

УДК 519.63 + 519.64 + 621.318

В. П. Ильин, В. М. Свешников*(Новосибирск)***ОЦЕНКИ ЭФФЕКТИВНОСТИ РАСПАРАЛЛЕЛИВАНИЯ
АЛГОРИТМОВ ДЕКОМПОЗИЦИИ ОБЛАСТЕЙ***

Рассмотрены алгоритмы декомпозиции областей с применением алгоритмов Якоби и Писмана – Речфорда. Приведены оценки их вычислительных сложностей и критериев эффективности распараллеливания. Даны результаты численных экспериментов на вычислительных системах МВС-100 (НИИ «Квант», Москва), Power Explorer фирмы "Parsytec" и RM-600-E30 ("Siemens – Nixdorf") для разных сеточных областей и процессорных топологий. Проведены их сравнительный анализ и обсуждение.

Введение. Алгоритмы декомпозиции областей являются одним из главных средств распараллеливания при решении многомерных краевых задач сеточными методами конечных разностей, конечных элементов или конечных объемов (см., например, работу [1] и цитируемую в ней литературу). Их сущность заключается в определении сеточных подобластей и в организации двухуровневого итерационного процесса: внутренние итерации проводятся для решения подзадач в подобластях, а внешние – для полного решения сеточной алгебраической системы. При естественном отображении топологии подобластей на топологию процессорной сети (одна подобласть – один процессор) такой подход эффективно распараллеливается, и для решения подзадач на соответствующих процессорах могут применяться быстрые решатели, например методы неполной факторизации [2], которые сами по себе распараллеливаются плохо. Однако введение двойных итераций неизбежно увеличивает общую вычислительную сложность получаемого алгоритма.

По-видимому, практически единственным исключением здесь являются блочные неявные методы переменных направлений, которые на основе декомпозиции областей распараллеливаются непосредственно, без искажения исходного алгоритма и, следовательно, без ухудшения присущей им высокой скорости сходимости итераций [1].

Цель настоящей работы – отображение неявного метода Писмана – Речфорда на архитектуру многопроцессорных вычислительных систем и экспериментальное исследование эффективности распараллеливания при использовании различных вариантов декомпозиции. В качестве критериев рас-

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (гранты № 99-01-00579 и № 99-07-90422).

считаются ускорение вычислений и коэффициент эффективности: $S_N = T_1/T_N$, $E_N = S_N/N$, где T_N – время решения задачи на N процессорах. Рассматриваются зависимости этих величин от числа процессоров, топологии подобластей и количества узлов сетки. Расчеты проводятся на модельной двумерной задаче Дирихле для уравнения Лапласа с простейшей пятиточечной аппроксимацией на квадратной сетке. Для сравнения аналогичные экспериментальные результаты приводятся для двойного итерационного процесса с явным методом Якоби (простой итерации), который идеально распараллеливается, но имеет очень низкую скорость сходимости итераций.

В разд. 1 приводится описание алгоритмов декомпозиции областей с применением алгоритмов Якоби и Писмана – Речфорда с оценками их вычислительных сложностей и критериев эффективности распараллеливания. В разд. 2 даны результаты численных экспериментов на вычислительных системах МВС-100 (НИИ «Квант», Москва), Power Explorer фирмы "Parsytec" и RM-600-E30 ("Siemens – Nixdorf") для разных сеточных областей и процессорных топологий. Проведены их сравнительный анализ и обсуждение.

1. Описание алгоритмов. Рассматривается задача отыскания решения системы пятиточечных сеточных линейных алгебраических уравнений

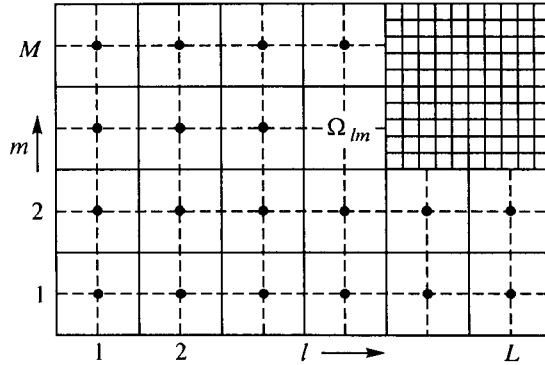
$$-a_{i,j}^1 u_{i-1,j} - a_{i,j}^2 u_{i,j-1} - a_{i,j}^3 u_{i+1,j} - a_{i,j}^4 u_{i,j+1} + a_{i,j}^0 u_{i,j} = f_{i,j},$$

