

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ РЕШЕНИЯ МНОГОМЕРНЫХ КРАЕВЫХ ЗАДАЧ ДЛЯ ПАРАБОЛИЧЕСКИХ УРАВНЕНИЙ

Т. А. Кудряшова, С. В. Поляков¹

В данной работе рассматривается проблема решения с помощью многопроцессорных вычислительных систем (МВС) с распределенной памятью многомерных краевых задач для параболических уравнений в областях прямоугольной формы. Для проведения детального анализа, возникающих при этом трудностей в качестве модельной выбрана краевая задача для линейного трехмерного уравнения теплопроводности с переменными коэффициентами. Для ее решения предложено использовать явную схему и неявные схемы суммарной аппроксимации. На их основе разработан комплекс параллельных программ для решения модельной задачи. По результатам расчетов проведен сравнительный анализ решения задачи по явной схеме и по неявным схемам расщепления с использованием параллельного алгоритма прогонки. При создании параллельных программ рассмотрены различные варианты разбиения расчетной области, изучено влияние разбиения на итоговую эффективность параллельных вычислений. В результате проведенного анализа удалось получить информацию о способах повышения эффективности различных параллельных алгоритмов решения модельной задачи за счет оптимального выбора разностной схемы, алгоритма ее реализации и способа организации межпроцессорных связей (в частности, объема и топологии обменов между процессорами).

Постановка модельной краевой задачи

Рассмотрим проблему решения трехмерного линейного уравнения теплопроводности, которое в безразмерных переменных имеет вид

$$\frac{\partial U}{\partial t} = \sum_{\alpha=1}^3 \frac{\partial}{\partial x_{\alpha}} \left(k_{\alpha}(\vec{x}, t) \frac{\partial U}{\partial x_{\alpha}} \right) + f(\vec{x}, t), \quad \vec{x} \in D, \quad t > 0. \quad (1)$$

Здесь $U = U(\vec{x}, t)$ - температура, функция пространственных координат $\vec{x} = (x_1, x_2, x_3)$ и времени t , $k_{\alpha} = k_{\alpha}(\vec{x}, t) > 0$ - коэффициенты теплопроводности, $f(\vec{x}, t)$ - плотность мощности источников/стоков тепла,

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект No 02-07-90168

D – некоторая ограниченная прямоугольная область трехмерного пространства. Уравнение (1) получается, например, при определении баланса тепла для некоторого объема D , ограниченного поверхностью S , за время $\Delta t = t_2 - t_1$ [1]. В общем случае уравнение (1) описывает нестационарный процесс распространения тепла в объеме при действии внутри него неоднородных нестационарных источников и стоков.

Решение уравнения (1) в аналитической форме можно получить лишь в самых простых случаях (см. например, [2]). Поэтому его находят численно, что является самостоятельной достаточно сложной и актуальной проблемой, особенно в случаях, когда коэффициент теплопроводности разрывен или близок к нулю.

Уравнение (1) замыкается добавлением некоторых начальных и граничных условий. В качестве начального возьмем однородное распределение температуры в объеме

$$U(\vec{x}, 0) = U_0 = \text{const} > 0. \quad (2)$$

Граничные условия могут быть различны в зависимости от температурного режима на границах. Типичными для данной постановки задачи могут быть граничные условия 1-го, 2-го, 3-го рода или смешанные. Однако возможна постановка и специальных дополнительных условий (периодических или интегральных). Без ограничения общности дальнейших исследований рассмотрим условия 3-го рода, как наиболее естественные для задач теплопроводности:

$$(k(\vec{x}, t) \nabla U, \vec{\sigma}) = -\eta (U - U_0), \quad \vec{x} \in S. \quad (3)$$

Здесь $\vec{\sigma}$ - вектор внешней нормали к S , η - коэффициент теплопередачи через поверхность S .

При проведении дальнейшего анализа задачи (1)-(3) без ограничения общности можно считать, что область D есть трехмерный единичный куб, интервал по времени ограничен значением $T > 0$. В этих предположениях граничные условия (3) можно переписать в виде

$$k_\alpha \frac{\partial U}{\partial x_\alpha} = +\eta (U - U_0), \quad x_\alpha = 0, \quad k_\alpha \frac{\partial U}{\partial x_\alpha} = -\eta (U - U_0), \quad x_\alpha = 1, \quad \alpha = 1, 2, 3. \quad (4)$$

В тестовых расчетах в качестве коэффициентов теплопроводности и источника тепла использовались следующие функции

$$k_\alpha(\vec{x}, t) = k_{\alpha,0} (1 + \lambda e^{-s(\vec{x})} g_m(t)), \quad f(\vec{x}, t) = q_0 e^{-s(\vec{x})} g_m(t), \quad m = 1, 2, \quad (5)$$

$$\text{где } s(\vec{x}) = \sum_{\alpha=1}^3 (x_\alpha - x_{\alpha,0})^2 / a_0^2, \quad g_m(t) = \begin{cases} 1 - e^{-t/\tau_0}, & m = 1 \\ 0.5 - 0.5 \cos(t/\tau_0), & m = 2 \end{cases}.$$

Зависимость коэффициентов уравнения (1.1) от времени $g_m(t)$ была подобрана как для стационарного ($m = 1$), так и для псевдопериодического ($m = 2$) решения с целью исследования точности и эффективности обсуждаемых ниже численных методов, применяющихся в этих двух случаях.

Методы численного решения уравнения теплопроводности

Сформулированную выше задачу (1), (2), (4) можно решать различными способами. Рассмотрим возможные конечно-разностные методы ее решения. Наша цель – предложить экономичный прямой метод или группу методов для решения поставленной задачи на МВС с распределенной памятью.

