

В. П. Ильин, В. М. Свешников, С. А. Литвиненко

(Новосибирск)

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ТРЕХМЕРНОГО АНАЛОГА МЕТОДА ПИСМАНА – РЕЧФОРДА*

Рассмотрен метод декомпозиции областей с применением трехмерного аналога алгоритма Писмана – Речфорда. Даны оценки эффективности распараллеливания и результаты численных экспериментов на вычислительных системах RM-600-E30 и MBS-1000 для разных сеточных областей и процессорных топологий.

Введение. Методы декомпозиции областей являются эффективным средством распараллеливания решения многомерных краевых задач при помощи сеточных алгоритмов конечных разностей, конечных элементов или конечных объемов (см., например, работу [1] и цитируемую в ней литературу). Их суть заключается в том, что исходная область разбивается на несколько подобластей, и в каждой из них независимо решаются вспомогательные задачи на различных процессорах, а также происходит периодический обмен данными между ними. При этом вычислительная сложность алгоритмов в целом, как правило, увеличивается, поскольку требуются дополнительные итерации по подобластям. Кроме того, существенный вклад во время решения задачи вносят межпроцессорные обмены. В связи с этим представляет интерес исследование распараллеливания на основе декомпозиции областей с применением быстрых итерационных методов, например метода Писмана – Речфорда, реализация которого осуществляется практически без увеличения общего объема вычислений.

Данная работа посвящена экспериментальным оценкам эффективности распараллеливания трехмерного аналога метода Писмана – Речфорда. В двумерном случае аналогичная проблема рассматривалась в работах [1, 2].

В предлагаемом методе решение трехмерной краевой задачи выполняется с помощью итераций, каждая из которых включает два полушага: на первом решаются двумерные задачи в плоскостях, перпендикулярных оси z , а на втором полушаге – одномерные задачи на линиях, параллельных оси z . Назовем эти итерации внешними в отличие от внутренних итераций, которые проводятся по классическому методу Писмана – Речфорда для решения двумерных задач. На втором полушаге внешних итераций требуется решение

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (гранты № 01-01-00819, № 02-01-01176).

одномерных задач. Наиболее экономично решать их методом прогонки, что обычно и делается при последовательных вычислениях. Однако прогонка непосредственно эффективно не распараллеливается. В связи с этим для решения одномерных задач применяется блочный метод четно-нечетной редукции без обратного хода, который легко поддается распараллеливанию.

При проведении вычислительного эксперимента на вычислительных системах (ВС) имеем дело со следующими основными операциями с точки зрения затрат времени на решение задачи:

- арифметико-логические операции (мы предполагаем, что все арифметические действия выполняются с двойной точностью, необходимой для обеспечения требований задач математического моделирования);
- обмен информацией между процессорами;
- инициализация обмена.

Если через τ_a обозначить среднее время выполнения одной арифметической операции с плавающей запятой, через τ_c – время передачи одного числа с двойной точностью, занимающего 8 байт, через τ_0 – время инициализации обмена, то общее время решения некоторой задачи на ВС с синхронной работой процессоров оценивается как

$$T = N_a \tau_a + N_i (\tau_0 + \tau_c N_c),$$

где N_a – количество арифметических операций, N_i – число информационных обменов, а N_c – усредненный объем одного обмена. Поскольку

$$\tau_0 \gg \tau_c \gg \tau_a,$$

то для повышения эффективности распараллеливания необходимо организовать вычислительный процесс с минимальным количеством обменов, а сами обмены совмещать во времени с выполнением арифметических операций. Отметим также, что асинхронная работа процессоров приводит к простоям некоторых из них, уменьшая в целом эффективность вычислительной системы.

В разд. 1 даны постановка задачи и описание трехмерного аналога метода Писмана – Речфорда, ориентированного на эффективную параллельную реализацию в простой топологии ВС. В разд. 2 приведены результаты численных экспериментов на вычислительных системах RM-600-E30 ("Siemens – Nixdorf") и МВС-1000 (ФГУП НИИ «Квант», Москва) для разных сеточных областей и разного числа процессоров, их сравнительный анализ и обсуждение.

