Вариант распределенного метода декомпозиции областей*

В.П.Ильин[1,2], Д.В.Перевозкин[1]

Институт вычислительной математики и математической геофизики СО РАН[1],
Новосибирский государственный университет[2]

Рассматриваются алгоритмы масштабируемого распараллеливания решения сверхбольших разреженных сеточных СЛАУ, представленных в универсальных сжатых форматах, в том смысле, что их реализация осуществляется без программных ограничений на порядки алгебраических систем и на количество используемых вычислительных узлов, процессоров и/или ядер.
Данная задача сводится к распределенному варианту алгебраической 3D-декомпозиции областей, в котором отсутствует чрезмерная расчетно-информационная нагрузка корневого процессора, т.е. все организуемые MPI-процессы, каждый из которых соответствует "своей" подобласти, являются практически равноправными. Описываемые методы реализованы в составе библиотеки алгебраических решателей KRYLOV. В работе приводятся некоторые оценки используемых ресурсов и особенности параллельных вычислительных технологий. Эффективность разработанных алгоритмов иллюстрируется результатами численных экспериментов по решению характерных алгебраических задач на различных конфигурациях многопроцессорной вычислительной системы.

1. Введение

Методы декомпозиции областей (МДО), как главное орудие масштабируемого распараллеливания численного решения больших задач математического моделирования на многопроцессорных вычислительных системах (МВС), является предметом изучения огромного количества статей и монографий, представленных, например, на специальном сайте [1].

Обширную область изучаемых здесь вопросов можно разбить на две основные части. Первая из них относится к построению новых алгоритмов для различных классов задач, их теоретическому обоснованию, оценкам скорости сходимости, оптимизации и сравнительному анализу. Относительно свежие математические результаты в данных направлениях можно найти на сайте [2] 22-й Международной конференции по декомпозиции областей, состоявшейся в сентябре 2013 г.

Вторая часть не менее актуальных проблем — это эффективное отображение МДО на архитектуру современных МВС, которая, несмотря на стремительное обновление суперкомпьютерного списка TOP-500, в ближайшую пятилетку, до рождения первого экзафлопсника, прогнозируется традиционной: кластер из гетерогенных узлов, содержащих по несколько центральных многоядерных процессоров (CPU) с общей памятью, а также по несколько "графических процессоров общего назначения" (GPGPU), сверхбыстрых, но с достаточно сложным использованием. Управление иерархической памятью с распределенным и общим доступом ограничивается средствами MPI, OpenMP и CUDA.

Цель данной работы заключается в том, чтобы хотя бы для достаточно ограниченного набора алгоритмов декомпозиции сконцентрироваться на вариантах технологических решений, обеспечивающих реализацию в определенном смысле главной задачи — создания высокопроизводительного математического и программного обеспечения, не содержащего формальных ограничений на число степеней свободы и на количество используемых вычислительных ядер и/или процессоров.

*Работа поддержана грантом РФФИ № 11-01-00205, а также грантами Президиума РАН № 15.9-4 и ОМН РАН № 1.3.3-4.

Рассматриваемая нами задача с функциональной точки зрения достаточно простая — это численное решение систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами больших порядков (до 10^8–10^{11}), возникающими при различных аппроксимациях многомерных междисциплинарных начально-краевых задач со сложной геометрической конфигурацией границ на неструктурированных сетках.

Особенность проблемы заключается в объеме данных, необходимых для хранения ненулевых элементов разреженных матриц, число которых в каждой из строк не зависит от размерности СЛАУ и составляет обычно несколько десятков (их количество растет с увеличением порядка точности аппроксимации). Например, система уравнений Навье–Стокса, описывающая гидро-газодинамические течения, включает 4 неизвестные скалярные функции — три компоненты вектора скорости и давление. Поэтому на трехмерной сетке с числом узлов $1000^3 = 10^9$ общий порядок СЛАУ составляет примерно $4 \cdot 10^9$, а количество ненулевых матричных элементов — около 10^{10}. Поскольку возникающие при этом вычислительные задачи являются очень плохо обусловленными, то расчеты необходимо проводить, как минимум, со стандартной двойной точностью (64 двоичных разряда, или 8 байтов, на одно число). Таким образом, размещение соответствующей СЛАУ на одном процессоре, объем оперативной памяти которого составляет обычно несколько гигабайт, становится проблематичным. В этом случае не проходит зачастую применяемая тактика распараллеливания алгебраических задач (наиболее легко программируемая), когда сначала "глобальная" матрица формируется на одном (головном) процессоре, а потом ее блоки рассылаются по остальным используемым процессам. Единственная альтернатива здесь — распределенное хранение матрицы, когда ее элементы изначально рассчитываются в разных процессорах.

Данная статья построена следующим образом. В п. 2 мы представляем основные структурные особенности рассматриваемых СЛАУ и применяемых МДО. В п. 3 описываются некоторые предлагаемые параллельные алгоритмы и их программные реализации. В п. 4 приводятся результаты характерных численных экспериментов по решению представительной серии методических задач.

