РАЗРАБОТКА НОВОГО ПЕРЕУПОРЯДОЧИВАТЕЛЯ СИММЕТРИЧНЫХ РАЗРЕЖЕННЫХ МАТРИЦ

А.Ю. Пирова

Нижегородский госуниверситет им. Н.И. Лобачевского

В работе рассматривается задача уменьшения заполнения фактора разреженной матрицы при разложении Холецкого. Предлагается модификация алгоритма переупорядочения на основе многоуровневого метода вложенных сечений. Приводятся результаты вычислительных экспериментов, показывающие сопоставимость реализации с библиотекой METIS.

Введение

Решение систем линейных алгебраических уравнений вида Ax=b с разреженной симметричной положительно определенной матрицей A лежит в основе многих вычислительных задач компьютерной алгебры [20]. Необходимость решения таких задач возникает при моделировании различных физических процессов, в частности, в конечно-элементных расчетах, а также при моделировании в экономике, химии и других областях. Принцип работы прямых методов основан на факторизации матрицы системы с последующим решением треугольных систем (для симметричных положительно определенных матриц – разложение методом Холецкого). В этом случае решение системы выполняется в несколько этапов.

1. *Переупорядочение матрицы*: перестановка строк и столбцов матрицы для уменьшения размера фактора Холецкого.
2. *Символическая фаза*: анализ структуры исходной матрицы и построение портрета фактора.
3. *Численная фаза*: заполнение полученного портрета численными значениями.
4. *Решение треугольных систем*.

При применении разложения Холецкого для решения разреженной СЛАУ число ненулевых элементов фактора матрицы в разы больше числа ненулевых элементов исходной матрицы – матрица претерпевает *заполнение*. Это приводит к значительным затратам памяти для хранения матрицы, а также увеличению времени на ее факторизацию. Для того чтобы снизить заполнение фактора, к исходной матрице применяется процедура перестановки ее строк и столбцов. Нахождение перестановки, минимизирующей размер фактора или число вычислительных операций для ее факторизации, является NP-полной задачей [18]. В практических приложениях для ее решения применяются эвристические методы.

Разработка новых методов переупорядочения симметричных разреженных матриц, а также их перенос на современные вычислительные архитектуры, представляют большой практический интерес. Исследования в указанном направлении ведут такие ученые как Дж. Карипис, В. Кумар, Дж. Лю, К. Эшкрафт, П. Амстой, Ф. Пеллигринии, Б. Хендриксон др. В ННГУ данную область исследуют М.Х. Прилуцкий, Н.В. Старостин и др. Среди разработанных библиотек, предназначенных для переупорядочения матриц и разделения графов, отметим METIS [8], Scotch [14] и их аналоги для систем с распределенной памятью ParMETIS [10] и PT-Scotch [4].

В данной работе представлены текущие результаты по разработке нового алгоритма для переупорядочения разреженных матриц, а также его высокопроизводительной программной реализации. Работа выполняется в рамках проекта по созданию прямого решателя СЛАУ на кафедре МО ЭВМ факультета ВМК ННГУ им. Н.И. Лобачевского [19, 20].

1. Постановка задачи

Рассмотрим постановку задачи переупорядочения в следующем виде [18]. Пусть дана разреженная симметричная матрица $A = (a_{ij})$ размера $n \times n$. Построим *граф матрицы* $G = (V, E)$, в котором каждая вершина v_i $(i = \overline{0, n-1})$ сопоставлена строке матрицы i, а ребро $(v_i, v_j) \in E$ тогда и только тогда, когда $a_{ij} \neq 0$ ($i, j \in \{0, 1, \dots, n-1\}$, $i \neq j$). Смоделируем процесс гауссова исключения при разложении Холецкого. При выполнении *исключения* вершины v из графа G в граф добавляются ребра таким образом, чтобы множество смежных с v вершин стало кликой, сама вершина v удаляется из множества вершин вместе с инцидентными ей ребрами. Пусть $\pi = (\pi_0, \pi_1, \dots, \pi_{n-1})$ – перестановка множества вершин V. *Заполнение* $F(\pi)$, порожденное перестановкой π, – это множество ребер, добавленных при исключении вершин $\pi_0, \pi_1, \dots, \pi_{n-1}$. Поставим задачу нахождения перестановки π^* такой, что $|F(\pi^*)| \to min$. Под *качеством перестановки* будем понимать число ненулевых элементов фактора Холецкого матрицы после применения к ней данной перестановки.

