УДК 519.178: 519.688 + 519.61 / 673

ПРИМЕНЕНИЕ МНОГОУРОВНЕВЫХ ВЛОЖЕННЫХ СЕЧЕНИЙ ДЛЯ РЕШЕНИЯ БОЛЬШИХ РАЗРЕЖЕННЫХ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ В ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЕ FORTU

Бувайло Д.П., к.ф.-м.н., доцент, Толок В.А., д.т.н., профессор

Запорожский государственный университет

В статье описываются алгоритмы для разложения разреженных матриц, которые существенно улучшают процесс прямого решения разреженных линейных систем – в терминах требуемой памяти и полного объема вычислений. Разрабатываемый пакет программ позволяет эффективно производить нахождение перестановки неизвестных, которая уменьшает размер графа заполнения, путем определения сепараторов приемлемого качества с помощью эвристического алгоритма многоуровневых вложенных сечений.

Ключевые слова: разделение графа; многоуровневые методы разделения графа; перестановки, сокращающие размер графа заполнения; разреженные матричные алгоритмы.

Бувайло Д.П., Толок В.О. ЗАСТОСУВАННЯ БАГАТОРІВНЕВИХ ВКЛАДЕНИХ ПЕРЕТИНІВ ДЛЯ РОЗВ'ЯЗУВАННЯ ВЕЛИКИХ РОЗРІДЖЕНИХ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ / Запорізький державний університет, Україна.

У статті описуються алгоритми для розкладання розріджених матриць, що істотно поліпшують процес прямого розв'язання розріджених лінійних систем – у термінах необхідної пам'яті і повного обсягу обчислень. Розроблюваний пакет програм дозволяє ефективно знаходити перестановки невідомих, котрі зменшують розмір графа заповнення, шляхом визначення сепараторів прийнятної якості за допомогою евристичного алгоритму багаторівневих вкладених перетинів.

Ключові слова: поділ графа; багаторівневі методи поділу графа; перестановки, що скорочують розмір графа заповнення; розріджені матричні алгоритми.

Buvailo D.P., Tolok V.A. APPLICATION OF THE MULTILEVEL NESTED DISSECTIONS FOR THE SOLUTION OF THE LARGE SPARSE LINEAR SYSTEMS / Zaporizhzhya State University, Ukraine

In paper described sparse matrix factorization algorithms, which essentially improve process of the sparse linear systems direct solution – in terms of required memory and calculations. Developed software package allows effectively finding permutation of unknowns, which reduce graph fill-in, by definition separators of acceptable quality with the help of heuristic algorithm of the multilevel nested dissections.

Key words: graph partitioning; multilevel partitioning methods; fill reducing ordering; sparse matrix algorithms.

ВВЕДЕНИЕ

Решение больших разреженных систем линейных уравнений является ядром многих вычислительных проблем в инженерно-конструкторских и научных вычислениях. Такие системы обычно решаются с помощью двух различных групп методов – итерационных и прямых. Характер проблемы определяет, какой метод является более подходящим. Прямой метод для решения разреженной линейной системы $Ax=b$ использует явное разложение разреженной матрицы коэффициентов в произведение нижней и верхней треугольных матриц L и U. Этот шаг требует много времени и потребляет много памяти. Однако, прямые методы важны из-за их общности и устойчивости к накоплению ошибок. Для линейных систем, возникающих в некоторых приложениях, как, например, линейное программирование или структурный инженерный анализ, они зачастую единственно применимые методы решения. Во многих других областях прямые методы также часто предпочтительнее, потому что усилия, потраченные на определение и вычисление хорошего начального приближения для итерационных методов, могут перевешивать стоимость прямого разложения. Кроме того, прямые методы обеспечивают эффективные средства для многократного решения системы с той же самой матрицей коэффициентов и различными правыми частями, потому что необходимое разложение может быть выполнено только однажды.

При разложении разреженной матрицы A общего вида для обеспечения численной устойчивости необходима та или иная форма *выбора главного элемента*, т. е. перестановки строк и/или столбцов. Таким образом, при заданной матрице A обычно получают разложение для $P \cdot A \cdot Q$, где P и Q – матрицы перестановок соответствующих размеров. Заметим, что умножение на P слева переставляет строки A, а умножение на Q справа переставляет столбцы A. При разложении разреженной матрицы общего вида эти перестановки определяются в процессе разложения путем компромисса между конкурирующими требованиями численной устойчивости и разреженности. Различные матрицы, хотя бы они и имели

одинаковую структуру нулей-ненулей, обычно приводят к различным P и Q и, следовательно, имеют множители с различной структурой разреженности. Другими словами, для разреженных матриц общего вида, как правило, нельзя предсказать, где произойдет заполнение, пока не начались собственно вычисления. Это вынуждает пользоваться различными схемами динамического хранения, в которых память для заполнения выделяется в ходе разложения.