Представим уравнение (1) в следующем виде

$$\frac{\partial U}{\partial t} = LU + f, \quad LU \equiv \sum_{\alpha=1}^3 L_{\alpha} U, \quad L_{\alpha} U \equiv \frac{\partial}{\partial x_{\alpha}} \left(k_{\alpha} \frac{\partial U}{\partial x_{\alpha}} \right). \quad (6)$$

В расчетной области D введем равномерную сетку $\Omega = \bar{\omega}_{x_1} \times \bar{\omega}_{x_2} \times \bar{\omega}_{x_3} \times \bar{\omega}_t$, состоящую из компонент $\bar{\omega}_{x_{\alpha}} = \{x_{\alpha} = x_{\alpha, i_{\alpha}} = i_{\alpha} h_{\alpha}, i_{\alpha} = 0, 1, \dots, N_{\alpha}, h_{\alpha} = 1/N_{\alpha}\}$ ($\alpha = 1, 2, 3$) и $\bar{\omega}_t = \{t = t_j = j\tau, j = 0, 1, \dots, N_t, \tau = T/N_t\}$. На этой сетке определим разностный аналог искомой функции U_h . Для нахождения функции U_h с помощью интегро-интерполяционного подхода [3] построим целое семейство разностных схем, называемое схемой с весами:

$$\frac{\hat{U}_h - U_h}{\tau} = \sigma \hat{\Lambda}_h \hat{U}_h + (1 - \sigma) \Lambda_h U_h. \quad (7_{\sigma})$$

Здесь использованы общепринятые обозначения для разностного пространственного оператора, верхнего и нижнего слоев по времени, а также веса схемы. Аппроксимация Λ_h пространственного оператора L

имеет вид $\Lambda_h y_h = \sum_{\alpha=1}^3 \Lambda_{h,\alpha} y_h$, $\Lambda_{h,\alpha} y_h = \left(\bar{k}_{\alpha} y_{h,x_{\alpha}} \right)_{\bar{x}_{\alpha}}$.

В семействе схем (7_σ) выделяют обычно явную ($\sigma = 0$) и неявную ($\sigma = 1$) разностные схемы, имеющие первый порядок точности по времени и второй - по пространству, а также симметричную схему Кранка-Николсона ($\sigma = 0.5$), имеющую второй порядок точности и по времени, и по пространству.

Явная схема (7₀) наиболее проста в реализации и экономична (то есть при ее использовании число арифметических действий Q необходимое для нахождения решения на очередном временном слое

пропорционально количеству узлов пространственной сетки $N = \prod_{\alpha=1}^3 N_{\alpha}$.

Наряду с этим, явная схема не является абсолютно устойчивой, и для проведения корректных вычислений необходимо соблюдать условие согласования шагов сетки [3]

$$\tau < 0.5\hbar^2 / (3k_0), \quad \hbar = \min_{\alpha=1,2,3} h_{\alpha}, \quad k_0 = \max_{x \in D, t \in [0, T], \alpha=1,2,3} k_{\alpha}(\bar{x}, t). \quad (8)$$

Неявная схема (7₁) является более трудоемкой в реализации и в общем случае не экономична. Однако она абсолютно устойчива, то есть вычисления по ней будут корректными при любых $\tau > 0$ и $h_{\alpha} > 0$. При нахождении решения на очередном временном слое в зависимости от выбранного метода (прямого или итерационного) и вида функции $k_{\alpha}(\bar{x}, t)$ может потребоваться от $O(N \log_2 N)$ до $O(N^{7/3})$ арифметических действий. Отметим также, что при выполнении условия (8) точность решений, полученных по явной и неявной схемам, практически совпадает.

Симметричная схема (7_{1/2}) по трудоемкости реализации близка к неявной, а по условию устойчивости аналогична явной схеме. Поэтому единственным ее преимуществом по сравнению с этими схемами является второй порядок точности по времени.

На практике предпочтение отдается либо явной схеме, как наиболее простой в реализации, либо экономичным неявным схемам, основанным на расщеплении пространственного оператора по координатам x_{α} . Примером экономичных неявных схем являются локально-одномерные схемы (ЛОС), предложенные А.А. Самарским [3]. Особенностью этих схем является свойство суммарной аппроксимации уравнения исходной задачи и абсолютная устойчивость.

Для решения сформулированной выше задачи предлагается использовать следующий вариант **неявной ЛОС**:

$$\frac{U_h^{j+\alpha/3} - U_h^{j+(\alpha-1)/3}}{\tau} = \Lambda_{h,\alpha}^{j+\alpha/3} U_h^{j+\alpha/3} + \frac{1}{3} f_h^{j+1}, \quad \alpha = 1, 2, 3. \quad (9)$$

Здесь, как принято в литературе, введены неполные индексные обозначения и промежуточные слои по времени $t_{j+\alpha/3} = t_j + \alpha\tau/3$. Схема (9) имеет первый порядок точности по времени и второй - по пространству. Она экономична и может быть реализована с помощью алгоритма прогонки на каждом промежуточном этапе по времени.

Если необходимо повысить точность вычислений по времени, сохранив при этом свойство экономичности, то можно использовать **неявные схемы двуциклического расщепления на основе схемы Кранка-Николсона** [4], которые также являются схемами суммарной

аппроксимации, но имеют второй порядок точности по времени. В нашем случае предлагается следующий вариант такой схемы:

$$\begin{aligned}
\frac{U_h^{j+1/8} - U_h^j}{\tau} &= \frac{1}{4} \Lambda_{h,1}^{j+1/2} (U_h^{j+1/8} + U_h^j), \\
\frac{U_h^{j+2/8} - U_h^{j+1/8}}{\tau} &= \frac{1}{4} \Lambda_{h,2}^{j+1/2} (U_h^{j+2/8} + U_h^{j+1/8}), \\
\frac{U_h^{j+3/8} - U_h^{j+2/8}}{\tau} &= \frac{1}{4} \Lambda_{h,3}^{j+1/2} (U_h^{j+3/8} + U_h^{j+2/8}), \\
\frac{U_h^{j+5/8} - U_h^{j+3/8}}{\tau} &= f_h^{j+1/2}, \\
\frac{U_h^{j+6/8} - U_h^{j+5/8}}{\tau} &= \frac{1}{4} \Lambda_{h,3}^{j+1/2} (U_h^{j+6/8} + U_h^{j+5/8}), \\
\frac{U_h^{j+7/8} - U_h^{j+6/8}}{\tau} &= \frac{1}{4} \Lambda_{h,2}^{j+1/2} (U_h^{j+7/8} + U_h^{j+6/8}), \\
\frac{U_h^{j+1} - U_h^{j+7/8}}{\tau} &= \frac{1}{4} \Lambda_{h,1}^{j+1/2} (U_h^{j+1} + U_h^{j+7/8}).
\end{aligned} \tag{10}$$

Здесь также введены промежуточные слои по времени: $t_{j+\beta/8} = t_j + \beta\tau/8$, $\beta = 1, \dots, 7$. Схема (10) абсолютно устойчива [4], экономична и реализуется также с помощью метода прогонки.

