

СКВОЗНОЙ ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ НЕПОЛНОГО ТРЕУГОЛЬНОГО РАЗЛОЖЕНИЯ ВТОРОГО ПОРЯДКА ТОЧНОСТИ С ДИНАМИЧЕСКИМ ВЫБОРОМ ДЕКОМПОЗИЦИИ И УПОРЯДОЧИВАНИЯ

И.Н. Коньшин, С.А. Харченко

Введение. В настоящее время многие отрасли промышленности используют моделирование задач математической физики как неотъемлемую часть процесса проектирования. К таким отраслям традиционно принадлежат авиастроение, космическая промышленность, автомобилестроение, и др. Все чаще моделирование используется в строительстве при проектировании сложных конструкций, а в последнее время моделирование требуется уже и в отраслях, производящих товары народного потребления — от санитарно-технических устройств до проектирования мобильных телефонов. Качественное моделирование разнообразных физических процессов в трехмерных областях со сложной геометрией требует расчетных сеток с большим числом элементов (от сотен тысяч до сотен миллионов ячеек и более), а значит зачастую моделирование может быть произведено только на параллельной вычислительной технике. Современные пакеты прикладных программ, которые используются при моделировании подобных задач, зачастую основаны на неявных схемах аппроксимации, что приводит к необходимости решать с высокой точностью плохо обусловленные разреженные системы линейных алгебраических уравнений (СЛАУ) большой размерности. При этом для пользователей пакетов прикладных программ важно быть уверенным в точности и надежности моделирования, а также в эффективности производимых параллельных вычислений. В работе будут рассмотрены некоторые вопросы, связанные с эффективностью решения СЛАУ на параллельной вычислительной технике. Работы выполнены ООО «ТЕСИС» по договору с ФГУП РФЯЦ-ВНИИЭФ (г. Саров), 2012 г.

Неполное треугольное разложение второго порядка точности. Рассмотрим СЛАУ

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

где A - невырожденная разреженная квадратная матрица большой размерности, b - вектор правой части, x - искомый вектор решения. Многие современные параллельные итерационные алгоритмы решения СЛАУ (1) основаны на построении предобуславливания матрицы A как результата вычисления ее неполного треугольного разложения. Для получения приближения к решению (1) как правило проводятся предобусловленные итерации подходящего алгоритма типа подпространства Крылова.

В работе [1] предложен алгоритм построения предобуславливания второго порядка точности, для которого имеют место соотношения:

$$A + E = L^{(I)} U^{(I)} + L^{(I)} U^{(II)} + L^{(II)} U^{(I)} \quad (2)$$

где A – матрица коэффициентов системы уравнений, $L^{(I)}$ и $U^{(I)}$ соответственно нижняя и верхняя треугольная части предобуславливания (элементы первого порядка точности), $L^{(II)}$ и $U^{(II)}$, соответственно, нижняя и верхняя треугольная части элементов второго порядка точности предобуславливания, и E – матрица ошибок. Предобусловленные итерации производятся с матрицей $A(L^{(I)}U^{(I)})^{-1}$. В процессе неполной факторизации в памяти одновременно хранятся только элементы матриц A , $L^{(I)}$ и $U^{(I)}$ и некоторая небольшая часть элементов матриц $L^{(II)}$ и $U^{(II)}$. Элементы матрицы ошибок E отбрасываются с корректировкой соответствующих диагональных значений. В работе [1] приводятся некоторые теоретические оценки качества предобуславливания в симметричном случае. Этот алгоритм показал высокую эффективность и надёжность при последовательном решении широкого класса задач с плохо обусловленными матрицами коэффициентов.

Существует несколько вариантов распараллеливания алгоритма построения неполного треугольного разложения второго порядка точности. Это прежде всего так называемый вариант предобуславливания типа блочного Шварца с односторонними перекрытиями [2], в котором для заданной начальной декомпозиции расчетной области на набор подобластей проводится факторизация вида (2) для специальным образом построенных подматриц с перекрытиями вперед. При этом в качестве предобуславливания всей задачи рассматривается соответствующее аддитивное предобуславливание. Этот алгоритм обладает очень хорошими свойствами масштабируемости и зачастую очень просто может быть реализован. Основным недостатком этого алгоритма является существенная зависимость сходимости от количества блоков и от степени их перекрытия. При этом для плохообусловленных задач иногда требуется достаточно большое перекрытие подобластей для обеспечения требуемых свойств сходимости, что может быть накладным, особенно для 3D задач. Кроме того, строго говоря, аддитивное предобуславливание, предложенное в работе [2], не является распараллеливанием алгоритма предобуславливания (2), а по существу только некоторый вариант его параллельного использования.

