

УДК 519.63

Параллельные алгоритмы и технологии декомпозиции расчетной области для решения трехмерных краевых задач на квазиструктурированных сетках*

В.Д. Корнеев^{1,2}, В.М. Свешников^{1,2}

¹Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук, просп. Акад. М.А. Лаврентьева, 6, Новосибирск, 630090

²Новосибирский государственный университет, ул. Пирогова, 2, Новосибирск, 630090
E-mails: korneev@ssd.sccc.ru (Корнеев В.Д.), victor@lapasrv.sccc.ru (Свешников В.М.)

Корнеев В.Д., Свешников В.М. Параллельные алгоритмы и технологии декомпозиции расчетной области для решения трехмерных краевых задач на квазиструктурированных сетках // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. — Новосибирск, 2016. — Т. 19, № 2. — С. 183–194.

Предлагается новый подход к методу декомпозиции трехмерной расчетной области на подобласти, сопрягаемые без наложения, основу которого составляет прямая аппроксимация уравнения Пуанкаре–Стеклова на границе сопряжения. Излагаются параллельные алгоритмы и технологии решения трехмерных краевых задач на квазиструктурированных сетках при помощи данного подхода. Даются экспериментальные оценки эффективности распараллеливания на примере решения модельной задачи на квазиструктурированных параллелепипедальных согласованных и несогласованных сетках.

DOI: 10.15372/SJNM20160205

Ключевые слова: *краевые задачи, методы декомпозиции области, уравнение Пуанкаре–Стеклова, квазиструктурированные сетки, алгоритмы и технологии распараллеливания.*

Korneev V.D., Sveshnikov V.M. Parallel algorithms and domain decomposition technologies for solving three-dimensional boundary value problems on quasi-structured grids // Siberian J. Num. Math. / Sib. Branch of Russ. Acad. of Sci. — Novosibirsk, 2016. — Vol. 19, № 2. — P. 183–194.

A new approach to the decomposition method of a three-dimensional computational domain into subdomains, adjoint without overlapping, which is based on a direct approximation of the Poincaré–Steklov equation at the conjugation interface, is proposed. With the use of this approach, parallel algorithms and technologies for three-dimensional boundary value problems on quasi-structured grids are presented. The experimental evaluation of the parallelization efficiency on the solution of the model problem on quasi-structured parallelepiped coordinated and uncoordinated grids is given.

Keywords: *boundary value problems, domain decomposition methods, Poincaré–Steklov equation, quasi-structured grids, algorithms and technologies of parallelization.*

1. Введение

Основа предлагаемого в настоящей работе подхода к методу декомпозиции расчетной области на подобласти, сопрягаемые без наложения с условиями Дирихле–Дирихле,

*Работа выполнена при поддержке Российского научного фонда (проект № 14-11-00485) и РФФИ (проект № 16-01-00168).

при решении краевых задач на многопроцессорных супер ЭВМ — это прямая аппроксимация уравнения Пуанкаре–Стеклова на границе сопряжения (интерфейсе). Впервые данный подход для метода конечных элементов был предложен и рассмотрен в работах [1, 2], а в настоящей работе применяется конечно-разностный метод. Достоинством прямой аппроксимации является то, что в подобластях строятся автономные подсетки, а в них — автономная аппроксимация краевой задачи, исходя из ее физических особенностей. Подсетки в подобластях образуют квазиструктурированную сетку, которая, во-первых, имеет возможность локально регулировать плотность узлов и, во-вторых, проста в использовании. Решение краевых задач предлагаемым методом в двумерной постановке рассматривалось в работах [3–5]. Настоящая работа посвящена трехмерным задачам. Для их решения предлагается двухуровневый итерационный процесс по решению уравнения Пуанкаре–Стеклова на гранях интерфейса и исходного уравнения на ребрах интерфейса и в макроузлах, эффективность распараллеливания которого исследуется экспериментально на модельной задаче.