В дальнейшем исследовании, посвященном решению модельной задачи на МВС, мы будем рассматривать три схемы: явную (которую обозначим как (7₀)), неявную ЛОС (9) и схему двуциклического расщепления (10). Выбор этих схем обусловлен требованием экономичности вычислений, которое является основным при решении многомерных задач. Для неявной схемы (7₁) в случае переменных коэффициентов экономичный прямой или итерационный метод пока не найден. Поэтому, далее она не рассматривается.

Отметим далее, что для расчета стационарных решений наиболее предпочтительной является неявная ЛОС, в которой шаг по времени ограничен только условием асимптотической устойчивости ($\tau \sim \hbar$). Для расчета псевдопериодических или существенно нестационарных решений более подходит схема (10), как более точная и позволяющая также брать шаг $\tau \sim \hbar$. При этом условии схема (10) имеет точность $O(\hbar^2)$, а неявная ЛОС - $O(\hbar)$. Явная же схема наиболее проста в параллельной реализации и является в определенном смысле эталоном эффективности.

Параллельные алгоритмы решения уравнения теплопроводности

Для реализации выбранных разностных схем на МВС необходимо применить один из подходов распараллеливания соответствующего «скалярного» (однопроцессорного) численного алгоритма. Наиболее эффективным в настоящее время для многомерных задач механики сплошной среды и МВС с распределенной памятью считается принцип геометрического параллелизма, который предполагает проведение декомпозиции расчетной области на равные (по числу узлов сетки) подобласти соответственно числу процессоров.

В нашем случае, учитывая геометрические свойства исходной расчетной области (а именно, ее прямоугольность), можно применить декомпозицию трех типов: линейную, квадратную и кубическую. Первая предполагает разбиение области по одной из трех координат, вторая – по двум координатам, третья – по трем (примеры различных типов разбиения показаны на рис. 1). В соответствии с типами декомпозиции возникают три различных алгоритма распараллеливания, а также три типа топологии межпроцессорных связей, которые условно называют «линейка», «квадратная решетка» и «кубическая решетка» соответственно типу декомпозиции.

Поясним сказанное на примере *параллельной реализации явной схемы* (1.7₀). При линейном разбиении области по одной из координат (например, по координате x_2 , как показано на рис. 1а) k -ый процессор вычисляет решение на верхнем слое по времени в диапазоне узлов

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = 0, \dots, N_1, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = 0, \dots, N_3\} \equiv I_1 \times I_2^{(k)} \times I_3.$$

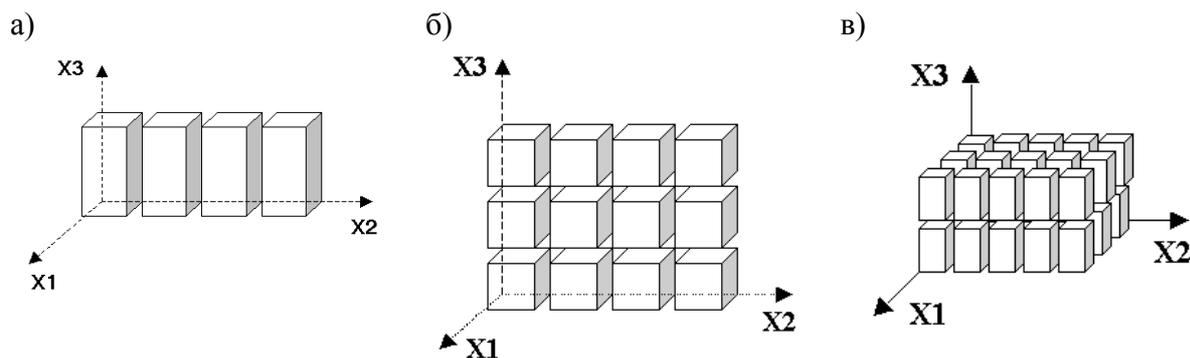


Рис. 1. Разбиение расчетной области по одной (а), двум (б) и трем координатам (в).

Для получения равномерной загрузки процессоров количества точек $N^{(k)} = N_1(i_{21}^{(k)} - i_{20}^{(k)} + 1)N_3$ в диапазонах $I^{(k)}$ должны быть примерно одинаковы. Расчетные формулы для вычисления решения на верхнем слое выглядят следующим образом:

$$U_{h,i_1,i_2,i_3}^{j+1} = U_{h,i_1,i_2,i_3}^j + \tau \left(\Lambda_{h,i_1,i_2,i_3}^j U_{h,i_1,i_2,i_3}^j + f_{h,i_1,i_2,i_3}^j \right), \quad (i_1, i_2, i_3) \in I^{(k)}. \quad (11)$$

Топология межпроцессорных связей в этом случае будет линейной, поскольку в формуле (11) для процессора k участвуют значения функции U_h в точках $(i_1, i_{20}^{(k)} - 1, i_3) \equiv (i_1, i_{21}^{(k-1)}, i_3)$ и $(i_1, i_{21}^{(k)} + 1, i_3) \equiv (i_1, i_{20}^{(k+1)}, i_3)$, лежащие в области данных процессоров $(k-1)$ и $(k+1)$. Перед началом очередного шага вычислений по формулам (11) на процессоре k необходимо получить требующиеся значения U_h от соседних процессоров. Для этого лучше всего произвести один групповой обмен данными для всех индексов i_1, i_3 . Этот обмен легко организуется в рамках линейной топологии. Количество передаваемых данных одним процессором в общем случае равно $M^{(k)} = 2N_1N_3$ и не зависит от числа процессоров. В случае равного количества узлов сетки по различным направлениям ($N_1 = N_2 = N_3 = \sqrt[3]{N}$) величина $M^{(k)} = 2N^{2/3}$.