2. Некоторые постановки, формализмы и алгоритмы декомпозиции

Рассматривается следующая задача: пусть матрица СЛАУ

$$Au = f, \;\; A = \{a_{i,j}\} \in \mathcal{R}^{N,N}, \tag{1}$$

разбита на блочные строки

$$A = \{A_p \in \mathcal{R}^{N_p,N}, \; p = 1, ..., P\}, \;\; N_1 + ... + N_p = N,$$

с приблизительно равным числом строк $N_p \gg 1$ в каждой. Соответствующим образом также разбиваются векторы искомого решения и правой части

$$u = \{u_p \in \mathcal{R}^{N_p}\}, \;\; f = \{f_p \in \mathcal{R}^{N_p}\},$$

так что система уравнений (1) может быть записана в форме совокупности P подсистем

$$A_{p,p}u_p + \sum_{\substack{q=1 \\ q \neq p}}^{P} A_{p,q}u_q = f_p, \;\; p = 1, ..., P, \tag{2}$$

где $A_{p,q} \in \mathcal{R}^{N_p,N_q}$ — прямоугольные матричные блоки, получаемые при разбиении каждой блочной строки (и всей матрицы A) на блочные столбцы:

$$A_p = \{A_{p,q}; q = 1, ..., P\}, \;\; A = \{A_{p,q}; p, q = 1, ..., P\}.$$

Пусть СЛАУ (1) задана в CSR-формате, см. [3], с указанием числа строк N_p в каждой из P подсистем (2), в соответствии с разбиением матрицы A на блочные строки A_p.

Будем считать, что СЛАУ (1) представляет собой систему сеточных уравнений, аппроксимирующих некоторую многомерную краевую задачу для дифференциального уравнения, так что каждая компонента векторов u, f соответствует узлу сетки, общее число которых в расчетной сеточной области $\Omega^h = \bigcup\limits_{p=1}^{P} \Omega_p$ равно N. Отметим, что в методах конечных элементов (МКЭ, [4]) такие системы называются узловыми, но более сложные сеточные уравнения мы пока не рассматриваем. При этом блочное разбиение матриц и векторов соответствует разбиению (декомпозиции) Ω^h на P сеточных непересекающихся подобластей Ω_p, в каждой из которых находится N_p узлов, $N_1 + ... + N_P = N$, см. рис. 1.

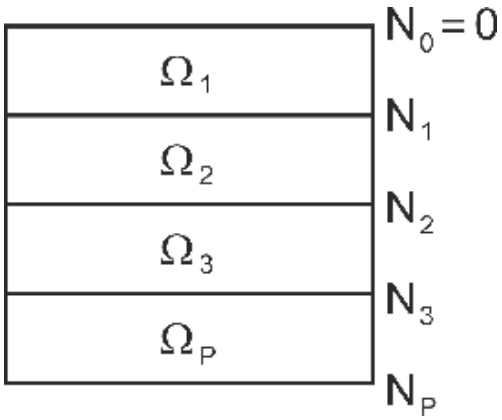

Рис. 1. Схема блочного разбиения матрицы

Описанная декомпозиция Ω^h не использует узлов-разделителей, т.е. условные границы подобластей не проходят через узлы сетки. Формальности ради можно считать, что внешность расчетной области есть неограниченная подобласть Ω_0 с числом узлов $N_0 = 0$.

Сеточное уравнение для i-го узла сетки может быть записано в виде

$$a_{i,i}u_i + \sum_{\substack{j\in\omega_i \\ j\neq i}} a_{i,j}u_j = f_i, \;\; i \in \Omega^h, \tag{3}$$

где через ω_i обозначается сеточный шаблон i-го узла, т.е. совокупность номеров всех узлов (как правило, являющихся геометрическими соседями), участвующих в i-м уравнении.

Если сеточные узлы и соответствующие переменные пронумерованы следующим образом: сначала идут подряд все узлы 1-й подобласти Ω_1 (неважно в каком внутреннем порядке), затем — узлы второй подобласти и т.д., — то вышеприведенное блочное разбиение матриц и векторов, приводящее к подсистемам (2), соответствует декомпозиции расчетной области без пересечения подобластей. При этом номера узлов из ω_i, соседних к i-у узлу, указываются в CSR-формате для i-й строки матрицы (фактически там указываются номера столбцов как для диагонального, так и внедиагональных элементов). Отметим, что взаимно однозначного соответствия алгебраической и геометрической (сеточной) интерпретации структуры СЛАУ может и не существовать. Типичный пример — одному узлу сетки может соответствовать $m > 1$ переменных в алгебраической системе. Соответствующая ситуация имеет место в междисциплинарных задачах, описываемых системами из m дифференциальных уравнений, у которых решение есть вектор-функция размерности m. Если такие "кратные" переменные нумеруются подряд, то структура СЛАУ приобретает "мелкоблочный" ($m \ll N$) вид, и на таких случаях мы остановимся в последующем особо.

При заданном блочном разбиении СЛАУ, или декомпозиции сеточной расчетной области без пересечения подобластей, для решения системы (1) можно построить хорошо

распараллеливаемый аддитивный метод Шварца, представимый итерационным процессом

$$A_{p,p}u_p^n = f_p - \sum_{\substack{q=1 \\ q\neq p}}^{P} A_{p,q}u_q^{n-1} \equiv g_p^{n-1}. \tag{4}$$

При вычисленных правых частях g_p^{n-1} нахождение всех u_p^n требует для каждой подобласти решения подсистем с порядками N_p, которые на каждой n-й итерации могут выполняться независимо и одновременно на разных процессорах. В данном случае перевычисление g_p^{n-1} фактически означает использование новых значений u_p^{n-1} для соседних подобластей в качестве граничного условия 1-го рода (Дирихле) для околограничных внутренних узлов из Ω_p.