Матрицы, получаемые в инженерно-конструкторских и научных вычислениях, обычно соответствуют минимуму некоторого энергетического функционала и являются симметричными, положительно определенными. Симметричное исключение в применении к симметричной положительно определенной матрице не требует перестановок для поддержания численной устойчивости. Прямые методы для таких матриц основаны на разложении матрицы A в произведение треугольных матриц и требуют хранения треугольного множителя (целиком или частично). Метод Холецкого [1] $\mathbf{A=L\cdot L^T}$, где L – нижняя треугольная матрица, применим для положительно определенных матриц. Метод Мартина [1] $\mathbf{A=L\cdot D\cdot L^T}$, где D – дополнительная диагональная матрица, применим как для положительно определенных, так и для более общих знаконеопределенных несингулярных эрмитовых матриц. Поскольку $P\cdot A\cdot P^T$ также симметрична и положительно определена при любой матрице перестановки P, это значит, что можно симметрично переупорядочить A, во-первых, не заботясь о численной устойчивости и, во-вторых, до начала реального численного разложения.

Эти возможности, обычно отсутствующие в случае матрицы A общего вида, имеют важнейшие практические последствия. Раз упорядочение можно определить до начала разложения, то можно определить также и местоположение заполнения, которое произойдет при разложении. Поэтому способ хранения L можно выбрать до реального численного разложения, так же как и зарезервировать место для элементов заполнения. Вычисления затем происходят при структуре хранения, остающейся статичной (неизменной). Таким образом, три задачи: 1) выбор надлежащего упорядочения; 2) формирование подходящей схемы хранения; 3) реальные вычисления – могут быть разделены как самостоятельные объекты исследования и как разные модули программного обеспечения. Эта независимость задач имеет ряд отчетливых преимуществ. Она поощряет модульность при составлении математического обеспечения. Во многих инженерно-конструкторских приложениях приходится решать большое количество различных задач с положительно определенными матрицами, имеющих одинаковую структуру. Ясно, что упорядочение и формирование схемы хранения можно выполнить лишь однажды; поэтому желательно, чтобы эти этапы были изолированы от реальных вычислений.

Трудно получить аналитические выражения для числа арифметических действий при разложении и для размера полученного треугольного множителя (в терминах числа вхождений отличных от нуля) для общей разреженной матрицы. Заполнение позиций ненулями в ходе разложения разреженной матрицы зависит от числа и позиций ненулей в первоначальной матрице. Пусть A – заданная симметричная матрица, а P – матрица перестановки. Хотя структуры ненулевых элементов A и PAP^T разные, их размеры одинаковы: $|Nonz(A)| = |Nonz(PAP^T)|$. Однако, и это обстоятельство большого значения, между $|Nonz(L(A))|$ и $|Nonz(L(P\cdot A\cdot P^T))|$ для некоторой перестановки P разница может быть очень велика.

В идеале мы хотели бы найти перестановку P^*, которая бы минимизировала размер структуры ненулевых элементов матрицы заполнения: $|Nonz(L(P^*AP^{*T}))| = \min |Nonz(L(P\cdot A\cdot P^T))|$. Не существует эффективного алгоритма для отыскания такой оптимальной P^* в случае произвольной матрицы – эта задача имеет сложность NP [8]. Поэтому приходится использовать эвристические алгоритмы, которые производят упорядочение с приемлемо малым, хотя и не обязательно минимальным размером заполнения $|Nonz(L(P\cdot A\cdot P^T))|$.

ВЛОЖЕННЫЕ СЕЧЕНИЯ

Существует эвристический подход к проблеме минимизации размера треугольного множителя, известный под общим названием метода вложенных сечений [9,10,11]. Пусть каждому неизвестному в исходной системе уравнений соответствует узел графа, а ненулевому элементу матрицы коэффициентов – ребро между узлами графа. Допустим, что каким-либо методом мы нашли в графе эффективный узловой сепаратор S, т.е. множество узлов, удаление которых разделяет граф на две приблизительно равные несвязные части A и B (рис. 1). Перенумеруем неизвестные так, чтобы вначале были расположены узлы из части A, затем из части B, и в конце – узлы из сепаратора. Разложение переупорядоченной таким образом матрицы проиллюстрировано на рис. 2.

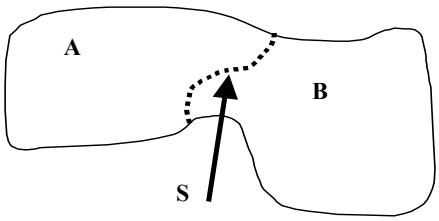


Рис.1. Схематическое изображение планарного графа и сепаратора в нем.

