

# Параллельные методы декомпозиции в пространствах следов

В.П. Ильин, Д.В. Кныш

Институт вычислительной математики и математической геофизики СО РАН

## 1. Введение

Альтернирующий метод Шварца, возникший изначально как средство теоретического исследования сложных краевых задач математической физики, впоследствии породил большое семейство алгоритмов декомпозиции областей, ставшим основным орудием распараллеливания при решении многомерных дифференциальных уравнений. Данным проблемам посвящено огромное количество работ российских и зарубежных авторов, а также имеется специальная регулярная международная конференция, активно работающая в течение многих лет, см. [1] – [15] и цитируемую там литературу.

Общая идея методов декомпозиции заключается в следующем. Пусть в заданной расчетной области  $\Omega$  с границей  $\Gamma$  требуется решить линейную дифференциальную краевую задачу

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

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

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

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

причем на внешней части границы  $\Gamma_q$ , если она существует, ставится исходное краевое условие из (1), а  $f_q(\vec{r})$  при  $r \in \Omega_q$ . Типичное условие сопряжения на внутренней границе записывается в виде краевого условия 3-го рода, или Робена:

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

где  $\alpha_q \cdot \beta_q \geq 0$ ,  $\frac{\partial u_q}{\partial n_q} \Big|_{\Gamma_{q,q'}}$  есть одностороннее значение (внутреннее по отношению к  $\Omega_q$ ) внешней нормальной производной к границе  $\Gamma_q$ , а случаи  $\beta_q = 0$  или  $\alpha_q = 0$  соответственно означают условия Дирихле или Ньютона. Сами коэффициенты  $\alpha_q, \beta_q$  являются функциями координат  $\vec{r}$  и обычно выбираются кусочно-постоянными, одинаковыми для фиксированной пары  $q, q'$ .

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

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

причем условия сопряжения имеют записанный вид для всех  $q' = \omega_q$ . Если  $u_q^n \rightarrow u_q$  при  $n \rightarrow \infty$ , что обеспечивается выбором соответствующих  $\Gamma_{q,q'}$ ,  $\alpha_q$  и  $\beta_q$ , то мы получаем параллельный алгоритм, поскольку каждая из задач (4) для фиксированных  $n, q$  может решаться одновременно при наличии  $P$  процессоров. Данный итерационный процесс представляет собой метод одновременных смещений, или Якоби. Отметим, что если в правой части условия сопряжения (4) брать  $u_{q'}^n$  вместо  $u_{q'}^{n-1}$  для  $q' < q$ , для уже рассчитанных на данной итерации подобластей  $\Omega_{q'}$ , 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. построении сетки и аппроксимации методами конечных разностей, конечных объемов или конечных элементов (МКР, МКО или МКЭ, см. [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.

При этом сеточные СЛАУ в подобластях  $\Omega_q$  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.

- Топология декомпозиции областей: в трехмерной краевой задаче подобласти 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 сильно зависеть от возможностей эффективного отображения алгоритмов на архитектуру конкретной МВС.
- Наличие и величина пересечений смежных подобластей. 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 подобластях (в предельном абсурдном случае, когда все подобласти  $\Omega_q$  совпадают между собой и с  $\Omega$ , 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, и для выбора практических рекомендаций для конкретных ситуаций свое слово должен сказать эксперимент.
- Тип граничных условий на внутренних границах  $\Gamma_{q,q'}$ , через которое идет обмен информацией между подобластями, т.е. фактически значения коэффициентов  $\alpha_q, \beta_q$  в (3). Вполне возможно, что их оптимальный выбор связан с предыдущей задачей о перехлесте  $\Delta_{q,q'}$ .
- Очередность пересчета подобластей: последовательная или одновременная. Получаемые блочные алгоритмы оказываются типа Зейделя или Якоби и называются также

мультипликативными или аддитивными соответственно. Очевидные здесь альтернативные предпочтения: первый – с точки зрения ускорения итераций, а второй – из эффективности распараллеливания, – могут быть дополнены какими-либо компромиссными вариантами типа шахматной упорядоченностью, или черно-белой раскраской под областей.

- Организация итерационных процессов. Наиболее эффективные современные подходы основаны на предобусловленных методах в подпространствах Крылова, реализующих вариационные и/или ортогональные принципы, см. [16], [17]. Сама декомпозиция области играет роль предобуславливателя, а оптимальный внешний итерационный процесс – это какой-то 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 предобуславливателями.

