Разработка прямого решателя для разреженных систем линейных уравнений с симметричной положительно определенной матрицей*

Е.А. Козинов, И.Г. Лебедев, С.А. Лебедев, А.В. Линев, А.Ю. Малова, И.Б. Мееров, А.В. Сысоев, Т.А. Сысоева, С.С. Филиппенко

Нижегородский государственный университет им. Н.И. Лобачевского

Рассмотрен подход к решению СЛАУ с разреженной симметричной положительно-определенной матрицей. Описана поэтапная схема решения СЛАУ с использованием методаХолецкого. Выполнена базовая последовательная программная реализация представленной схемы. Реализована модификация решения на основе супернодального подхода, выполнены ее последовательная и параллельная реализации. Приведены результаты экспериментов, дано их сравнение с результатами, полученными с помощью некоторых известных библиотек.

1. Введение

Одной из актуальных задач алгебры разреженных матриц является решение систем линейных алгебраических уравнений $Ax = b$ с разреженной симметричной положительно определенной матрицей A. Разработка алгоритмов решения таких задач и их высокопроизводительная программная реализация, ориентированная на современные вычислительные архитектуры, представляет большой практический интерес.

На сегодняшний день в мире разработано большое количество специализированного программного обеспечения для решения больших разреженных СЛАУ – так называемые «решатели» СЛАУ. В соответствии с используемыми методами эти решатели подразделяются на прямые и итерационные, некоторые из них имеют высокопроизводительные реализации. В данной работе идет речь о разработке прямого решателя разреженных СЛАУ. Среди известныхпрямых решателей – MKL PARDISO, SuperLU, MUMPS, CHOLMOD и многие другие. На сегодняшний день существуют решатели, ориентированные на различные режимы работы: последовательный, параллельный для систем с общей памятью, параллельный для систем с распределенной памятью. Некоторые из решателей поддерживают режим работы out-of-core, используя жесткий диск в качестве «продолжения» оперативной памяти.Часть решателейработает только для симметричных положительно определенных матриц, другие – для матриц общего вида.Постоянно обновляемый обзор прямых решателей от авторов SuperLU можно найти по ссылке http://crd.lbl.gov/~xiaoye/SuperLU/SparseDirectSurvey.pdf.Сравнительный анализ функциональности программного обеспечения, реализующего численные методы линейной алгебры, включая прямые и итерационные решатели разреженных СЛАУ, расположен на странице Дж. Донгарра (J. Dongarra):http://www.netlib.org/utk/people/JackDongarra/la-sw.html.

Задачей коллектива авторов является разработка собственной реализации высокопроизводительного прямого решателя разреженных СЛАУ с симметричной положительно определенной матрицей A. Мотивация для создания своего программного комплексасостоит в потенциальной возможности создания конкурентноспособногоинструмента, а также перспективах его использования в учебном процессе.В данной работе приведены текущие результаты проекта, дано их сравнение с результатами некоторых известных библиотек, а также определены пути дальнейшего развития.

* Работа выполнена в лаборатории «Информационные технологии» ВМК ННГУ впри поддержке проекта «Подготовка и переподготовка профильных специалистов на базе центров образования и разработок в сфере информационных технологий», госконтракт № 07.P20.11.0030.

2. Постановка задачи

Дана система линейных уравнений:

$$Ax = b \quad (1)$$

Здесь A – разреженная симметричная положительно определенная матрица, b– плотныйвектор, x – вектор неизвестных. Необходимо найти решение системы x.

3. Метод решения

Прямые методы решения задачи (1), как правило, основаны на применении разложения Холецкого к матрице A в виде:

$$A = U^T U \quad (2)$$

где U – верхнетреугольная матрица. В этом случае решение системы сводится к последовательному решению двух треугольных систем:

$$U^T y = b, Ux = y \quad (3)$$

Особенностью процедуры разложения Холецкого для разреженной матрицы является то, что матрица обычно претерпевает заполнение, что на практике может привести к неудовлетворительным требованиям по памяти. Степень заполненности матрицы можно уменьшить с помощью переупорядочивания ее строк и столбцов. Это соответствует нахождению матрицы перестановки Pи переходу к эквивалентной системе (4):

$$\bar{A}(Px) = Pb, \bar{A} = PAP^T \quad (4)$$

Таким образом, при численном решении разреженной системы с использованием методаХолецкого можно выделить следующие этапы: переупорядочивание – вычисление матрицы перестановки P и переход к системе (4); символическое разложение – построение портрета матрицы U, выделение памяти для хранения ненулевых элементов; численное разложение – вычисление значений матрицы U и размещение их в выделенной памяти; обратный ход – решение треугольных систем уравнений (3).