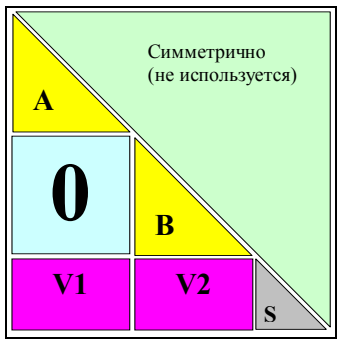


Рис.2. Нумерация неизвестных при разложении матрицы.

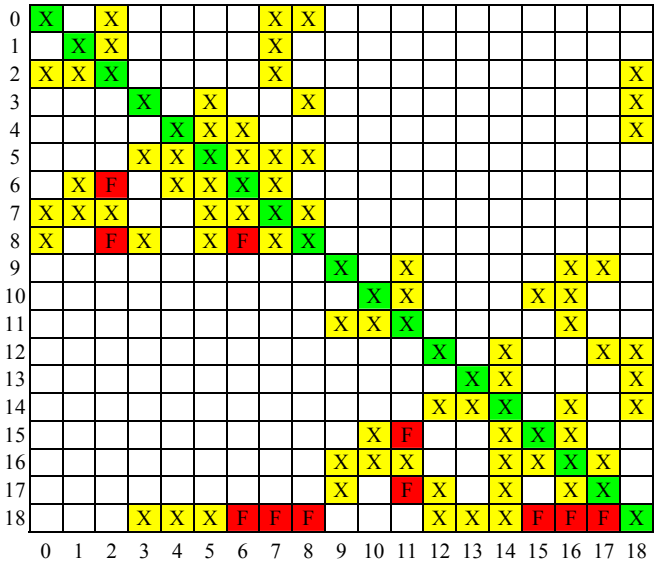
Основным результатом перестановки неизвестных является появление большого нулевого блока, элементы которого всегда будут равны нулю, независимо от реальных числовых значений ненулевых элементов исходной матрицы. Процесс нахождения узловых сепараторов и соответствующих нулевых блоков можно продолжить, разделив несвязные части A и B. Следуя стратегии рекурсивного разбора, можно процесс деления продолжать и далее, маркируя узлы графа и выделяя его части в меньшие подграфы, пока, наконец, мы не придем к далее неделимому графу, которому соответствует небольшая плотная матрица. После удаления столбец, соответствующий любому из узлов в двух подграфах может создавать заполнение только в столбцах, соответствующих узлам разделителя, но не узлам другого подграфа. Интуитивно меньшие разделители означают меньшее заполнение и работу при разложении. Для конечно-элементных графов с некоторыми непрерывными свойствами было доказано [9], что эта методика выдает упорядочения, отличные от оптимального в пределах некоторого постоянного множителя: $L_{эвр} < C \cdot L_{опт}$. Такие пределы не могут быть доказаны для произвольных матриц без четкой структуры, вроде разреженных матриц, возникающих в LP вычислениях. Однако стратегия на практике работает хорошо даже для матриц с произвольными образцами разреженности.

Кроме уменьшения количества требуемой памяти и объема необходимых вычислений, при поиске оптимальной перестановки преследуются и другие цели. Для разложения столбца, соответствующего какому-то неизвестному k, необходимо ранее разложить все столбцы, аккумуляция которых необходима для вычисления k-го столбца. Это соответствие порядка можно представить в виде древовидного графа, в котором отношения потомок-предок строятся на основе порядка разложения столбцов. Разложение разреженной матрицы можно рассматривать как просмотр этого дерева исключения, при котором сначала раскладываются столбцы, соответствующие всем потомкам данного узла k, а затем строится разложение для самого узла k. В общем случае для произвольной, в том числе и оптимальной, перестановки неизвестных дерево исключения не обладает какими-либо специальными свойствами. Оно может быть достаточно несбалансированным.

Анализ дерева исключения позволяет выявить в процессе разложения матрицы значительное количество вычислений, выполнение которых может быть произведено параллельно. Действительно, поддеревья одного корня могут быть вычислены одновременно. При параллельном разложении разреженной матрицы на первое место выходит не просто уменьшение количества требуемой памяти и объема необходимых вычислений, но создание такого дерева исключения, которое хорошо сбалансировано и имеет достаточно простой вид. Сбалансированность дерева исключения дает сбалансированность распределения вычислительной нагрузки на процессоры параллельной системы, а простота дерева – уменьшает сложность алгоритма назначения заданий процессорам.

Алгоритм вложенных сечений обладает несомненным преимуществом перед всеми другими алгоритмами переупорядочения разреженной матрицы. Порождаемое им дерево исключения всегда является на нескольких своих верхних уровнях бинарным деревом. Это позволяет достаточно просто распределять вычислительную нагрузку делением задачи на две подзадачи. Разделения графа на две приблизительно равные части дает хорошую сбалансированность дерева исключения. Таким образом, применение вложенных сечений позволяет эффективно масштабировать процесс разложения разреженной матрицы на доступное количество процессоров. Рис. 3 дает пример разреженной матрицы и соответствующего ей «хорошего» дерева исключения.

