

УДК 519.632

ПАРАЛЛЕЛЬНЫЕ МЕТОДЫ ДЕКОМПОЗИЦИИ В ПРОСТРАНСТВАХ СЛЕДОВ

В. П. Ильин¹, Д. В. Кныш²

Предлагаются двухуровневые крыловские итерационные методы сопряженных направлений для следов искомым решений на внутренних границах подобластей при пространственной декомпозиции многомерных краевых задач. Внешний итерационный процесс представляет собой решение уравнения Пуанкаре–Стеклова с наложением или без наложения подобластей, а внутренний — решение независимых вспомогательных задач в подобластях. Экспериментально исследуется влияние размеров пересечений подобластей, типов итерируемых внутренних граничных условий, а также точности решения вспомогательных краевых задач на скорость сходимости методов декомпозиции. Приводятся результаты решений методических краевых задач, демонстрирующие эффективность распараллеливания методов декомпозиции на МВС с распределенной и общей памятью в зависимости от значений расчетных параметров итерационных процессов. Работа выполнена при финансовой поддержке РФФИ и Отделения математических наук РАН (коды проектов 11-01-00205-а и 1.3.4 соответственно). Статья рекомендована к публикации Программным комитетом Международной научной конференции “Параллельные вычислительные технологии” (ПАВТ-2011; <http://agora.guru.ru/pavt2011>).

Ключевые слова: уравнение Пуанкаре–Стеклова, перехлест, наложение, декомпозиция, уравнение Пуассона, альтернирующий метод Шварца.

1. Введение. Альтернирующий метод Шварца, разработанный изначально как средство теоретического исследования сложных краевых задач математической физики, впоследствии породил большое семейство алгоритмов декомпозиции областей, ставших основным орудием распараллеливания при решении многомерных дифференциальных уравнений. Указанным проблемам посвящены многочисленные публикации российских и зарубежных авторов [1–14], а также регулярные международные конференции под общим названием “Domain Decomposition” [15].

Общая идея методов декомпозиции заключается в следующем. Пусть в заданной расчетной области Ω с границей Γ требуется решить линейную дифференциальную краевую задачу

$$Lu = f(\mathbf{r}), \quad \mathbf{r} \in \Omega; \quad lu|_{\Gamma} = g, \quad (1)$$

где u , f , g — искомая и заданные функции, а L и l — операторы, определяющие исходное уравнение и граничные условия.

Разобьем Ω на P пересекающихся или непересекающихся подобластей: $\Omega = \bigcup_{q=1}^P \Omega_q$, границу каждой из которых обозначим через Γ_q , а замыкание области — через $\bar{\Omega}_q = \Omega_q \cup \Gamma_q$. Если $\Gamma_{q,q'}$ при $q' \neq q$ означает часть границы Γ_q , принадлежащую $\bar{\Omega}_{q'}$, то можно записать $\Gamma_q = (\Gamma \cap \bar{\Omega}_q) \cup_{q' \in \omega_q} \Gamma_{q,q'}$, где выражение в скобках представляет собой внешнюю границу подобласти Ω_q (если последняя является околограничной), а остальные $\Gamma_{q,q'}$ составляют внутреннюю границу, причем ω_q есть совокупность номеров подобластей $\Omega_{q'}$, соседних к Ω_q . В случае $\Gamma_{q,q'} = \Gamma_{q,q'}$ области Ω_q и $\Omega_{q'}$ касаются (сопрягаются) без наложения, т.е. их пересечение $\Delta_{q,q'} = \Omega_q \cap \Omega_{q'}$ есть пустое множество. Исходной задаче (1) сопоставим следующие совокупности вспомогательных краевых задач в подобластях:

$$Lu_q(\mathbf{r}) = f_q(\mathbf{r}), \quad \mathbf{r} \in \Omega_q, \quad l_{q,q'}u_q|_{\Gamma_{q,q'}} = g_{q,q'} \equiv l_{q,q'}(u_{q'})|_{\Gamma_{q',q}}, \quad q' \in \omega_q, \quad q = 1, \dots, P. \quad (2)$$

На внешней части границы Γ_q , если она существует, ставится исходное краевое условие из (1). Типичное условие сопряжения на внутренней границе записывается в виде краевого условия 3-го рода, или Робина:

$$\alpha_q u_q + \beta_q \frac{\partial u_q}{\partial n_q} \Big|_{\Gamma_{q,q'}} = g_{q,q'} \equiv \alpha_q u_{q'} + \beta_q \frac{\partial u_{q'}}{\partial n_q} \Big|_{\Gamma_{q',q}}. \quad (3)$$

¹ Институт вычислительной математики и математической геофизики СО РАН, просп. Лаврентьева, 6, 630090, Новосибирск; гл. науч. сотр., e-mail: ilin@sscc.ru

² Новосибирский государственный университет, механико-математический факультет, ул. Пирогова, д. 2, 630090, Новосибирск; аспирант, e-mail: dasha_knysh@gorodok.net

Здесь $\alpha_q \beta_q \geq 0$, $\frac{\partial u_q}{\partial n_q} \Big|_{\Gamma_{q,q'}}$ — одностороннее значение (внутреннее по отношению к Ω_q) внешней нормальной производной к границе Γ_q , а случаи $\beta_q = 0$ или $\alpha_q = 0$ соответствуют условиям Дирихле или Неймана. Сами коэффициенты α_q и β_q являются функциями координат \mathbf{r} и обычно выбираются кусочно-постоянными, одинаковыми для фиксированной пары q и q' .