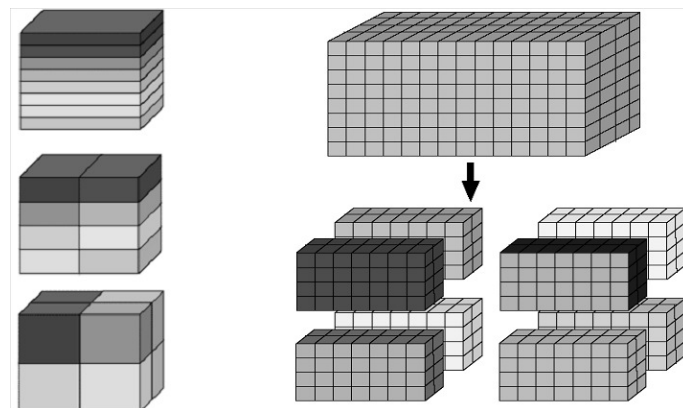


Рис. 1. Схемы 1D, 2D и 3D декомпозиции области

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  $M_x M_y M_z$  параллелепипедоидальной сетки, которая (область) разбита на число  $P = P_x \cdot P_y \cdot P_z$  подобластей такой же конфигурации. Один из принципиальных вопросов – масштабируемость параллельного алгоритма, т.е. допускает ли он приемлемую величину ускорения вычислений с ростом числа процессоров? Например, при  $M_x, M_y, M_z > 10^3$  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 эксафлопных вычислителей [18] проблема массивного параллелизма перейдет и в практическую плоскость.

Данная работа посвящена экспериментальному исследованию многопараметрической одномерной декомпозиции трехмерных краевых задач для уравнения Пуассона. В п. 2 мы приводим алгебраическое представление рассматриваемых методов. Пункт 3 посвящен вопросам оценок 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  $\Omega = [0 \times 1]^3$ :

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

которая на кубической сетке аппроксимируется семиточечной СЛАУ порядка  $M^3$ :

$$\begin{aligned} (Au^h)_{i,j,k} &= 6u_{i,j,k}^h - u_{i-1,j,k}^h - u_{i,j-1,k}^h - u_{i+1,j,k}^h - u_{i,j,k-1}^h - u_{i,j,k+1}^h = \\ &= f_{i,j,k}^h; \quad i, j, k = 1, \dots, M. \end{aligned} \quad (6)$$

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.

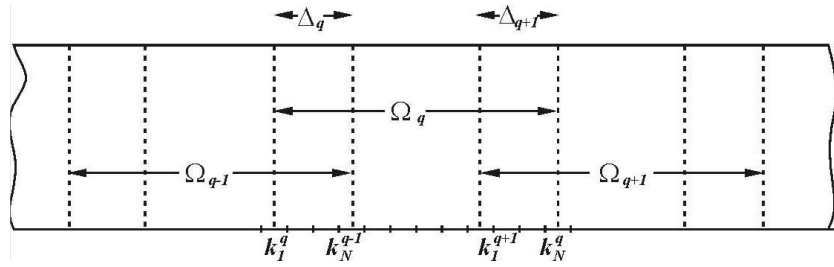


Рис. 2. Схема одномерной декомпозиции с пересечением

Здесь  $\Delta_q$  означает общую часть подобластей  $\Omega_{q-1}$  и  $\Omega_q$  причем предполагается, что границы  $\Gamma_{q,q\pm 1}$  проходят посередине между сеточными узлами. В каждой из сеточных подобластей  $\Omega_q^h$  обозначим через  $k_1^q$  и  $k_N^q$  номера первой и последней  $z$ -х координатных сеточных плоскостей, а через  $N$  и  $m$  – количество таких плоскостей в  $\Omega_q$  и  $\Delta_q$ , так что для всех  $q$  выполняются соотношения

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

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

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

где  $u_q$  и  $f_q$  суть 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  $\Omega_q$ .