Успех применения метода вложенных сечений целиком зависит от эффективности нахождения сепараторов. Известно, что проблема определения наилучшего сепаратора в графе является NP-полной проблемой. Таким образом, от одной NP-полной задачи мы перешли к последовательности NP-полных задач. Однако в последнее время разработаны эффективные эвристические алгоритмы, направленные на поиск сепараторов приемлемого качества в графах.



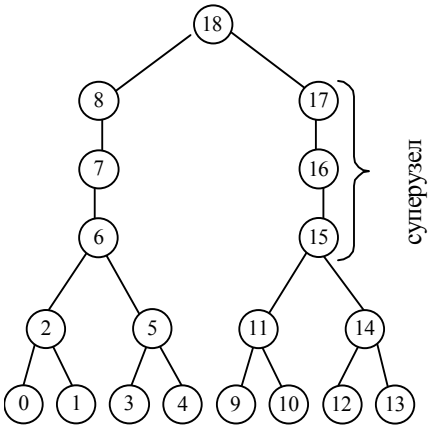


Рис.3. Дерево исключения

X – Диагональные ненулевые элементы

X – Внедиагональные ненулевые элементы исходной матрицы

F – Внедиагональные ненулевые элементы заполнения

МНОГОУРОВНЕВЫЙ АЛГОРИТМ БИСЕКЦИИ ГРАФА

В последнее время интенсивно развиваются многоуровневые алгоритмы – быстрое и эффективное средство вычисления разделителей в графе [11,12]. Этот класс алгоритмов разделения графа уменьшает размер графа до приемлемой величины, огрубляя граф путем стягивания его узлов и ребер, делит меньший граф, и затем уточняет разделитель, чтобы построить разделение для первоначального графа. Следует отметить, что многоуровневые алгоритмы не решают сами по себе проблему разделения графа. Под этим названием скрывается общий подход, основанный на переходе от проблемы разделения большого графа к проблеме разделения меньшего графа. Для меньшего графа затем гораздо дешевле использовать любой метод разделения. При этом уменьшается время вычисления разделения за счет несколько худшего качества разделения. Типичный многоуровневый разделяющий граф алгоритм имеет четыре компоненты: стягивание, начальное выделение разделителя, восстановление графа, усовершенствование разделителя.

Цель фазы стягивания состоит в том, чтобы уменьшить размер графа при сохранении его свойств, которые существенны и необходимы для нахождения хорошего разделителя. Первоначальный граф представляется как взвешенный граф с назначенными каждому ребру и узлу весами. На каждом шаге стягивания строится паросочетание – максимальный набор ребер, таких что никакие два ребра не имеют общей вершины. Ребра из этого набора удаляются, и два узла, связанные ребром из паросочетания, сокращаются в один узел, чей вес – сумма весов составляющих узлов. Стягивание также приводит к сокращению многих ребер и пересчету их весов.

Для взвешенного графа после любой стадии стягивания, имеется несколько возможных путей построения паросочетания для следующего шага стягивания. Karypis и Kumar описывают эвристику [6], известную как стягивание тяжелого ребра (HEM), чтобы помочь в выборе соответствия, которое не только уменьшает время выполнения компоненты усовершенствования выделения разделов графа, но также и имеет тенденцию генерировать части с маленькими разделителями. Стратегия состоит в том, чтобы беспорядочно выбрать необработанный узел, выбрать ребро с самым большим весом среди ребер, инцидентных этому узлу, которые соединяют этот узел с другим необработанные узлом, и отмечают оба узла, связанный этим ребром как обработанные. Вес ребра, соединяющего два узла в стянутой версии графа – число ребер в первоначальном графе, которые соединяют два набора первоначальных узлов, сокращенные в два грубых узла. HEM, поглощая более тяжелые грани, генерирует грубые графы, чьи узлы свободно связаны оставшимися легкими ребрами, таким образом, гарантируя, что разделитель грубого графа соответствует хорошему разделителю первоначального графа.

HEM может пропускать некоторые тяжелые ребра в графе, потому что узлы посещаются беспорядочно. Например, рассмотрим узел i, самый тяжелый инцидент которого – ребро, соединяющее его с узлом j. Если i посещен прежде j, i и j необработаны, ребро (i, j) будет включено в соответствие. Если существует ребро (j, k), такое, что i и k не связаны, оно не войдет в соответствие, даже если оно намного более тяжелое чем (i, j), потому что j более не доступен для соответствия. Чтобы преодолеть эту проблему, после первых немногих шагов стягивания, переходят к тому, что называется соответствием самого тяжелого ребра (SHEM). Ребра сортируют по их весам и посещают в порядке уменьшения веса, чтобы осмотреть их для возможного включения в соответствие. Связи распределены беспорядочно. Выгода паросочетаний из самых тяжелых ребер над паросочетанием из просто тяжелых ребер проявляется на более поздних стадиях стягивания, где сортировка не слишком дорога, потому что граф значительно сжался от первоначального размера.