Предполагается, что решения задач (1) и (2) существуют, являются единственными и совпадают. Решение исходной задачи (1) (сложной, или “большой”) ищется с помощью итерационного решения последовательности более простых вспомогательных задач (2):

$$Lu_q^n = f_q, \quad l_{q,q'} u_q^n \Big|_{\Gamma_{q,q'}} = l_{q,q'} u_{q'}^{n-1} \Big|_{\Gamma_{q',q}}, \quad q = 1, \dots, P, \tag{4}$$

причем условия сопряжения имеют выписанный вид для всех $q' = \omega_q$. Если $u_q^n \rightarrow u_q$ при $n \rightarrow \infty$, что обеспечивается выбором соответствующих $\Gamma_{q,q'}$, α_q и β_q , то мы получаем параллельный алгоритм, поскольку каждая из задач (4) для фиксированных n и q может решаться одновременно при наличии P процессоров. Данный итерационный процесс представляет собой метод одновременных смещений, или метод Якоби. Отметим, что если в правой части условия сопряжения (4) брать $u_{q'}^n$ вместо $u_{q'}^{n-1}$ при $q' < q$ для уже рассчитанных на данной итерации под областей $\Omega_{q'}$, то мы приходим к методу последовательных смещений Зейделя. Последний может иметь более быструю сходимость, но для него неприменимо естественное распараллеливание.

Численные алгоритмы решения задач (1)–(4) основываются на дискретизации, т.е. на построении сетки и аппроксимации методами конечных разностей, конечных объемов или конечных элементов (МКР, МКО или МКЭ [16, 17]), в результате чего мы приходим к системам линейных алгебраических уравнений (СЛАУ) и алгебраическим методам декомпозиции, представляющим собой специальные блочные итерационные алгоритмы для решения больших систем с разреженными матрицами.

Сеточные СЛАУ в под областях Ω_q решаются, как правило, также итерационными методами, и мы приходим к проблеме оптимизации трудоемких двойных итерационных процессов. При этом необходимо решать два основных вопроса:

ускорение сходимости внешних и внутренних итераций, а также минимизация коммуникационных потерь при реализации распараллеливания на многопроцессорных вычислительных системах (МВС).

Среди многообразия возникающих здесь подходов можно выделить следующие.

1. Топология декомпозиции областей: в трехмерной краевой задаче под области могут составлять одномерную, двумерную или трехмерную сеть (1D, 2D или 3D, см. рис. 1). Теоретически предпочтительней 3D-декомпозиция с точки зрения как сокращения внешних итераций, так и уменьшения отношения “объем пересылок/объем вычислений”, однако реальная картина может существенно зависеть от возможностей эффективного отображения алгоритмов на архитектуру конкретной МВС.

2. Наличие и величина пересечений смежных под областей. Как и для классического альтернирующего метода Шварца, интуитивно ясно, что с увеличением перехлеста будет ускоряться внешний итерационный процесс, но зато дорожает реализация вспомогательных задач в под областях (в предельном абсурдном случае, когда все под области Ω_q совпадают между собой и с Ω , требуется всего одна итерация, но вспомогательная задача совпадает с исходной). По-видимому, получение строгих теоретических оценок оптимального размера пересечения $\Delta_{q,q'}$ является в общем случае проблематичным, а для выбора практических рекомендаций для конкретных ситуаций свое слово должен сказать эксперимент.

3. Тип граничных условий на внутренних границах $\Gamma_{q,q'}$, через которые идет обмен информацией между под областями, т.е. фактически задание значений коэффициентов α_q и β_q в (3). Вполне возможно, что их оптимальный выбор связан с предыдущей задачей о перехлесте $\Delta_{q,q'}$.

4. Очередность пересчета под областей: последовательная или одновременная. Получаемые блочные

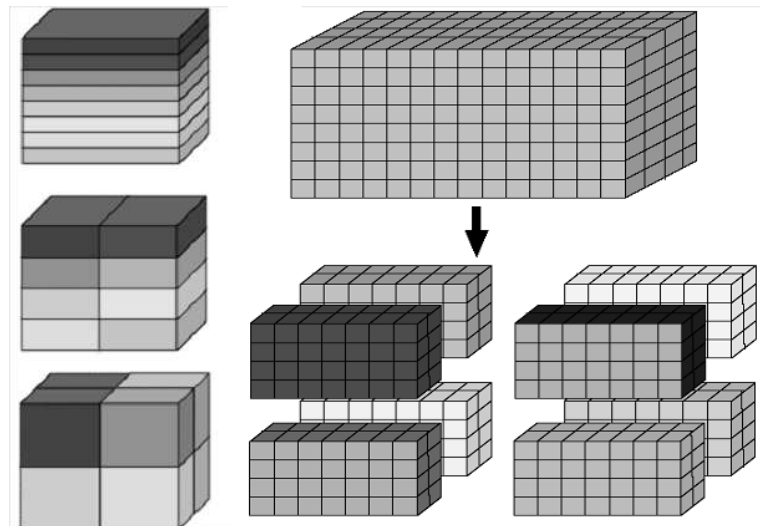


Рис. 1. Схемы 1D-, 2D- и 3D-декомпозиции области

алгоритмы оказываются типа Зейделя или Якоби и называются также мультипликативными или аддитивными соответственно. Очевидные здесь альтернативные предпочтения (первое — с точки зрения ускорения итераций, а второе — с точки зрения эффективности распараллеливания) могут быть дополнены какими-либо компромиссными вариантами типа шахматной упорядоченности, или черно-белой раскраски подобластей.

5. Организация итерационных процессов. Наиболее эффективные современные подходы основаны на предобусловленных методах в подпространствах Крылова, реализующих вариационные и/или ортогональные принципы [16, 17]. Сама декомпозиция области играет роль предобусловливателя, а оптимальный внешний итерационный процесс — это один из крыловских методов. Важный момент заключается в построении двойного итерационного алгоритма. Естественно, в каждой из подобластей целесообразно применять также крыловские методы, причем на первых внешних итерациях бессмысленно решать вспомогательные задачи в подобластях с высокой точностью. В таком случае приходим к динамическому внешнему итерационному процессу с переменными предобусловливателями.

