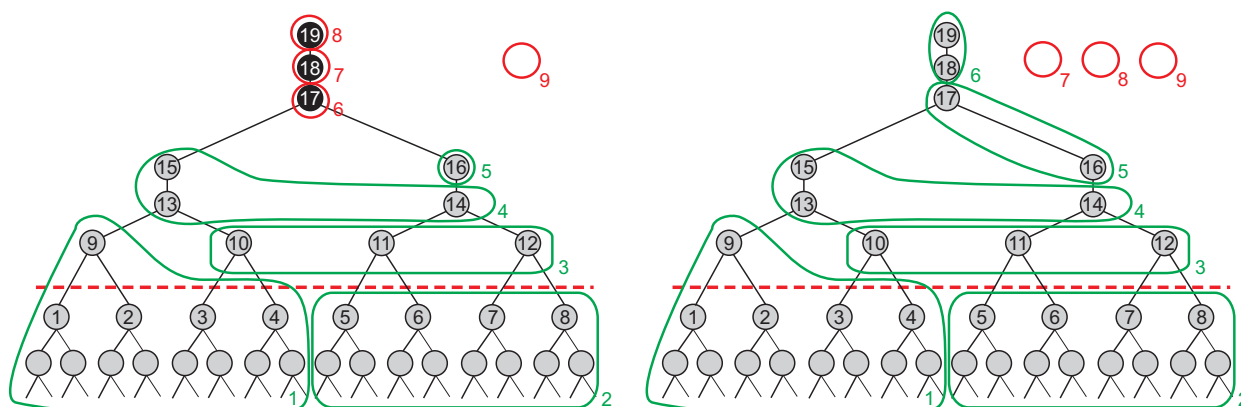


Рис. 6. Распределение данных по вычислительным узлам кластера на третьем (слева) и заключительном (справа) шагах этапа факторизации

Дальнейший ход алгоритма подобен описанному. Вычислительные узлы кластера назначаются для обработки узлов дерева исключений в соответствии с порядком, определенным деревом исключений. Они получают данные для вычислений от узла с номером 0 (в частности, столбцы исходной матрицы, соответствующие обрабатываемому узлу дерева исключений), а также от узлов, хранящих потомков обрабатываемых узлов дерева исключений (части L -фактора, необходимые для вычисления дополнения Шура для текущего узла дерева исключений). Результаты обработки (факторизации) сжимаются с использованием малоранговой аппроксимации и HSS-формата и собираются на отдельных вычислительных узлах с целью освобождения других узлов кластера для дальнейших вычислений. Таким образом, по мере продвижения по дереву исключений количество узлов кластера, назначенных для вычислений, уменьшается, а результаты оказываются собранными на части узлов, первоначально игравшими роль вычислительных узлов. Рис. 6 призван иллюстрировать дальнейшие шаги алгоритма. На нем серыми узлами дерева исключений показаны обработанные данные, черными — данные в процессе обработки. Система обозначений та же, что и на рис. 5, где присутствуют еще и белые узлы, обработка которых еще не начата. Зелеными контурами обозначаются узлы кластера, хранящие результаты факторизации части столбцов матрицы L ; красными — узлы в процессе факторизации. По окончании факторизации результаты (столбцы матрицы L) неким образом распределены в памяти вычислительных узлов кластера, при этом могут остаться свободные узлы без данных (рис. 6 справа).

На этапе решения вектор правой части b (в общем случае плотная матрица B) распределяется между кластерными узлами в соответствии с распределением матрицы L . Мы не останавливаемся на детальном описании процессов исключения неизвестных, ограничившись лишь указанием на то, что он состоит из трех шагов:

- прямой шаг (обращение L);
- диагональный (обращение D);
- обратный шаг (обращение L^t).

Первый и третий шаги этой схемы требуют пересылки промежуточных данных между кластерными узлами. В результате получим решение, распределенное между кластерными узлами таким же образом, как и вектор правой части b .

4. Численные эксперименты. Приведем результаты двух тестов, иллюстрирующих эффективность предложенного алгоритма. В обоих тестах используется однородная скоростная модель, а в каче-

стве источника — дельта-функция Дирака. При этих условиях известно аналитическое решение исходного уравнения Гельмгольца, с которым и производится сравнение численного решения.