Оставляя в левой части уравнения (8) только “диагональный” член с  $u_q$ , блочный метод Якоби (4) можно переписать в форме

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

Участвующие здесь векторы имеют блочный вид  $u_q = (u_{k_1^q}, \dots, u_{k_N^q})^t$ , где  $u_{k_1^q} = \{u_{i,j,k_1^q}; i, j = 1, \dots, M\}$ , например, есть подвектор размерности  $M^2$ , соответствующий сеточной плоскости

с координатой  $z = z_{k_1^q}$ . Для более детального представления матриц в (9) перепишем его 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  $\Omega_{q \pm 1}$ , реализуя 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  $\theta$  может быть выражен через коэффициенты  $\alpha_q$  и  $\beta_q$  из условий сопряжения в (3), которые для простоты считаем одинаковыми во всех подобластях  $\Omega_q$  (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, только соответствующие сеточным плоскостям в окрестности внутренних границ  $\Gamma_{q,q \pm 1}$ . После вычисления вектора  $u_q^n$  (из решения краевой подзадачи в  $\Omega_q$ ) 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  $u_q \in \mathcal{R}^{M^2 N}$  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

$$v_q = C_{q,q-1}u_q, \quad w_q = C_{q,q+1}u_q, \quad A_{q,q \pm 1} = Q_{q,q \pm 1}C_{q,q \pm 1}, \quad (11)$$

где  $C_{q,q \pm 1} \in \mathcal{R}^{M^2 N, M^2}$ ,  $Q_{q,q \pm 1} \in \mathcal{R}^{M^2 M^2 N}$  суть 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

$$\begin{aligned} v_q^n &= \hat{B}_{q,q-1}w_{q-1}^{n-1} + \hat{B}_{q,q+1}v_{q+1}^{n-1} + \hat{g}_q, \quad q = 2, \dots, P, \\ w_q^n &= \check{B}_{q,q-1}w_{q-1}^{n-1} + \check{B}_{q,q+1}v_{q+1}^{n-1} + \check{g}_q, \quad q = 1, \dots, P-1, \\ \hat{B}_{1,0} &= \hat{B}_{P,P+1} = 0, \end{aligned} \quad (12)$$

в которых введены обозначения

$$\begin{aligned} \hat{g}_q &= C_{q,q-1}A_{q,q}^{-1}f_q, \quad \check{g}_q = C_{q,q+1}A_{q,q}^{-1}f_q, \\ \hat{B}_{q,q \pm 1} &= C_{q,q-1}A_{q,q}^{-1}Q_{q,q \pm 1}, \quad \check{B}_{q,q \pm 1} = C_{q,q+1}A_{q,q}^{-1}Q_{q,q \pm 1}. \end{aligned} \quad (13)$$

Мы не будем вдаваться в детали представления матриц  $\hat{B}_{q,q \pm 1}$ ,  $\check{B}_{q,q \pm 1}$  и отметим, что они в алгебраическом смысле являются матричными аналогами операторов Пуанкаре–Стеклова, реализуя итерационные интерфейсы только между внутренними граничными условиями на  $\Gamma_{q,q \pm 1}$ . При этом, естественно, на каждом шаге решаются задачи в подобластях  $\Omega_q$ , 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  $g = (\hat{g}_1, \hat{g}_2, \hat{g}_2, \dots, \hat{g}_{P-1}, \hat{g}_{P-1}, \hat{g}_P)^t$  векторы порядка  $2M^2(P-1)$ , итерационный процесс (12) можно записать в виде

$$s^n = Ts^{n-1} + g, \quad n = 1, 2, \dots, \quad (14)$$

где матрица перехода  $T$  является блочно-двудиagonalной, имея 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

$$\bar{A}s \equiv (I - T)s = g, \quad (15)$$

и для ускорения итераций (14) естественно применить какой-либо из методов в подпространствах Крылова.

Например, при использовании алгоритмов сопряженных градиентов (СГ) или сопряженных невязок (СН) получаем следующие формулы, см. [17]:

$$\begin{aligned} r^0 &= g - \bar{A}s^0 = \hat{s}^1 - s^0, \quad \hat{s}^1 = Ts^0 + g, \quad p^0 = r^0, \\ s^{n+1} &= s^n + \alpha_n^{(\nu)} p^n, \quad \alpha_n^{(\nu)} = \rho_n^{(\nu)} / \delta_n^{(\nu)}, \quad \rho_n^{(\nu)} = (\bar{A}^\nu r^n, r^n), \quad \delta_n^{(\nu)} = (\bar{A} p^n, \bar{A}^{(\nu)} p^n), \\ r^{n+1} &= r^n - \alpha_n^{(\nu)} \bar{A} p^n, \quad p^{n+1} = r^{n+1} + \beta_n^{(\nu)} p^n, \quad \beta_n^{(\nu)} = \rho_{n+1}^{(\nu)} / \rho_n^{(\nu)}. \end{aligned} \quad (16)$$