$$i=1,2,\dots,I; \quad j=1,2,\dots,J, \quad (1)$$

где I, J – число узлов регулярной прямоугольной сетки по разным координатам; $a_{i,j}^k$, $k=0,1,\dots,4$; $f_{i,j}$ – известные величины; $u_{i,j}$ – искомое сеточное решение. Система (1) в матричном виде записывается как $Au = f$, где $A = \{a_{i,j}^k\}$ – квадратная матрица порядка IJ , которая возникает при аппроксимации двумерных краевых задач для эллиптических или параболических дифференциальных уравнений.

Пусть Ω_h – сеточная расчетная область, топологически представляющая собой прямоугольную конечную сеть, каждый из узлов которой с индексами (i, j) лежит на пересечении i -й вертикальной и j -й горизонтальной сеточных линий ($i=1,2,\dots,I$; $j=1,2,\dots,J$). Обозначим через $u = \{u_{i,j}\}$ вектор размерности IJ , компоненты которого суть значения некоторой сеточной функции в (i, j) -х узлах.

Предполагаем, что исходная расчетная сеточная область Ω_h разбивается на $N = L \times M$ прямоугольных подобластей $\Omega_{l,m}$, $l=1,\dots,L$; $m=1,\dots,M$, каждая из которых содержит pq узлов сетки с номерами $i=(l-1)p+1, (l-1)p+2, \dots, lp$; $j=(m-1)q+1, (m-1)q+2, \dots, mq$. Соответственно общее число узлов сетки есть $IJ = pLqM$, а «полный» вектор значений сеточной функции u можно разбить на подвекторы $\bar{u}_{l,m}$ из pq компонент каждый. Предполагается, естественно, что данные каждой (l, m) -й подобласти будут обрабатываться соответствующим процессором прямоугольной вычислительной сети. Схема декомпозиции расчетной области и соответствующей вычислительной сети приведена на рисунке. Сплошными линиями обозначены границы подобластей, штриховыми – межпроцессорные связи, символами \bullet – сами процессоры, а заштрихованный квадрат представляет собой фрагмент исходной расчетной сетки.



Исходная система уравнений (1) в блочном представлении имеет вид

$$-A_{l,m}^{(1)} \bar{u}_{l-1,m} - A_{l,m}^{(2)} \bar{u}_{l,m-1} - A_{l,m}^{(3)} \bar{u}_{l+1,m} - A_{l,m}^{(4)} \bar{u}_{l,m+1} + A_{l,m}^{(0)} \bar{u}_{l,m} = \bar{f}_{l,m}, \quad (2)$$

где матрицы $A_{l,m}^{(k)}$ ответственны за коммуникации каждой подобласти (и процессора) с одним из четырех своих соседей.

Блочный метод Якоби для решения уравнения (2) может быть записан в форме

$$A_{l,m}^{(0)} \bar{u}_{l,m}^n = A_{l,m}^{(1)} \bar{u}_{l-1,m}^{n-1} + A_{l,m}^{(2)} \bar{u}_{l,m-1}^{n-1} + A_{l,m}^{(3)} \bar{u}_{l+1,m}^{n-1} + A_{l,m}^{(4)} \bar{u}_{l,m+1}^{n-1} + \bar{f}_{l,m}, \quad (3)$$

где n – номер внешней итерации. В свою очередь, для каждой пары l, m (т. е. для соответствующей подобласти) система (2) решается с помощью внутренних итераций.

Для простоты рассмотрим поточечный метод Якоби, описываемый в аналогичной (3) скалярной форме:

$$u_{i,j}^{n,k} = (f_{i,j}^{n-1} + a_{i,j}^1 u_{i-1,j}^{n,k-1} + a_{i,j}^2 u_{i,j-1}^{n,k-1} + a_{i,j}^3 u_{i+1,j}^{n,k-1} + a_{i,j}^4 u_{i,j+1}^{n,k-1}) / a_{i,j}^0. \quad (4)$$

Здесь k – номер внутренней итерации, а начальное приближение $u_{i,j}^{n,0}$ выбирается естественным образом как конечный результат предыдущей $(n-1)$ -й внешней итерации. Отметим, что зависимость $f_{i,j}^{n-1}$ от n в (4) проистекает от фактического наличия в этом члене для околограничных узлов подобласти значений $u_{i\pm 1, j\pm 1}^{n-1}$, соответствующих узлам из соседних подобластей.