Эвристики на основе паросочетаний уменьшают число узлов в графе приблизительно вдвое на каждой стадии стягивания. Следовательно, количество шагов стягивания и рафинирования для разделения графа из n узлов – $\log_2(n)$. Если r (вместо 2) узлов графа стягиваются в один на каждом шаге стягивания, общее количество шагов может быть уменьшено до $\log_r(n)$. Меньшее количество шагов, вероятно, уменьшит и полное время выполнения. Однако, если r увеличивается, задача усовершенствования после каждого шага восстановления становится тяжелее. Это влияет и на время выполнения и на качество усовершенствования. HTM объединяет три узла одновременно, выбирая необработанный узел наугад и согласовывая это с двумя из соседей так, что сумма весов из трех ребер, соединяющих три узла максимизируется над всеми парами соседей выбранного узла. Несуществующее ребро между двумя соседями расценивается как ребро весом ноль.

Фазу стягивания останавливают, когда граф уменьшен от первоначальных нескольких тысяч до нескольких сотен узлов, и используют любую эвристику, чтобы вычислить первоначальный разделитель в грубом графе за существенную часть времени выполнения. Gupta рекомендует продолжать стягивание графа, пока он не содержит точно 2 узла. Этот грубый граф служит как хорошее начальное выделение разделителя, при условии, что алгоритм стягивания не позволяет узлу участвовать в соответствии, если его вес превышает n/2.

Компоненты восстановления графа и усовершенствования разделителя работают совместно. Первоначально, 2 узлам стянутого графа назначаются метки, указывающие, что они принадлежат различным начальным частям. Узлы восстанавливаемого графа наследуют метки из их родительских узлов в предыдущем грубом графе. На любой стадии восстановления, ребра, соединяющие пару узлов, принадлежащих к различным частям, составляют реберный разделитель графа. После каждого шага восстановления разделитель усовершенствуется. При усовершенствовании реберного разделителя делаются попытки минимизировать его общий вес, перемещая часть узлов, если это переключение уменьшает размер разделителя и не нарушает баланса между частями. После стягивания первоначального графа из n узлов к 2 узлам, восстанавливаем его до q узлов, где $2<q<n$, при усовершенствовании реберного разделителя после каждого шага восстановления. В этой стадии мы сохраняем разделитель, огрубляем q-узловой граф обратно к k узлам, и повторяем восстановление и усовершенствование разделителя. Этот процесс повторяется несколько раз, и из всех разделителей сгенерированного q-графа, мы выбираем самый лучший для дальнейшего восстановления, чтобы получить разделитель первоначального n-графа. Рандомизация в процессе стягивания гарантирует, что различные испытания заканчиваются различными разделителями. Большая часть стоимости вычисления разделителя приходится на первые несколько шагов стягивания первоначального графа и последние шаги восстановления, когда размер графа велик. Следовательно, многократные циклы стягивания и восстановления на меньшем, грубом графе с q узлами имеют небольшое воздействие на полное время выполнения всего алгоритма. Для рафинирования разделителя используется Fiduccia-Mattheyses [5] – линейно-затратная разновидность эвристики Kernighan-Lin [4].

Недавно было предложено несколько многоуровневых алгоритмов [11,12], которые улучшают процесс уточнения разделителя на стадии обратного увеличения стянутого графа. Эти схемы имеют тенденцию давать хорошие разделения по разумной стоимости. Bui и Jones используют случайное максимальное паросочетание к последовательно огрубляемым графам вплоть до предельного размера в несколько сотен узлов. Далее они делят самый маленький граф и затем рафинируют граф уровень за уровнем, применяя Kernighan-Lin эвристику, чтобы улучшить разделение. Hendrickson и Leland расширяют этот подход, используя веса ребер и узлов, чтобы контролировать стягивание узлов и ребер. Эта последняя работа показала, что многоуровневые схемы могут обеспечивать лучшее разделение, чем спектральные методы, при более низкой стоимости для самых разнообразных конечно-элементных проблем.