В работах [3]-[5] была предпринята попытка явного распараллеливания по общей и по распределенной памяти алгоритма из работы [1]. При этом распараллеливание осуществлялось на основе построения для всей задачи упорядочивания типа вложенных сечений и выделения в разделителях специального дополнительного блочного биения на основе структурной информации о задаче. В терминах этого расширенного блочного биения проводилось распараллеливание вычислений типа (2) с априорным ограничением заполнения для обеспечения параллельной эффективности вычислений в 3D случае. Подчеркнем, что внутри блока никаких ограничений на заполнение не накладывается. К достоинством этого подхода можно отнести тот факт, что если убрать ограничения на блочное заполнение, то этот алгоритм по-существу является параллельной реализацией алгоритма (2) для специального упорядочивания матрицы. К сожалению, снятие блочных ограничений на заполнение в 3D случае приводит к существенному ухудшению качества распараллеливания. С другой стороны, при введении простых естественных ограничений на блочное заполнение (типа дублирования блочной структуры матрицы) параллельность вычислений вполне достаточна для многих классов прикладных задач, однако при этом для очень плохообусловленных задач в некоторых случаях имеет место существенная зависимость свойств сходимости от блочного биения.

Для демонстрации негативного влияния ограничения блочного заполнения на сходимость рассмотрим результаты для следующей симметричной тестовой задачи, возникающей при моделировании задачи механики «resh2_4», любезно предоставленной РФЯЦ ВНИИЭФ (г. Саров): $N = 569\,217$, число ненулевых элементов $NZA = 37\,826\,212$, эксперименты проводились для различных режимов параллельных вычислений по общей памяти. Строилось мелкоблочное предобусловливание с размером каждого мелкого блока 3×3 , предобусловливание хранилось в формате "float". СЛАУ (1) решалась с использованием предобусловленного алгоритма CG с относительной точностью $\epsilon = 10^{-9}$ по невязке. Эксперименты проводились на процессоре Intel Core I7 Nehalem 920, 2.67GHz, 4 ядра, 12 Gb. В таблицах ниже использованы следующие обозначения: «Режим, MPI x threads» - режим параллельных вычислений в формате 'число MPI процессов' x 'число нитей'; «%» - плотность заполнения предобусловливания в процентах по отношению к заполнению матрицы; «FctTime, сек.» - время неполной факторизации в секундах, «Niter» - число итераций предобусловленного алгоритма CG для достижения заданной сходимости; «IterTime, сек.» - время итераций в секундах.

Таблица 1 - Результаты для алгоритма факторизации с блочным ограничением заполнения

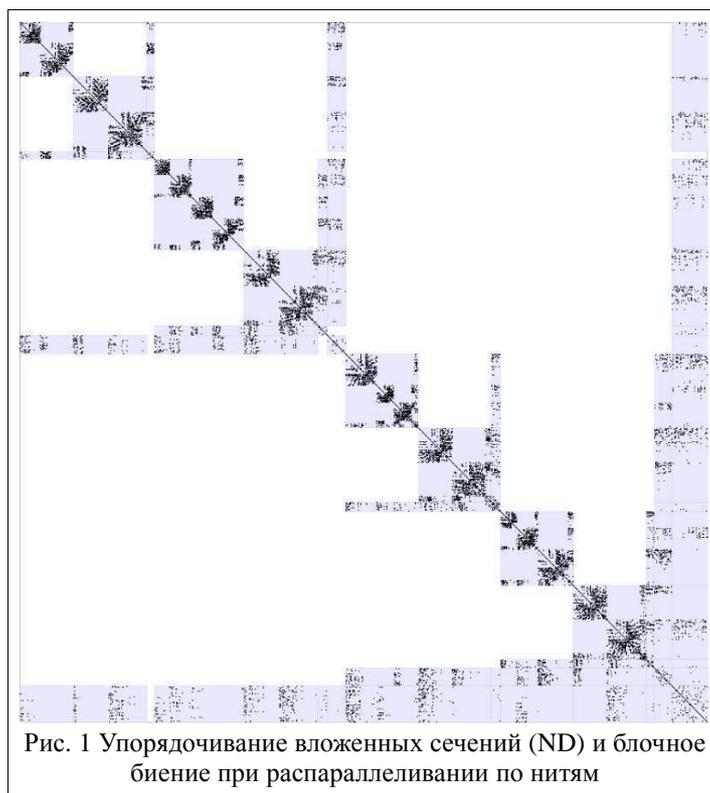
Режим, MPI x threads	%	FctTime, сек.	Niter	IterTime, сек.
1x1	517.2	91.4	94	39.7
1x2	516.3	48.5	107	32.9
1x3	515.2	32.6	129	27.0
1x4	516.8	22.3	129	22.3
1x6	517.0	27.9	144	22.4
1x8	513.1	24.4	155	21.1
1x12	514.5	24.1	191	26.4
1x16	506.9	21.5	215	28.4