4. Программная реализация

4.1 Базовый подход

На основе описанного вышеметода решения выполнена последовательная программная реализация решателя СЛАУ и ее модификация с применением супернодального подхода. Подготовленапараллельная версия для систем с общей памятью. Программная реализация выполнена на языке C. Для хранения разреженных матриц выбран формат CRS. Вычисления проводились в двойной точности. Рассмотрим подробнее реализацию каждого этапа метода:

Этап 1. Для оптимизации заполнения фактора был использован *метод вложенных сечений* (nesteddissection)[6], основанный на разбиении графа матрицы при помощи разделителей. По сравнению с методом минимальной степени, он позволяет достичь большей степени параллелизма на стадиях факторизации. Реализация метода вложенных сечений выполнена в соответствии с описанием, приведенномв [14]. Реализована модификация метода с применением алгоритма уменьшения размера разделителя, предложенного в [2].

Этап 2. Во время *символической фазы* подготавливаются структуры данных для факторизации. Задачамиэтого этапа являются вычисление шаблонарасположения ненулевых элементов и построение вспомогательных структур данных для численной фазы.Так, для нахождения числа ненулевых элементов в факторе и выполнения параллельных вычислений строится специальная структура – *дерево исключения*. Оно отражает зависимость между строками матрицы: если вершины дерева не соединены ребром, то соответствующие им строки фактора могут вычисляться параллельно. В рассматриваемой реализациипостроение дерева исключения выполняется по алгоритму, описанному в [13]. Кроме того, к переупорядоченной на этапе 1 матрице дополнительно применяется *постперестановка*на основеалгоритма, приведенного в [3]. Этот прием повышает структурированность портрета фактора, не ухудшая его заполненности, что позволяет ускорить численную фазу факторизации. Далее выполняется *оценка общего числане-*

нулевых элементов в факторе и выделяется память для массива столбцовых индексов по алгоритму, предложенномув [7]. Непосредственно *символическая факторизация* выполняется на основе процедуры, описанной в [15].
Этапы 3-4. Во время *численной фазы* выполняется нахождение значений элементов верхнего треугольника. Это самая трудоемкая часть факторизации, состоящая в последовательном рассмотрении всех строк исходной матрицы и проведении операции Гауссова исключения для всех ненулевых элементов, лежащих левее главной диагонали. Базовая версия численной части (этап 3), а также *обратный ход*(этап 4) реализованы в соответствии с работой [15].

4.2 Суперноdальный подход

Недостатком базового подхода к разложению Холецкого является низкая производительность на матрицах больших размерностей из-за возникновения существенного количества кеш-промахов. Для решения этой проблемы существует два широко распространенных подхода: суперноdальный (supernodal, «суперэлементный») и мультифронтальный (multifrontal).

*Суперноdальный подход*для алгоритма факторизации использует так называемые «*суперноды*» (supernode) и позволяет производить факторизацию поблочно с применением матричных операций BLAS. Повышение производительности в этом случае можно получить путем применения оптимизированных реализаций BLAS для плотных подматриц, повышения эффективности работы с памятью, создания базы для распараллеливания. Подобный подход используется в таких решателях, как MKL Pardiso, CHOLMOD и др.

Мультифронтальный подход (авторы Дюфф и Рэйд [5]) состоит в разбиении всего процесса факторизации на факторизацию небольших плотных матриц [11, 12]. При этом используются те же механизмы повышения производительности. Одним из наиболее известных решателей, реализующих данный подход, является MUMPS.