Рассмотренное двойственное (матрично-алгебраическое и сеточно-геометрическое) представление структуры СЛАУ (1) может быть дополнено до триады путем введения графового описания этой же задачи. Каждому сеточному узлу с номером i (или i-й строке матрицы A) можно сопоставить вершину v_i некоторого графа G, а сеточному ребру, соединяющему узлы i и $j \in \omega_i$ (или j-й матричной строке) — сопоставить ребро графа $G = (V, E)$ и в соответствии с [5] определить множества вершин и ребер:

$$V = \{v_i;\ i = 1, ..., N\},\ \ E = \{(v_i, v_j)|a_{i,j} \neq 0\}. \tag{5}$$

Отметим следующий существенный технологический аспект, возникающий при реализации численного решения подсистемы (4). Традиционно универсальные программные "решатели" СЛАУ, ориентированные на матричный CSR-формат, предполагают сплошную нумерацию векторных компонент и соответствующих матричных строк, начиная с единицы или нуля. Однако, если для исходной СЛАУ информация дана в глобальной нумерации переменных, то в подобласти Ω_p требуется своя локальная нумерация. Таким образом, здесь возникает техническая задача переформирования CSR_p-формата для диагонального блока $A_{p,p}$ с переходом из глобальной нумерации в локальную. А после вычисления подвекторов u_p в подобластях, естественно, потребуется сборка глобального вектора u, т.е. перевод векторных компонент из локальной нумерации в глобальную.

Известно, что скорость сходимости итераций метода Шварца (4) возрастет, если декомпозицию расчетной области сделать с пересечением подобластей.

В силу этого мы дадим определение расширенной сеточной подобласти $\bar{\Omega}_p \supset \Omega_p$ с пересечениями, величину которых мы будем определять в терминах количества околограничных сеточных слоев, или фронтов, см. рис. 2.

На этой иллюстрации рассматриваемая сеточная подобласть Ω_p представляет собой квадрат, окруженный соседними сеточными подобластями $\Omega_1, ..., \Omega_8$ (сетка для простоты рассматривается квадратная, а шаблон сеточного уравнения — пятиточечный). Проводимые ниже рассуждения легко переносятся на общий случай неструктурированных сеток и произвольной конфигурации трехмерных подобластей.

Обозначим через $\Gamma_p^0 \in \Omega_p$ множество внутренних околограничных узлов из Ω_p, т.е. таких узлов $P_i \in \Omega_p$, у которых один из соседей не лежит в $\Omega_p (P_j \notin \Omega_p,\ j \in \omega_i,\ j \neq i))$. В Γ_p^0 определим подмножество узлов $\Gamma_{p,q}^0$, у которых соседние по шаблону узлы принадлежат смежной подобласти $\Omega_q, q \in Q_p^0$, где Q_p^0 — совокупность номеров подобластей, соседних с Ω_p. Таким образом,

$$\Gamma_p^0 = \bigcup_{q \in Q_p^0} \Gamma_{p,q}^0,$$

причем подмножества $\Gamma_{p,q}^0$ могут пересекаться, т.е. содержать околограничные узлы, у которых есть соседи из различных смежных подобластей.

Обозначим далее через Γ_p^1 множество узлов, соседних с узлами из Γ_p^0, но не принадлежащих Ω_p, через Γ_p^2 — множество узлов, соседних с узлами из Γ_p^1, но не принадлежащих

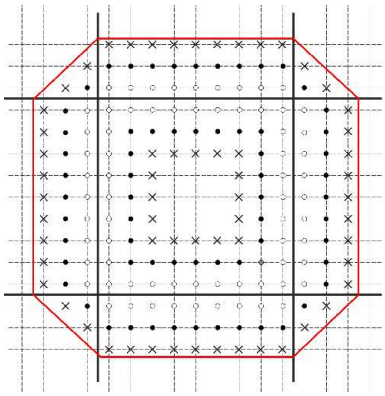

Рис. 2. Построение расширенных сеточных подобластей

объединению $\Gamma_p^1 \bigcup \Omega_p$, через Γ_p^3 — множество узлов, соседних с Γ_p^2, но не принадлежащих $\Gamma_p^2 \bigcup \Gamma_p^1 \bigcup \Omega_p$, и т.д. Соответственно эти множества назовем первым внешним слоем (фронтом) узлов, вторым слоем, третьим и т.д. Получаемое объединение узлов

$$\Omega_p^\Delta = \Omega_p \bigcup \Gamma_p^1 ... \bigcup \Gamma_p^\Delta$$

будем называть расширенной p-й сеточной подобластью, а целую величину Δ определяем как величину расширения, или пересечения (в терминах количества сеточных слоев). Случай $\Delta = 0$ фактически означает декомпозицию области Ω^h на подобласти без пересечений ($\Omega_p^0 = \Omega_p$), а обведенная на рис. 2 восьмиугольной условной границей расширенная подобласть $\bar{\Omega}_p = \Omega_p^\Delta$ соответствует значению $\Delta = 3$.