В качестве расчетной области в первом тесте использован куб с равномерной по пространству дискретизацией уравнения Гельмгольца (на сетке $256 \times 256 \times 256$). Уровень погрешности аппроксимации матрицами малого ранга задавался 10^{-6} , число ОМР-потоков равно 4 на каждом вычислительном узле. Тест предназначен проиллюстрировать преимущества кластерной версии программного кода перед “одноузловым” кодом — кодом без использования кластера. Одноузловая версия тестировалась на сервере с общей памятью 512 GB, в то время как кластерная — на 32 вычислительных узлах с той же самой суммарной оперативной памятью. Характеристики использованных вычислительных систем вместе с временем решения задачи приведены в таблице. Для того чтобы как-то нивелировать эффект использования процессоров разной мощности на этих двух системах, в таблице присутствует строка “Отношение производительностей процессоров”. Соответствующие значения определяются экспериментально на задачах меньшей размерности как соотношение производительности одного процессора на сервере и одного из процессоров кластерного узла. Таким образом, коэффициент ускорения, получаемый как отношение времен $17198/7715 \approx 2.3$, следует умножить еще на коэффициент 2.2, учитывающий, во сколько раз процессор сервера мощнее процессоров кластера.

Характеристики сравнения одноузловой и кластерной версии

	Сервер 512G RAM	Кластер НСК-30Т 32x16G RAM
Процессор Intel(R) Xeon(R)	E5-2690 v2 @ 3.00GHz AVX	E5540 @ 2.53GHz SSE4.2
Отношение производительностей процессоров	x2.2	x1.0
Время факторизации, сек.	17 198	7 715

Кроме того, следует отметить тот факт, что использование малоранговой аппроксимации в этой задаче позволяет ограничиться в три раза меньшим объемом оперативной памяти (около 240 GB), чем без использования этого приема (около 800 GB).

Второй тест демонстрирует возможность решать большие задачи с использованием кластерной версии программы. В качестве вычислительных ресурсов использован неоднородный набор узлов кластера НСК-30Т Сибирского суперкомпьютерного центра [8], состоящего из одного мощного узла с 8 десятиядерными процессорами Intel(R) Xeon(R) CPU E7-4870 @ 2.40GHz, 1000GB RAM и четырех менее мощных с двумя 6-ядерными процессорами Intel(R) Xeon(R) CPU X5675 @ 3.07GHz, 96 GB RAM каждый. Предполагая, что операционная система кластера использует порядка 10% RAM для своих внутренних ресурсов, считаем, что размер всей свободной памяти составляет около 1250 GB.

Расчетная область — однородный параллелепипед размером $12 \text{ км} \times 12 \text{ км} \times 6 \text{ км}$ со скоростью звука 2400 м/с. Шаг дискретизации выбран равным 30 м во всех направлениях, что при частоте 4 Гц дает 30 точек на длину волны. С трех сторон область окружена слоем PML шириной 300 м (10 сеточных узлов), на верхней границе задано однородное условие Дирихле. В итоге получаем разностную сетку $460 \times 460 \times 230$, общее число неизвестных в СЛАУ примерно равно 5×10^7 . Источник (носитель дельта-функции в правой части уравнения) располагался в верхнем углу расчетной области.

Для решения задачи с использованием малоранговой аппроксимации нам потребовалось 860 GB RAM, в то время как без его использования объем необходимой памяти возрастает до 3500 GB. Факторизация матрицы потребовала 14 часов, этап решения (обращение LDL^t) для одной правой части занимает порядка 3 минут.

Описанная выше конфигурация кластера (один узел с огромной памятью плюс 4 сравнительно небольших) позволила организовать вычисления так, что этот особенный узел с огромной памятью был использован для сбора всех результатов, в то время как вычисления проводились на четырех оставшихся узлах. В итоге матрица L оказалась расположенной на одном узле, и процесс нахождения компонент вектора неизвестных проходил уже на одном узле.

Вещественная часть решения в горизонтальной плоскости источника, а также вдоль горизонтальной линии, проходящей через источник, показаны на рис. 7. Сравнение полученного решения с аналитическим дает относительную ошибку порядка 1% в норме C . Сравнение происходит в области на удалении четверти длины волны от положения источника, чтобы исключить бесконечность в источнике в аналитическом решении. При этом норма относительной невязки $\frac{\|F - AU\|}{\|F\|} \simeq 10^{-5}$ в норме C .