В данной работе для модификации численной части использовался суперноdальный подход. Для выделения блоков применялись так называемые *ослабленные суперноды* (relaxedsupernode [1]) – группы строк, имеющих различие в структуре левее плотного треугольного блока не более чем в фиксированном числе элементов. *Выделение супернодов* выполняется *после символической части* на основе процедур, введенных в [1].При этом задается параметр огрубления, отвечающий за количество нулей, допускаемых в BCRS-блоке матрицы, что позволяет сформировать блоки большего размера и сократить время работы матричных операций. Однако сильное увеличение параметра может приводить к неудовлетворительным требованиям по памяти, его оптимальное значение может быть подобрано под конкретную задачу.

Модифицированная численная фаза при суперноdальном подходе состоит из нескольких этапов, а именно: перевод исходной матрицы в формат BCRS на основе выделенных супернодов; выполнение факторизации с использованием матричных операций (BLAS 3 уровня).Для устранения необходимости перевода результата факторизации в формат CRS реализован алгоритм обратного хода метода Гаусса для формата BCRS. Реализация указанных алгоритмовоснована наработах [4], [8-10]. Использовались функцииBLASиз библиотеки IntelMKL.

4.3 Параллельная версия для систем с общей памятью

Авторами выполнена реализация параллельной версии решателя для систем с общей памятью на основе суперноdального подхода. На данном этапе работы распараллелена численная фаза разложения Холецкого, как наиболее трудоемкий этап вычислений.

Схема параллельных вычислений формировалась на основе модификации дерева исключения, построенного по BCRS-портрету матрицы. В этом случае дерево исключений показывает, над какими группами строк операции могут выполняться независимо. Для минимизации накладных расходов при организации паралеллизма и равномерной загрузки потоков по дереву исключения строится так называемое дерево распараллеливания. При этомобъединеняются некоторые вершины дерева исключения, принадлежащие одному и тому же поддереву. Вычисления, соответствующие отдельному узлу, могут выполняться независимо.

Распараллеливаниевыполнено на основе потоков Windowsи шаблона параллельного программирования «мастер – рабочий». Назначение задач «потокам-рабочим» происходит согласс-

но дереву распараллеливания. Обработка вершин дерева начинается с листьев. Очередная вершина становится доступной для независимых вычислений и назначается «потоку-рабочему» после завершения вычислений для ее потомков.

5. Результаты экспериментов

Для анализа производительности программного комплекса был проведен ряд экспериментов на матрицах из коллекции [16] университета Флориды. Характеристики тестовых матриц представлены в таблице ниже (Таблица 1). Все они являются симметричными положительно определенными.

Таблица 1. Характеристики тестовых матриц

Название матрицы	Порядок	Число ненулевых элементов	Заполненность, %
pwtk	217918	5926171	0,0125
msdoor	415863	10328399	0,0060
parabolic_fem	525825	2 100225	0,0008
tmt_sym	726713	2903837	0,0005
G3_circuit	1585 478	4 623152	0,0002
ecology2	999999	2997995	0,0003
audikw_1	943695	39297771	0,0044

Параметры тестовой инфраструктуры приведены в следующей таблице (Таблица 2):

Таблица 2. Параметры тестового окружения

Процессор	2 четырехъядерных процессора Intel® Xeon E5520 (2.27 GHz)
Память	16 GB
Операционная система	Windows 7
Среда разработки	MicrosoftVisualStudio 2008
Компилятор, библиотеки	Intel® ParallelStudio XE 2011

Были проведеныэксперименты для базовой и супернодальной последовательной версии. Для сокращения времени работы супернодальной реализации использованы матричные операции BLAS библиотеки Intel® MKL, а также настройка параметра построения супернодов (параметра огрубления, описанного выше). Далее были проведены эксперименты для параллельной версии решателя с настройкой параметра алгоритма –числа поддеревьев.

Также были проведеныэксперименты на тех же тестовых матрицах с использованием библиотекиMKL PARDISO из пакета Intel® MKL, SuperLUVersion 4.1, MUMPSна той же аппаратной платформе под управлением операционной системыLinuxopensuse 11.2.

В дальнейшем для решателя авторов указано лучшее время работы из тестируемых модификаций. Для матриц parabolic_fem, ecology2, audikw_1 это базовая версия решателя, для матриц pwtk, msdoor, tmt_sym, G3_circuit – супернодальная версия. Для матрицы G3_circuit применялся модифицированный метод вложенных сечений, для остальных матриц – базовая версия алгоритма переупорядочивания.

