

Об одном варианте метода декомпозиции области для решения краевых задач на многопроцессорных суперЭВМ *

В.М. Свешников

*Институт вычислительной математики
и математической геофизики СО РАН*

Аннотация

Переложен новый вариант метода декомпозиции расчетной области на подобласти, сопрягаемые без наложения с условиями типа Дирихле-Дирихле. Его основой является непосредственная аппроксимация уравнения Пуанкаре-Стеклова на сетке, которая вводится на границе сопряжения подобластей, при помощи системы линейных алгебраических уравнений (СЛАУ). Элементы матрицы СЛАУ можно не вычислять явно, а использовать лишь ее действие на некоторое приближение, что дает итерационный метод по подобластям. Рассмотрены технологические особенности реализации данного подхода при решении краевых задач на многопроцессорных суперЭВМ.

1 Введение

Метод декомпозиции области служит основным инструментом распараллеливания решения краевых задач на многопроцессорных суперЭВМ [1]. Наиболее естественным образом разбиение расчетной области на подобласти происходит без наложения подобластей, так как в этом случае отсутствует такая нефизическая деталь, как зона перехлеста подобластей. Основным отличием предлагаемого в настоящей работе подхода к построению методов декомпозиции расчетной области на подобласти, сопрягаемые без наложения, является следующее. Уравнение Пуанкаре-Стеклова для искомой функции на границе сопряжения γ аппроксимируется при помощи системы линейных алгебраических уравнений (СЛАУ), построенных в узлах сетки, введенной на γ . Элементы матрицы данной СЛАУ явно не вычисляются, а используется лишь ее действие на какой-либо вектор в итерационном процессе. Для этих целей наиболее подходят быстросходящиеся итерационные методы в подпространствах Крылова. Используя разбиение на подобласти как макросетку, можно получить хорошее начальное приближение, что значительно сокращает число итераций, необходимых для достижения заданной точности.

Приводятся технологические особенности распараллеливания данного подхода при решении краевых задач на многопроцессорных суперЭВМ.

Настоящая работа является обобщением работ автора [2,3], в которых данный подход рассматривался для случая квазиструктурированных прямоугольных локально-модифицированных сеток.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 10-01-90009)

2 Постановка задачи и описание алгоритма

Пусть требуется найти функцию $\varphi(T)$, являющуюся решением краевой задачи

$$\begin{aligned} L\varphi(T) &= g(T), \quad T \in D, \\ l\varphi(T)_\Gamma &= 0, \end{aligned} \quad (1)$$

где D – расчетная область, Γ – ее граница ($D \cup \Gamma = \bar{D}$), L – эллиптический дифференциальный оператор, l – оператор граничных условий, T – точка наблюдения. (рассматривается двумерная область). В дальнейшем мы будем предполагать существование и единственность достаточно гладкого решения задачи (1).

Разобьем область D на заданное число N_D подобластей D_K ($K = \overline{1, N_D}$) и обозначим через $\gamma_{n,m}$ ($n \neq m, \quad n, m = \overline{1, N_D}$) границу сопряжения подобластей D_n и D_m , причем положим $\gamma_{n,m} = \gamma_{m,n}$. Обозначим через Λ множество пар индексов n, m , которые определяют неповторяющиеся, непустые множества $\gamma_{n,m}$. Пусть T_P ($P = \overline{1, N_P}$) – точки пересечения границ $\gamma_{n,m}$ (макроузлы), N_P – число макроузлов, а $\gamma = \bigcup_{n,m \in \Lambda} \gamma_{n,m}$ – объединение этих границ. Тогда справедливо равенство

$$D = \bigcup_{K=1}^{N_D} D_K \bigcup_{P=1}^{N_P} T_P \bigcup \gamma.$$

Запишем на γ уравнение Пуанкаре-Стеклова

$$d_n(w_{n,m}) - d_m(w_{n,m}) = 0, \quad (2)$$

где $w_{n,m}$ – функция, определенная на $\gamma_{n,m}$, $d_r(w_{n,m}) = \left(\frac{\partial \varphi(w_{n,m})}{\partial \vec{n}} \right)_i$ ($r = n, m$), \vec{n} – нормаль к $\gamma_{n,m}$, а величины в скобках относятся соответственно к подобластям D_n и D_m . Направление нормали в данном случае не важно, но для определенности положим, что \vec{n} – это внутренняя нормаль по отношению к D_n .