2. Методы переупорядочения разреженных матриц

Основные подходы к переупорядочению симметричной разреженной матрицы с целью снижения заполнения фактора Холецкого можно разделить на две группы – подходы, основанные на методе вложенных сечений и подходы, основанные на методе минимальной степени. Так, метод минимальной степени был предложен Тиннеем и Уолкером в 1969 г. [17]. Этот метод основан на локальной стратегии уменьшения заполнения фактора матрицы и в некотором смысле моделирует процесс гауссова исключения. Ряд модификаций алгоритма, направленных на сокращение времени его работы, был предложен Лю (1985 г.), Амстеем, Дэвисом и Даффом (1996 г.). Метод вложенных сечений, впервые предложенный Джорджем [5] в 1973 г., основан на многократном разбиении графа матрицы при помощи вершинных разделителей. Модификации метода различаются способами выделения разделителя.

Начиная с 1996 г., широко используются модификации метода вложенных сечений, основанные на многоуровневой процедуре разделения графа. Такой подход был применен в работах Кариписа и Кумара [9], Эшкрафта и Лю [2], Хендриксона и Ротберга [7] и др. В многоуровневом методе выделение разделителя графа $G = (V, E)$ выполняется в три этапа:

- *Огрубление графа (coarsening)*: построение последовательности сжимающихся графов $G_1, G_2, \dots, G_m$ с помощью построения паросочетания. При этом $|V| > |V_1| > \dots > |V_m|$ и структура каждого следующего в последовательности графа отражает структуру предыдущего.
- *Разделение графа (partitioning)*: поиск разделителя и выделение двух несвязных компонент наименьшего графа G_m.
- *Развертывание графа (uncoarsening)*: проекция разделения наименьшего графа G_m на исходный G через последовательность графов $G_m, G_{m-1}, \dots, G_1$. На каждом шаге i выполняется улучшение разделения графа G_{m-i}, полученного проекцией разделения графа G_{m-i+1}. Для этого используются модификации итерационного метода Кернигана–Лина [11].

3. Программная реализация

В рамках данной работы выполняется разработка и высокопроизводительная программная реализация новой модификации алгоритма переупорядочения симметричных разреженных матриц, основанного на многоуровневом методе вложенных сечений.

В настоящий момент выполнена программная реализация последовательной версии переупорядочения на базе классического многоуровневого метода, описанного выше, с модификацией отдельных стадий. Предварительно выполняется сжатие структуры графа [4] для ускорения работы переупорядочения. В многоуровневом методе на этапе сжатия графа используются методы случайных паросочетаний и паросочетаний тяжелых ребер [5], на этапе разделения – построение структуры уровней смежности графа [3], на этапе улучшения разделения предлагается использовать модификацию итерационного метода, основанного на работах Эшкрафта–Лю [1] и Хендриксона–Ротберга [4]. Суть модификации заключается в изменении условий выполнения итерации алгоритма (перемещения вершины из одной части разделения в другую), а также использовании функции оценки разделения [3]. В ряде случаев для улучшения качества получаемых разделений модифицированный метод Эшкрафта–Лю используется с перезапуском.

4. Результаты вычислительных экспериментов

Для текущей программной реализации был проведен ряд вычислительных экспериментов по переупорядочению матриц из широко распространенной коллекции университета Флориды [16]. Параметры тестовых матриц и аппаратного окружения приведены ниже (таблица 1, таблица 2).