2. Постановка задачи и основы алгоритма ее решения

Пусть в замкнутой трехмерной области $\bar{G} = G \cup \Gamma$ с границей Γ требуется решить краевую задачу:

$$\Delta u = g_1, \quad lu|_{\Gamma} = g_2. \quad (1)$$

Здесь $u = u(T)$ — искомая функция, $g_1 = g_1(T)$, $g_2 = g_2(T)$ — заданные функции ($T = (x, y, z)$ — текущая точка, где x, y, z — декартовы координаты), Δ — оператор Лапласа, l — оператор граничных условий. Рассматриваются граничные условия Дирихле, Неймана, а также смешанные краевые условия. Предполагается, что граница Γ и функции g_1, g_2 таковы, что существует единственное решение задачи (1), обеспечивающее гладкость, достаточную для проведения дальнейших рассуждений.

Построим в расчетной области \bar{G} макросетку $\bar{\Omega}_H$ с шагами, намного превышающими максимальный шаг результирующей сетки, на которой ищется решение исходной задачи. Тем самым мы проведем декомпозицию \bar{G} на непересекающиеся подобласти \bar{G}_m , $m = \bar{1}, \bar{M}$, где M — известное целое число. Граница сопряжения подобластей γ , в свою очередь, разбивается на грани γ_f , ребра γ_e и макроузлы γ_m , являющиеся узлами макросетки $\bar{\Omega}_H$ так, что $\gamma = \gamma_f \cup \gamma_e \cup \gamma_m$. Для дальнейшего удобно ввести объединение $\gamma_{e,m} = \gamma_e \cup \gamma_m$ и область $G_0 = G/\gamma$.

На гранях γ_f справедливо уравнение Пуанкаре–Стеклова:

$$Fv = 0. \quad (2)$$

Здесь оператор F определяется как

$$Fv \equiv \left(\frac{\partial u(v)}{\partial \vec{n}} \right)_{\gamma_f}^{(+)} - \left(\frac{\partial u(v)}{\partial \vec{n}} \right)_{\gamma_f}^{(-)} = f(v), \quad (3)$$

где v — след функции u на γ (в том числе на γ_f). Уравнение (2) следует из условий сопряжения для функции u и ее нормальных производных. Отметим, что на решении краевой задачи на γ_f выполняется как уравнение Пуанкаре–Стеклова, так и уравнение Пуассона (1).

Исходную краевую задачу (1) переформулируем следующим образом: в замкнутой области \bar{G} требуется найти решение операторных уравнений:

$$Fv(T) = 0, \quad T \in \gamma_f, \quad (4)$$

$$\Delta v(T) = g_1(T), \quad T \in \gamma_{e,m}, \quad (5)$$

совместно с решением краевой задачи:

$$\Delta u(T) = g_1(T), \quad lu|_{\Gamma} = g_2, \quad u|_{\gamma} = v, \quad T \in G_0, \quad (6)$$

относительно функций u и v .

Решение задачи (4)–(6) будем проводить методом итераций по подобластям, состоящим из следующих этапов.

1. Задается начальное приближение $v_f^{(0)}$ на гранях γ_f .
2. Из уравнения (5) находят значения функции $v_{e,m}^{(n)}$ ($n = 0, 1, \dots$ — номер итерации) на ребрах и в макроузлах $\gamma_{e,m}$ (подробнее см. ниже).
3. Из решения краевой задачи (6) с граничными условиями Дирихле $v = v^{(n)} = v_f^{(n)} \cup v_{e,m}^{(n)}$ находят значения искомой функции на n -м приближении в подобластях.
4. Рассчитываются производные, входящие в выражение (3).
5. Делается очередной $(n + 1)$ -й шаг по решению уравнения Пуанкаре–Стеклова (4) и находят значения функции $v_f^{(n+1)}$ на гранях (подробнее см. ниже).
6. Если сходимость итерационного процесса достигнута, то находится окончательное решение на ребрах, в макроузлах и в подобластях, если же это не так, то повторяются этапы 2–5.