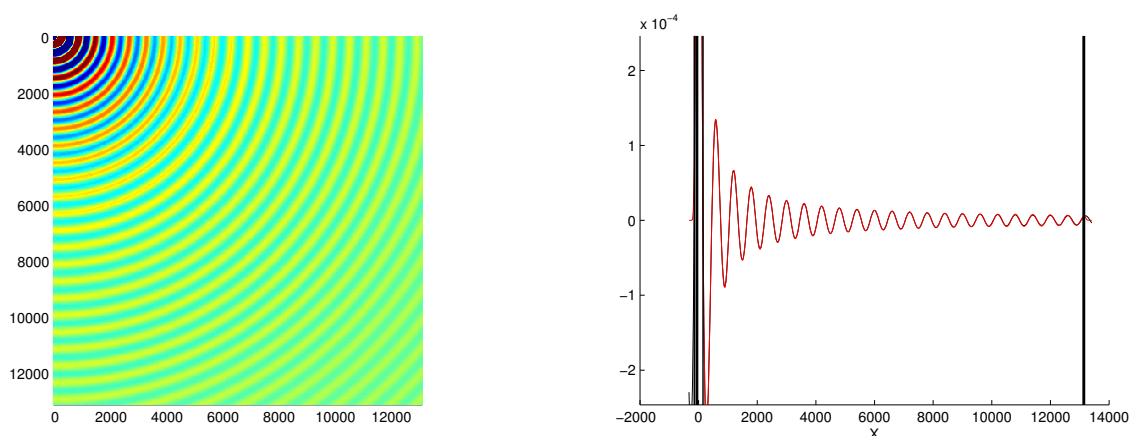


Рис. 7. Вещественная часть решения в горизонтальной плоскости источника (справа), а также вдоль горизонтальной линии, проходящей через источник (слева)

5. Заключение. Разработана версия алгоритма для кластера с гетерогенной архитектурой для решения СЛАУ, возникающих при аппроксимации краевых задач для уравнений в частных производных. Алгоритм основан на использовании вложенных сечений для переупорядочивания матрицы коэффициентов, LDL^t -факторизации этой матрицы, аппроксимации блоков матрицы L матрицами малого ранга и представлении ее диагональных блоков в HSS -формате. Динамический параллелизм алгоритма позволяет эффективно перераспределять имеющиеся ресурсы для вычисления факторизации. В частности, в ситуации неоднородного кластера для хранения промежуточных результатов в первую очередь используются узлы с большим объемом свободной памяти, а расчеты производятся на узлах с меньшей памятью.

Численные эксперименты продемонстрировали корректность работы программы и ускорение расчетов по сравнению с одноузловой версией. Показано также, что использование малоранговой аппроксимации позволяет решать задачи, недоступные для решения традиционными методами без использования малоранговой аппроксимации.

Работа выполнена при финансовой поддержке РФФИ (гранты 13-07-00589, 14-05-00049, 14-05-93090, 16-01-00460, 16-05-00800). Все расчеты проводились на кластере НКС-30Т Сибирского суперкомпьютерного центра.

СПИСОК ЛИТЕРАТУРЫ

1. Соловьев С.А. Решение разреженных СЛАУ прямым методом Гаусса с использованием техники аппроксимации матрицами малого ранга // Вычислительные методы и программирование. 2014. **15**. 441–460.
2. George A. Nested dissection of a regular finite element mesh // SIAM Journal on Numerical Analysis. 1973. **10**, N 2. 345–363.
3. Solov'yev S., Vishnevsky D., and Liu H. Multifrontal hierarchically semi-separable solver for 3D Helmholtz problem using 27-point finite-difference scheme // Extended Abstracts of 77th EAGE Conference and Exhibition. Vol. 77. 2015. doi 10.3997/2214-4609.201413101.
4. Chandrasekaran S., Dewilde P., Gu M., Somasunderam N. On the numerical rank of the off-diagonal blocks of Schur complements of discretized elliptic PDEs // SIAM J. Matrix Anal. Appl. 2010. **31**, N 5. 2261–2290.
5. Collino F., Tsogka C. Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media // Geophysics. 2001. **66**, N 1. 294–307.
6. Xia J. Robust and efficient multifrontal solver for large discretized PDEs // High-Performance Scientific Computing. London: Springer, 2012. 199–217.
7. Solov'yev S.A., Tordeux S. An efficient truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems // Siberian Electronic Mathematical Reports. 2015. **12**. 592–609.
8. Глинский Б.М., Черных И.Г., Кучин Н.В., Ломакин С.В., Макаров И.Н. Управление вычислительными ресурсами Сибирского Суперкомпьютерного Центра // Суперкомпьютерные дни в России: Труды международной конференции (28–29 сентября 2015 г. Москва). М.: Изд-во МГУ, 2015. 667–675.