Множество $\Gamma_p^\Delta \in \Omega_p^\Delta$ представляет совокупность внутренних околограничных узлов расширенной подобласти Ω_p^Δ, а $\Gamma_p^{\Delta+1}$ есть множество внешних околограничных узлов. Условная граница Ω_p^Δ проходит между Γ_p^Δ и $\Gamma_p^{\Delta+1}$.

Аналогично Γ_p^0, множество Γ_p^Δ можно разбить на подмножества околограничных узлов

$$\Gamma_p^\Delta = \Gamma_{p,q_1}^\Delta \bigcup \Gamma_{p,q_2}^\Delta ... \bigcup \Gamma_{p,q_{m_p}}^\Delta$$

у которых соседние (по шаблону) находятся соответственно в подобластях $\Omega_{q_1}, \Omega_{q_2}, ..., \Omega_{q_{m_p}}$ (здесь m_p означает количество подобластей, пересекающихся с Ω_p^Δ, а $q_1, q_2, ..., q_{m_p}$ — номера этих подобластей). Отметим, что Γ_p^Δ может содержать "кратные" околограничные узлы, принадлежащие одновременно разным подмножествам $\Gamma_{p,q}^\Delta$, т.е. у которых есть соседи из различных смежных подобластей.

Если обозначить через $\Omega_{p,q}^\Delta$ пересечение какой-то подобласти Ω_q с расширенной подобластью $\bar{\Omega}_p$, то последнюю можно представить в форме объединения

$$\bar{\Omega}_p = \Omega_p \bigcup \Omega_{p,q_1}^\Delta \cdots \bigcup \Omega_{p,q_{m_1}}^\Delta, \tag{6}$$

причем каждое пересечение составляется из частей внешних околограничных фронтов:

$$\Omega_{p,q}^\Delta = \Gamma_{p,q}^1 \bigcup \Gamma_{p,q}^2 ... \bigcup \Gamma_{p,q}^\Delta, \;\; q = q_1, ..., q_{m_k}. \tag{7}$$

Таким образом, сборка матрицы СЛАУ для расширенной подобласти может быть интерпретирована как идентификация, объединение и перенумерация множества узлов, участвующих в (6), (7).

На формальном алгебраическом языке расширенные подсистемы уравнений и итерационный метод Шварца записываются аналогично (4):

$$\bar{A}_{p,p} \bar{u}_p^n = \bar{f}_p - \sum_{\substack{q=1 \\ q \neq p}}^{P} \bar{A}_{p,q} \bar{u}_q^{n-1} = \bar{g}_p^{n-1}. \tag{8}$$

Здесь размерность векторов $\bar{u}_p^n$ и $\bar{f}_p$ равна числу узлов $\bar{N}_p$ в расширенной подобласти $\bar{\Omega}_p$. При этом надо иметь в виду, что в пересечениях $\Omega_{p,q}^{\Delta}$ один узел принадлежит двум или более подобластям, а аналогичная изображенной на рис. 1 блочная структура матрицы A трудно представима, поскольку расширенный блок $\bar{A}_{p,p}$ в (8) состоит из всех строк блока $A_{p,p}$, а также из отдельных строк блоков, соответствующих соседним (геометрически) подобластям.

Остановимся теперь подробнее на формировании правых частей $\bar{g}_p^{n-1}$ или g_p^{n-1} в (7), (4). Для этого рассмотрим сеточное уравнение вида (3) для расположенного в $\bar{\Omega}_p$ i-го околограничного узла, у которого некоторые из соседей находятся в других подобластях $\Omega_q, q \neq p$, см. рис. 3:

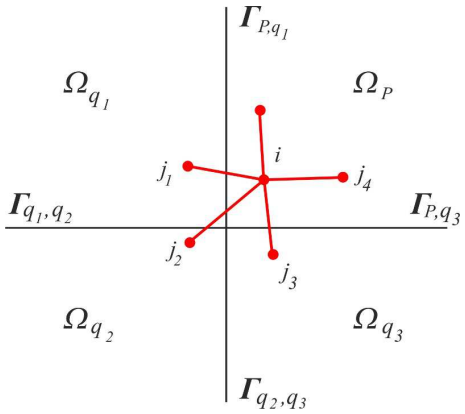

Рис. 3. Иллюстрация сеточного шаблона для околограничного узла

$$a_{i,i}u_i + \sum_{j\in\Omega_p} a_{i,j}u_j = f_i - \sum_{j\notin\Omega_p} a_{i,j}u_j. \tag{9}$$

Отметим, что в (9) i-й узел располагается и в $\bar{\Omega}_p$, и в какой-то из соседних подобластей. Отсюда строим итерационный процесс, несколько модифицировав данное уравнение:

$$\Big(a_{i,i} + \theta \sum_{j\notin\Omega_p} a_{i,j}\Big)u_i^n + \sum_{j\in\Omega_p} a_{i,j}u_j^n = f_i + \sum_{j\notin\Omega_p} a_{i,j}(\theta u_i^{n-1} - u_j^{n-1}). \tag{10}$$

Здесь $\theta \in [0,1]$ — некоторый итерационный параметр, который при $\theta = 0$ соответствует условию Дирихле в околограничном i-м узле, при $\theta = 1$ — условию Неймана, а при $0 < \theta < 1$ — условию 3-го рода, или Робена. Значения u_i^{n-1}, u_j^{n-1} в правой части (10) берутся с предыдущей итерации в узлах из соседних к Ω_p подобластей. Очевидно, что если итерационный процесс (10) сходится, то при любом θ его предел будет решением исходной системы (9).