Здесь  $\nu = 0$  и  $\nu = 1$  соответствуют алгоритмам СГ и СН, а индексы “ $\nu$ ” у векторов  $r^n, s^n$  и  $p^n$  для краткости опущены. Отметим, что реализация каждой итерации (16) связана с одним умножением матрицы  $T$  на вектор (в том числе при  $\nu = 1$ , в силу соотношения  $\bar{A} p^{n+1} = \bar{A} r^{n+1} + \beta_n^{(\nu)} \bar{A} p^n$ ). А вследствие определения  $T$ , это связано с вычислением частичных решений в подобластях  $\Omega_q$  (параллельно для всех  $q = 1, \dots, P$ ). Однако заметим, что скалярные произведения в (16) определяются для векторов размерности  $2(P-1)M^2$ , равной общему количеству узлов во всех интерфейсных сечениях  $\Gamma_q$ . Традиционное условие окончания итераций имеет вид  $(r^n, r^n) / (g, g) \leq \varepsilon_e^2 \ll 1$ , при заданном значении  $\varepsilon_e$ .

Решение вспомогательных задач в подобластях  $\Omega_q$  целесообразно (при достаточно большом значении  $M^2N$ ) также проводить с помощью предобусловленного метода в подпространствах Крылова. В качестве предобуславливателя естественным образом выбирается какой-то из алгоритмов неполной факторизации. Если через  $r_{in}^{nq}$  обозначить величину невязки на  $n_q$ -м шаге внутреннего итерационного процесса вида (16), то критерием его окончания можно выбрать

$$(r_{in}^{nq}, r_{in}^{nq}) / (f_q^n, f_q^n) \leq (\varepsilon_{in}^{(n)})^2 \quad (17)$$

где величина  $\varepsilon_{in}^{(n)}$  может выбираться динамически в зависимости от номера внешней итерации  $n$ .

Отметим, что при этом фактически для внешнего итерационного процесса используется динамическое предобуславливание (в [15] используется термин *flexible*, т.е. гибкое). Строго говоря, в (15) матрица зависит от  $n$ , что может интерпретироваться, как приближенное умножение на  $\bar{A}$  в формулах (16).

### 3. Особенности распараллеливания алгоритмов

В соответствии с определением блочной структуры матрицы  $T$  и векторов, параллельная реализация крыловского метода (16) (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  $\Omega_q$ .

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

$$\begin{aligned} v_q^0 &= C_{q,q-1}u_q^0 = \{u_{i,j,k_N^{q-1}+1}^0 - \theta u_{i,j,k_N^{q-1}}^0; i, j, = 1, \dots, M\}, \\ w_q^0 &= C_{q,q+1}u_q^0 = \{u_{i,j,k_1^{q+1}-1}^0 - \theta u_{i,j,k_1^{q+1}}^0; i, j, = 1, \dots, M\}. \end{aligned} \quad (18)$$

Отметим, что  $v_q^0$  представляет правое краевое условие для  $\Omega_{q-1}$ , а  $w_q^0$  – левое краевое условие для  $\Omega_{q+1}$ . Затем осуществляются встречные пересылки всех компонент  $v_{q,i,j}^0, w_{q,i,j}^0$  данных векторов по следующей схеме:

$$v_q^0, w_q^0 : \Omega_q \rightarrow \Omega_{q\pm 1}; \quad v_{q+1}^0, w_{q-1}^0 : \Omega_{q\pm 1} \rightarrow \Omega_q. \quad (19)$$

Далее формируются и решаются вспомогательные СЛАУ вида (9) в  $\Omega_q$ :

$$A_{q,q}\hat{u}_q^1 = \bar{f}_q = \begin{cases} f_{k_1^q} + w_{q-1}^0, & k = k_1^q, \\ f_{k_N^q} + v_{q+1}^0, & k = k_N^q, \\ f_k, & k = k_1^q + 1, \dots, k_N^q - 1. \end{cases} \quad (20)$$

По найденным решениям на интерфейсных границах  $\Gamma_{q,q\pm 1}$ , с помощью формул (18) (при замене  $u^0$  на  $\hat{u}^1$ ) находятся новые краевые условия  $\hat{v}_q^1, \hat{w}_q^1$ , составляющие блочные компоненты вектора  $\hat{s}^1$ , из которого и находятся  $r^0 = p^0 = \hat{s}^1 - s^0$ .