Отдельный важный методологический и практический вопрос — это выбор количества подобластей и соответствующего числа процессоров. Рассмотрим для простоты трехмерную сеточную расчетную область-параллелепипед с числом узлов $M_x M_y M_z$ параллелепипедаидальной сетки, которая (область) разбита на $P = P_x \times P_y \times P_z$ подобластей той же конфигурации. Один из принципиальных вопросов — масштабируемость параллельного алгоритма, т.е. допускает ли он приемлемую величину ускорения вычислений с ростом числа процессоров, например при M_x, M_y и $M_z > 10^3$ и $P_x \rightarrow M_x, P_y \rightarrow M_y, P_z \rightarrow M_z$. Ответ здесь зависит, очевидно, не только от алгоритма, но и от архитектуры МВС. Зачастую в реальных условиях вопрос заключается в тривиальной доступности нужного объема компьютерных ресурсов, но с предстоящим появлением экзафлопных вычислителей [18] проблема массивного параллелизма перейдет и в практическую плоскость.

Настоящая статья посвящена экспериментальному исследованию многопараметрической одномерной декомпозиции трехмерных краевых задач для уравнения Пуассона. В разделе 2 мы приводим алгебраическое представление рассматриваемых методов. Раздел 3 посвящен вопросам оценки эффективности их распараллеливания, а в последнем разделе обсуждаются результаты методических экспериментов на представительной серии тестовых задач с различными шагами сеток и варьируемыми алгоритмическими параметрами.

2. Алгебраическое представление методов декомпозиции. Рассмотрим для простоты первую краевую задачу для уравнения Пуассона в единичном кубе $\Omega = [0 \times 1]^3$:

$$-\Delta u = f, \quad u|_{\Gamma} = g, \tag{5}$$

которая на кубической сетке аппроксимируется семиточечной СЛАУ порядка M^3 при $i, j, k = 1, \dots, M$:

$$(Au^h)_{i,j,k} = 6u_{i,j,k}^h - u_{i-1,j,k}^h - u_{i,j-1,k}^h - u_{i,j,k-1}^h - u_{i+1,j,k}^h - u_{i,j+1,k}^h - u_{i,j,k+1}^h = f_{i,j,k}^h. \tag{6}$$

Здесь предполагается, что в левой части уравнений члены со значениями индексов 0 или M , известные из граничного условия, перенесены в правую часть. Очевидно, что такая алгебраическая система имеет единственное решение при любом векторе $f^h = \{f_{i,j,k}^h\}$. Далее верхний индекс “ h ” будем для краткости опускать. Разобьем расчетную сеточную область $\Omega^h = \{i, j, k\}$ вдоль оси z на P одинаковых подобластей, как это указано на рис. 2. Здесь Δ_q означает общую часть подобластей Ω_{q-1} и Ω_q , причем предполагается, что границы $\Gamma_{q,q\pm 1}$ проходят посередине между сеточными узлами. В каждой из сеточных подобластей Ω_q^h обозначим через k_1^q и k_N^q номера первой и последней z -координатных сеточных плоскостей, а через N и m — количество таких плоскостей в Ω_q и Δ_q , так что для всех q выполняются соотношения

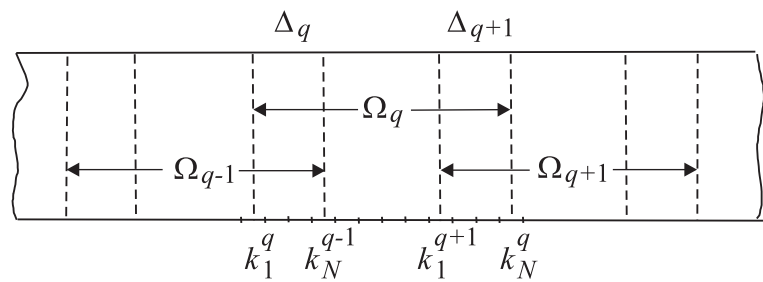


Рис. 2. Схема одномерной декомпозиции с пересечением

$N = k_N^q - k_1^q + 1, \quad m = k_N^q - k_1^{q+1} + 1, \quad M = PN - (P - 1)m.$ (7)

Систему (6) с учетом интерфейсных условий сопряжения запишем в блочном виде

$$-A_{q,q-1}u_{q-1} + A_{q,q}u_q - A_{q,q+1}u_{q+1} = f_q, \quad q = 1, \dots, P, \quad A_{1,0} = A_{P,P+1} = 0, \tag{8}$$

где u_q и f_q — векторы, а $A_{q,q}$ и $A_{q,q\pm 1} = A_{q\pm 1,q}^t$ — квадратные матрицы порядка M^2N , равного числу узлов в Ω_q .

Оставляя в левой части уравнения (8) только “диагональный” член с u_q , блочный метод Якоби можно переписать в форме [19]

$$A_{q,q}u_q^n = \bar{f}_q^{n-1} \equiv f_q + \hat{f}_q^{n-1} + \check{f}_q^{n-1}, \quad \hat{f}_q^{n-1} = A_{q,q-1}u_{q-1}^{n-1}, \quad \check{f}_q^{n-1} = A_{q,q+1}u_{q+1}^{n-1}. \quad (9)$$

Здесь векторы имеют блочный вид $u_q = (u_{k_1^q}, \dots, u_{k_N^q})^t$, где, например, $u_{k_1^q} = \{u_{i,j,k_1^q}; i, j = 1, \dots, M\}$ — подвектор размерности M^2 , соответствующий сеточной плоскости с координатой $z = z_{k_1^q}$. Для более детального представления матриц в (9) перепишем метод в блочно-компонентной форме

$$(A_{q,q}u_q^n)_k = \begin{cases} (C - \theta I)u_{k_1^q}^n - u_{k_1^{q+1}}^n = f_{k_1^q} + v_{q-1}^{n-1}, & v_{q-1}^{n-1} = u_{k_1^{q-1}}^{n-1} - \theta u_{k_1^q}^{n-1}, & k = k_1^q, \\ (C - \theta I)u_{k_N^q}^n - u_{k_N^{q-1}}^n = f_{k_N^q} + w_{q+1}^{n-1}, & w_{q+1}^{n-1} = u_{k_N^{q+1}}^{n-1} - \theta u_{k_N^q}^{n-1}, & k = k_N^q, \\ -u_{k-1}^n + Cu_k^n - u_{k+1}^n = f_k, & & k = k_1^q + 1, \dots, k_N^q - 1, \end{cases} \quad (10)$$

где C — матрица порядка M^2 , а выражение Cu_k соответствует первым четырем членам в левой части уравнения (6), т.е. не содержащим индексы $k \pm 1$.