При вычислении упорядочения, уменьшающего заполнение разреженной матрицы, критерием бисекции является размер разделителя, который должен быть минимизирован. Современные GP-основанные алгоритмы упорядочения следуют двум различным подходам к нахождению маленького разделителя. Karypis и Kumar совершенствуют разделитель между двумя подграфами после каждого шага

восстановления так, чтобы немногочисленная группа ребер соединяла узлы различных подграфов в заключительном выделении частей первоначального графа. Затем они используют алгоритм для нахождения минимального одностороннего покрытия, чтобы вычислить узловой разделитель из реберного разделителя. Этот подход очень полагается на условие, что размер узлового разделителя пропорционален размеру реберного разделителя. Это предположение часто неправильно, особенно для очень неструктурированных матриц. Ashcraft и Liu находят узловой разделитель уже внутри грубого графа и далее совершенствуют узловой разделитель. Не все составляющие узлы грубого узла разделителя могут иметь ребра, пересекающие границы раздела. Поскольку граф рафинируется, такие узлы удаляются из узлового разделителя.

Как только разделитель первоначального графа найден и удален, вся процедура рекурсивно повторяется для двух разъединенных подграфов. Рекурсия завершается, если подграф слишком мал. Далее используется разновидность AMD или MMD эвристики минимальной степени для упорядочения этих маленьких подграфов. После того, как узлы из двух подграфов в любом уровне рекурсии были помечены, и узлы разделителя в том уровне помечены, алгоритм возвращается к предыдущему уровню.

ПРИМЕР ПРИМЕНЕНИЯ РАЗРЕЖЕННОГО РАЗЛОЖЕНИЯ

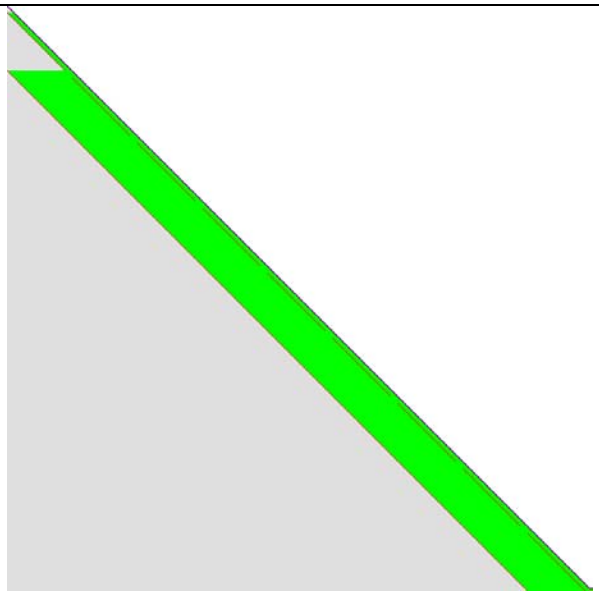
Разрабатываемый пакет программ позволяет использовать несколько различных эвристических алгоритмов для поиска приемлемого переупорядочения неизвестных.

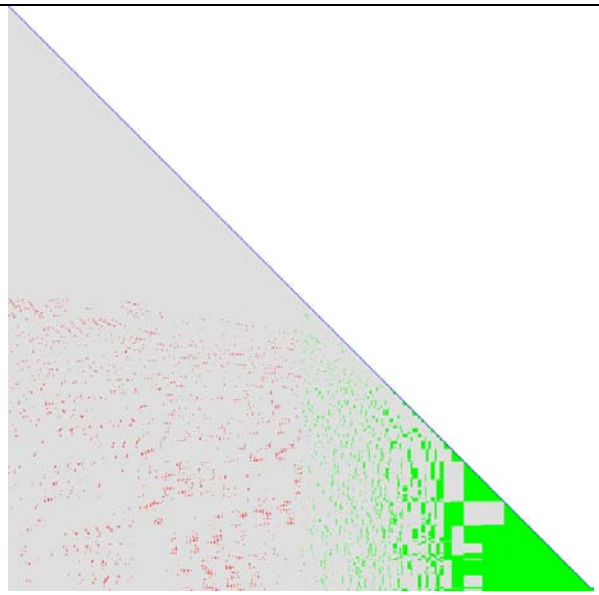
Самой популярной схемой уменьшения заполнения среди используемых является алгоритм минимальной степени (MD) [7,8]. В основе алгоритма – следующее наблюдение. Пусть помечены узлы $\{x_1,x_2,...,x_{i-1}\}$. Число ненулевых элементов в этих столбцах в дальнейшем не меняется. Ясно, что для уменьшения числа ненулевых элементов i-го столбца в еще не факторизованной матрице на место i-го столбца нужно перевести столбец с наименьшим числом ненулевых элементов. Другими словами, схему можно рассматривать как метод уменьшения заполненности матрицы путем локальной минимизации $|Nonz(L_k)|$ для разлагаемой матрицы. Начальный успех MD-основанных эвристик вызвал их интенсивное развитие, улучшение времени выполнения и качества. Лучшими представителями семейства MD-основанных эвристик на сегодня являются multiple minimal degree (MMD) алгоритм (George, Liu) [8] и approximate minimal degree (AMD) алгоритм (Davis, Amestoy, Duff) [7]. В инструментальной системе FORTU реализованы два алгоритма на основе эвристики минимальной степени и агрессивной абсорбции MMD и AMD. Для представления последовательности графов исключения использовалась структура фактор-графа.