1. Постановка задачи и описание алгоритма. 1.1. *Постановка задачи.* Пусть требуется найти функцию $\bar{u}(x, y, z)$, являющуюся решением краевой задачи

$$\begin{aligned} \mathcal{L}\bar{u}(x, y, z) &= g(x, y, z), & (x, y, z) \in G, \\ \ell\bar{u}(x, y, z) &= 0, & (x, y, z) \in \Gamma, \end{aligned} \tag{1}$$

где G – расчетная область, которую для простоты считаем параллелепипедом, Γ – ее граница ($G \cup \Gamma = \bar{G}$), \mathcal{L} – эллиптический дифференциальный оператор, ℓ – оператор граничных условий. Рассмотрим также нестационарные задачи. Тогда в (1) \mathcal{L} – параболический оператор, ℓ – оператор граничных и

начальных условий, а искомая функция \bar{u} зависит не только от пространственных координат, но и от времени. Построим в \bar{G} параллелепипедоидальную сетку Ω^h при помощи пересечения плоскостей, перпендикулярных координатным линиям. На этой сетке аппроксимируем задачу (1) методом конечных разностей, конечных элементов или конечных объемов [3]. В результате получим систему семиточечных сеточных линейных алгебраических уравнений:

$$\begin{aligned} & -a_{i,j,k}^1 u_{i-1,j,k} - a_{i,j,k}^2 u_{i+1,j,k} - a_{i,j,k}^3 u_{i,j-1,k} - a_{i,j,k}^4 u_{i,j+1,k} - \\ & - a_{i,j,k}^5 u_{i,j,k-1} - a_{i,j,k}^6 u_{i,j,k+1} + a_{i,j,k}^0 u_{i,j,k} = f_{i,j,k}, \\ & i=1,2,\dots,I; \quad j=1,2,\dots,J; \quad k=1,2,\dots,K, \end{aligned} \quad (2)$$

где $u_{i,j,k}$ – приближенное значение искомой функции $\bar{u}(x_i, y_j, z_k)$ в узлах сетки; $a_{i,j,k}^m$, $m=0,1,\dots,6$, $f_{i,j,k}$ – известные величины; I, J, K – число узлов разностной сетки по координатам x, y, z соответственно. Запишем (2) в матричном виде:

$$Au = f,$$

где A – квадратная матрица порядка IJK , а $u = \{u_{i,j,k}\}$, $f = \{f_{i,j,k}\}$ – искомый и заданный векторы.

В случае нестационарных задач при использовании неявных по времени аппроксимаций системы вида (2) требуется решать на каждом временном шаге, и рассмотренные далее оценки эффективности распараллеливания остаются в силе. Мы не будем останавливаться на распараллеливании явных методов для параболических дифференциальных уравнений, так как они не связаны с решением алгебраических систем и имеют свои проблемы обеспечения устойчивости и минимизации коммуникационных потерь.

При характерных современных размерах сетки в несколько миллионов (десятков или даже сотен миллионов) узлов применение прямых методов решения ленточных систем типа (2) (если только они не имеют частный простой вид) является неэффективным в силу высоких требований к вычислительным ресурсам: количество арифметических операций и объем памяти равны примерно $I^3 J^3 K$ и $I^2 J^2 K$ соответственно.

Использование быстрых итерационных алгоритмов с предобуславливанием (методов неполной факторизации [3]) и ускорением с помощью сопряженных градиентов или чебышевских параметров позволяет снизить указанные оценки до $O(IJK^{3/2})$ и $O(IJK)$ соответственно.

Если для решения системы (2) высокого порядка применить «классический» метод декомпозиции областей, то это эквивалентно применению итерационного альтернирующего метода Шварца с наложением соседних подобластей, равным характерному шагу сетки h (мы предполагаем, что I, J, K являются величинами одного порядка $O(h^{-1})$). Это обуславливает число внешних итераций порядка $O(h^{-1})$, что неизбежно приведет к значительным лишним вычислительным затратам (такой алгоритм, очевидно, нецелесообразно применять на однопроцессорных компьютерах).