Зависящие от u_k^{n-1} величины в (10) фактически берутся как подвекторы из $\Omega_{q\pm 1}$ и реализуют пограничный интерфейс между подобластями, при этом нижние индексы надо поменять формально в соответствии с (7): $k_1^q = k_N^{q-1} - m + 1$, $k_N^q = k_1^{q+1} + m - 1$. Параметр θ может быть выражен через коэффициенты α_q и β_q из условий сопряжения в (3), которые для простоты считаем одинаковыми во всех подобластях Ω_q (в частности, $\theta = 0$ и $\theta = 1$ соответствуют условиям Дирихле и Неймана). Векторы $A_{q,q\pm 1}u_{q\pm 1}^{n-1}$ содержат ненулевые подвекторы, но только соответствующие сеточным плоскостям в окрестности внутренних границ $\Gamma_{q,q\pm 1}$. После вычисления вектора u_q^n (из решения краевой подзадачи в Ω_q) в дальнейших итерациях фактически будут участвовать только значения правых частей граничных условий из (2), (3), определяемых алгебраически как $A_{q\pm 1,q}u_q^n$ и используемых в подобластях $\Omega_{q\pm 1}$.

Отсюда нетрудно видеть, что итерационный процесс (9) для векторов $u_q \in \mathcal{R}^{M^2N}$ можно переформулировать в терминах подвекторов меньшей размерности $v_q, w_q \in \mathcal{R}^{M^2}$, соответствующих внутренним граничным интерфейсам на $\Gamma_{q,q\pm 1}$. Для этого можно ввести обозначения $v_q = C_{q,q-1}u_q, w_q = C_{q,q+1}u_q, A_{q,q\pm 1} = Q_{q,q\pm 1}C_{q,q\pm 1}$, где $C_{q,q\pm 1} \in \mathcal{R}^{M^2NM^2}, Q_{q,q\pm 1} \in \mathcal{R}^{M^2M^2N}$ — прямоугольные матрицы “сужения” и “продолжения” соответствующих векторных подпространств.

Умножая обе части равенства (9) один раз на $C_{q,q-1}$, а второй — на $C_{q,q+1}$, получаем соотношения вида

$$\begin{aligned} v_q^n &= \hat{B}_{q,q-1}w_{q-1}^{n-1} + \hat{B}_{q,q+1}v_{q+1}^{n-1} + \hat{g}_q, & q = 2, \dots, P, \\ w_q^n &= \check{B}_{q,q-1}w_{q-1}^{n-1} + \check{B}_{q,q+1}v_{q+1}^{n-1} + \check{g}_q, & q = 1, \dots, P-1, \quad \hat{B}_{1,0} = \hat{B}_{P,P+1} = 0, \end{aligned} \quad (11)$$

где введены следующие обозначения: $\hat{g}_q = C_{q,q-1}A_{q,q}^{-1}f_q, \check{g}_q = C_{q,q+1}A_{q,q}^{-1}f_q, \hat{B}_{q,q\pm 1} = C_{q,q-1}A_{q,q}^{-1}Q_{q,q\pm 1}, \check{B}_{q,q\pm 1} = C_{q,q+1}A_{q,q}^{-1}Q_{q,q\pm 1}$. Мы не будем вдаваться в детали представления матриц $\hat{B}_{q,q\pm 1}$ и $\check{B}_{q,q\pm 1}$ и отметим, что они в алгебраическом смысле являются матричными аналогами операторов Пуанкаре–Стеклова и реализуют итерационные интерфейсы только между внутренними граничными условиями на $\Gamma_{q,q\pm 1}$. При этом, естественно, на каждом шаге решаются задачи в подобластях Ω_q , но как вспомогательные (реально вычисляются частичные решения, т.е. векторы правых частей имеют “очень мало” ненулевых компонент, а в искомым векторах также представляют интерес только некоторые компоненты). После сходимости с достаточной точностью итерационных приближений $v_q^n \rightarrow v_q, w_q^n \rightarrow w_q$ во всех подобластях могут быть восстановлены “полные” векторы u_q .

Обозначая через $s = (w_1, v_2, \dots, v_{P-1}, w_{P-1}, v_P)^t$ и $g = (\check{g}_1, \hat{g}_2, \check{g}_2, \dots, \hat{g}_{P-1}, \check{g}_{P-1}, \hat{g}_P)^t$ векторы порядка $2M^2(P-1)$, итерационный процесс (11) можно записать в виде

$$s^n = Ts^{n-1} + g, \quad n = 1, 2, \dots, \quad (12)$$

где матрица перехода T является блочно-двудиagonalной и имеет нулевую главную блочную диагональ и составленные из блоков $\hat{B}_{q,q\pm 1}, \check{B}_{q,q\pm 1}$ соседние к ней диагонали.