Таблица 1. Характеристики тестовых матриц

Название матрицы	Порядок	Число ненулевых элементов в верхнем треугольнике (NZ)	Заполненность, %
Pwtk	217 918	5 926 171	0,000245
Msdoor	415 863	10 328 399	0,000117
parabolic_fem	525 825	2 100 225	0,000013
tmt_sym	726 713	2 903 837	0,000010
ecology2	999 999	2 997 995	0,000005
G3_circuit	1 585 478	4 623 152	0,000003

Таблица 2. Параметры тестового окружения

Процессор	Четырехъядерный процессор Intel® Xeon L5630 (2.13 GHz)
Память	24 GB
Операционная система	Windows Server SP2
Компилятор	Intel® C++ Composer (в составе Intel® Parallel Studio XE 2013)

Также был проведен ряд экспериментов на указанных тестовых матрицах с использованием библиотек METIS из ParMetis 4 [8], а также решателей СЛАУ SuperLU v. 4.1 [12], MUMPS v. 4.10 [1] (переупорядочиватель PORD [15]). В них использовались параметры переупорядочения по умолчанию. Качество выполненной реализации оценивалось по получаемому числу ненулевых элементов фактора матрицы, а также времени, необходимом для переупорядочения матрицы. Сравнение результатов переупорядочения по числу ненулевых элементов фактора представлено ниже (таблица 3). Для собственной реализации приведено два результата: с лучшим числом ненулевых элементов фактора и с лучшим временем работы (при приемлемом размере фактора). Сравнение времени получения перестановки собственной реализации и в пакете METIS представлено ниже (таблица 4).

Таблица 3. Сравнение результатов переупорядочения по числу ненулевых элементов

Матрица	Порядок, n	Собственное переупорядочение: Лучший фактор	Собственное переупорядочение: Лучшее время + фактор	Metis из Parmetis 4	SuperLU	PORD
pwtk	217 918	46 970 027	50 061 339	49 542 235	108 904 188	48 988 990
msdoor	415 863	51 780 765	55 250 683	51 716 753	111 245 898	54 793 824
parabolic_fem	525 825	25 281 824	25 882 293	27 287 157	52 361 272	28 496 994
tmt_sym	726 713	28 759 956	34 342 694	32 353 685	88 165 169	36 156 598
ecology2	999 999	30 452 184	34 905 653	38 980 242	68 677 613	32 956 605
G3_circuit	1 585 478	92 360 925	103 071 640	95 701 654	Ошибка	130 972 526

Таблица 4. Сравнение времени переупорядочения (в секундах)

Матрица	Порядок, n	Собственное переупорядочение: Лучший фактор	Собственное переупорядочение: Лучшее время + фактор	Metis из Parmetis 4
pwtk	217 918	0,953	0,563	0,83
msdoor	415 863	1,765	1,219	1,35
parabolic_fem	525 825	7,953	3,313	5,82
tmt_sym	726 713	17,344	6,500	8,17
ecology2	999 999	15,656	4,093	8,85
G3_circuit	1 585 478	21,047	10,219	18,22

Как видно из результатов экспериментов, для всех тестовых матриц удалось получить лучшее в смысле минимизации заполнения переупорядочение в сравнении с лидером данной области – пакетом METIS, в котором использовались настройки переупорядочения по умолчанию. Выигрыш в размерах фактора составляет 5–12% на большинстве матриц, на матрице ecology2 – 28%. Время получения такой перестановки в 1,2 – 2,1 раза больше, чем у METIS (в среднем – в 1,5 раза). При этом для каждой матрицы за время в 1,5–2 раза меньшее, чем у METIS, можно получить перестановку, дающую заполнение фактора не хуже, чем на 8%. Для матриц parabolic_fem и ecology2 полученные заполнения на 5 – 10% лучше и в этом случае.

Перестановки, полученные в результате работы метода, использовались при решении СЛАУ одним из академических решателей – MUMPS. Время решения системы при собственном переупорядочении и при переупорядочении из METIS очень близко, что говорит о приемлемости разработанной реализации для задачи решения СЛАУ.

Заключение

В настоящий момент разработана программная реализация алгоритма переупорядочения разреженных матриц, показывающая результаты, близкие по качеству и скорости работы к библиотеке METIS, – одной из наиболее известных и распространенных библиотек в данной области. Данная реализация пока не показывает однозначного превосходства над METIS, но допускает настройку параметров для достижения лучшего, чем у METIS, качества или скорости работы при некритическом ухудшении значения второго критерия.