Если ввести диагональную $D_p = diag\{\sum_{j\neq\Omega_p} a_{i,j}\}$, то итерационный процесс (10) можно записать в форме, обобщающей соотношение (8):

$$\bar{B}_p(\bar{u}_p^n - \bar{u}_p^{n-1}) = \bar{r}_p^{n-1} \equiv \bar{f}_p - (A\bar{u}^{n-1})_p, \tag{11}$$

где предобуславливающая матрица $\bar{B}_p$ имеет вид

$$\bar{B}_p = \bar{A}_{p,p} + \theta D_p. \tag{12}$$

В приведенных выше рассуждениях мы рассмотрели расширение подобласти Ω_p в наружную сторону. Однако такие же процедуры осуществляются и для соседних подобластей, что приводит к построению пересечений внутри самой Ω_p. Их реализация может осуществляться по сеточных слоям (фронтам), как и при расширении Ω_p.

3. Особенности реализации распределенных алгоритмов

В высокопроизводительных версиях декомпозиции областей существуют два ключевых момента: формирование CSR-форматов алгебраических подсистем для соответствующих подобластей и реализация параллельных по подобластям крыловских итерационных процессов якобиевого типа.

3.1. Метод построения квазисбалансированных сеточных подобластей

В начале данного пункта остановимся на одном факторе, имеющем большое значение для скорости сходимости итерационных процессов вида (4) по подобластям. Речь идет о топологии и некоторых других характеристиках разбиения расчетной сеточной области Ω^h на сеточные подобласти $\Omega_p^h,\ p = 1, ..., P$. Здесь следует отметить, что вообще говоря, нужно различать исходную непрерывную расчетную область Ω, в которой решаются дифференциальные и/или интегральные уравнения, и ее дискретный аналог Ω^h. Более того, декомпозицию можно делать в первую очередь: сначала разбить область на подобласти $\left(\Omega = \bigcup\limits_{p=1}^{P} \Omega_p\right)$, а затем в каждой из них строить "подсетку" Ω_p^h, которая геометрически "привязана" к соответствующей Ω_p. Однако мы эти аспекты сознательно опускаем, ограничиваясь только декомпозицией сеточной области.

Совокупность подобластей образует собой макросеть, или макрограф, у которого каждая подобласть представляет собой макровершину, а связи с геометрически соседними подобластями — макроребра. Как известно [6], скорость сходимости итераций по подобластям будет тем выше, чем меньше диаметр макрографа (максимальное из минимально возможных расстояний между макроузлами). При разбиении трехмерной области на достаточно большое число P подобластей диаметр макрографа есть $O(P)$ при одномерной декомпозиции, $O(P^{1/2})$ — при двумерной и $O(P^{1/3})$ — при трехмерной. Поэтому естественно, что оптимальной является трехмерная сбалансированная декомпозиция, при которой число узлов в каждой подобласти является примерно одинаковым, чтобы при одновременном решении алгебраических подсистем в подобластях соответствующие процессоры имели минимальные простои.

В некотором смысле альтернативным подходом к декомпозиции сеточной области является следующий. В соответствии с вышеприведенными рассуждениями, оптимальное разбиение кубической области с кубической же сеткой — это декомпозиция на кубические сеточные подобласти. Особенность этой конфигурации заключается в том, что при заданном числе узлов она имеет относительно малый диаметр соответствующего сеточного подграфа, т.е. сравнительно близкий к идеальному варианту — сфере. Именно это наблюдение мы и положим в основу рассматриваемого "локального" принципа декомпозиции: строить сеточные подобласти с примерно равным количеством узлов и имеющие относительно малые диаметры "своих" сеточных подграфов.

Итак, описываемый алгоритм декомпозиции преследует цель создания сеточных подобластей Ω_p, $p = 1, ..., P$, по возможности меньшего диаметра, где диаметр понимается как диаметр подграфа $G_p(V_p, E_p)$, порожденного [7] множествами содержащихся в подобласти вершин V_p и ребер E_p. Вообще говоря, поиск оптимального в терминах какого-либо критерия разбиения графа G является задачей как минимум полиномиальной сложности, хотя для решения больших СЛАУ нежелательна даже квадратичная зависимость от количества элементов графа N^V, N^E.

С этой точки зрения интересен следующий ход: с помощью какого-либо экономичного метода выполнить на первом этапе разбиение графа на связные непересекающиеся подмножества сравнительно небольшой мощности и малого диаметра, впоследствии рассматривая их как макровершины и декларируя наличие ребра между двумя "соседними" макровершинами в том и только том случае, если есть хотя бы одно ребро, соединяющее некоторые

элементы этих двух подмножеств. В дальнейшем предлагается интерпретировать полученный таким образом граф как огрубленную (агрегированную) копию исходного, применяя более сложные (или, наоборот, простые) алгоритмы для формирования последующих и конечного разбиений.