Если последовательные приближения сходятся, т.е. $s^n \rightarrow s$ при $n \rightarrow \infty$, то мы приходим к решению предобусловленной системы уравнений

$$\bar{A}s \equiv (I - T)s = g. \quad (13)$$

Для ускорения итераций (12) естественно применить какой-либо из методов в подпространствах Крылова. Например, при использовании алгоритмов сопряженных градиентов (СГ) или сопряженных невязок (СН) получаем следующие формулы [17]:

$$\begin{aligned} r^0 &= g - \bar{A}s^0 = \hat{s}^1 - s^0, \quad \hat{s}^1 = Ts^0 + g, \quad p^0 = r^0, \quad s^{n+1} = s^n + \alpha_n^{(\nu)} p^n, \\ \alpha_n^{(\nu)} &= \frac{\rho_n^{(\nu)}}{\delta_n^{(\nu)}}, \quad \rho_n^{(\nu)} = (\bar{A}^\nu r^n, r^n), \quad \delta_n^{(\nu)} = (\bar{A} p^n, \bar{A}^{(\nu)} p^n), \\ r^{n+1} &= r^n - \alpha_n^{(\nu)} \bar{A} p^n, \quad p^{n+1} = r^{n+1} + \beta_n^{(\nu)} p^n, \quad \beta_n^{(\nu)} = \frac{\rho_{n+1}^{(\nu)}}{\rho_n^{(\nu)}}. \end{aligned} \quad (14)$$

Здесь $\nu = 0$ и $\nu = 1$ соответствуют алгоритмам СГ и СН, а индексы “ ν ” у векторов r^n , s^n и p^n для краткости опущены. Отметим, что реализация каждой итерации (14) связана с одним умножением матрицы T на вектор (в том числе при $\nu = 1$, поскольку $\bar{A} p^{n+1} = \bar{A} r^{n+1} + \beta_n^{(\nu)} \bar{A} p^n$). В силу структуры матрицы T это связано с вычислением частичных решений в подобластях Ω_q (параллельно для всех $q = 1, \dots, P$). Однако заметим, что скалярные произведения в (14) определяются для векторов размерности $2(P-1)M^2$, равной общему количеству узлов во всех интерфейсных сечениях Γ_q . Традиционное условие окончания итераций имеет вид $\frac{(r^n, r^n)}{(g, g)} \leq \varepsilon_e^2 \ll 1$ при заданном значении ε_e .

Решение вспомогательных задач в подобластях Ω_q целесообразно (при достаточно большом значении M^2N) также проводить с помощью предобусловленного метода в подпространствах Крылова. В качестве предобусловливателя естественным образом выбирается какой-то из алгоритмов неполной факторизации. Если через r_{in}^{nq} обозначить величину невязки на nq -м шаге внутреннего итерационного процесса вида (14), то критерием его окончания можно выбрать $\frac{(r_{\text{in}}^{nq}, r_{\text{in}}^{nq})}{(f_q^n, f_q^n)} \leq (\varepsilon_{\text{in}}^{(n)})^2$, где величина $\varepsilon_{\text{in}}^{(n)}$ может выбираться динамически в зависимости от номера внешней итерации n .

Отметим, что фактически для внешнего итерационного процесса используется динамическое предобусловливание (в [15] используется термин *flexible*, т.е. гибкое). Строго говоря, в (13) матрица зависит от n , что может интерпретироваться как приближенное умножение на \bar{A} в формулах (14).

3. Особенности распараллеливания алгоритмов. В соответствии с определением блочной структуры матрицы T и векторов, параллельная реализация крыловского метода (14) (при выбранном начальном векторе u^0) осуществляется следующим образом. Будем предполагать, что вычисления проводятся на МВС с распределенной памятью, имеющей P процессоров, каждый из которых соответствует “своей” подобласти Ω_q .

1. До проведения итераций вычисляется вектор начальной невязки r^0 (и равный ему начальный направляющий вектор $p^0 = r^0$). Для этого в каждой подобласти Ω_q сначала находятся подвекторы начальных граничных условий, представляющие собой блочные компоненты s^0 :

$$\begin{aligned} v_q^0 &= C_{q,q-1} u_q^0 = \{u_{i,j,k_N^{q-1}+1}^0 - \theta u_{i,j,k_N^{q-1}}^0; i, j = 1, \dots, M\}, \\ w_q^0 &= C_{q,q+1} u_q^0 = \{u_{i,j,k_1^{q+1}-1}^0 - \theta u_{i,j,k_1^{q+1}}^0; i, j = 1, \dots, M\}. \end{aligned} \quad (15)$$

Здесь v_q^0 — правое краевое условие для Ω_{q-1} , а w_q^0 — левое краевое условие для Ω_{q+1} . Затем осуществляются встречные пересылки всех компонент $v_{q,i,j}^0$ и $w_{q,i,j}^0$ данных векторов по следующей схеме:

$$v_q^0, w_q^0 : \Omega_q \rightarrow \Omega_{q\pm 1}; \quad v_{q+1}^0, w_{q-1}^0 : \Omega_{q\pm 1} \rightarrow \Omega_q. \quad (16)$$

Далее формируются и решаются вспомогательные СЛАУ вида (9) в Ω_q :

$$A_{q,q} \hat{u}_q^1 = \bar{f}_q = \begin{cases} f_{k_1^q} + w_{q-1}^0, & k = k_1^q, \\ f_{k_N^q} + v_{q+1}^0, & k = k_N^q, \\ f_k, & k = k_1^q + 1, \dots, k_N^q - 1. \end{cases} \quad (17)$$

По найденным решениям на интерфейсных границах $\Gamma_{q,q\pm 1}$, с помощью формул (15) (при замене u^0 на \hat{u}^1) находятся новые краевые условия \hat{v}_q^1 и \hat{w}_q^1 , составляющие блочные компоненты вектора \hat{s}^1 , из которого и находятся $r^0 = p^0 = \hat{s}^1 - s^0$.