Уравнение (2) будем решать приближенно. Для этого построим на $\gamma_{n,m}$ сетку $\omega_{n,m} = \{T_k^{(n,m)} \in \gamma_{n,m}, \quad k = \overline{1, M_{n,m}}\}$ с шагами $h_k^{(n,m)} = s(T_{k+1}^{(n,m)}) - s(T_k^{(n,m)}) > 0$, $k = \overline{0, M_{n,m}}$, где $M_{n,m}$ – заданное целое число, а $s(T_k^{(n,m)})$ – длина отрезка $[T_0^{(n,m)}, T_k^{(n,m)}]$ границы $\gamma_{n,m}$, отсчитываемая от точки $T_0^{(n,m)}$. Подчеркнем, что точки $T_0^{(n,m)}, T_{M_{n,m}+1}^{(n,m)}$ совпадают с макроузлами и не являются узлами сетки $\omega_{n,m}$.

Заменим задачу (2) отыскания функции $w_{n,m}$ непрерывного аргумента заменим приближенной задачей нахождения функции $v_{n,m}$ дискретного аргумента, состоящей в решении уравнений

$$f_{n,m} \equiv \tilde{d}_n - \tilde{d}_m = 0, \quad n, m \in \Lambda. \quad (3)$$

Здесь $\tilde{d}_i = \{\tilde{d}_k^{(r)}\}$ – вектор с компонентами $\tilde{d}_k^{(r)} = \left(\frac{\partial \tilde{\varphi}(v_{n,m})}{\partial \vec{n}} \right)_k^{(r)}$, $r = n, m, \quad k = \overline{1, M_{n,m}}$, $\tilde{\varphi}$ – решение приближенной задачи во всей области D . Введем сетку $\omega = \bigcup_{n,m \in \Lambda} \omega_{n,m}$, на которой определим вектор f как объединение векторов $f_{n,m}$. Дополнив ее макроузлами, образуем сетку $\bar{\omega} = \{\omega, T_1, T_2, \dots, T_P\}$ и вектор v в ее узлах. Введем единую нумерацию $k = \overline{1, M + P}$, где $M = \sum_{n,m \in \Lambda} M_{n,m}$, для всех узлов сетки $\bar{\omega}$. В силу линейности исходной задачи величины $f_i = f_i(v_1, v_2, \dots, v_M, v_{M+1}, \dots, v_{M+P})$ являются линейными функциями, которые можно записать как

$$f = Av + b, \quad (4)$$

где $A = \{a_{i,j}, i = \overline{1, M}, j = \overline{1, M+P}\}$ – прямоугольная матрица, а $f = \{f, i = \overline{1, M}\}$, $b = \{b_i, i = \overline{1, M}\}$, $v = \{v_j, j = \overline{1, M+P}\}$ – векторы. Тогда уравнения (3) представляются в виде системы линейных алгебраических уравнений (СЛАУ)

$$Av + b = 0. \quad (5)$$

Дополнив (5) уравнениями в макроузлах

$$(L^{(P)}v)_j = g_j, \quad j = \overline{M+1, M+P},$$

где $L^{(P)}$ – аппроксимация исходного дифференциального оператора на сеточном шаблоне, включающем узлы сетки ω и один из макроузлов, получим СЛАУ

$$\bar{A}v + \bar{b} = 0, \quad (6)$$

с квадратной матрицей \bar{A} порядка $M+P$, и вектором $\bar{b} = (b, g)'$.