Поступила в редакцию
30.11.2015

Aspects of Parallel Computing to Solve Helmholtz Equation by a Direct Solver with Low-Rank Approximation and the HSS Format of Data Storage

B. M. Glinskiy¹, V. I. Kostin², V. V. Kuchin³, S. A. Solovyev⁴, and V. A. Cheverda⁵

¹ *Institute of Computational Mathematics and Mathematical Geophysics, Siberian Supercomputer Center, Siberian Branch of Russian Academy of Sciences; prospekt Lavrentyeva 6, Novosibirsk, 630090, Russia; Dr. Sci., Professor, Executive Director, e-mail: gbm@sscc.ru*

² *Trofimuk Institute of Petroleum Geology and Geophysics, Siberian Branch of Russian Academy of Sciences; prospekt Koptyuga 3, Novosibirsk, 630090, Russia; Ph.D., Senior Scientist, e-mail: vctrkstn@mail.ru*

³ *Institute of Computational Mathematics and Mathematical Geophysics, Siberian Supercomputer Center, Siberian Branch of Russian Academy of Sciences; prospekt Lavrentyeva 6, Novosibirsk, 630090, Russia; Senior Specialist, e-mail: kuchin@sscc.ru*

⁴ *Trofimuk Institute of Petroleum Geology and Geophysics, Siberian Branch of Russian Academy of Sciences; prospekt Koptyuga 3, Novosibirsk, 630090, Russia; Ph.D., Scientist, e-mail: 511ssa@mail.ru*

⁵ *Trofimuk Institute of Petroleum Geology and Geophysics, Siberian Branch of Russian Academy of Sciences; prospekt Koptyuga 3, Novosibirsk, 630090, Russia; Dr. Sci., Head of Laboratory, e-mail: CheverdaVA@ipgg.sbras.ru*

Received November 30, 2015

Abstract: An algorithm for solving systems of linear algebraic equations based on the Gaussian elimination method is proposed. The algorithm is aimed to solve boundary value problems for the Helmholtz equation in 3D heterogeneous media. In order to solve linear systems raised from geophysical applications, we developed a parallel version targeted on heterogeneous high-performance computing clusters (MPP and SMP architecture). Using the low-rank approximation technique and the HSS format allows us to solve problems larger than by the use of traditional direct solvers with saving the L -factor in full rank (FR). Using the proposed approach reduces computation time; it is the key-point of 3D geophysical problems. Numerical experiments demonstrate a number of advantages of the proposed low-rank approach in comparison with direct solvers (FR-approaches).

Keywords: Helmholtz equation, algorithms for sparse systems of linear algebraic equations, Gaussian elimination method, low-rank approximation, HSS matrix representation, distributed parallel systems, heterogeneous high-performance computing systems.

References

1. S. A. Solovyev, "Application of the Low-Rank Approximation Technique in the Gauss Elimination Method for Sparse Linear Systems," *Vychisl. Metody Programm.* **15**, 441–460 (2014).
2. A. George, "Nested Dissection of a Regular Finite Element Mesh," *SIAM J. Numer. Anal.* **10** (2), 345–363 (1973).
3. S. Solovyev, D. Vishnevsky, and H. Liu, "Multifrontal Hierarchically Semi-Separable Solver for 3D Helmholtz Problem Using 27-Point Finite-Difference Scheme," in *Proc. 77th EAGE Conference and Exhibition, Madrid, Spain, June 1–4, 2015*, doi 10.3997/2214-4609.201413101
4. S. Chandrasekaran, P. Dewilde, M. Gu, and N. Somasunderam, "On the Numerical Rank of the Off-Diagonal Blocks of Schur Complements of Discretized Elliptic PDEs," *SIAM J. Matrix Anal. Appl.* **31** (5), 2261–2290 (2010).
5. F. Collino and C. Tsogka, "Application of the Perfectly Matched Absorbing Layer Model to the Linear Elastodynamic Problem in Anisotropic Heterogeneous Media," *Geophysics* **66** (1), 294–307 (2001).
6. J. Xia, "Robust and Efficient Multifrontal Solver for Large Discretized PDEs," in *High-Performance Scientific Computing* (London, Springer, 2012), pp. 199–217.
7. S. A. Solovyev and S. Tordeux, "An Efficient Truncated SVD of Large Matrices Based on the Low-Rank Approximation for Inverse Geophysical Problems," *Siberian Electronic Math. Rep.* **12**, 592–609 (2015).
8. B. Glinsky, I. Chernykh, N. Kuchin, et al., "Control and Managing the HPC Cluster in Siberian Supercomputer Center," in *Proc. Int. Conf. Russian Supercomputing Days, Moscow, Russia, September 28–29, 2015* (Mosk. Gos. Univ., Moscow, 2015), pp. 667–675.