В случае "квадратного" разбиения области, пример которого показан на рис. 1б, формулы (11) остаются в силе, но изменяются диапазон индексов $I^{(k)}$ и количество точек $N^{(k)}$, обрабатываемых процессором k :

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = 0, \dots, N_1, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = i_{30}^{(k)}, \dots, i_{31}^{(k)}\} \equiv I_1 \times I_2^{(k)} \times I_3^{(k)},$$

$$N^{(k)} = N_1 (i_{21}^{(k)} - i_{20}^{(k)} + 1) (i_{31}^{(k)} - i_{30}^{(k)} + 1).$$

Соответственно изменяется и топология обменов и межпроцессорных связей, которые образуют теперь решетку. Для определения соседей процессора k , с которыми ему придется обмениваться данными, можно ввести двойную нумерацию процессоров и установить взаимно однозначное соответствие исходных и двойных процессорных номеров. В данном случае можно использовать следующие соотношения:

$$k = p_3 k_2 + k_3, \quad k_2 = 0, \dots, p_2 - 1, \quad k_3 = 0, \dots, p_3 - 1,$$

$$k_2 = \lfloor k / p_3 \rfloor, \quad k_3 = k \bmod p_3, \quad k = 0, \dots, p - 1, \quad p = p_2 p_3.$$

Здесь p - общее число процессоров, образующих решетку $p_2 \times p_3$. В этих обозначениях процессор $k \leftrightarrow (k_2, k_3)$ перед началом очередного шага вычислений будет обмениваться недостающими данными с четырьмя соседями $k^{left} \leftrightarrow (k_2 - 1, k_3)$, $k^{right} \leftrightarrow (k_2 + 1, k_3)$, $k^{down} \leftrightarrow (k_2, k_3 - 1)$, $k^{up} \leftrightarrow (k_2, k_3 + 1)$ в рамках топологии "решетка". Количество передаваемых данных $M^{(k)} = 2N_1(N_2 / p_2 + N_3 / p_3)$ теперь зависит от числа процессоров.

При равном числе узлов сетки по различным направлениям и квадратной решетке процессоров ($p_2 = p_3 = \sqrt{p}$) величина $M^{(k)} = 4N^{2/3}p^{-1/3}$.

В случае "кубического" разбиения области (см. пример на рис. 1в) также получим расчетные формулы (11), в которых

$$I^{(k)} = \{(i_1, i_2, i_3), i_1 = i_{10}^{(k)}, \dots, i_{11}^{(k)}, i_2 = i_{20}^{(k)}, \dots, i_{21}^{(k)}, i_3 = i_{30}^{(k)}, \dots, i_{31}^{(k)}\} \equiv I_1^{(k)} \times I_2^{(k)} \times I_3^{(k)},$$

$$N^{(k)} = (i_{11}^{(k)} - i_{10}^{(k)} + 1)(i_{21}^{(k)} - i_{20}^{(k)} + 1)(i_{31}^{(k)} - i_{30}^{(k)} + 1).$$

Для определения соседей и обмена с ними данными вводится новая тройная нумерация процессоров:

$$k = p_2 p_3 k_1 + p_3 k_2 + k_3, \quad k_1 = 0, \dots, p_1 - 1, \quad k_2 = 0, \dots, p_2 - 1,$$

$$k_3 = 0, \dots, p_3 - 1, \quad k_1 = [k / (p_2 p_3)], \quad k_2 = [(k - p_2 p_3 k_1) / p_3],$$

$$k_3 = (k - p_2 p_3 k_1) \bmod p_3, \quad k = 0, \dots, p - 1, \quad p = p_1 p_2 p_3.$$

В соответствии с ней каждый процессор в кубической решетке имеет в общем случае 6 соседей, с которыми и обменивается необходимыми данными в начале каждого шага вычислений. Количество передаваемых данных равно $M^{(k)} = 2N_1 N_2 / (p_1 p_2) + 2N_1 N_3 / (p_1 p_3) + 2N_2 N_3 / (p_2 p_3)$. При равном числе узлов сетки по различным направлениям и равномерной кубической решетке процессоров ($p_1 = p_2 = p_3 = \sqrt[3]{p}$) $M^{(k)} = 6N^{2/3}p^{-2/3}$.

Теоретическая эффективность (не учитывающая время обменов) рассмотренных трех алгоритмов одинакова и равна 100%. На практике реальная эффективность алгоритма зависит от схемы обменов и числа процессоров. При прочих равных условиях использование линейной топологии при большом числе процессоров будет менее эффективно, чем использование топологий типа «решетка», поскольку у последних размер передаваемых данных снижается с ростом числа процессоров.

Рассмотрим теперь *параллельную реализацию неявной ЛОС* (9). Для этого тоже можно использовать три типа разбиения расчетной области и три типа топологии межпроцессорных связей.

При линейном разбиении области, например, по координате x_2 , получается следующий параллельный алгоритм. Для вычисления решения на новом временном слое каждый процессор последовательно решает три одномерные задачи:

$$(E - \tau \Lambda_{h,\alpha}^{j+\alpha/3}) U_h^{j+\alpha/3} = U_h^{j+(\alpha-1)/3} + \frac{\tau}{3} f_h^{j+1}, \quad \alpha = 1, 2, 3, \quad (12)$$

в своем диапазоне индексов $I^{(k)}$. При этом оказывается, что первый и третий этапы ЛОС можно реализовать независимо от данных на других

процессорах и использовать для этого алгоритм скалярной прогонки. На втором этапе ЛОС данные оказываются распределенными между процессорами. Поэтому на этом этапе необходимо использовать какой-либо параллельный алгоритм. В качестве такового предлагается использовать алгоритм параллельной прогонки предложенный в [5] и детально рассмотренный нами в [6]. Он применяется по индексу i_2 для всех фиксированных значений i_1 и i_3 .