Не останавливаясь пока на методах агрегации каждого этапа, отметим, что предлагаемый подход является многошаговым и фактически образует последовательность вложенных сеток, или макросеток, из подобластей. Если обозначить через s номер шага в рассматриваемой иерархической структуре, то для формализации данного процесса можно ввести следующие обозначения: $G^{(s)}(V^{(s)}, E^{(s)})$, $s = 0, 1, ..., M$, — граф макросетки s-го уровня ($s = 0$ соответствует начальному сеточному графу, в котором вершина представляет простой узел сетки, а M есть количество стадий агрегации, которое заранее неизвестно), $N_V^{(s)}$ и $N_E^{(s)}$ — число макровершин и макроребер макросетки s-го уровня.

В целях построения начального разбиения предлагается использовать поиск в ширину [5]. Данный метод обладает привлекательным свойством помечать в первую очередь вершины с наименьшим расстоянием от исходной, что соответствует цели создания подмножеств малого диаметра. Поиск ведется среди вершин, еще не относящихся к какому-либо подмножеству, до тех пор, пока количество помеченных вершин не достигнет определенного предела (желаемого размера макровершины $N_{max}^{(s)}$, где $N_{max}^{(0)}$ означает число узлов сетки, которое допускается включать в макровершину первого уровня). Все помеченные на j-м шаге алгоритма вершины объявляются принадлежащими j-му подмножеству, после чего процедура повторяется до исчерпания множества вершин.

Формализуем алгоритм поиска в ширину. Через $Adj(v)$ обозначим множество вершин, смежных вершине v. Пусть также Q — структура данных типа очередь [5]. Обозначим через $Q \leftarrow S$ добавление множества S к очереди (в произвольном порядке), $v \leftarrow Q$ — изъятие элемента из очереди. Для выполнения поиска в ширину дополнительно введем C и W — целочисленные массивы (списки) цветов и весов вершин (смысл последних будет раскрыт позднее). Через $C(v)$ и $W(v)$ обозначим текущие цвет и вес вершины v. Перед стартом алгоритма цвет всех вершин инициализируется нулем, а их вес полагаем равным единице. Также введем n_{max} — желаемый размер макровершины. Тогда алгоритм заключается в начальном выборе произвольной вершины v из множества V и в дальнейших преобразованиях графа в соответствии со следующим псевдокодом.

$i = 1$
while $\{u \in V \mid C(u) = 0\} \neq \emptyset$
  pick any v *from* $\{u \in V \mid C(u) = 0\}$
  $Q := \{v\}$
  $n = 0$
  while ($n < n_{max}$ *and* $Q \neq \emptyset$)
    $v \leftarrow Q$
    $C(v) := i$
    $Q \leftarrow (Adj(v) \cap \{u \in V \mid C(u) = 0\}) \setminus Q$
    $n = n + W(v)$
  end while
  $i = i + 1$
end while

После работы алгоритма объявим $V_i = \{u \mid C(u) = i\}$. Сформируем граф с макровершинами $G' = (V', E')$, где $V' = \{V_1, \dots V_{N'}\}$, N' — количество получившихся макровершин, а множество ребер E' определяется так:

$$E' = \{(V_i, V_j) \mid \exists u \in V_i, v \in V_j \mid (u, v) \in E\}.$$

Указанный выше алгоритм можно использовать и для формирования собственно подобластей нужного размера, положив $W(V_i) = |V_i|$, а n_{max} равным желаемому размеру подобласти. На данный момент именно такой алгоритм используется в библиотеке Krylov, хотя, вообще говоря, представляется целесообразным использование некоторого аналога очереди с приоритетом, где, например, приоритет тем выше, чем больше ребер связывают рассматриваемую макровершину с текущим составом формируемой подобласти, и тем ниже, чем меньше вес (мощность) этой макровершины (стоит заметить, что в таком случае разумно вычислять приоритет макровершин в момент извлечения элемента из очереди, а не в момент добавления). Такой алгоритм будет отдавать предпочтение макровершинам с размером меньше типичного, выполняя их поглощение, и тем макровершинам, включение которых способствует уменьшению "площади" границы подобласти (т.е. количества ребер, связывающих эту подобласть с остальными).

Из эмпирических соображений предполагается, что созданные таким образом подобласти будут обеспечивать хорошую скорость сходимости метода Шварца. Отметим, что хотя рассмотренный алгоритм является детерминированным, качественные особенности декомпозиции заранее являются трудно предсказуемыми, хотя ее количественные характеристики (количества узлов и диаметры подобластей можно определить апостериори).

3.2. Особенности программной реализации

Рассмотрим более подробно некоторые вопросы, касающиеся непосредственно организации вычислительного процесса, причем остановимся для краткости на алгоритмах без пересечения подобластей.

В первую очередь заметим, что выполнение одного шага аддитивного метода Шварца (8) состоит из P независимых операций решения СЛАУ, для реализации каждой из которых формируется MPI-процесс, а выполнение ее осуществляется одним из процессоров с указываемым количеством доступных вычислительных ядер, или потоков N_{th}, которыми выполняется "внутреннее" распараллеливание над общей памятью средствами OpenMP.