Для пакета SuperLU приведено время работы при использовании одного из встроенных алгоритмов перестановки, дающегоменьшее время работы. Для матриц pwtk, parabolic_fem, tmt_sym, ecology2 это приближенный столбцовый метод минимальной степени (COLAMD), для матрицы msdoor – множественный метод минимальной степени (MMD). Отметим, что на матрице G3_circuit программа завершила работу из-за внутренней ошибки работы с памятью; на матрице audikw_1 – в связи с нехваткой ресурсов для хранения фактора.

Рассмотрим текущие результаты последовательной версии решателя в сравнении с последовательными версиями SuperLU, MUMPS и MKL PARDISO (Таблица 3).Как видно из таблицы ниже, последовательная версия решателя в основном показывает лучшие результаты, чем SuperLU, но отстает от MKL PARDISO и MUMPS.

Таблица 3. Сравнение работы последовательной версии решателя авторов с некоторыми сторонними решателями.

Матрица	Решатель авторов		SuperLU		MUMPS		MKL	
	Число элементов в факторе	t, сек	Число элементов в факторе	t, сек	Число элементов в факторе	t, сек	Число элементов в факторе	t, сек
pwtk	61 678 703	18,6	108 904 188	79,8	48 988 990	11,9	52 190 144	5,5
msdoor	180 142 589	91,9	111 245 898	80,5	54 793 824	11,8	57 478 740	6,1
parabolic_fem	26 872 825	9,7	52 361 272	37,6	28496994	8,7	28 125 024	6,5
tmt_sym	45 181 460	28,1	88 165 169	76,9	36156598	17,2	33 075 548	8,7
ecology2	33 203 232	56,0	68 677 613	38,0	32956605	12,6	38 516 392	10,4
G3_circuit	213 111 677	468,4	Ошибка		102828595	76,8	104 692 916	25,5
audikw_1	2006 889 336	20 273,0	Ошибка		1 270 735 946	2856,5	1 270 292 396	793,0

Рассмотрим текущие результаты параллельной версии решателя в сравнении с параллельными версиями SuperLU и MKL PARDISO при работе в 8 потоков (Таблица 4). Для решателя авторов отдельно приведено ускорение численной фазы разложения. Как видно из таблицы, параллельная версия решателя авторов в целом несколько уступаетSuperLUисущественно отстает от MKLPARDISO.

Таблица 4. Сравнение работы параллельной версии решателя авторов с некоторыми сторонними решателями.

Матрица	Решатель авторов				SuperLU			MKL		
	Число элементов в факторе	t, сек	Ускорение численной фазы	Ускорение общее	Число элементов в факторе	t, сек	Ускорение	Число элементов в факторе	t, сек	Ускорение
pwtk	61 678 703	9,14	2,8	2,0	109 716 912	13,6	5,9	51 050 753	1,5	3,8
msdoor	180 142 589	55,69	1,8	1,7	111 395 559	18,7	4,3	57 096 124	1,9	3,2
parabolic_fem	26 872 825	6,27	3,8	1,8	54 270 042	7,4	5,1	27 941 564	2,5	2,6
tmt_sym	45 181 460	15,82	3,9	1,8	88 177 959	14,8	5,2	32 816 445	3,5	2,5
ecology2	33 203 232	56,09	4,4	1,2	68 688 337	8,0	4,7	38 876 742	4,5	2,3
G3_circuit	213 111 677	303,82	2,8	1,5	Ошибка			102 828 595	8,7	2,9

Анализ результатов экспериментов позволяет сделать следующие выводы:

- Время работы определяется стадиями переупорядочивания и численной фазой разложения Холецкого.
- Решатели, существенно обгоняющие по скорости работы реализацию авторов, используют в работе для переупорядочивания библиотеку METIS – мировой лидер в указанной области на протяжении многих последних лет.В нашей вычислительной схеме применяется собственная реализация метода вложенных сечений. На всех тестовых примерах заполнение фактора лучше, чем у SuperLU, но хуже, чем у MKL PARDISO.
- По сравнению с MUMPS, решатель авторов показывает незначительное отставание на матрицах размером до миллиона (parabolic_fem, tmt_sym, pwtk). На матрицах большего размера отставание возрастает, но не превышает одного порядка.
- Отставание решателя по отношению к MKL колеблется в зависимости от размера матрицы и ее заполнения. Большее отставание на ряде матриц (msdoor, G3_circuit, audikw_1) вызвано прежде всего тем, что MKL PARDISO имеет лучшую реализацию переупорядочивателя (METIS), а также более эффективно работает с памятью.