При использовании этого алгоритма возникают обмены данными, связанные с формированием массивов коэффициентов "коротких задач" (количество которых равно $N_1 N_3$), а также при передаче значений решения этих задач на все процессоры (см. [6]). В обоих случаях обмены производятся либо в рамках топологии "звезда" (обмены всех процессоров с процессором-мастером), либо в рамках топологии "клика" (обмены "каждый с каждым") (Заметим, что линейная топология при такой организации обменов фактически не используется, однако она лежит в основе разбиения области.). При использовании стандарта коммуникаций MPI лучше всего использовать функцию "коллективное суммирование" (MPI_Reduce или MPI_Allreduce) для сбора информации со всех процессоров и функцию "разбрызгивания" (MPI_Broadcast) для распределения информации по всем процессорам. Количество передаваемых данных с одного процессора в данном случае равно $M^{(k)} = O(N_1 N_3 p)$ и растет с увеличением числа процессоров. При равном числе узлов сетки по различным направлениям $M^{(k)} = O(N^{2/3} p)$.

В случае "квадратного" разбиения области, например, по координатам x_2, x_3 , первый этап схемы реализуется с помощью скалярной прогонки, второй и третий – с помощью параллельной. Обмены данными между процессорами на каждом параллельном этапе организуются описанным выше способом. Общее количество передаваемых данных (по двум параллельным этапам) равно $M^{(k)} = O(N_1 N_2 p_3 / p_2 + N_1 N_3 p_2 / p_3)$ или $M^{(k)} = O(N^{2/3})$ - при равном количестве узлов сетки по различным направлениям и квадратной решетке процессоров.

В случае "кубического" разбиения области все этапы схемы реализуются с помощью параллельного алгоритма прогонки. Обмены данными производятся с помощью коллективных процедур на каждом этапе. Общее количество передаваемых данных с одного процессора равно $M^{(k)} = O(N_1 N_2 p_3 / (p_1 p_2) + N_1 N_3 p_2 / (p_1 p_3) + N_2 N_3 p_1 / (p_2 p_3))$. При равном количестве узлов сетки и количестве процессоров по различным направлениям $M^{(k)} = O(N^{2/3} p^{-1/3})$ и убывает с увеличением p .

Оценим эффективность рассмотренных трех алгоритмов. Для этого заметим, что в скалярном (однопроцессорном) варианте алгоритм расчета по неявной ЛОС на каждом временном шаге имеет арифметическую сложность $Q_1 = (C_1 + 3C_2)N$. Здесь C_1 - количество арифметических операций, связанное с вычислением коэффициентов уравнений в одном узле пространственной сетки, C_2 - количество операций, связанное с вычислениями по алгоритму прогонки на каждом этапе схемы, приходящееся на один узел сетки. На практике обычно выполняется соотношение $C_1 \gg C_2$. При параллельной реализации на системе с p процессорами без учета операций обмена данными между процессорами получаем следующие оценки для арифметической сложности $Q_p^{(\alpha)}$ ($\alpha = 1, 2, 3$) рассмотренных выше алгоритмов:

$$Q_p^{(1)} = (C_1 + 2C_2) \frac{N}{p} + C_2 N_1 N_3 \left(3 \frac{N_2}{p} + 2p \right) = \frac{N}{p} \left(C_1 + 5C_2 + 2C_2 \frac{p^2}{N_2} \right),$$

$$Q_p^{(2)} = \frac{N}{p} \left(C_1 + 7C_2 + 2C_2 \frac{p_2^2}{N_2} + 2C_2 \frac{p_3^2}{N_3} \right),$$

$$Q_p^{(3)} = \frac{N}{p} \left(C_1 + 9C_2 + 2C_2 \frac{p_1^2}{N_1} + 2C_2 \frac{p_2^2}{N_2} + 2C_2 \frac{p_3^2}{N_3} \right).$$

Сравнивая теперь эти величины с Q_1 , в предположениях $C_1 \sim C_2$ (наихудшая ситуация) и $p_\beta^2 \ll \sqrt[3]{N_\beta}$ ($\beta = 1, 2, 3$) получаем оценки для теоретического ускорения $S_p^{(\alpha)} = Q_1 / Q_p^{(\alpha)}$ и теоретической эффективности $E_p^{(\alpha)} = (S_p^{(\alpha)} / p) \cdot 100\%$ приблизительно равные $4p/6$, $4p/8$, $4p/10$ и 67, 50, 40 % для $\alpha = 1, 2, 3$.

Рассмотрим далее кратко **параллельную реализацию схемы двуциклического расщепления** (10). В ее основе лежат те же принципы, что и в алгоритме для неявной ЛОС. Только теперь в скалярном и в параллельном случаях для поиска решения на новом временном слое мы решаем семь подзадач (семь этапов схемы (10)), шесть из которых решаются методом прогонки, а седьмой аналогичен этапу явной схемы:

$$\begin{aligned} \left(E - \frac{\tau}{4} \Lambda_{h,\alpha}^{j+1/2} \right) V_h^{j+\alpha/8} &= 2U_h^{j+(\alpha-1)/8}, U_h^{j+\alpha/8} = V_h^{j+\alpha/8} - U_h^{j+(\alpha-1)/8}, \quad \alpha = 1, 2, 3, \\ U_h^{j+5/8} &= U_h^{j+3/8} + \tau f_h^{j+1/2}, \\ \left(E - \frac{\tau}{4} \Lambda_{h,\alpha}^{j+1/2} \right) V_h^{j+\beta/8} &= 2U_h^{j+(\beta-1)/8}, U_h^{j+\beta/8} = V_h^{j+\beta/8} - U_h^{j+(\beta-1)/8}, \quad \beta = 9 - \alpha, \alpha = 3, 2, 1. \end{aligned} \quad (13)$$