1.2. *Трехмерный аналог метода Писмана – Речфорда.* Рассмотрим аналог итерационного неявного метода Писмана – Речфорда для трехмерного

случая. При решении системы (2) он формально представляет собой алгебраическую задачу, состоящую в вычислении последовательности векторов

$$\begin{aligned} \mathbf{u}^{n-1/2} &= \mathbf{u}^{n-1} - \omega_z (A_{xy} \mathbf{u}^{n-1/2} + A_z \mathbf{u}^{n-1} - f); \\ \mathbf{u}^n &= \mathbf{u}^{n-1/2} - \omega_z (A_{xy} \mathbf{u}^{n-1/2} + A_z \mathbf{u}^n - f), \end{aligned} \quad (3)$$

где $n = 1, 2, \dots$ – номера внешних итераций; ω_z – итерационный внешний числовой параметр; A_{xy} и A_z – пятидиагональная и трехдиагональная матрицы, определяемые как

$$\begin{aligned} A_{xy} + A_z &= A, \\ (A_{xy} \mathbf{u})_{i,j,k} &= -a_{i,j,k}^1 \mathbf{u}_{i-1,j,k} + a_{i,j,k}^{0,x} \mathbf{u}_{i,j,k} - a_{i,j,k}^2 \mathbf{u}_{i+1,j,k} - \\ &\quad - a_{i,j,k}^3 \mathbf{u}_{i,j-1,k} + a_{i,j,k}^{0,y} \mathbf{u}_{i,j,k} - a_{i,j,k}^4 \mathbf{u}_{i,j+1,k}, \\ (A_z \mathbf{u})_{i,j,k} &= -a_{i,j,k}^5 \mathbf{u}_{i,j,k-1} + a_{i,j,k}^{0,z} \mathbf{u}_{i,j,k} - a_{i,j,k}^6 \mathbf{u}_{i,j,k+1}, \\ a_{i,j,k}^{0,x} + a_{i,j,k}^{0,y} + a_{i,j,k}^{0,z} &= a_{i,j,k}^0. \end{aligned} \quad (4)$$

Соотношения (3), (4) представляют собой векторную запись метода переменных направлений Писмана – Речфорда, в котором каждая n -я итерация выполняется за два полушага. Рассмотрим реализацию каждого из них более подробно на покомпонентном уровне. На первом полушаге решается K двумерных систем порядка IJ :

$$(E + \omega_z A_{xy}) \mathbf{u}^{n-1/2} = f^{xy}, \quad (5)$$

а на втором – IJ одномерных систем порядка K :

$$(E + \omega_z A_z) \mathbf{u}^n = f^z. \quad (6)$$

Здесь и далее E – единичная матрица, а правые части

$$f^{xy} = \{f_{i,j,k}^{xy}\}, \quad f^z = \{f_{i,j,k}^z\}$$

вычисляются как

$$\begin{aligned} f_{i,j,k}^{xy} &= \omega_z f_{i,j,k} + \omega_z a_{i,j,k}^5 \mathbf{u}_{i,j,k-1}^{n-1} + \omega_z a_{i,j,k}^6 \mathbf{u}_{i,j,k+1}^{n-1} + (1 - \omega_z a_{i,j,k}^{0,z}) \mathbf{u}_{i,j,k}^{n-1}, \\ f_{i,j,k}^z &= \omega_z f_{i,j,k} + \omega_z a_{i,j,k}^1 \mathbf{u}_{i-1,j,k}^{n-1/2} + \omega_z a_{i,j,k}^2 \mathbf{u}_{i+1,j,k}^{n-1/2} + (1 - \omega_z a_{i,j,k}^{0,x}) \mathbf{u}_{i,j,k}^{n-1/2} + \\ &\quad + \omega_z a_{i,j,k}^3 \mathbf{u}_{i,j-1,k}^{n-1/2} + \omega_z a_{i,j,k}^4 \mathbf{u}_{i,j+1,k}^{n-1/2} + (1 - \omega_z a_{i,j,k}^{0,y}) \mathbf{u}_{i,j,k}^{n-1/2}. \end{aligned}$$

Двумерная задача (5), в свою очередь, решается также методом Писмана – Речфорда:

$$\begin{aligned} \mathbf{v}^{\bar{n}-1/2} &= \mathbf{v}^{\bar{n}-1} - \omega_{xy}(A_x \mathbf{v}^{\bar{n}-1/2} + A_y \mathbf{v}^{\bar{n}-1} - f^{xy}), \\ \mathbf{v}^{\bar{n}} &= \mathbf{v}^{\bar{n}-1/2} - \omega_{xy}(A_x \mathbf{v}^{\bar{n}-1/2} + A_y \mathbf{v}^{\bar{n}} - f^{xy}), \\ \mathbf{v}^0 &= \mathbf{u}^{n-1}, \end{aligned} \quad (7)$$