Внешние итерации (3) проводятся до получения решения с заданной точностью, а критерием их окончания может служить условие

$$\|r^n\| \equiv \|f - Au^n\| = \sum_{i,j} (r_{i,j}^n)^2 \leq \varepsilon \|r_{i,j}^0\|, \quad \varepsilon \ll 1, \quad (5)$$

где сумма берется по всем узлам сетки Ω_n . На каждой внешней итерации внутренние итерации по подобластям могут проводиться до выполнения

аналогичных (5) условий в каждой (l, m) -й подобласти:

$$\|r^{n,k}\|_{l,m} = \sum_{i,j \in \Omega_{l,m}} (r_{i,j}^{n,k})^2 \leq \varepsilon_n \|r^{n,0}\|_{l,m}, \quad l=1, \dots, L; \quad m=1, \dots, M, \quad (6)$$

где величины ε_n в общем случае зависят от n . Поскольку количества внутренних итераций k могут отличаться для разных подобластей, что приводит к простому некоторым процессорам, другим естественным подходом является проведение одинакового количества итераций k_n в каждой из подобластей $\Omega_{l,m}$.

Количество внешних итераций $n(\varepsilon)$, необходимых для обеспечения условия (5), оценивается величиной

$$n(\varepsilon) = C |\ln \varepsilon| / |\ln \rho| = \sum_{m=1}^n k_m, \quad (7)$$

где C – некоторая константа, а ρ (при точном обращении матриц $A_{l,m}^{(0)}$ в (3)) есть коэффициент подавления ошибки, или спектральный радиус матрицы $\bar{A}^{-1} (A - \bar{A}^{-1})$. Здесь $\bar{A} = \text{diag}\{\bar{A}_{l,m}^{(0)}\}$ представляет собой блочно-диагональную матрицу блочного порядка LM , а величина $\rho = 1 - O(h^2)$ (h – характерный шаг сетки) зависит от числа узлов и сеточных подобластей. При $\varepsilon_n \ll 1$ количество внутренних итераций k_n определяется по аналогичной (7) формуле, в которой надо заменить ε на ε_n , а ρ – на $\rho_h = 1 - O(h^2)$, т. е. коэффициент подавления ошибки внутреннего итерационного процесса. Как внутренние, так и внешние итерации можно ускорять с помощью методов сопряженных градиентов или чебышевских параметров, что на порядок сокращает число итераций, которое в данном случае будет $n(\varepsilon) \approx h^{-1}$. В целом вычислительная сложность алгоритма с двойными итерациями оценивается достаточно трудоемко, и его оптимизация наиболее полно проводится экспериментально.

Перейдем теперь к рассмотрению итерационного неявного метода Писмана – Речфорда, который формально представляет собой алгебраическую задачу, состоящую в вычислении последовательности векторов

$$\begin{aligned} u^{n-1/2} &= u^{n-1} - \omega(A_1 u^{n-1/2} + A_2 u^{n-1} - f), \\ u^n &= u^{n-1/2} - \omega(A_1 u^{n-1/2} + A_2 u^n - f), \end{aligned} \quad (8)$$

где $n=1, 2, \dots$; ω – числовой параметр; A_1, A_2 – трехдиагональные матрицы, определяемые соотношениями

$$\begin{aligned} A_1 + A_2 &= A, \quad a_{i,j}^{0,x} + a_{i,j}^{0,y} = a_{i,j}^0, \\ (A_1 u)_{i,j} &= -a_{i,j}^1 u_{i-1,j} + a_{i,j}^{0,x} u_{i,j} - a_{i,j}^3 u_{i+1,j}, \\ (A_2 u)_{i,j} &= -a_{i,j}^2 u_{i,j-1} + a_{i,j}^{0,y} u_{i,j} - a_{i,j}^4 u_{i,j+1}. \end{aligned} \quad (9)$$

Соотношения (8), (9) представляют собой векторную запись неявного метода переменных направлений Писмана – Речфорда, в котором каждая n -я итерация выполняется за два полушага. На первом полушаге обращается J одномерных систем I -го порядка:

$$-\omega a_{i,j}^1 u_{i-1,j}^{n-1/2} - \omega a_{i,j}^3 u_{i+1,j}^{n-1/2} + (1 + \omega a_{i,j}^{0,x}) u_{i,j}^{n-1/2} = f_{i,j}^x, \quad (10)$$