В параллельной реализации формулы (13) относятся к соответствующему индексному пространству $I^{(k)}$. Обмены данными организуются также как и в схеме (9). Для оценки эффективности алгоритма заметим, что для данной схемы $Q_1 = (C_1 + 6C_2)N$. В соответствующих параллельных реализациях имеем

$$Q_p^{(1)} = (C_1 + 4C_2) \frac{N}{p} + 2C_2 N_1 N_3 \left(3 \frac{N_2}{p} + 2p \right) = \frac{N}{p} \left(C_1 + 10C_2 + 4C_2 \frac{p^2}{N_2} \right),$$

$$Q_p^{(2)} = \frac{N}{p} \left(C_1 + 14C_2 + 4C_2 \frac{p_2^2}{N_2} + 4C_2 \frac{p_3^2}{N_3} \right),$$

$$Q_p^{(3)} = \frac{N}{p} \left(C_1 + 18C_2 + 4C_2 \frac{p_1^2}{N_1} + 4C_2 \frac{p_2^2}{N_2} + 4C_2 \frac{p_3^2}{N_3} \right).$$

Сравнивая эти величины с Q_1 , в предположениях $C_1 \sim C_2$ и $p_\beta^2 \ll \sqrt[3]{N_\beta}$ ($\beta = 1, 2, 3$) получаем значения теоретических ускорения и эффективности приблизительно равные $7p/11$, $7p/15$, $7p/19$ и 64, 47, 37 % для $\alpha = 1, 2, 3$.

Результаты тестовых расчетов

Основные тестовые расчеты проводились по явной схеме (1.7₀) и неявной ЛОС (9) на сетках, указанных в Таблице 1. При этом рассматривались все три типа разбиения расчетной области. Расчеты проводились на однородной вычислительной системе МВС-1000М, имеющую 768 процессоров. Результаты расчетов представлены на рисунках 2-10. Проанализируем приведенные данные.

Таблица 1. Конфигурации расчетных сеток.

Номер сетки	1	2	3
Число узлов	100x100x100	200x200x200	300x300x300

Первая серия расчетов была проведена по **явной схеме** (рис. 2, 3) для трех обсуждавшихся выше типов разбиения области и трех сеток. Расчеты в целом выявили высокую эффективность распараллеливания. Вместе с тем выявлены следующие тенденции, которые необходимо учитывать при решении конкретных задач.

Во-первых, реальная эффективность распараллеливания тем выше, чем больше узлов расчетной сетки попадает на процессор, то есть для более подробной сетки загрузка процессоров получается более равномерной.

Во-вторых, эффективность при линейном разбиении ниже, чем на квадратной или кубической решетке, что определяется уменьшением накладных расходов при пересылках с ростом числа процессоров на конфигурациях типа «решетка». Преимущества кубической решетки по сравнению с квадратной проявляются лишь при умеренном числе процессоров. Это связано с тем, что количество обменов для кубической решетки больше, чем для квадратной, а также с тем, что при большом числе процессоров характеристики МВС, на которой проводились расчеты, заметно ухудшаются. Однако при очень больших размерах сетки ситуация должна измениться в пользу кубической решетки вследствие того, что уменьшение размера пересылаемых данных с ростом числа процессоров происходит для этой топологии существенно быстрее.

В-третьих, применение линейной топологии сильно ограничено числом точек по одному из направлений (на практике, можно брать число процессоров не более, чем $N_\alpha/2$). Более гибкими в этом смысле оказываются «квадратная» и «кубическая» решетки. Однако при их использовании следует следить за тем, чтобы величины N_α/p_α были примерно одинаковы. На сетке с одинаковым числом узлов по всем направлениям это требование можно интерпретировать как уменьшение коэффициента «неквадратности» решетки.

В-четвертых, при проведении массовых расчетов на сетке фиксированного размера следует определить оптимальную топологию и число процессоров, при которых задача решается максимально быстро и эффективно. Например, для сеток 2 (3) оптимальной конфигурацией оказалась квадратная решетка процессоров $14 \times 14 = 196$ ($19 \times 19 = 361$), дающая ускорение 167.6 (327.2) и эффективность 85.5 (90.6) %.

Вторая серия расчетов была проведена по *неявной ЛОС* (рис. 4-6). При их анализе можно сделать следующие выводы.

Во-первых, эффективность расчетов с ростом числа процессоров здесь падает быстрее, чем в случае явной схемы. Это связано с тем, что накладные расходы (отношение суммарного времени обменов данными к общему времени расчета в %), связанные с обменами данных здесь растут быстрее и могут быть близки к 100% при большом числе процессоров (см. рис. 4).

Во-вторых, ввиду более сложной и объемной процедуры обменов, преимущества кубической решетки по сравнению с «линейкой» и «квадратной» решеткой процессоров проявляются сразу во всем диапазоне процессорных конфигураций (от 1 до 500), несмотря на ухудшение характеристик МВС при большом числе процессоров.

В-третьих, при малом и умеренном числе процессоров (1-100) очень сильно проявляется фактор «неквadratности» решетки (см. рис. 4, пики на графиках накладных расходов), который следует учитывать в расчетах.

В-четвертых, по сравнению с явной схемой эффективность расчетов по неявной ЛОС оказывается в 2-3 раза ниже. Учитывая это, а также различие расчетных формул, получаем, что при одинаковых: размере пространственной сетки, шаге по времени и числе процессоров неявный алгоритм работает в 1.5-3 раза медленнее. Однако, если рассчитывается стационарное решение, то шаг в неявной схеме можно взять в N_α раз больший, что в десятки раз окупает ее более низкую эффективность распараллеливания.

При расчетах существенно нестационарных решений следует использовать схему двуциклического расщепления (10), где также можно увеличить шаг по времени на несколько порядков при той же точности вычислений, что и в явной схеме.