где $\bar{n} = 1, 2, \dots$ – номера внутренних итераций; ω_{xy} – итерационный внутренний числовой параметр; A_x, A_y – трехдиагональные матрицы, определяемые соотношениями

$$\begin{aligned} A_x + A_y &= E + \omega_z A_{xy}, \\ (A_x \mathbf{u})_{i,j,k} &= -\omega_z a_{i,j,k}^1 \mathbf{u}_{i-1,j,k} - \omega_z a_{i,j,k}^2 \mathbf{u}_{i+1,j,k} + (1/2 + \omega_z a_{i,j,k}^{0,x}) \mathbf{u}_{i,j,k}, \\ (A_y \mathbf{u})_{i,j,k} &= -\omega_z a_{i,j,k}^3 \mathbf{u}_{i,j-1,k} - \omega_z a_{i,j,k}^4 \mathbf{u}_{i,j+1,k} + (1/2 + \omega_z a_{i,j,k}^{0,y}) \mathbf{u}_{i,j,k}. \end{aligned}$$

Оба полушага в алгоритме (7), реализующем первый полушаг алгоритма (3), требуют обращения трехдиагональных матриц, что наиболее экономично проводится методом прогонки.

Формулы (3)–(7) ориентированы на параллельную реализацию в топологии ВС типа линейка. При этом сеточная расчетная область разбивается на горизонтальные слои, каждому из которых соответствует один процессор, и наиболее трудоемкие внутренние итерации осуществляются без обменов.

Рассмотрим подробнее второй полушаг алгоритма (3). Система уравнений (6) представляет собой LJ независимых одномерных систем, которые решаются по однотипным алгоритмам, описанным далее. Занумеруем их индексами (i, j) , $i = 1, 2, \dots, I$, $j = 1, 2, \dots, J$, и запишем (i, j) -ю систему в блочном виде:

$$-A_l \bar{\mathbf{u}}_{l-1} + B_l \bar{\mathbf{u}}_l - C_l \bar{\mathbf{u}}_{l+1} = \bar{\mathbf{f}}_l, \quad l = 1, 2, \dots, L, \quad A_1 = C_L = 0, \quad (8)$$

где L – заданное целое число; $\bar{\mathbf{u}}_l, \bar{\mathbf{f}}_l$ – векторы размера $p = K/L$: $\bar{\mathbf{u}}_l = \{\mathbf{u}_m^l\}$, $\bar{\mathbf{f}}_l = \{\mathbf{f}_m^l\}$; $\mathbf{u}_m^l = \mathbf{u}_{i,j,(l-1)p+m}$, $\mathbf{f}_m^l = \mathbf{f}_{i,j,(l-1)p+m}$, $m = 1, 2, \dots, p$; A_l, B_l, C_l – квадратные матрицы порядка p , причем B_l трехдиагональные, а A_l, C_l имеют по одному ненулевому элементу a_l, c_l в правом верхнем и левом нижнем углах соответственно.

Для решения системы (8) применяется блочный метод четно-нечетной редукции без обратного хода [4]. Если $L = 2^R$ (а именно такой случай мы рассматриваем в настоящей работе), то выполняется R этапов редукции. На r -м этапе проводится превычисление a_l, c_l :

$$\mathbf{a}_l^{(r)} = (\mathbf{a}_l \hat{\mathbf{b}}_{p,1}^{l-} \mathbf{a}_l^{(r-1)})^{(r-1)}, \quad \mathbf{c}_l^{(r)} = (\mathbf{c}_l \hat{\mathbf{b}}_{1,p}^{l+} \mathbf{c}_l^{(r-1)})^{(r-1)}, \quad (9)$$

а также элементов матрицы B_l :

$$b_{1,1}^{(r)} = (b_{1,1} - a_l \hat{b}_{p,p}^{l^-} c_{l^-})^{(r-1)}, \quad b_{p,p}^{(r)} = (b_{p,p} - c_l \hat{b}_{1,1}^{l^+} a_{l^+})^{(r-1)} \quad (10)$$