а на втором – I одномерных систем порядка J :

$$-\omega a_{i,j}^2 u_{i,j-1}^n - \omega a_{i,j}^4 u_{i,j+1}^n + (1 + \omega a_{i,j}^{0,y}) u_{i,j}^n = f_{i,j}^y. \quad (11)$$

Здесь правые части вычисляются как

$$f_{i,j}^x = \omega f_{i,j} + \omega a_{i,j}^2 u_{i,j-1}^{n-1} + \omega a_{i,j}^4 u_{i,j+1}^{n-1} + (1 - \omega a_{i,j}^{0,y}) u_{i,j}^{n-1}, \quad (12)$$

$$f_{i,j}^y = \omega f_{i,j} + \omega a_{i,j}^1 u_{i-1,j}^{n-1/2} + \omega a_{i,j}^3 u_{i+1,j}^{n-1/2} + (1 - \omega a_{i,j}^{0,x}) u_{i,j}^{n-1/2}.$$

Рассмотрим последовательность вычислений для первого полушага. Запишем (10) в блочном виде:

$$-A_l \bar{u}_{l-1} + B_l \bar{u}_l - C_l \bar{u}_{l+1} = \bar{f}_l, \quad l=1,2,\dots,L, \quad A_1 = C_L = 0, \quad (13)$$

где \bar{u}_l, \bar{f}_l – векторы размерности $p = I/L$, $\bar{u}_l = u_k^l, \bar{f}_l = f_k^l$; $u_k^l = u_{(l-1)p+k,j}$, $f_k^l = f_{(l-1)p+k,j}^x$, $k=1,2,\dots,p$; 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 (13) применяется блочный метод четно-нечетной редукции без обратного хода [1, 3]. Если $L = 2^R$ (а именно такой случай мы рассматриваем в настоящей работе), то выполняется R этапов редукции. На r -м этапе проводится переычисление a_l, c_l по формулам

$$a_l^{(r)} = (a_l \hat{b}_{pl}^{l-1} a_{l-1})^{(r-1)}, \quad c_l^{(r)} = (c_l \hat{b}_{1p}^{l-1} c_{l-1})^{(r-1)}, \quad (14)$$

элементов матрицы B_l по формулам

$$b_{11}^{(r)} = (b_{11} - a_l \hat{b}_{pp}^{l-1} c_{l-1})^{(r-1)}, \quad b_{pp}^{(r)} = (b_{pp} - c_l \hat{b}_{11}^{l+1} a_{l+1})^{(r-1)} \quad (15)$$

и, наконец, правой части по формулам

$$(f_1^l)^{(r)} = (f_1^l + a_l w_1^{l-1})^{(r-1)}, \quad (f_p^l)^{(r)} = (f_p^l + c_l w_p^{l+1})^{(r-1)}. \quad (16)$$

Здесь $b_{ts}^l, t, s=1,2,\dots,p$, – элементы матрицы B_l ; \hat{b}_{ts}^l – элементы обратной матрицы B_l^{-1} ; $w_1^l = \sum_{k=1}^p \hat{b}_{pk}^l f_k^l$; $w_p^l = \sum_{k=1}^p \hat{b}_{1k}^l f_k^l$.

Элементы матриц B_l, B_l^{-1} и компоненты правой части \bar{f}_l , не вошедшие в формулы (14)–(16), остаются неизменными.

На последнем R -м этапе редукции мы получим L независимых под-систем

$$B_l^{(R)} \bar{u}_l = \bar{f}_l^{(R)}, \quad l=1,2,\dots,L, \quad (17)$$

каждая из которых решается методом прогонки.