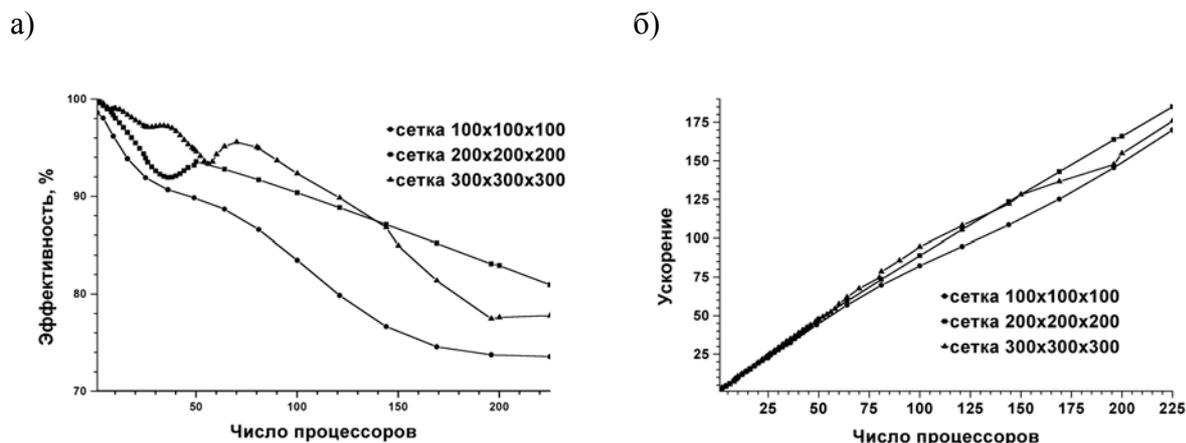


Рис. 2. Эффективность (а) и ускорение (б) в расчетах по явной схеме на различных сетках для топологии “решетка”.

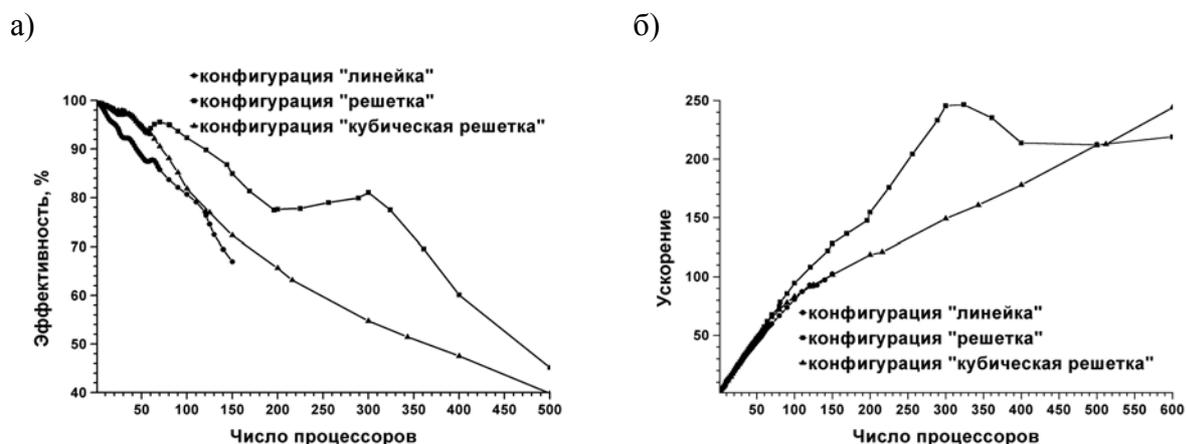


Рис. 3. Эффективность (а) и ускорение (б) на сетке 3 для различных Топологий обменов в расчетах по явной схеме

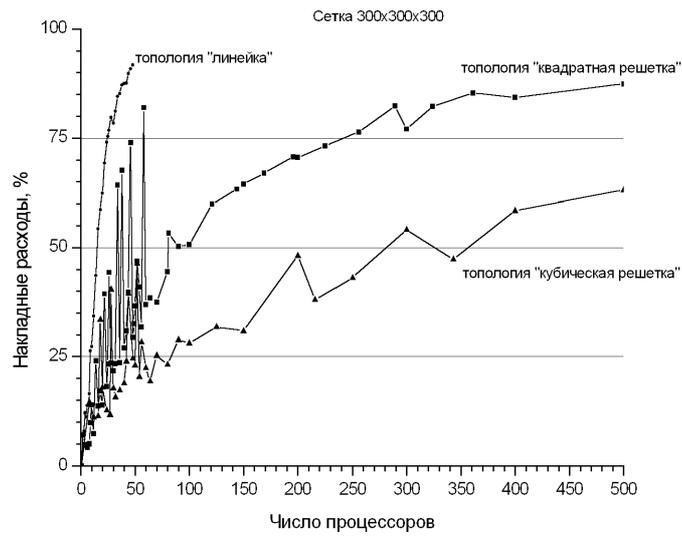


Рис. 4. Доля накладных расходов по обмену данными между процессорами во времени решения задачи.

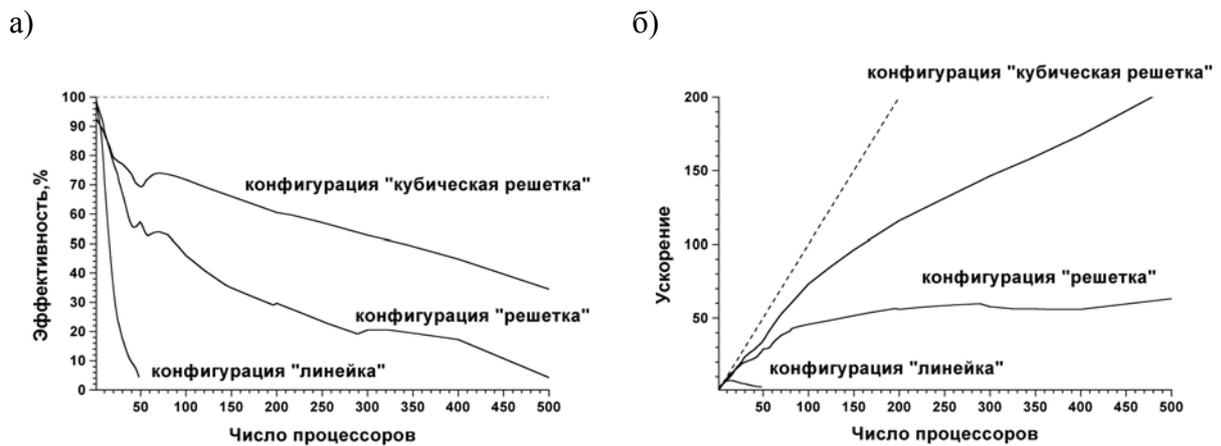


Рис. 5. Эффективность (а) и ускорение (а) на сетке 3 для различных топологий обменов в расчетах по неявной ЛОС.