6. Заключение

На данный момент авторами реализована базовая и супернодальнаяпоследовательные версии решателя и ряд их модификаций. Также разработана базовая параллельная супернодальная версия решателя.Анализ результатов вычислительных экспериментов показал, что последовательная версия решателя авторов на большинстве тестовых задач опережает по скорости решатель SuperLU, но проигрывает решателям MUMPS и Intel MKL PARDISO на матрицах порядка $10^5 – 10^6$. Величина отставания зависит от задачи идля некоторых матриц составляет 1,5-5 раз, что наряду со сравнением с SuperLU подтверждает качество текущей реализации и создает предпосылки ее дальнейшего развития. Результаты работы используются в учебном процессе факультета ВМК в рамках курса «Параллельные численные методы».

Целью дальнейшего исследования является повышение производительности решателя. Основные направления работы:

- Реализация многоуровневого параллельного переупорядочивателя на базе метода вложенных сечений с возможным подключением алгоритма минимальной степени на некотором шаге метода.
- Оптимизация последовательной и параллельной версии численной фазы разложения Холецкого.

Литература

1. Ashcraft C., Grimes R. The influence of relaxed supernode partitions on the multifrontal method // ACM Trans. Math. Software. – 1989. – Vol. 15, No. 4. – P. 291-309.
2. Aschcraft C., Liu J.W.H. A partition improvement algorithm for generalized nested dissection // Technical Report BCSTECH-92-020 Boeing computer Services. – 1994.
3. Davis T.A. Direct methods for sparse linear systems. – Philadelphia: SIAM, 2006. – 217 pp.
4. Demmel J.W., Eisenstat S.C., Gilbert J.R., Li X.S., Liu J.W.H. A supernodal approach to sparse partial pivoting // SIAM J. Matrix Anal. Appl. – 1999. – Vol. 20, No. 3. – P. 720-755.
5. Duff I. S., Reid J. K. The multifrontal solution of indefinite sparse symmetric linear equations // ACM Bans. Math. Software, 9 (1983). – P. 302-325.
6. George A., Liu J.W.H. An Automatic Nested Dissection Algorithm for Irregular Finite Element Problems // SIAM J. on Numerical Analysis. – 1978. – Vol. 15, No. 5. – P. 1053–1069.
7. Gilbert J.R., Ng E.G., Peyton B.W. An efficient algorithm to compute row and column counts for sparse Cholesky factorization // SIAM J. Matrix Anal. Appl. – 1994. – Vol. 15, No. 4. – P. 1075-1091.
8. Hogg J.D. Efficient sparse Cholesky factorization – [http://www.maths.ed.ac.uk/~s0455378/EfficientCholesky.pdf].
9. Hogg J.D. Elimination Trees and Up-/Down-dating of Sparse Cholesky Factorizations – [http://www.maths.ed.ac.uk/~s0455378/ETreesUpDown.pdf].
10. Hogg J.D., Reid J.K., Scott J.A. A DAG-based Sparse Cholesky Solver for Multicore Architectures // Tech. report RAL-TR-2009-004, Rutherford Appleton Laboratory. – 2009.
11. Liu J.W.H. The multifrontal method and paging in sparse Cholesky factorization // ACM Trans. Math. Softw. – 1989. – P. 310-325.
12. Liu J. W. H. The multifrontal method for sparse matrix solution: Theory and practice // SIAM Review. – 1992. Vol. 34. – P. 82-109.
13. Liu. J. W. H. The role of elimination trees in sparse factorization // Matrix Anal Appl. – 1990. – P. 134-172.
14. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. – М.: Мир, 1984.
15. Писсанецки С. Технология разреженных матриц. – М.: Мир,1988.
16. The University of Florida Sparse Matrix Collection –[www.cise.ufl.edu/research/sparse/matrices/].