и, наконец, правой части:

$$(f_1^l)^{(r)} = (f_1^l + a_l w_p^{l^-})^{(r-1)}, \quad (f_p^l)^{(r)} = (f_p^l + c_l w_1^{l^+})^{(r-1)}. \quad (11)$$

Здесь $l_r^{\pm} = l \pm 2^{r-1}$; $b_{i,s}^l$ ($i, s = 1, 2, \dots, p$) – элементы матрицы B_l ; $\hat{b}_{i,s}^l$ – элементы обратной матрицы B_l^{-1} , а также введены обозначения

$$w_p^l = \sum_{k=1}^p \hat{b}_{p,k}^l f_k^l, \quad w_1^l = \sum_{k=1}^p \hat{b}_{1,k}^l f_k^l.$$

При недопустимых значениях $l_r^- < 1$, $l_r^+ > L$ соответствующие вычисления в формулах (9)–(11) не проводятся. Элементы матриц B_l , B_l^{-1} и компоненты правой части \tilde{f}_l , не вошедшие в формулы (9)–(11), остаются неизменными. На последнем R -м этапе редукции мы получим L независимых подсистем

$$B_l^{(R)} \bar{u}_l = \bar{f}_l^{(R)}, \quad l = 1, 2, \dots, L, \quad (12)$$

каждая из которых решается методом прогонки.

Запишем систему одномерных трехточечных уравнений в виде

$$-\alpha_i v_{i-1} + \beta_i v_i - \gamma_i v_{i+1} = \eta_i, \quad i = 1, 2, \dots, p, \quad \alpha_1 = \gamma_p = 0,$$

где $\alpha_i, \beta_i, \gamma_i$ – известные коэффициенты, а v_i – искомые величины. Метод прогонки для ее решения реализуется по формулам

$$\theta_i = (\beta_i - \alpha_i \delta_{i-1})^{-1}, \quad \delta_i = \gamma_i \theta_i, \quad \kappa_i = (\eta_i + \alpha_i \kappa_{i-1}) \theta_i, \quad i = 1, 2, \dots, p, \quad (13)$$

$$v_i = \delta_i v_{i+1} + \kappa_i, \quad i = p, p-1, \dots, 1.$$

Для нахождения элементов обратной матрицы $\hat{b}_{1,p}, \hat{b}_{p,p}$ решаем систему уравнений

$$B_l v = \bar{\xi}_p \quad (14)$$

с правой частью $\bar{\xi}_p = (0, 0, \dots, 0, 1)$ относительно неизвестного вектора $v = \{v_i, i = 1, 2, \dots, p\}$. При этом в (13) значительно сокращается объем арифметических операций, выполняемых фактически по формулам

$$\kappa_1 = \kappa_2 = \dots = \kappa_{p-1} = 0, \quad \kappa_p = \theta_p, \quad v_p = \kappa_p,$$

$$v_i = \delta_i v_{i+1}, \quad i = p-1, p-2, \dots, 1.$$

Отыскание $\hat{b}_{1,1}, \hat{b}_{p,1}$ приводит к решению (14) с правой частью $\bar{\xi}_p = (1, 0, \dots, 0)$, что упрощает в (13) вычисление κ_i :

$$\kappa_1 = \theta_1, \quad \kappa_i = \alpha_i \kappa_{i-1} \theta_i, \quad i = 1, 2, \dots, p.$$

Решая (14) с правой частью \bar{f}_i , находим

$$w_p^l = v_p, \quad w_1^l = v_1.$$

Каждая из L систем (12) формируется и решается на отдельном процессоре. Для реализации формул (9)–(11) на каждом шаге редукции требуется провести между процессорами обмен величинами, которые объединим в две группы:

$$D^l = (a_l, c_l, \hat{b}_{1,1}^l, \hat{b}_{1,p}^l, \hat{b}_{p,1}^l, \hat{b}_{p,p}^l), \quad W^l = (w_1^l, w_p^l).$$

Величины D^l не зависят от номера итерации, а W^l должны перевычисляться на каждой внешней итерации, так как они зависят от \bar{f}^l .