Запишем систему одномерных трехточечных уравнений в виде

$$-\alpha_i v_{i-1} + \beta_i v_i - \gamma_i v_{i+1} = \eta_i, \quad i=1,2,\dots,p; \quad \alpha_1 = \gamma_p = 0. \quad (18)$$

Метод прогонки для ее решения реализуется по формулам

$$\theta_i = (\beta_i - \alpha_i \delta_{i-1})^{-1}, \quad \delta_i = \gamma_i \theta_i, \quad z_i = (\eta_i + \alpha_i z_{i-1}) \theta_i, \quad i=1,2,\dots,p; \quad (19)$$

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

Для нахождения элементов обратной матрицы \hat{b}_{1p} , \hat{b}_{pp} решается система уравнений

$$B_l v = \hat{\xi}_p \quad (20)$$

с правой частью $\hat{\xi}_p = (0, 0, \dots, 0, 1)$ и при этом в (19) значительно сокращается объем арифметических операций, выполняемых фактически с помощью простых соотношений

$$z_1 = z_2 = \dots = z_{p-1} = 0, \quad z_p = \theta_p; \quad v_p = z_p; \quad (21)$$

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

Отыскание \hat{b}_{11} , \hat{b}_{p1} приводит к решению (20) с правой частью $\hat{\xi}_p = (1, 0, \dots, 0)$, что упрощает в (19) вычисление z_i :

$$z_1 = \theta_1, \quad z_i = \alpha_i z_{i-1} \theta_i, \quad i=1,2,\dots,p. \quad (22)$$

Решая (20) с правой частью \bar{f}_l , находим

$$w_1^l = v_p, \quad w_p^l = v_1. \quad (23)$$

Аналогично проводится реализация и второго полушага одной итерации метода Писмана – Речфорда: формируется M систем одномерных уравнений, которые решаются по изложенному выше алгоритму.

Метод Писмана – Речфорда (8), (9) обладает высокой скоростью сходимости итераций: при оптимальном постоянном итерационном параметре ω коэффициент подавления ошибки есть $\rho = 1 - 2h$, а при перестановочности матриц A_1, A_2 существует последовательность параметров ω_n , обеспечивающая число итераций $n(\epsilon) \sim |\ln h|^{-1}$. Если же A_1, A_2 не перестановочны, но симметричны и положительно определены, то к итерационному процессу можно применить ускорение сопряженными градиентами или дополнитель-

ными чебышевскими параметрами, когда число итераций оценивается величиной $n \sim h^{-1/2}$ [2].

2. Распараллеливание алгоритмов. Рассматриваемые алгоритмы реализованы на 8-процессорных вычислительных системах МВС-100, созданных в НИИ «Квант» (Москва), Power Explorer фирмы "Parsytec" и RM-600-E30 ("Siemens – Nixdorf").

Для оценки эффективности распараллеливания алгоритмов декомпозиции области необходимо учитывать следующую общую схему вычислительного процесса, включающую выполнение каждым процессором арифметических операций для одной или нескольких итераций в соответствующей подобласти и обмен данными между топологически соседними процессорами. При этом принимаем простейшую модель вычислений, предполагающую арифметическую часть процессорного времени T_a пропорциональной общему количеству N_a арифметических действий (независимо от их типа), а время одного коммуникационного сеанса T_c состоящим из времени инициализации обмена τ_0 плюс время самого обмена, пропорциональное числу передаваемых чисел: $T_a = \tau_a N_a$, $T_c = \tau_0 + \tau_c N_c$.

Измерения средних времен выполнения арифметической операции τ_a и передачи одного числа τ_c на рассматриваемых вычислительных системах показывают очевидное соотношение $\tau_c \gg \tau_a$, что определяет сокращение коммуникационных потерь как одну из главных проблем повышения эффективности распараллеливания.

Отметим, что независимо от методов решения подзадач в подобластях и организации внешних итераций между подобластями объемы вычислений в подобластях и обменов между ними определяются площадью и периметром одной подобласти:

$$Q_{p,q} \approx C_{p,q}^{(1)}(pq)^\gamma, \quad P_{p,q} \approx C_{p,q}^{(2)}(p+q), \quad (24)$$

где $\gamma \geq 1$ и $C_{p,q}^{(1)}$, $C_{p,q}^{(2)}$ – некоторые постоянные. Таким образом, для минимизации относительных коммуникационных потерь оптимальным является формирование квадратных сеточных подобластей, т. е. $p = q$, когда отношение площади к периметру является максимальным.

С точки зрения тонких особенностей отображения алгоритмов на архитектуру вычислительной системы интересными являются вопросы об использовании неоднородной оперативной памяти (при наличии в процессоре быстрой кэш-памяти) и о совмещении обменов с арифметическими вычислениями, но эти аспекты выходят за рамки данной работы.