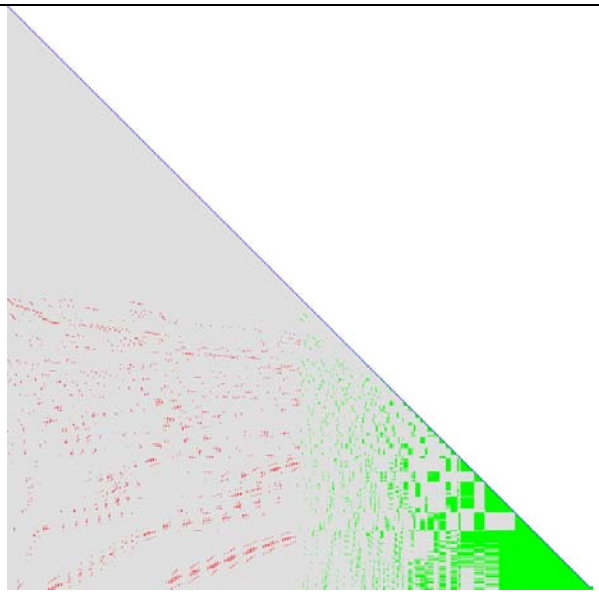
На основе метода вложенных сечений для инструментальной системы FORTU реализованы два эвристических алгоритма с определением сепараторов по многоуровневой схеме, предложены эвристические решающие правила для этапов стягивания, начального выделения разделителя, усовершенствования разделителя.

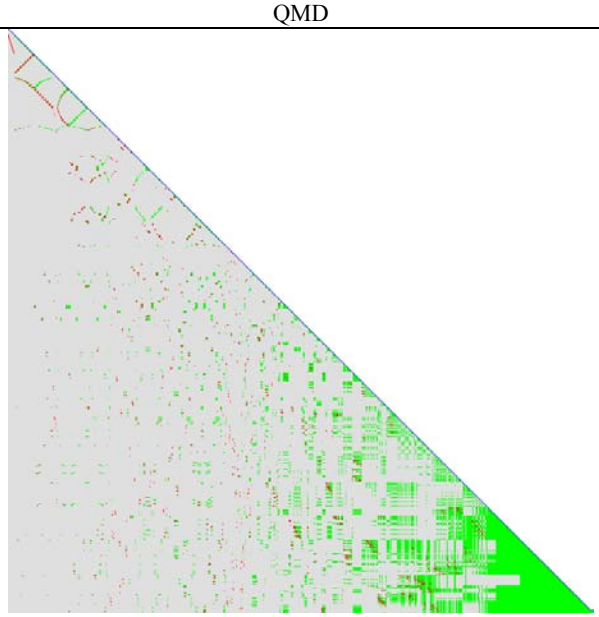
В качестве иллюстрации разных эвристик приведем результаты их применения для кубической решетки 9x9x9. Для этой небольшой задачи можно наглядно показать структуру полученного разными методами треугольного множителя. На рисунках разными оттенками показаны: исходные ненулевые элементы, элементы заполнения, диагональные элементы, нули. В таблице дано сравнение характеристик треугольного множителя для разных эвристик:

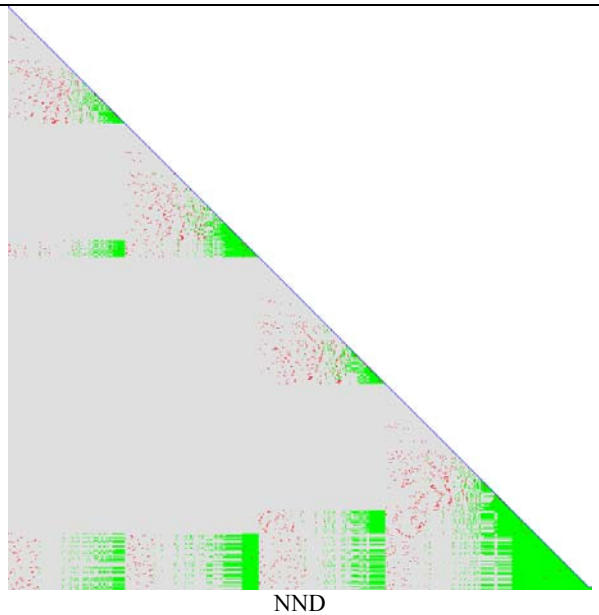
Название эвристики	Размер множителя L	Размер структуры индексов	Число операций	Память (байт)
NAT – наилучшая ленточная структура	53144	не используется	4132640	509276
QMD – quotient minimal degree	19833	5927	1270450	266892
MMD – multiple minimal degree	17761	5620	949223	249084
AMD – approximate minimal degree	24867	13072	1350200	335740
END – multilevel nested dissection (edge refinement)	17167	5972	858705	245740
NND – multilevel nested dissection (node refinement)	20410	7002	1130150	275804