Во вторых, реально итерации по подобластям проводятся не по алгоритму Шварца (8), а с помощью предобусловленного "гибкого" обобщенного метода минимальных невязок FGMRES [8], в котором блочно-диагональная матрица с блоками $A_{p,p}$ на главной диагонали выступает в качестве предобуславливателя. Решение подсистемы для каждой p-й подобласти осуществляется прямым алгоритмом или итерационным, и в последнем случае предобуславливатель фактически является переменным. Реализация FGMRES осуществляется в распределенном варианте, т.е. "глобальные" векторные и векторно-матричные операции выполняются в основном "своими" процессорами, при минимальных обменах между ними.

Учитывая все вышесказанное, логичной представляется следующая схема реализации вычислений. Будем хранить векторы так же, как и матрицу (см. рис. 1), по N_p компонент в каждом процессе, соответствующем "своей" подобласти. В таком случае векторно-векторные операции и обращение предобуславливающего оператора можно выполнять без обмена данными между MPI-процессами (исключение составляет лишь скалярное произведение, где объем пересылаемых данных невелик). Для вычисления действия оператора A на вектор будем пересылать лишь те данные, которые требуются для вычисления соответствующих частей вектора, т.е. только между соседними подобластями.

Итерации по подобластям продолжаются до выполнения условий

$$\|r^n\| = \|f - Au^n\| \le \varepsilon^e \|f\|, \quad \varepsilon^e \ll 1, \quad \text{или} \quad n \le n^e_{max},$$

где ε^e и n^e_{max} — задаваемые критерии останова. Если решение вспомогательных СЛАУ в подобластях осуществляется итерационным методом, то для него задаются "внутренние" критерии ε^i, n^i_{max}.

4. Примеры численных экспериментов

Главная задача данного пункта заключается в том, чтобы продемонстрировать эффективность итерационного метода по подобластям, которая в значительной степени зависит от качества декомпозиции. Расчеты проводились на кластере ИВМиМГ СО РАН [11] для количества подобластей $P = 1, 2, 4, 8, 16, 32$, причем случай $P = 1$ соответствует решению задачи без применения декомпозиции.

В таблице 1 мы приводим результаты по решению модельной задачи Дирихле для диффузионно-конвективного уравнения в декартовой системе координат x, y, z с постоянными конвективными коэффициентами $p = q = r = 16$, см. [9, 10], в кубической области $\Omega = [0, 1]^3$, аппроксимируемой на кубической сетке с числом узлов $N = N_x^3$ с помощью экспоненциальной семиточечной схемы. Граничные условия соответствуют точному решению $u(x, y, z) = 1$, а начальное приближение выбрано $u^0(x, y, z) = 0$.

Отметим, что в приведенных расчетах использовался двухуровневый вариант декомпозиции ($s = 0, 1$ в пункте 2), причем выбирались значения $N_{max}^{(0)} = \sqrt{N}$, $N_{max}^{(1)} = N/P$.

Таблица 1. Решение модельной задачи с использованием PARDISO в подобластях, $\varepsilon^e = 10^{-7}$

$N \setminus P$	1	2	4	8	16	32
64^3	1	29	32	39	54	78
	32.2	14.2	6.39	2.88	3.42	1.58
	164	93.5	55.1	46.3	232	200
128^3	1	40	44	53	75	108
	885	221	86.6	30.1	20.4	12.6
	10340	4987	1785	782	1232	1615
256^3	—	—	60	72	102	142
			2222	332	212	138
			100009	13450	18400	18210

В каждой клетке таблицы 1 представлены сверху вниз: количество внешних итераций, астрономическое и процессорное время (t_a и t_{cpu}, последнее — суммарное по всем MPI-процессам и по всем вычислительным потокам). В качестве решателя в подобластях использовался PARDISO из библиотеки Intel MKL [3], при заданном числе потоков $N_{th} = 12$. Из-за высоких требований данного решателя к объему оперативной памяти оказалось невозможным получить на имеющемся оборудовании результаты, соответствующие незаполненным клеткам таблицы.

В таблице 2 приводятся результаты для решения той же задачи, однако для решения СЛАУ в подобластях использовался предобусловленный метод бисопряженных градиентов со стабилизацией (BiCGStab), причем для предобуславливания использовалась модификация Айзенштата [4, 8] алгоритма неполной факторизации, без использования какого-либо "внутреннего" распараллеливания. Критерии останова полагались следующими: $\varepsilon^i = 0.1$, $n_{max}^i = 10$.

В таблице 3 представлена зависимость между критериями останова ε^i, n_{max}^i, фактически отвечающих за точность решения СЛАУ в подобластях, и величинами, отражающими скорость сходимости внешнего итерационного процесса, а также общую вычислительную сложность. В каждой клетке таблицы перечислены: количество внешних итераций n^e, общее количество внутренних итераций n^i (суммарное по всем подобластям и по всем внешним итерациям), астрономическое и процессорное время в секундах. Для сравнения также приводятся величины, получаемые при решении СЛАУ в подобластях прямым методом PARDISO.