Мы рассмотрим результаты численных экспериментов по распараллеливанию описанных алгоритмов на разностных сетках с числом узлов $I \times J = 32 \times 32, 64 \times 64, 128 \times 128, 256 \times 256$ для МВС-100 (результаты для Power Explorer практически совпадают и не приводятся) и на сетках $256 \times 256, 512 \times 512, 1024 \times 1024, 2048 \times 2048$ для RM-600. Сравнительные времена расчетов приводятся для одного процессора и для конфигураций процессоров $L \times M = 2 \times 1, 2 \times 2, 4 \times 1, 4 \times 2, 8 \times 1$.

В табл. 1, 2 приводятся значения T_N (в секундах), S_N и E_N , соответствующие выполнению 100 итераций метода Якоби на $N = L \times M$ процессорах для МВС-100 и RM-600. При этом обмены между процессорами проводились на каждой итерации, т. е. фактически алгоритм декомпозиции области осуществлялся без внутренних итераций.

Таблица 1

Простая итерация (MBC-100)

$L \times M$	Критерий	$I \times J$			
		32×32	64×64	128×128	256×256
1×1	T_1	0,36	1,45	5,84	23,70
2×1	T_N	0,28	0,90	3,27	12,73
	S_N	1,29	1,61	1,79	1,86
	E_N	0,64	0,81	0,89	0,93
2×2	T_N	0,34	0,63	1,83	6,56
	S_N	1,05	2,31	3,19	3,61
	E_N	0,26	0,58	0,80	0,90
4×1	T_N	0,26	0,60	1,84	6,62
	S_N	1,36	2,40	3,17	3,58
	E_N	0,34	0,60	0,79	0,89
4×2	T_N	0,63	0,68	1,36	3,67
	S_N	0,57	2,12	4,62	6,45
	E_N	0,07	0,26	0,58	0,81
8×1	T_N	0,27	0,45	1,12	3,60
	S_N	1,34	3,18	5,21	6,58
	E_N	0,17	0,40	0,65	0,82

Аналогичные данные для 100 итераций метода Писмана – Речфорда приводятся в табл. 3, 4.

Поскольку эти данные свидетельствуют только о скорости параллельного выполнения итераций, но ничего не говорят о быстродействии решения алгебраической системы в целом, в табл. 5 приводятся количества итераций и времена (первое и второе числа в каждой клетке), полученные при решении

Таблица 2

Простая итерация (RM-600)

$L \times M$	Критерий	$I \times J$			
		256×256	512×512	1024×1024	2048×2048
1×1	T_1	3,6	22,2	97,6	390,5
2×1	T_N	1,7	9,0	32,3	77,1
	S_N	2,1	2,4	3,0	5,0
	E_N	1,1	1,2	1,5	2,5
2×2	T_N	0,77	6,2	23,6	32,3
	S_N	4,5	3,6	4,2	12,0
	E_N	1,2	0,9	1,0	3,0
4×1	T_N	0,79	5,9	21,0	37,0
	S_N	4,5	3,3	4,6	10,6
	E_N	1,2	0,9	1,16	2,6

Таблица 3

Метод Писмана – Речфорда (MBC-100)

$L \times M$	Критерий	$I \times J$			
		32×32	64×64	128×128	256×256
1×1	T_1	1,25	5,63	23,4	106,4
2×1	T_N	0,83	3,65	14,40	63,30
	S_N	1,52	1,54	1,62	1,68
	E_N	0,76	0,77	0,81	0,84
2×2	T_N	1,05	2,61	9,30	36,06
	S_N	1,19	2,16	2,51	2,95
	E_N	0,30	0,54	0,63	0,74
4×1	T_N	1,24	2,86	9,69	38,4
	S_N	1,00	1,96	2,41	2,78
	E_N	0,25	0,50	0,60	0,69
4×2	T_N	2,20	2,93	6,85	21,99
	S_N	0,57	1,91	3,41	4,80
	E_N	0,07	0,24	0,43	0,60
8×1	T_N	2,68	3,49	7,62	24,70
	S_N	0,47	1,61	3,07	4,31
	E_N	0,06	0,20	0,38	0,54

систем уравнений методами Якоби и Писмана – Речфорда с точностью $\varepsilon = 10^{-3}$ на одном процессоре RM-600. Отметим, что в экспериментах с методом Писмана – Речфорда здесь использовались постоянные, но разные значения итерационных параметров ω (почти оптимальные величины 8,75, 17,5, 35, 70 для трех рассмотренных сеток соответственно).