NAT


QMD


MMD


AMD


NND

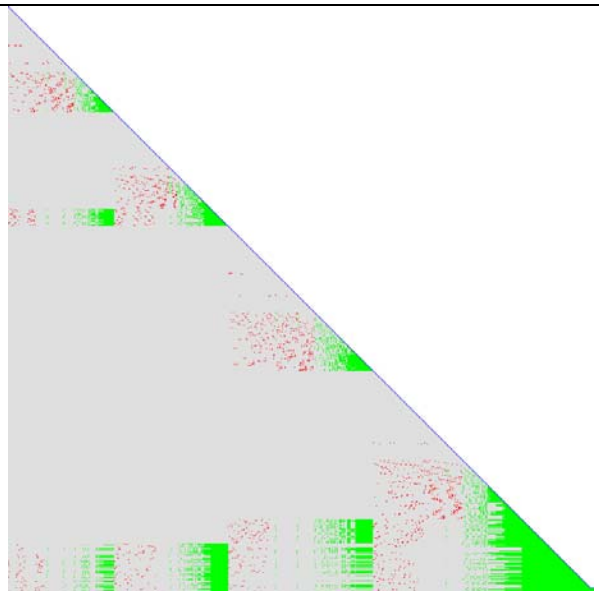

END

Рис.4 Портреты треугольного множителя при разных эвристиках переупорядочения неизвестных

ЗАКЛЮЧЕНИЕ

Эксперименты с многоуровневыми схемами показали, что они хорошо работают для многих различных типов огрубления, начального разделения, рафинирования. В частности, все схемы огрубления хорошо учитывают глобальные свойства графа, а Kernighan-Lin алгоритм или его варианты, используемые для обработки на стадии восстановления, хорошо учитывают локальные свойства графа. Из-за объединения глобального и локального взгляда на граф, обеспеченного схемами огрубления и восстановления, выбор алгоритма для деления грубого графа мало влияет на общее качество разделения.

Для анализа качества получаемых упорядочений производилось их сравнение с перестановками, производимыми эвристиками минимальной степени [10]. Лучшими представителями семейства MD-основанных эвристик являются multiple minimal degree (MMD) алгоритм (George, Liu) и approximate minimal degree (AMD) алгоритм (Davis, Amestoy, Duff). Сравнение показало, что GP-основанные эвристики способны к созданию упорядочений лучшего качества, чем MD-основанные эвристики для конечно-элементных задач при несколько худших, но того же порядка, затратах времени. Но время, потерянное на этапе поиска перестановки неизвестных, с лихвой компенсируется за счет уменьшения времени численного разложения матрицы коэффициентов.

ЛИТЕРАТУРА

1. Джордж А., Лиу Дж. Численное решение больших разреженных систем уравнений. Пер. с англ. – Москва: Мир,1984.–333 с.
2. J. R. Bunch, D. J. Rose, ed., Sparse Matrix Computations, Academic Press (1976)
3. Cleve Ashcraft and Joseph W.-H. Liu, Generalized Nested Dissection: Some Recent Progress, Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, Snowbird, UT, June 1994.
4. B. W. Kernighan and S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, Bell Syst. Tech. J., 1970.
5. C. M. Fiduccia and R. M. Mattheyses, A Linear Time Heuristic for Improving Network Partitions, Proceedings of the 19th IEEE Design Automation Conference, 1982, pp. 175 181.
6. George Karypis and Vipin Kumar, METIS: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Technical report, Department of Computer Science, University of Minnesota, Minneapolis, 1995.
7. Gupta, Fast and effective algorithms for graph partitioning and sparse-matrix ordering, IBM Journal of Research and Development, vol. 41, No. ½
8. M. Yannakakis, Computing the Minimum Fill-In Is NP-Complete, SIAM J. Alg. Discrete Meth. 2, 77-79 (1981).
9. R. J. Lipton, D. J. Rose, and R. E. Tarjan, "Generalized Nested Dissection," SIAM J. Numer. Anal. 16, 346-358 (1979).
10. Alan George and Joseph W.-H. Liu, The Evolution of the Minimum Degree Ordering Algorithm, SIAM Rev. 31, 1-19 (March 1989).
11. Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. Technical Report SAND93-1301, Sandia National Laboratories, 1993.
12. T. Bui and C. Jones. A heuristic for reducing fill in sparse matrix factorization. In *6th SIAM Conf. Parallel Processing for Scientific Computin*g, 1993. - P. 445–452.