Таблица 2 - Результаты для алгоритма факторизации без блочного ограничения на заполнение

Режим, MPI x threads	%	FctTime, сек.	Niter	IterTime, сек.
1x1	517.7	98.8	94	40.0
1x2	517.7	52.4	94	29.3
1x3	517.7	38.7	94	20.1
1x4	517.7	31.2	94	16.9
1x6	517.7	32.7	94	15.2
1x8	517.7	38.7	94	16.1
1x12	517.7	39.5	94	13.6
1x16	517.7	47.8	94	14

Из Таблицы 1 хорошо видно, что при учете блочных ограничений на заполнение для разного числа нитей строятся разные упорядочивания для окаймлений и получаются существенно разные предобусловливания, причем число итераций существенно увеличивается с увеличением числа нитей. Из Таблицы 2 следует, что при отсутствии блочных ограничений на заполнение во всех режимах строится одно и то же предобусловливание (базой параллельных вычислений является одно и то же упорядочивание вложенных сечений ND, см. Рис. 1, распараллеливание по нитям осуществлялось в терминах бинарного дерева

вычислений). При использовании блочных ограничений на заполнение, вычислительные затраты при факторизации ниже и масштабируемость факторизации лучше, а число итераций при этом больше, чем во втором случае. Во всех вариантах при числе нитей > 4 наблюдается увеличение или неубывание времени счета, поскольку в вычислительной системе имеется только 4 физических ядра.



Таким образом, снятие ограничений на блочное заполнение в предобуславливании позволяет избежать или ослабить деградацию качества предобуславливания при использовании большого числа процессоров/нитей. С другой стороны, необходимо предпринять дополнительные усилия, чтобы снятие ограничений на заполнение не приводило к потере качества масштабируемости параллельных вычислений при неполной факторизации матрицы в случае разделителей большого размера, как это может иметь место при решении 3D задач.

Сквозной параллельный алгоритм построения неполного треугольного разложения второго порядка точности.

Идея сквозного параллельного алгоритма построения предобуславливания типа (2) заключается в том, чтобы перед факторизацией диагонального блока разделителя на основе разреженности дополнения по Шуру для этого блока динамически вычислить упорядочивание вложенных сечений ND и его новую декомпозицию по процессорам/нитям.

Рассмотрим подробнее предлагаемый параллельный алгоритм для простейшего случая двух процессоров, на этом примере проясним основные идеи параллельной организации алгоритма.

Пусть имеются два объема и разделитель между ними, как это имеет место в случае упорядочивания типа ND. Разделитель предварительно распределяем на 2 процессора на основе простых априорных вычислений с учетом геометрической локальности. Независимо вычисляются неполные треугольные разложения в исходных объемах. Далее, каждый процессор вычисляет свою и чужую части дополнения по Шуру. В вычисленных частях дополнений по Шуру проводится фильтрация очень маленьких элементов (меньших второго порога) с соответствующей коррекцией главной диагонали для обеспечения устойчивости вычислений. Производится обмен частями дополнения по Шуру и суммирование результатов. Затем для разреженной матрицы дополнения по Шуру (она будет квази-2D матрицей) строится декомпозиция на две части и сепаратор (квази-1D) по построенному дополнению по Шуру. Важно отметить, что это будет происходить динамически, в зависимости от того, как строились разложения в объемах. Для двух квази-2D матриц при необходимости все повторяется, и т.д.