Таким образом, отметим, что описанный внешний метод Писмана – Речфорда при оптимальных значениях итерационного параметра ω_z требует для получения достаточной точности проведения числа итераций порядка $O(h^{-1})$, а если к нему дополнительно применить метод чебышевского ускорения или сопряженных градиентов (которые легко распараллеливаются и незначительно увеличивают общий объем вычислений), то число итераций сокращается до $O(h^{-1/2})$. Что касается внутренних итераций (7) по горизонтальным плоскостям, то за счет улучшения числа обусловленности решаемых здесь алгебраических систем требуемое число итераций сокращается до $O(h^{-1/2})$, а при дополнительном использовании ускорения – до $O(h^{-1/4})$. Итак, общий объем вычислений в указанном двухступенчатом методе имеет порядок $O(h^{-4,5})$, а при использовании ускорений на обоих уровнях – $O(h^{-3,75})$. Следует также указать, что при перестановочности операторов A_x, A_y, A_z количество итераций можно сократить за счет использования оптимальных последовательностей итерационных параметров, но это осуществимо только при решении исходных краевых задач частного вида.

2. Оценки эффективности распараллеливания. Представим сетку Ω^h в виде

$$\Omega^h = \bigcup_{l=1}^L \Omega_l^h,$$

где L – число подсеток Ω_l^h , причем каждая из них имеет одинаковое число узлов IJp (I, J, p – число узлов по x, y, z соответственно). Каждой подсетке ставится в соответствие один процессор, который проводит решение двумерных задач в горизонтальных плоскостях и решение редуцированных одномерных задач, определенных на данной подсетке. Вычисление коэффициентов разностных уравнений $a_{i,j,k}^m, m = 0, 1, \dots, 6$, на подсетке Ω_l^h проводится l -м процессором. Время расчета коэффициентов в приводимые далее оценки не

входит, поскольку при достаточно большом числе итераций его вклад незначителен.

При проведении расчетов на l -м процессоре на каждой внешней итерации требуется совершать обмены с соседними процессорами значениями компонент искомого вектора, лежащими на крайних плоскостях разностной сетки. Программа, реализующая эти расчеты, построена так, что обмены проводятся на фоне вычислений (см. шаг 2).

В целом решение исходной системы уравнений на каждом процессоре можно записать в виде последовательности шагов.

Шаг 1. До начала итераций рассчитываются и запоминаются в виде числовых массивов следующие величины для всех R этапов редукции по оси z :

– θ_k, δ_k – прогоночные коэффициенты для $k = 1, 2, \dots, p$ при проведении прогонок по оси z ;

– a_l, c_l – коэффициенты в формулах (9)–(11).

Шаг 2. Выполняется первый полушаг внешних итераций, на котором проводятся:

– инициализации обменов значениями $u_{i,j,k}$, лежащими на крайних плоскостях сетки, с соседними процессорами;

– решение двумерных задач методом Писмана – Речфорда для внутренних плоскостей, не ожидая окончания обменов;

– решение двумерных задач для крайних плоскостей после завершения обменов.

Шаг 3. Выполняется второй полушаг внешних итераций, на котором осуществляются:

– R этапов редукции, на каждом из которых вычисляются величины W^l для всех линий сетки и проводится обмен этими значениями с соответствующими процессорами;

– решение системы уравнений (12).

В качестве критериев эффективности распараллеливания рассматриваются ускорение вычислений $P = T_1/T_L$ и коэффициент эффективности $Q = P/L$, где T_L – время решения задачи на L процессорах. Зависимости этих величин от числа процессоров и от количества узлов сетки можно оценить на основе анализа одной внешней итерации, поскольку распараллеливание каждой из них проводится одинаково.

Реализация одной внешней итерации на одном процессоре (без распараллеливания) требует проведения числа арифметических действий в расчете на один узел сетки

$$Y_1^1 = c_1 + c_2 N + c_3,$$

где c_1 – количество арифметических действий для вычисления правых частей f^{xy} в (7), c_2 – число операций при выполнении одной внутренней итерации, N – число внутренних итераций, c_3 – количество операций для вычисления прогоночных коэффициентов вида (13). Данные вычисления проводятся за время

$$T_1^1 = Y_1^1 IJK\tau_a.$$

Заметим, что $T_1 = T_1^1 n_z$, где n_z – число внешних итераций.