Таблица 2. Решение модельной задачи с использованием предобусловленного итерационного метода BiCGStab в подобластях, $\varepsilon^e = 10^{-7}, \varepsilon^i = 0.1, n^i_{max} = 10$

$N \setminus P$	1	2	4	8	16	32
64^3	9	29	34	47	65	98
	4.05	6.35	3.44	1.54	1.59	2.26
	4.05	12.7	13.2	12.3	20.9	77.3
128^3	18	58	56	67	92	132
	64.9	115	57.4	25.2	15.9	18.3
	64.9	230	229	201	215	510
256^3	21	64	74	92	143	92
	606	1045	647	443	271	231
	606	2079	2588	3553	3916	6627

Таблица 3. Влияние точности решения СЛАУ в подобластях на скорость сходимости и ресурсоемкость

Задача \ критерии	$\varepsilon^i = 0.1$ $n^i_{max} = 10$	$\varepsilon^i = 0.01$ $n^i_{max} = 20$	$\varepsilon^i = 0.001$ $n^i_{max} = 40$	PARDISO
$P = 8$ $N = 128^3$	67	57	54	53
	3297	5280	7040	424
	25.2	39.9	52.2	30.1
	201	318	415	782
$P = 16$ $N = 256^3$	143	111	108	102
	15929	18843	25489	1632
	271	319	427	212
	3916	5112	6824	18400

Отметим, что во всех приведенных экспериментах итоговая погрешность численного решения была примерно одинаковой и равнялась 10^{-6}, что достаточно хорошо соответствует критерию $\varepsilon^e = 10^{-7}$.

По приведенным результатам можно сделать следующие выводы.

- С увеличением количества процессоров P, а точнее говоря — MPI-процессов, или подобластей, до определенного числа, наблюдается существенное ускорение решения СЛАУ как для прямого, так и итерационного алгоритмов в подобластях. Однако, в силу роста числа внешних итераций n^e с увеличением P, а также ввиду роста количества обменов при этом, наблюдается минимум $t_a(P)$, который на грубой сетке с $N = 64^3$ наступает при $P = 8$, а с увеличением N проявляется при гораздо большем числе P. Уменьшения числа n^e можно достичь за счет применения декомпозиции с пересечением подобластей и использования других приемов ускорения итераций по подобластям, однако вопрос о возможности монотонного роста ускорения вычислений с ростом P остается открытым.

- Сравнение времен решения СЛАУ при использовании для решения СЛАУ в подобластях прямого или итерационного алгоритмов показывает, что в различных ситуациях выигрывает то один, то другой подход. Как и следовало ожидать из теоретических оценок вычислительной сложности, прямой алгоритм проигрывает итерационному, ес-

ли число узлов в подобласти достаточно велико. Однако с ростом значений P ресурсоемкость прямого метода убывает быстрее, чем итерационного. Справедливости ради следует заметить, что времена счета соответствуют реализации прямого алгоритма с распараллеливанием по 12 потокам, а итерационного — без распараллеливания.

- Что касается решения СЛАУ с применением итерационного алгоритма в подобластях, то здесь существенные резервы ускорения вычислений заложены в оптимизации критериев окончания внутренних итераций. Как видно из таблицы 3, уменьшение ε^i и увеличение n^i_{max} приводит к излишним итерациям в подобластях и увеличению времени счета. Приведенные результаты носят предварительный характер и оставляют открытым вопрос об оптимальной стратегии выбора критериев останова внутренних итераций.

- В приведенных экспериментах характер зависимости величины t_{cpu} от значений N и P, которая фактически отражает общий объем арифметических операций, примерно такой же, что и t_a. Однако следует иметь в виду, что эти показатели имеют относительно косвенный смысл, поскольку, например, они зависят от загрузки МВС.

В заключение заметим, что данная работа затрагивает лишь малую толику вопросов, касающихся изучения и оптимизации алгоритмов решения больших разреженных СЛАУ с помощью декомпозиции, что подчеркивает актуальность и насущную потребность дальнейшего исследования этой проблемы.

Литература

1. Domain Decomposition Methods. URL: http://ddm.org (дата обращения : 14.03.2012)

2. 22nd International Conference on Domain Decomposition Methods (DD22). URL: http://dd22.ics.usi.ch/ (дата обращения : 31.11.2013)

3. Intel (R) Math Kernel Library from Intel. URL: http://software.intel.com/en-us/articles/intel-mkl/

4. Ильин В.П. Методы и технологии конечных элементов / В.П.Ильин — Новосибирск, изд. ИВМиМГ СО РАН, 2007.

5. Писсанецки С. Технология разреженных матриц / С.Писсанецки — М.: Мир, 1988.

6. Bramble J.H. Convergence estimates for product iterative methods with applications to domain decomposition / J. Bramble, J. Pasciak, J. Wang, J. Xu // Math. Comp. — 1991. — V. 57, № 195. — P. 1–21.

7. Берж К. Теория графов и ее применения / К.Берж — М.: Изд-во иностр. лит., 1962.

8. Saad Y. Iterative methods for sparse linear systems // NY: PWS Publish., 1996.

9. Andreeva M.Yu., Il'in V.P., Itskovich E.A. Two solvers for nonsymmetric SLAE // Bull. NCC, series: "Num. Anal.", iss. 12, 2003, P. 1–16.

10. Бутюгин Д.С., Ильин В.П., Перевозкин Д.В. Методы параллельного решения СЛАУ на системах с распределенной памятью в библиотеке Krylov // Вестник ЮУрГУ. Серия "Вычислительная математика и информатика", т. 47, № 306, 2012, С. 5–19.

11. Кластер НКС-30Т: URL: http://www2.sscc.ru/НКС-30Т/НКС-30Т.htm (дата обращения: 15.02.2013).