2. Реализация итераций для каждого n состоит из одного умножения матрицы на вектор $\overline{A}p^n$ (или $\overline{A}r^n$), вычисления двух скалярных произведений и поиска линейных комбинаций векторов. Первая из этих операций, вследствие очевидного соотношения

$$t^n \equiv \overline{A}p^n = p^n - q^n, \quad q^n = Tp^n, \tag{18}$$

выполняется в соответствии с формулами (15)–(17): сначала определяются аналоги краевых условий (в (15) надо только величины v_q^0 , w_q^0 и u^0 поменять на \hat{t}_q^n , \check{t}_q^n и p^n), затем в подобластях Ω_q решаются аналогичные (17) подзадачи

$$A_{q,q}\varphi_q^n = \psi_q^n = \begin{cases} \check{t}_{q-1}^n, & k = k_1^q, \\ \hat{t}_{q+1}^n, & k = k_N^q, \\ 0, & k = k_1^q + 1, \dots, k_N^q - 1, \end{cases} \tag{19}$$

где в правых частях отсутствуют не зависящие от n члены f_k , и наконец, по найденным значениям φ_q^n вычисляются блочные компоненты вектора $q^n = (\hat{q}_1^n, \hat{q}_2^n, \check{q}_2^n, \dots, \hat{q}_{P-1}^n, \check{q}_{P-1}^n, \hat{q}_P^n)$. Формально последний этап тоже заключается в нахождении краевых условий по формулам (15), нужно в данном случае только заменить векторы

$$\begin{aligned} \hat{q}_q^n &= \{\varphi_{i,j,k_N^{q-1}+1}^n - \theta\varphi_{i,j,k_N^n}^n; \quad i, j = 1, \dots, M\}, \\ \check{q}_q^n &= \{\varphi_{i,j,k_1^{q+1}-1}^n - \theta\varphi_{i,j,k_1^n}^n; \quad i, j = 1, \dots, M\}. \end{aligned} \tag{20}$$

Итоговые блочные компоненты результата операции (18) имеют вид $\hat{t}_q^n = \hat{p}_q^n - \hat{q}_q^n$, $\check{t}_q^n = \check{p}_q^n - \check{q}_q^n$.

Скалярные произведения векторов в (14) вычисляются путем суммирования по всем узлам интерфейсных сечений Γ_q , т.е., например,

$$\begin{aligned} (t^n, p^n) &= \sum_{i,j=1}^M \left(\sum_{q=2}^p \hat{t}_{q,i,j}^n \hat{p}_{q,i,j}^n + \sum_{q=1}^{p-1} \check{t}_{q,i,j}^n \check{p}_{q,i,j}^n \right) = (\check{t}^n, \check{p}^n)_1 + \sum_{q=2}^{p-1} (t^n, p^n)_q + (\hat{t}^n, \hat{p}^n)_p, \\ (t^n, p^n)_q &= (\hat{t}^n, \hat{p}^n)_q + (\check{t}^n, \check{p}^n)_q, \quad (\hat{t}^n, \hat{p}^n)_q = \sum_{i,j=1}^M \hat{t}_{q,i,j}^n \hat{p}_{q,i,j}^n. \end{aligned} \tag{21}$$

3. После сходимости итерационных приближений v_q^n и w_q^n с достаточной точностью мы легко получаем искомое решение из вспомогательных задач вида (17): в правой части надо только поменять w_{q-1}^0 и v_{q+1}^0 на w_{q-1}^{n-1} и v_{q+1}^{n-1} , в результате чего вместо \hat{u}_q^1 слева будем иметь u_q^n .

Обозначая традиционно через T_P время реализации данного алгоритма на P процессорах, для коэффициентов ускорения и эффективности использования МВС имеем выражения $S_P = \frac{T_1}{T_P}$, $E_P = \frac{S_P}{P}$, где полное время решения задачи складывается из времени выполнения арифметических операций T_P^a и времени обмена данными T_P^c между подобластями Ω_q : $T_P = T_P^a + T_P^c \approx \tau_a V_a + N_a(\tau_0 + \tau_c V_c)$. Здесь τ_a обозначает среднее время исполнения одного арифметического действия, V_a — общий объем выполняемых одним процессором операций (предполагается, что все процессоры работают синхронно), N_a — количество информационных обменов, τ_0 и τ_c — время задержки (настройки) одного обмена и время передачи одного числа, а V_c — объем передаваемого за один раз массива данных, причем имеют место неравенства $\tau_0 \gg \tau_c \gg \tau_a$.

Поскольку время реализации этапов 1 и 3 сравнимо со временем выполнения одной итерации на этапе 2, то для оценки величин S_P и E_P при $n \gg 1$ достаточно проанализировать качество распараллеливания одного итерационного шага.

Решение линейной системы (19) в каждой q -й подобласти занимает время $T_a^{(1)}$, пропорциональное числу узлов M^2N в Ω_q и количеству внутренних итераций $n_{in}(\varepsilon_{in})$, которое для современных предобусловленных крыловских методов составляет $N^\gamma |\ln \varepsilon_{in}|$, $\gamma \geq 1/2$:

$$T_a^{(1)} = C_1 M^2 N^{\gamma+1} |\ln \varepsilon_{in}| \tau_a, \tag{22}$$

где C_1 — количество арифметических действий на выполнение одной итерации, которое для разных предобусловливателей можно оценить величиной 20 или 30.

Время вычисления “краевых условий” \hat{q}_q и \check{q}_q из (20) малó по сравнению с (22): $T_a^{(2)} \leq 6M^2\tau_a$. Выполнение векторных операций на одной внешней итерации (14) в Ω_q (их всего пять) занимает время

$T_a^{(3)} \approx 20M^2 + C_2$, где последнее слагаемое относится к “довычислению” компонент векторных произведений (21), т.е. суммированию частичных произведений вида $(t^n, p^n)_q$, расположенных в разных процессорах. Если все процессоры передают их одновременно в головной процессор, где они складываются, а результат (опять одновременно) рассылается по всем процессорам, то $C_2 \approx P$. Времена пересылок “краевых условий” между процессорами оцениваются величиной $T_P^c \leq C_3(\tau_0 + 2\tau_c M^2 N)$.

Здесь значение C_3 не зависит от M, N и P в идеальном случае, когда все пересылки типа (16) между соседними процессорами происходят одновременно.