- б. Реализация итераций для каждого  $n$  включает одно умножение матрицы на вектор  $\bar{A}p^n$  (или  $\bar{A}r^n$ ), вычисление двух скалярных произведений и нахождение линейных комбинаций векторов. Первая из этих операций, вследствие очевидного соотношения

$$t^n \equiv \bar{A}p^n = p^n - q^n, \quad q^n = Tp^n, \quad (21)$$

выполняется в соответствии с формулами (18)–(20): сначала определяются аналоги краевых условий (в (18) 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  $\Omega_q$  решаются аналогичные (20) подзадачи –

$$A_{q,q}\varphi_q^n = \psi_q^n = \begin{cases} \check{t}_{q-1}^n, & k = k_1^q, \\ \hat{t}_{q+1}^n, & k = k_N^q, \\ 0, & k = k_1^q + 1, \dots, k_N^q - 1. \end{cases} \quad (22)$$

где в правых частях отсутствуют не зависящие от  $n$  члены  $f_k$ , и наконец, по найденным значениям  $\varphi_q^n$  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 (18), надо в данном случае только заменить векторы:

$$\begin{aligned} \hat{q}_q^n &= \{\varphi_{i,j,k_N^{q-1}+1}^n - \theta \varphi_{i,j,k_N^{q-1}}^n; i, j, = 1, \dots, M\}, \\ \check{q}_q^n &= \{\varphi_{i,j,k_1^{q+1}-1}^n - \theta \varphi_{i,j,k_1^{q+1}}^n; i, j, = 1, \dots, M\}. \end{aligned} \quad (23)$$

Итоговые блочные компоненты результата операции (21) имеют очевидный 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 (16) вычисляются путем суммирования по всем узлам интерфейсных сечений  $\Gamma_q$ , т.е. например,

$$\begin{aligned} (t^n, p^n) &= \sum_{i,j=1}^M \left( \sum_{q=2}^p \hat{t}_{q,i,j}^n \cdot \hat{p}_{q,i,j}^n + \sum_{q=1}^{p-1} \check{t}_{q,i,j}^n \cdot \check{p}_{q,i,j}^n \right) = (\check{t}^n, \check{p}^n)_1 + \sum_{q=2}^{p-1} (t^n, p^n)_q + (\hat{t}^n, \hat{p}^n)_p, \\ (t^n, p^n)_q &= (\hat{t}^n, \hat{p}^n)_q + (\check{t}^n, \check{p}^n)_q, \quad (\hat{t}^n, \hat{p}^n)_q = \sum_{i,j=1}^M \hat{t}_{q,i,j}^n \hat{p}_{q,i,j}^n. \end{aligned} \quad (24)$$

- в. После сходимости итерационных приближений  $v_q^n, w_q^n$  с достаточной точностью мы легко получаем искомое решение из вспомогательных задач вида (20): 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

$$S_P = T_1/T_P, \quad E_P = S_P/P, \quad (25)$$

где полное время решения задачи складывается из времени выполнения арифметических операций  $T_P^a$  и времени обмена данными  $T_P^c$  между подобластями  $\Omega_q$ :

$$T_P = T_P^a + T_P^c \approx \tau_a V_a + N_a(\tau_0 + \tau_c V_c). \quad (26)$$

Здесь  $\tau_a$  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,  $\tau_0$  и  $\tau_c$  – 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 времена реализации пп. а, в сравнимы со временем выполнения одной итерации в п.б, то для оценки величин  $S_P$  и  $E_P$  при  $n \gg 1$  достаточно проанализировать качество распараллеливания одного итерационного шага.

Решение СЛАУ (22) в каждой  $q$ -й подобласти занимает время  $T_a^{(1)}$ , пропорциональное числу узлов  $M^2 N$  в  $\Omega_q$  и количеству внутренних итераций  $n_{in}(\varepsilon_{in})$ , которое для современных предобусловленных крыловских методов составляет  $N^\gamma |ln \varepsilon_{in}|$ ,  $\gamma \geq 1/2$ :

$$T_a^{(1)} = C_1 M^2 N^{\gamma+1} |ln \varepsilon_{in}| \tau_a, \quad (27)$$

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

Время вычисления “краевых условий”  $\hat{q}_q$  и  $\check{q}_q$  из (23) мало по сравнению с (27):  $T_a^{(2)} \leq 6M^2 \tau_a$ . Выполнение векторных операций на одной внешней итерации (16) в  $\Omega_q$  (их всего пять) занимает время

$$T_a^{(3)} \approx 20M^2 + C_2, \quad (28)$$

где последнее слагаемое относится к “довычислению” компонент векторных произведений (24), 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

$$T_P^c \leq C_3(\tau_0 + 2\tau_c M^2 N). \quad (29)$$

Здесь значение  $C_3$  не зависит от  $M, N$  и  $P$  в идеальном случае, когда все пересылки типа (19) 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.