Балансировка вычислений для двух процессоров в таком алгоритме осуществляется автоматически и, в некотором смысле динамически за счет вложенных вызовов декомпозиции для разделителей каждого уровня. При этом для 3D задачи число уровней вложения не окажется больше трех. Число уровней вложения может оказаться и меньше если размер блока достаточно мал или если заполнение для текущего блока таково, что не удастся разделить задачу на независимые части.

Рассмотрим теперь распараллеливание в случае большого числа процессоров. Пусть имеется разделитель на границе между объемами для двух групп процессоров. Вычисления в объемах проводим в соответствии с предположением рекурсии вычислений. Для разделителя действуем аналогично случаю двух процессоров. Сначала разделитель распределяется по всем процессорам на основе априорной информации с учетом локальности. Все процессоры проводят необходимые вычисления по построению дополнения по Шуру для разделителя, с необходимыми фильтрациями и обходами. В результате имеем распределенную разреженную матрицу дополнения по Шуру. Для нее строим квази-2D декомпозицию с выделением квази-1D разделителей на то число процессоров, на которое удастся выстроить. Если подматрицы сильно разрежены, то проводится аналогичная динамическая редукция. На всех уровнях вычислений используется мелко блочный формат хранения матрицы и предобуславливания. При отсутствии элементов в блоке он считается нулевым и убирается из блочной структуры разреженности. Перед началом вычислений со следующим уровнем дерева происходит обмен блочной структурой разреженности между процессорами для минимизации в дальнейшем межпроцессорных взаимодействий.

В контексте двухуровневого параллелизма вычислений (MPI+threads) представляется естественным дополнительно модифицировать упомянутый алгоритм следующим образом. Описанный выше алгоритм будем использовать в терминах верхнего уровня параллелизма — в терминах MPI процессов. В этом контексте число уровней квази-бинарного дерева вычислений и крупно блочное разбиение задачи определяется числом MPI процессов. С другой стороны, в каждом крупном блоке перед началом вычислений с этим блоком также строится упорядочивание вложенных сечений. Соответственно, при факторизации каждого такого крупного блока можно использовать такой же многоуровневый параллельный алгоритм с динамическим упорядочиванием разделителей на основе разреженности матриц дополнения по Шуру, только уже для распараллеливания вычислений по нитям. При этом число обычных блоков в каждом крупном блоке и число уровней квази-бинарного дерева вычислений в каждом крупном блоке теперь определяется числом нитей вычислений, ассоциированных с соответствующим MPI процессом. Кроме того, за счет использования общей памяти при работе с нитями возможно естественное использование динамической балансировки вычислений по нитям.

В финальной версии статьи предполагается представить результаты численных экспериментов для комбинированной MPI+threads параллельной реализации сквозного алгоритма построения неполного треугольного разложения второго порядка точности.

Заключение. Таким образом, в работе представлен сквозной параллельный MPI+threads алгоритм построения неполного разложения второго порядка точности на основе динамического упорядочивания и декомпозиции разделителей. Результаты численных экспериментов подтверждают эффективность предложенного алгоритма для набора тестовых задач.

ЛИТЕРАТУРА:

1. I.E. Kaporin, High Quality Preconditioning of a General Symmetric Positive Definite Matrix Based on its $U^T U + U^T R + R^T U$ decomposition // Numer. Linear Algebra Appl. – 1998. – Vol. 5. – P. 483-509.
2. И.Е. Капорин, И.Н. Коньшин, "Параллельное решение симметричных положительно определенных систем на основе перекрывающегося разбиения на блоки" // Ж. Вычисл. Матем. и Матем. Физ. 2001. Т. 41, 4, с. 515-528.
3. С.А. Харченко, "Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики". Материалы международной научной конференции "Параллельные вычислительные технологии" (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с. 494-499.
4. Г.Б. Сушко, С.А. Харченко, "Многопоточная параллельная реализация итерационного алгоритма решения систем линейных уравнений с динамическим распределением нагрузки по нитям вычислений". Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с. 452-457.
5. Г.Б. Сушко, С.А. Харченко, "Экспериментальное исследование на СКИФ МГУ "Чебышев" комбинированной MPI+threads реализации алгоритма решения систем линейных уравнений, возникающих во FlowVision при моделировании задач вычислительной гидродинамики". Материалы международной научной конференции "Параллельные вычислительные технологии" (ПаВТ'2009), Нижний Новгород, 30 марта – 3 апреля 2009 г., Челябинск: Изд. ЮУрГУ, 2009, с. 316-324.