Таблица 4

Метод Писмана – Речфорда (RM-600)

$L \times M$	Критерий	$I \times J$			
		256×256	512×512	1024×1024	2048×2048
1×1	T_1	22,7	109,2	615,0	3353,0
2×1	T_N	9,9	7,0	435,0	3048,0
	S_N	2,3	1,6	1,4	1,1
	E_N	1,1	0,8	0,7	0,5
2×2	T_N	5,5	33,3	177,0	1134,0
	S_N	4,1	3,3	3,5	2,8
	E_N	1,0	0,8	0,9	0,7
4×1	T_N	5,8	29,4	173,0	1065,0
	S_N	3,9	3,7	3,5	3,15
	E_N	1,0	0,9	0,9	0,8

Т а б л и ц а 5

**Количество итераций и времена счета при решении систем
методами Якоби и Писмана – Речфорда**

$L \times M$	Метод Якоби	Метод Писмана – Речфорда
64 × 64	6325; 14	47; 0,43
128 × 128	24921; 225	95; 3,52
256 × 256	98919; 4246	189; 29,2

Анализ результатов численных экспериментов позволяет делать следующие выводы.

1. Рост числа процессоров (и подобластей) для решения задачи на фиксированной сетке приводит к уменьшению показателя загрузки процессоров E_N : при восьми процессорах он падает до значений менее 0,1. Коэффициент ускорения S_N при этом ведет себя нелинейно: для малого числа узлов (сетки $32 \times 32, 64 \times 64$) он падает, достигая в случае восьми процессоров всего лишь значения единицы и даже менее, а для сеток $128 \times 128, 256 \times 256$ он растет. Таким образом, увеличение числа процессоров усложняет проблему эффективного распараллеливания, что естественно объясняется увеличением числа межпроцессорных обменов. Рост числа узлов разностной сетки увеличивает значения S_N, E_N при любой конфигурации процессоров. Коэффициент загрузки E_N растет с уменьшением числа процессоров, достигая значений, близких к единице. Повышение эффективности распараллеливания алгоритма декомпозиции области можно объяснить тем, что с увеличением размера (т. е. числа узлов) сеточной подобласти отношение числа ее околограничных узлов к числу внутренних уменьшается.

2. Выполнение метода Якоби на процессорах RM-600 имеет существенную особенность, заключающуюся в проявлении сверхускорения, т. е. в наличии значений E_N , больших единицы. Данный факт объясняется неоднородностью оперативной памяти, что приводит к эффективному использованию быстрой кэш-памяти, если количество узлов в одной подобласти (процессоре) уменьшается. Это говорит об актуальности развития средств MPI по управлению кэш-памятью, которые в данных экспериментах не использовались. Отметим еще тот факт, что в силу малости относительных коммуникационных потерь проведение двойных итераций в методе Якоби оказывается на рассматриваемых примерах нецелесообразным, так как они могут только замедлить общую скорость сходимости итерационного процесса.

3. Реализация метода Писмана – Речфорда на MVS-100 имеет в целом те же особенности, что и метод Якоби (см. п. 1). Сравнение табл. 1 и 3 показывает, что в неявном алгоритме коэффициент эффективности использования процессоров E_N оказывается меньше на 20–50 %, что объясняется наличием дополнительных обменов при выполнении метода циклической редукции.

4. При выполнении метода Писмана – Речфорда на RM-600 эффект сверхускорения отсутствует, но коэффициент E_N оказывается больше 0,5, т. е. достаточно высоким.

5. С учетом огромного преимущества метода Писмана – Речфорда в скорости сходимости итераций по сравнению с методом Якоби (см. табл. 5), алгоритм декомпозиции области на базе неявного метода представляется наи-

более перспективным для распараллеливания рассматриваемого класса задач. Здесь особенно важным является тот факт, что его многопроцессорная реализация осуществляется без какого-либо искажения алгоритма, могущего снизить скорость сходимости итераций.

СПИСОК ЛИТЕРАТУРЫ

1. **Ильин В. П.** Параллельные неявные методы переменных направлений // Журн. вычисл. математики и мат. физики. 1997. **37**, № 8. С. 899.
2. **Ильин В. П.** Методы неполной факторизации для решения алгебраических систем. М.: Наука, 1995.
3. **Ильин В. П., Кузнецов Ю. И.** Трехдиагональные матрицы и их приложения. М.: Наука, 1985.

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

*Поступила в редакцию
12 января 2001 г.*

Подписка на наш журнал – залог Вашего успеха!