Оценим теперь время $T_{l,L}^1$ выполнения одной внешней итерации на l -м процессоре при распараллеливании на L процессорах (напомним, что число подобластей также равно L и $l=1,2,\dots,L$) с учетом коммуникационных потерь. Отметим, что

$$T_L = \frac{n_z}{L} \sum_{l=1}^L T_{l,L}^1.$$

Назовем внутренним процессор, у которого есть оба соседних процессора, и крайним – процессор, у которого имеется только один такой сосед. Эти понятия вводятся относительно какого-либо этапа вычислительного алгоритма, например, один и тот же процессор может быть на одном шаге редукции внутренним, а на другом – крайним.

Общее число инициализаций обменов (или просто обменов) на одной внешней итерации на l -м процессоре

$$Z_l = Z_l^I + Z_l^R,$$

где Z_l^I – число обменов, необходимых для вычисления f^{xy} в (7), а Z_l^R – число обменов, проводимых при реализации формул (9)–(11) метода редукции. В том и другом случае на одном этапе алгоритма на каждом внутреннем процессоре инициализируются 4 обмена (2 приема + 2 посылки), а на крайнем процессоре – 2 обмена (1 прием + 1 посылка). Тогда

$$Z_l^I = \begin{cases} 4 & \text{при } 1 < l < L, \\ 2 & \text{при } l=1, L; \end{cases}$$

$$Z_l^R = \begin{cases} 4R-2 & \text{при } R \leq l \leq L-R+1, \\ 4R-2(r+1) & \text{при } l=R-r, L-R+1+r, \quad r=1,2,\dots,R-1, \end{cases}$$

и время проведения обменов

$$T_{l,L}^{1,Z} = Z_l \tau_0 + (Z_l^I + 2Z_l^R) \mathcal{L} p \tau_c.$$

Число арифметических операций в рассматриваемых вычислениях можно выразить как

$$Y_{l,L}^1 = Y_1^1 + \sum_{r=1}^L c_r^l,$$

где c_r^l – количество операций для формирования систем (12) на одном этапе редукции. Время, которое затрачивается на их выполнение, определяется как

$$T_{l,L}^{1,J} = Y_{l,L}^1 \mathcal{L} p \tau_a.$$

Для искомого времени $T_{l,L}^1$ справедлива оценка

$$T_{l,L}^1 \leq T_{l,L}^{1,Z} + T_{l,L}^{1,J}.$$

Знак неравенства здесь обусловлен тем, что в программной реализации во время обменов проводятся вычисления. Для коэффициента ускорения P в силу последних соотношений имеем

$$P \leq \eta L, \quad \eta = \frac{1}{1 + \frac{c_4 R}{\gamma_1^1}},$$

где $c_4 = \min c_r^l$. Отсюда видно, что коэффициент $\eta \leq 1$, и он уменьшается с увеличением L . Следовательно, даже в идеальном случае, когда число арифметических действий таково, что на их фоне успевают завершиться обмены, т. е. $T_{l,L}^1 = T_{l,L}^{1,l}$, мы имеем $P \leq L$, причем знак равенства достигается при $L = 1$.

Рассмотрим результаты численных экспериментов, проводившихся на примере решения модельной задачи Дирихле для уравнения Лапласа

$$\Delta u = 0, \quad u|_{\Gamma} = 1,$$

в кубе $\{0 < x < 1, 0 < y < 1, 0 < z < 1\}$. Расчеты выполнялись на сетках $\Omega^{h,k}$, $k = 1, 2, \dots$, с различным числом узлов. На них аппроксимировалась исходная краевая задача конечно-разностным методом. Начальное приближение выбиралось $u^0 = 0$. Внешний итерационный процесс прекращался при достижении заданного числа внешних итераций n_z . На каждой внешней итерации проводилось фиксированное число n_{xy} внутренних итераций. Завершение итерационного процесса по заданному числу итераций, а не по заданной точности обусловлено следующими соображениями. Во-первых, для оценки эффективности распараллеливания необходимо, по сути, время выполнения одной итерации. Во-вторых, на густой сетке число итераций может быть большим, что влечет за собой большое время счета. Эксперименты проводились с $n_z = 100$ и $n_{xy} = 10$.