Реализация арифметических действий в рассматриваемом алгоритме на одном процессоре занимает в P раз больше времени, и при $P \gg 1$ мы имеем практически линейное ускорение, т.е. $S_P = \frac{T_P^a P}{T_P^a + T_P^c} \approx P$, $E_P \approx 1$.

Необходимо иметь в виду, что общий объем вычислений в двойном итерационном процессе может оказаться больше, чем при использовании “обычного” одноуровневого предобусловленного крыловского метода для решения полной исходной СЛАУ. Это связано, во-первых, с тем, что реально $n_{in} n_e(\varepsilon) > n(\varepsilon)$, (n_{in} — среднее число внутренних итераций, а $n(\varepsilon)$ — количество итераций одноуровневого алгоритма), а во-вторых — с неизбежными “накладными расходами” метода декомпозиции.

Однако принцип “разделяй и властвуй” является неизбежным при решении больших задач (значение M исчисляется тысячами или десятками тысяч), когда главная проблема заключается в интеграции вычислительных ресурсов, в силу тривиальной нехватки памяти одного процессора.

Таблица 1
Характеристики двойных итераций без внешнего ускорения
для $P = 2$ и $\Delta = 2, 4$

θ/M	9		19		29		49		99	
0	8	4	15	8	21	11	34	18	64	33
	100	73	338	214	659	399	1631	957	5542	3094
		0		5.07-02		0.3		3.54		137.78
0.25	6	4	11	7	16	10	26	15	49	29
	82	75	264	197	534	377	1312	838	4430	2803
		0		5.07-02		0.28		3.13		124.07
0.5	4	3	8	5	11	8	18	12	33	23
	77	69	210	159	402	321	976	710	3164	2325
		0		3.9-02		0.25		2.66		103.7
0.75	4	3	4	3	5	5	9	7	17	14
	84	71	151	140	251	252	579	477	1844	1583
		0		3.51-02		0.2		1.84		72.37
1.00	∞		∞		∞		∞		∞	

4. Примеры численных экспериментов. Мы проиллюстрируем скорость сходимости и эффективность распараллеливания рассмотренных алгоритмов в применении к решению симметричных систем сеточных семиточечных уравнений (6), аппроксимирующих задачу Дирихле для уравнения Пуассона (5) в единичном кубе $\Omega = [0, 1]^3$ на различных кубических сетках с шагами $h = \frac{1}{M+1}$. Функции f и g в (5) выбирались из условия, что точное решение $u(x, y, z)$ равно единице. В качестве начального приближения выбиралась функция $u^0(x, y, z) = 0$, а критерий окончания внешних и внутренних итераций полагался одинаковым и равным $\varepsilon_e = \varepsilon_{in} = 10^{-3}$, за исключением оговоренных ниже случаев. Все вычисления проводились со стандартной двойной точностью, а разделение областей было выбрано одномерным по направлению оси z в соответствии со схемой на рис. 2. Расчеты выполнялись на кластере с процессорами Itanium-2, а внутренние итерации для решения задач в подобластях для простоты реализовывались с помощью “явного” метода сопряженных градиентов, т.е. без предобусловливания.

В табл. 1 приведены характеристики двойного итерационного процесса при декомпозиции расчетной области на две подобласти с величинами перехлеста $\Delta = 2, 4$, которые измеряем в количестве шагов сетки. При этом общее количество шагов сетки бралось равным $M = 9, 19, 29, 49, 99$, а значение итерационного параметра в (10), характеризующее тип условия внутреннего сопряжения, выбиралось $\theta = 0, 0.25, 0.5, 0.75, 1.0$.

В каждой клетке таблицы приведены пять чисел: количество потребовавшихся внешних итераций

Таблица 2
Характеристики двойных итераций с внешним ускорением
методом СГ для $P = 2$ и $\Delta = 2, 4$

θ/M	9	19	29	49	99
0	3 2 57 48 0	5 3 176 114 3.9-02	6 4 292 214 0.17	7 5 511 400 1.55	10 7 1197 920 43.83
0.25	3 2 60 50 0	4 3 145 118 3.12-02	5 4 259 218 0.17	6 5 463 418 1.6	9 7 1157 990 46.52
0.5	2 2 46 52 0	4 3 150 129 3.12-02	4 3 220 172 0.14	6 4 495 358 1.41	8 6 1102 990 42.88
0.75	2 2 50 54 0	3 2 126 102 2.34-02	3 3 175 198 0.16	4 4 372 376 1.46	6 5 961 861 40.73
1.00	∞	∞	∞	∞	∞

Таблица 3
Характеристики итерационных процессов для $P = 3$ и
 $\Delta = 2, 4$ без внешних сопряженных градиентов

θ/M	9	19	29	49	99
0	9 4 98 72 —	17 8 290 183 4.68-02	24 12 560 334 0.24	39 20 1390 780 2.25	73 38 4572 2573 95.88
0.25	6 4 77 71 —	13 7 241 172 4.68-02	18 11 449 322 0.23	30 18 1129 729 2.11	56 33 3631 2325 87.10
0.5	4 3 70 61 —	8 6 176 159 4.68-02	12 9 369 289 0.21	20 14 819 607 1.82	38 26 2653 1931 73.39
0.75	4 3 73 64 —	4 3 148 122 3.12-02	6 5 262 222 0.164	10 8 498 423 1.32	20 16 1692 1354 52.74
1.00	8 4 61 43 —	18 6 219 110 34.68-02	∞	∞	∞

(верхние два числа), общее число внутренних итераций (вторая строка) и время счета в секундах (для вариантов, соответствующих значениям $\Delta = 4$). В данном случае внешние итерации проводились без применения СГ и представляли собой фактически просто блочный алгоритм Якоби.

В табл. 2 приводятся аналогичные данные с тем отличием, что внешние итерации выполнялись по формулам (14) метода сопряженных градиентов.