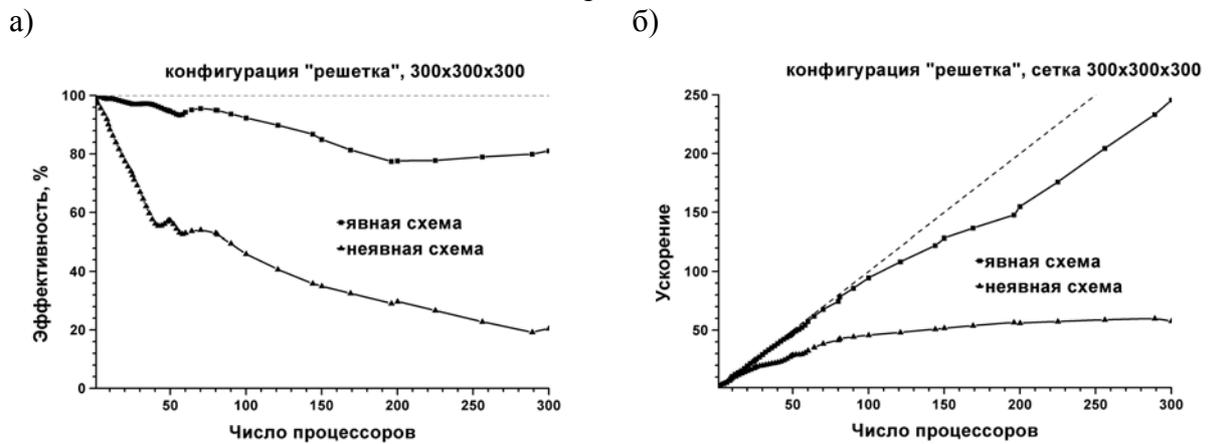


Рис. 6. Кривые эффективности (а) и ускорения (б) для явной и неявной схем.

Выводы

Суммируя приведенные в работе результаты, можно сделать следующие выводы.

При расчетах на МВС стационарных и медленно меняющихся со временем нестационарных решений краевых задач для параболических уравнений на грубых сетках следует использовать явные схемы. В этом случае условие устойчивости схемы не является обременительным, а эффективность ее параллельной реализации максимальна.

При расчетах на МВС стационарных и слабо нестационарных решений на подробной сетке достаточно использовать неявные ЛОС первого порядка точности по времени. При этом необходимо следить, чтобы выполнялось лишь условие их асимптотической устойчивости.

При расчетах на МВС существенно нестационарных решений следует использовать неявные схемы двуциклического расщепления или аналогичные им, имеющие второй порядок точности по времени. При этом шаг по времени должен определяться необходимой точностью решения и удовлетворять условию асимптотической устойчивости.

Во всех случаях выбор конкретного разбиения области, топологии и способа межпроцессорных обменов определяется соотношением размеров расчетной и «процессорной» сеток. При этом следует иметь в виду, что для фиксированной расчетной сетки всегда существуют оптимальные топология обменов и количество процессоров, при которых достигается скорейшее решение задачи в целом.

Литература

1. А.Н. Тихонов, А.А. Самарский. Уравнения математической физики. – М.: Изд-во МГУ, 1999. – 798 с.
2. А.Д. Полянин, А.В. Вязьмин, А.И. Журов, Д.А. Казенин. Справочник по точным решениям тепло- и массопереноса. – М.: Факториал, 1998. – 368 с.
3. А.А. Самарский, Теория разностных схем. – М.: Наука, 1989. – 616 с.
4. Г.И. Марчук. Методы расщепления. – М.: Наука, 1988. – 264 с.
5. Н.Н. Яненко, А.Н. Коновалов, А.Н. Бугров, Г.В. Шустов. Об организации параллельных вычислений и ”распараллеливании” прогонки. В сб. научных трудов ”Численные методы механики сплошной среды”. - Новосибирск, 1978, том 9, № 7, с.139-146.
6. Т.А. Кудряшова, С.В. Поляков. О некоторых методах решения краевых задач на многопроцессорных вычислительных системах. Труды четвертой международной конференции по математическому моделированию, 27 июня - 1 июля 2000 г., г. Москва (под ред. Л.А. Уваровой), том 2, с. 134-145. - М., Изд-во "СТАНКИН", 2001.

УДК 519.63. Параллельные алгоритмы решения многомерных краевых задач для параболических уравнений. Кудряшова Т.А., Поляков С.В. (Институт математического моделирования РАН).

Рассматривается проблема решения с помощью многопроцессорных вычислительных систем с распределенной памятью многомерных краевых задач для параболических уравнений в областях прямоугольной формы. Анализ проводится на примере краевой задачи для линейного трехмерного уравнения теплопроводности с переменными коэффициентами. Для ее решения предлагается использовать явную схему и неявные схемы суммарной аппроксимации. На их основе разработан комплекс параллельных программ для решения модельной задачи. По результатам расчетов проведен сравнительный анализ решения задачи по явной схеме и по неявным схемам расщепления с использованием параллельного алгоритма прогонки. При создании параллельных программ рассмотрены различные варианты разбиения расчетной области, изучено влияние разбиения на итоговую эффективность параллельных вычислений. В результате исследования предложены пути повышения эффективности параллельных алгоритмов решения модельной задачи за счет оптимального выбора разностной схемы, алгоритма ее реализации и способа организации межпроцессорных связей (в частности, объема и топологии обменов между процессорами).

Библ. 6 назв., 6 рис., 1 табл.