Таблица 1

Метод Писмана – Речфорда (RM-600-E30)

L	Критерий	I × J × K		
		16 × 16 × 16	32 × 32 × 32	64 × 64 × 64
1	T_1	9,52	83,53	740,77
2	T_N	5,25	44,28	373,5
	P	1,81	1,89	1,98
	Q	0,90	0,94	0,99
4	T_N	2,91	22,73	189,03
	P	3,27	3,67	3,92
	Q	0,82	0,92	0,98
8	T_N	2,25	12,62	100,72
	P	4,24	6,61	7,35
	Q	0,53	0,83	0,92

Метод Писмана – Речфорда (МВС-1000)

L	Критерий	I × J × K				
		16 × 16 × 16	32 × 32 × 32	64 × 64 × 64	128 × 128 × 128	256 × 256 × 256
1	T_1	0,55	4,26	43,47	398,48	5696,10
2	T_N	0,30	4,11	34,91	352,87	4980,42
	P	1,81	1,03	1,25	1,13	1,13
	Q	0,90	0,52	0,62	0,56	0,56
4	T_N	0,18	2,07	17,19	153,37	2605,23
	P	2,97	2,05	2,53	2,60	2,19
	Q	0,74	0,51	0,63	0,65	0,55
8	T_N	0,16	0,97	9,94	78,97	1138,73
	P	3,45	4,39	4,37	5,05	5,05
	Q	0,43	0,55	0,55	0,63	0,63

Заметим, что при проведении вычислений на одном процессоре редукция фактически не применяется, так как решение одномерных систем осуществляется методом прогонки на всех этапах алгоритма.

Результаты расчетов по предложенному алгоритму на различных сетках и на разном числе процессоров вычислительной системы RM-600-E30 приведены в табл. 1.

Из полученных результатов можно сделать следующие выводы.

1. Обмены между процессорами увеличивают время решения задачи, что особенно заметно на сетке с малым числом узлов.

2. С увеличением числа процессоров эффективность распараллеливания, как и ожидалось, падает, что объясняется затратами на организацию параллельных вычислений. Однако при большом количестве процессоров достижение коэффициента $Q > 0,8$ следует считать достаточно высоким показателем.

Результаты численных экспериментов на МВС-1000 по решению той же задачи приведены в табл. 2. Как следует из табл. 2, показатели эффективности уже не так монотонны, как в табл. 1. Это факт, видимо, объясняется большим объемом кэш-памяти МВС-1000 и в связи с этим увеличением ее роли в расчетах на редкой сетке.

В целом для задач с достаточно большим числом узлов при разном количестве процессоров коэффициент эффективности распараллеливания $Q \approx 0,6$, что можно считать вполне приемлемым.

Отметим, что скорость сходимости внешних итераций можно значительно повысить, например, сопряженными градиентами [5]. В табл. 3 приведены коэффициенты эффективности распараллеливания с ускорением сопряженными градиентами на сетке $16 \times 16 \times 16$. Ускорение сопряженными градиентами не должно существенно влиять на эффективность распараллеливания, поскольку необходимые дополни-

Таблица 3
Ускорение сопряженными градиентами (МВС-1000)

L	P	Q
2	1,75	0,88
4	2,67	0,67
8	3,66	0,46

тельные вычисления составляют небольшую долю от трудоемкости одной итерации и легко распараллеливаются, что и подтверждается данной таблицей.

СПИСОК ЛИТЕРАТУРЫ

1. Ильин В. П., Свешников В. М. Экспериментальные оценки эффективности распараллеливания некоторых итерационных методов // Тр. ИВМиМГ СО РАН. Сер. Вычислительная математика. 1998. Вып. 6. С. 58.
2. Ильин В. П., Свешников В. М. Оценки эффективности распараллеливания алгоритмов декомпозиции областей // Автометрия. 2002. № 1. С. 31.
3. Ильин В. П. Методы конечных разностей и конечных объемов. Новосибирск: Изд-во ИМ СО РАН, 2000.
4. Ильин В. П. Параллельные неявные методы переменных направлений // Журн. вычисл. математики и мат. физики. 1997. 37, № 8. С. 899.
5. P'in V. P., Itskovich E. A. Two explicit incomplete factorization methods // Bull. of the Novosibirsk Computing Center. Ser. Numerical Analysis. Novosibirsk: ICMMG, 2002. Issue 11. P. 51.

*Институт вычислительной математики
и математической геофизики СО РАН,
E-mail: ilin@comcen.nsk.su*

*Поступила в редакцию
27 января 2003 г.*