Направлениями дальнейших исследований являются развитие алгоритма и программной реализации с целью повышения качества и скорости решения задачи, а также разработка параллельной реализации для систем с общей памятью.

Работа выполнена при организационной поддержке лаборатории Intel-ННГУ «Информационные технологии». Автор благодарит И.Б. Меерова, А.В. Сысоева, Е.А. Козинова и А.В. Линева за полезные обсуждения и внимание к работе.

Литература

1. Amestoy P. R., Duff I., L'Excellent J.Y. MUMPS MUltifrontal Massively Parallel Solver. – Version 2.0. // Technical Report TR/PA/98/02. – 1998.
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. Ashcraft C., Liu J.W.H. Using domain decomposition to find graph bisectors //BIT Numerical Mathematics. – 1997. – Vol. 37. – No. 3. – P. 506–534.
4. Chevalier C., Pellegrini F. PT-Scotch: A tool for efficient parallel graph ordering // Parallel Computing. – 2008. – Vol. 34, No. 6. – P. 318–331.
5. George A. Nested dissection of a regular finite element mesh // SIAM J. on Numerical Analysis. – 1973. – Vol. 10, No. 2. – P. 345–363.
6. George A., Liu J.W.H. An automatic nested dissection algorithm for irregular finite element problems // SIAM J. on Num. Anal. – 1978. – Vol. 15, No. 5. – P. 1053–1069.
7. Hendrickson B., Rothberg. Improving the runtime and quality of nested dissection ordering // SIAM J. on Scientific Computing. – 1999. – Vol. 20. – P. 468–489.
8. Karipis G. METIS. A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Version 5.0. // Technical report, University of Minnesota. – 2011. URL: [http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/manual.pdf].
9. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs // SIAM J. on Scientific Computing. – 1999. – Vol. 20, No. 1. – P. 359–392.
10. Karypis G., Kumar V. ParMETIS: Parallel graph partitioning and sparse matrix ordering library // Technical Report TR 97-060, University of Minnesota. – 1997.
11. Kernighan B.W., Lin S. An efficient heuristic procedure for partitioning graphs // The Bell System Technical J. – 1970. – Vol. 29. – P. 291–307.
12. Li X.S. An overview of SuperLU: Algorithms, implementation, and user interface // ACM Transactions on Math. Software. – 2005. – Vol. 31, No. 3. – P. 302–325.
13. Pellegrini F. Shared memory parallel algorithms in Scotch 6. – URL: [http://graal.ens-lyon.fr/MUMPS/doc/ud_2013/Pellegrini.pdf].
14. Pellegrini F., Roman J. Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs // High-Performance Computing and Networking. – Springer Berlin Heidelberg, 1996. – P. 493–498.
15. Schulze J. Towards a tighter coupling of bottom-up and top-down sparse matrix ordering methods // Bit Numerical Mathematics. – 2001. – Vol. 41, No. 4. – P. 800–841.
16. The University of Florida Sparse Matrix Collection – URL.: [www.cise.ufl.edu/research/sparse/matrices/].
17. Tinney W., Walker J. Direct solutions of sparse network equations by optimally ordered triangular factorization // Proceedings of the IEEE. – 1967. – Vol. 55, No. 11. – P. 1801–1809.
18. Yannakakis M. Computing the minimum fill-in is NP-complete // SIAM J. on Algebraic and Discrete Methods. – 1981. – Vol. 2, No. 1. – P. 77–79.
19. Bastrakov S., Meyerov I., Gergel V. et al. High performance computing in biomedical applications // Procedia Computer Science, 2013. – Vol. 18. – P. 10–19.
20. Козинов Е.А., Лебедев И.Г., Лебедев С.А., Малова А.Ю. и др. Новый решатель для алгебраических систем разреженных линейных уравнений с симметричной положительно определенной матрицей // Вестник ННГУ. – Н. Новгород: Изд-во ННГУ, 2012. – №5(2) – С. 376-384.