$$S_P = T_P^a \cdot P / (T_P^a + T_P^c) \approx P, \quad E_P \approx 1. \quad (30)$$

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  $n_{in} \cdot n_e(\varepsilon) > n(\varepsilon)$ , ( $n_{in}$  – 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 ( $M$  исчисляется тысячами или десятками тысяч), когда главная проблема заключается в интеграции вычислительных ресурсов, в силу тривиальной нехватки памяти одного процессора.

## 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  $h = 1/(M + 1)$ . 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 осуществлялось одномерное 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 предобуславливания.

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 5 чисел: количества потребовавшихся внешних итераций (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 (16) метода сопряженных градиентов.

Табл. 3 и 4 содержат те же характеристики итерационных процессоров, но при декомпозиции расчетной области на три одинаковые подобласти, с теми же величинами пересечений

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

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

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

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

$\Delta$  в два и четыре шага сетки.

$\theta/M$	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	$\infty$	$\infty$	$\infty$

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

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

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

Анализ приведенных результатов позволяет сделать следующие выводы:

- с увеличением отношения  $M/\Delta$  (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  $\theta$  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 “ $\infty$ ”), 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 более плохие 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 при этом трудоемкость решения каждой вспомогательной задачи в подобласти уменьшается, общий объем вычислений в методе декомпозиции, по сравнению с решением исходной задачи на одном процессоре (если это позволяет его память), будет увеличиваться.

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.—М., ОВМ РАН, 1981, препринт N 14.
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 Y.A., Matsokin A.M. Fictitious domain and domain decomposition method.—Sov.J. Num. Anal. Math. Mod., 1986, v. 1, N 1, 3-35.
4. Quarteroni A., Valli A. Domain Decomposition Methods for Partial Differential Equations.—Oxford Science Publications/Clarendon Press., Oxford, 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., v. 57, N 195, 1991, 1-21.
6. Dryja M., Widlund O.B. Domain decomposition algorithms with small overlap.—SIAM J. Sci. Comp., 1994, 15(3), 604-620.

7. Ильин В.П. О стратегиях распараллеливания в математическом моделировании.// Программирование, N 1, 1999, 41-46.
8. Ильин В.П., Свешников В.М. Оценки эффективности распараллеливания алгоритмов декомпозиции областей.//Автометрия, N 1, 2002, 31-41.
9. Ильин В.П. Параллельные алгоритмы для больших прикладных задач: проблемы и технологии.//Автометрия, т. 43, N 2, 2007, 3-21.
10. Nepomnyashchikh S.V. Domain decomposition methods.–Radon Series Comp. Appl. Math., 2007, 1, 89-159.
11. Василевский Ю.В., Ольшанский М.А. Краткий курс по многосеточным методам и методам декомпозиции области.–М., изд. фак. ВМиК МГУ; МАКС Пресс, 2007.
12. Loisel S., Szyld D.B. On the geometric convergence of optimized Schwarz methods with application to elliptic problems.–Numer. Math., v. 114, 2010, 697-728.
13. Свешников В.М. Построение прямых и итерационных методов декомпозиции.// СибЖИМ, 2009, т.12, N 3(39), с. 99 - 109.
14. Ильин В.П. Проблемы высокопроизводительных технологий решения больших разреженных СЛАУ.–Вычислительные методы и программирование.–М., МГУ, т. 10, 2009, 141-147.
15. <http://www.ddm.org/>
16. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений.–Новосибирск, изд. ИВМиМГ СО РАН, 2001.
17. Ильин В.П. Методы и технологии конечных элементов.–Новосибирск, изд. ИВМиМГ СО РАН, 2007.
18. Ильин В.П. Об экзапроблемах математического моделирования.–CAD/CAM/CAE Observer, N 2 (54), 2010, 85-92.
19. Saad Y. Iterative methods for sparse linear systems. - PWS Publishing: New York, 1996