Таблицы 3 и 4 содержат те же характеристики итерационных процессов, но при декомпозиции расчетной области на три одинаковые подобласти с теми же величинами пересечений Δ в два и четыре шага сетки.

Анализ приведенных результатов позволяет сделать следующие выводы:

— с увеличением отношения M/Δ (т.е. с относительным уменьшением размера перехлеста подобластей) количество внешних итераций и общее число внутренних итераций значительно увеличиваются, причем этот рост существенно слабее при использовании СГ;

— при изменении параметра θ от 0 до 1 характерна следующая тенденция: при использовании СГ и без него количество внешних и суммарное число внутренних итераций уменьшаются для всех M , но при $\Delta = 1$ обнаруживается, как правило, расходимость процесса (в таблицах обозначаемая символом “ ∞ ”),

Таблица 4
Характеристики итерационных процессов с внешними СГ
($P = 3, \Delta = 2, 4$)

θ/M	9		19		29		49		99	
0	6	4	7	5	8	6	10	8	14	10
	94	80	199	162	307	267	578	512	328	1061
	—		3.9-02		0.19		1.51		43.56	
0.25	5	4	6	5	8	6	9	7	12	9
	82	80	181	166	324	274	545	469	1215	1010
	—		4.68-02		0.19		1.40		41.05	
0.5	5	5	5	5	6	5	7	7	10	8
	85	88	162	170	271	243	465	487	1120	964
	—		4.29-02		0.17		1.44		38.92	
0.75	6	5	5	6	5	4	5	5	7	7
	95	94	174	205	255	224	393	403	914	958
	—		5.07-02		0.16		1.21		38.49	
1.00	∞		∞		∞		∞		∞	

в результате чего обнаруживается наличие минимума (в том числе и по времени счета) для значений $\theta = 0.5 \div 0.75$;

— применение ускорения внешних итераций дает значительный эффект, особенно в “плохих” случаях, т.е. при увеличении общего числа узлов и количества подобластей (процессоров); однако при этом наблюдается парадоксальное на первый взгляд уменьшение коэффициента ускорения S_P (в сравнении без применения СГ); тем не менее, этот факт еще раз подтверждает то известное явление, что менее эффективные алгоритмы легче и эффективнее распараллеливаются.

Очевидно, что при увеличении количества подобластей и соответствующего числа процессоров P количество внешних итераций (для фиксированного общего числа узлов в расчетной области) будет расти. Хотя трудоемкость решения каждой вспомогательной задачи в подобласти уменьшается, общий объем вычислений в методе декомпозиции будет увеличиваться по сравнению с решением исходной задачи на одном процессоре при достаточно большом объеме его оперативной памяти.

Безальтернативность метода декомпозиции заключается в его распараллеливании и конечном росте производительности МВС. Однако вопросы оптимизации таких вычислительных процессов являются во многом открытыми и требуют серьезных исследований. Например, теоретически является естественным переход к двумерной и трехмерной декомпозиции, однако реальная эффективность такого подхода будет зависеть от качества отображения получаемых алгоритмов на архитектуру вычислительной системы (в частности, от реализации коммуникаций).

СПИСОК ЛИТЕРАТУРЫ

1. Смелов В.В., Журавлева Т.Б. Принципы итерирования по подобластям с эллиптическим уравнением. Препринт № 14. М.: ОВМ РАН, 1981.
2. Лебедев В.И., Агошков В.И. Операторы Пуанкаре–Стеклова и их приложения в анализе. М.: ОВМ АН СССР, 1983.
3. Marchuk G.I., Kuznetsov Yu.A., Matsokin A.M. Fictitious domain and domain decomposition method // Sov. J. Num. Anal. Math. Mod. 1986. 1, № 1. 3–35.
4. Quarteroni A., Valli A. Domain decomposition methods for partial differential equations. Oxford: Clarendon Press, 1999.
5. Bramble J.H., Pasciak J.E., Wang J., Xu J. Convergence estimates for product iterative methods with applications to domain decomposition // Math. Comp. 1991. 57, N 195. 1–21.
6. Dryja M., Widlund O.B. Domain decomposition algorithms with small overlap // SIAM J. Sci. Comp. 1994. 15, N 3. 604–620.
7. Ильин В.П. О стратегиях распараллеливания в математическом моделировании // Программирование. 1999. № 1. 41–46.
8. Ильин В.П., Свешников В.М. Оценки эффективности распараллеливания алгоритмов декомпозиции областей // Автометрия. 2002. № 1. 31–41.

9. Ильин В.П. Параллельные алгоритмы для больших прикладных задач: проблемы и технологии // Автометрия. 2007. **43**, № 2. 3–21.
10. Nepomnyashchikh S.V. Domain decomposition methods // Radon Series Comp. Appl. Math. 2007. N 1. 89–159.
11. Василевский Ю.В., Ольшанский М.А. Краткий курс по многосеточным методам и методам декомпозиции области. М.: МАКС Пресс, 2007.
12. Loisel S., Szyld D.B. On the geometric convergence of optimized Schwarz methods with application to elliptic problems // Numer. Math. 2010. **114**. 697–728.
13. Свешников В.М. Построение прямых и итерационных методов декомпозиции // Сиб. журн. индустриальной матем. 2009. **12**, № 3. 99–109.
14. Ильин В.П. Проблемы высокопроизводительных технологий решения больших разреженных СЛАУ // Вычислительные методы и программирование. 2009. **10**, № 1. 130–136.
15. <http://www.ddm.org/>
16. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: ИВМиМГ СО РАН, 2001.
17. Ильин В.П. Методы и технологии конечных элементов. Новосибирск: ИВМиМГ СО РАН, 2007.
18. Ильин В.П. Об экзатрешах математического моделирования // CAD/CAM/CAE Observer. 2010. N 2 (54). 85–92.
19. Saad Y. Iterative methods for sparse linear systems. New York: PWS Publishing Co., 1996.

Поступила в редакцию
02.03.2011