Из формулы (4) следует, что вектор b можно вычислить из решения исходной краевой задачи, положив в узлах ω искомую функцию равной нулю. Элементы матрицы \bar{A} можно не вычислять, а использовать лишь действие этой матрицы на какой-либо вектор согласно формуле (4). Тогда решение системы уравнений (6) проводится итерационным методом, использующим данное действие и сам вектор. Таким требованиям удовлетворяет семейство быстроходящихся итерационных методов в подпространствах Крылова, изложенных в монографии [4]. На каждом приближении ξ при этом решаются краевые задачи в подобластях D_K с условиями Дирихле на γ , которые определяются как $u^{(\xi)}|_\gamma = Sv^{(\xi)}$, где S – оператор интерполирования. Число итераций может быть существенно сокращено путем построения хорошего начального приближения $v^{(0)}$. Для этого решается исходная краевая задача на макросетке, координатные линии которой суть линии разбиения на подобласти, и полученное решение интерполируется в узлы сетки ω .

3 Параллельные технологии проведения расчетов

Распараллеливанию в первую очередь подлежит итерационный процесс по подобластям, который занимает подавляющую часть времени решения всей задачи. При решении подзадач в подобластях сеточными методами важным звеном является отображение сеточных данных на вычислительную сеть. Обычно принятый способ отображения: одна подобласть – один процессор в данном случае не эффективен, так как подобласти могут содержать различное число узлов, в которых вычисляются значения искомой функции (в дальнейшем – счетных узлов), что приводит к разбалансировке загрузки процессоров. Поэтому подобласти группируются в объединения $U_l, l = \overline{1, L}$, где L – известное число, с целью обеспечения приблизительно равной загрузки процессоров. Для этого в каждое объединение $U_l = \bigcup_k \bar{\Omega}_{h,k}^{(l)}$ включаются такие подсетки $\bar{\Omega}_{h,k}^{(l)}$, которые давали бы в сумме $N_l = \sum_k N_k^{(l)}$ число счетных узлов $N_l \approx N_U$, где N_U – заданная величина, а $N_k^{(l)}$ – число счетных узлов в подобластях. Алгоритм группировки подобластей для идеальной балансировки должен строиться на решении задачи линейного программирования. При этом может оказаться, что в объединение включаются не только соседние подобласти, но и подобласти, разделенные подобластями из других объединений. Последнее обстоятельство может привести к значительному увеличению объема медленных операций пересылки в системах с разделенной памятью. Обмены не происходят между соседними подобластями одного объединения. В связи с этим был принят следующий алгоритм построения объединений. Определяется число N_U , которое может быть равно максимальному числу узлов в подобласти или вычисляться как $N_U = N_h/L$, где N_h

– суммарное число счетных узлов во всех подсетках. Для каждой свободной, то есть не включенной ни в одно объединение, подобласти просматриваются свободные соседи. Количество узлов просмотренных подсеток суммируется. Если после просмотра k -й подсетки в l -м объединении окажется $N_l > N_U$, то процесс группировки заканчивается, причем при $N_l - N_U > 0.5N_k^{(l)}$ последняя подсетка не включается в данное объединение.

На текущем процессоре, который отвечает за одно объединение, выполняются следующие вычислительные работы: 1) вычисление искомой функции в подобластях, 2) расчет нормальных производных на сторонах подобластей, входящих в объединение, 3) вычисление разностей производных (3) на смежных сторонах подобластей, 4) реализация очередного шага итерационного процесса по подобластям. На каждом из перечисленных технологических этапов происходит обмен информацией с соседними объединениями. Здесь следует отметить, что, несмотря на некоторое усложнение алгоритмов и технологии распараллеливания по сравнению с расчетами на равномерных сетках, рассмотренный подход оправдан, так как позволяет проводить декомпозицию области, исходя из физических особенностей исходной краевой задачи.

Литература

1. A. Quarteroni., A Valli. Domain Decomposition Methods for Partial Differential Equations // Oxford: Clarendon Press. 1999.
2. В.М. Свешников "Построение прямых и итерационных методов декомпозиции" // Сибирский журнал индустр. матем. Т12, №3. 2009. С.99-109.
3. Свешников В.М. Параллельные технологии решения краевых задач на квазиструктурированных сетках. // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20-25 сентября 2010 г., г. Новосибирск). М.: Изд-во МГУ, 2010. С. 238 – 241.
4. В.П. Ильин Методы и технологии конечных элементов. // Новосибирск: ИВМиМГ (ВЦ) СО РАН. 2007.