Построим в подобластях \bar{G}_m подсетки $\bar{\Omega}_{h,m}$. Объединение этих подсеток составляет квазиструктурированную сетку

$$\bar{\Omega}_h = \bigcup_{m=1}^M \bar{\Omega}_{h,m}.$$

На квазиструктурированной сетке $\bar{\Omega}_h$ краевая задача (6) методом конечных разностей, конечных элементов или конечных объемов заменяется приближенной задачей:

$$\Delta_h u_h = g_1, \quad l_h u_h|_{\Gamma} = g_2, \quad u_h|_{\gamma} = v_h, \quad (7)$$

где u_h, v_h — приближенные значения функций u, v , а Δ_h, l_h — аппроксимации оператора Лапласа и оператора граничных условий. Ее решение сводится к решению дискретных подзадач на подсетках $\bar{\Omega}_{h,m}$. Тем самым мы реализуем на каждой итерации по подобластям этап 3 описанного выше алгоритма. Основной проблемой, алгоритм решения которой приведен ниже, является вычисление v_h .

Отметим, что при известной функции v_h решение подзадач в подобластях может проводиться параллельно.

3. Решение подзадач на интерфейсе

Введем на гранях γ_f сетку ω_f , на ребрах γ_e — сетку ω_e . Для единообразия сетку из макроузлов обозначим как $\omega_m = \gamma_m$. В объединении $\gamma_{e,m}$ будем рассматривать сетку $\omega_{e,m} = \omega_e \cup \omega_m$.

3.1. Решение подзадач на гранях

Решение подзадач на гранях составляет этапы 4 и 5 описанного выше алгоритма. Рассмотрим разности

$$f_i = \left(\frac{\partial u}{\partial \vec{n}} \right)_i^{(+)} - \left(\frac{\partial u}{\partial \vec{n}} \right)_i^{(-)}, \quad i = \overline{1, N}, \quad (8)$$

в узлах сетки ω_f . Пусть $\vec{\eta}$ — внутренняя нормаль к γ_f в подобластях. Тогда соотношение (8) приобретет вид

$$f_i = \left(\frac{\partial u}{\partial \vec{n}} \right)_i^{(+)} + \left(\frac{\partial u}{\partial \vec{n}} \right)_i^{(-)}, \quad i = \overline{1, N}.$$

Заменим производные здесь конечно-разностными соотношениями:

$$\left(\frac{\partial u}{\partial \vec{n}} \right)_h \approx d_h u_h,$$

где d_h — линейный оператор, определяющий конкретную схему, которая по порядку точности должна быть согласована с порядком точности решения приближенных задач (7). При этом величины f_i заменяются их разностными аналогами $f_{h,i}$, которые определяются как

$$f_{h,i} = \left(d_h^{(+)} u_h^{(+)} \right)_i + \left(d_h^{(-)} u_h^{(-)} \right)_i, \quad i = \overline{1, N}, \quad (9)$$

где $d_h^{(+)}, d_h^{(-)} = d_h$. По аналогии с функциями f непрерывного аргумента величины $f_{h,i}$ являются функциями, зависящими от значений $v_{f,i}$ в узлах ω_f . Здесь и далее индекс h в сеточных функциях v для простоты обозначений будем опускать. В силу линейности исходной задачи, $f_{h,i}$ — линейные функции, поэтому их можно представить в виде линейной комбинации величин $v_{f,i}$, а именно

$$f_{h,i} = \sum_{j=1}^N a_{i,j} v_{f,j} + b_i, \quad (10)$$

где $a_{i,j}, b_i$ — неизвестные пока коэффициенты. Запишем (10) в матричном виде

$$f_h = A v_f + b, \quad (11)$$

где $A = \{a_{i,j}, i = \overline{1, N}, j = \overline{1, N}\}$ — квадратная матрица, а $f_h = \{f_{h,i}, i = \overline{1, N}\}$, $b = \{b_i, i = \overline{1, N}\}$, $v_f = \{v_{f,i}, i = \overline{1, N}\}$ — векторы. Для нахождения функции v_f потребуем, чтобы разность приближенных производных в узлах сетки ω_f обращалась в нуль, что дает систему линейных алгебраических уравнений:

$$A v_f + b = 0. \quad (12)$$

Элементы матрицы A и вектора b в формуле (11) неизвестны. Известно лишь действие оператора F_h , аппроксимирующего оператор F , на какую-либо функцию \tilde{v}_f , заданную в узлах ω_f . Это действие определяется формулой

$$F_h \tilde{v}_f = f_h = A \tilde{v}_f + b. \quad (13)$$

Вычисление компонент вектора b проводится по следующему алгоритму:

1) дадим \tilde{v}_f пробное значение $\tilde{v}_f = e^{(0)} = 0$ и решим краевую задачу

$$\Delta_h \psi_h^{(0)} = g_1, \quad l_h \psi_h^{(0)} \big|_{\Gamma} = g_2, \quad \psi_h^{(0)} \big|_{\gamma} = e^{(0)}, \quad (14)$$

относительно функции $\psi_h^{(0)}$;

2) рассчитаем компоненты $f_{h,i} = f_{h,i}^{(0)}$ вектора $f_h = f_h^{(0)}$ согласно формуле (11), в которой вместо v_f фигурирует $\psi_h^{(0)}$. Тогда из (13) следует, что

$$b = f_h^{(0)}.$$

Решение краевой задачи (14) сводится к решению подзадач вида (7).

Решение системы линейных алгебраических уравнений (12) будем проводить при помощи итерационного процесса по подобластям. Для построения начального приближения найдем грубое решение исходной краевой задачи (1) на макросетке $\bar{\Omega}_H$ из вспомогательной задачи:

$$\Delta_H v'_H = g_1(T_p), \quad l_H v'_H \big|_{\Gamma} = g_2(T_p), \quad (15)$$

где Δ_H, l_H — аппроксимации оператора Лапласа и оператора граничных условий на данной макросетке, v'_H — искомая сеточная функция, T_p — узлы макросетки $\bar{\Omega}_H$. Полученное решение v'_H интерполируется в узлы сетки ω_f , что дает функцию $v_f^{(0)}$, которая принимается за начальное приближение. Кроме того, рассчитывается вектор b по алгоритму, изложенному выше.

Окончательное решение системы (12) осуществляется при помощи итерационного процесса вида

$$v_f^{(n+1)} = \Lambda(v_f^{(n)}, Av_f^{(n)}), \quad n = 0, 1, \dots, \quad (16)$$

где Λ — функция, определяющая конкретный алгоритм с начальным приближением $v_f^{(0)}$. Смысл формулы (16) заключается в том, что в данном итерационном процессе используются лишь сам вектор $v_f^{(n)}$ и действие $Av_f^{(n)}$ матрицы на вектор (вместо $v_f^{(n)}$ может быть какой-либо другой вспомогательный вектор). Этим требованиям удовлетворяют итерационные методы в подпространствах Крылова [6]. Из (11) следует, что

$$Av_f^{(n)} = f_h^{(n)} - b,$$

а величину $f_h^{(n)}$ можно вычислить, решив подзадачи вида (7) в подобластях (поэтому (16) называют итерационным процессом по подобластям). Для этого требуются согласованные значения функции на ребрах и в макроузлах, вычисление которых приведено ниже. Подчеркнем, что для реализации итерационного процесса по решению уравнения Пуанкаре–Стеклова в данном случае нет необходимости вычислять элементы матрицы A .

Положительным свойством трудоемкой процедуры итераций по подобластям, на каждом шаге которого необходимо решать краевые подзадачи в подобластях, является то, что она допускает параллельную реализацию без дополнительных вычислительных затрат.

3.2. Решение подзадач на ребрах и в макроузлах

Решение подзадач на ребрах и в макроузлах составляет этап 2 описанного выше алгоритма и проводится на каждой n -й итерации по подобластям. К моменту проведения данных вычислений значения функции $v_f^{(n)}$ на гранях считаются известными.

На сетке $\omega_{e,m}$ с привлечением узлов сетки ω_f аппроксимируем исходное уравнение (5), в результате чего получим систему сеточных уравнений:

$$B\tilde{v} = f \quad (17)$$

с матрицей B относительно функции \tilde{v} , определенной на ребрах и в макроузлах, т. е. на сетке $\omega_{e,m}$.

Аппроксимация на ребрах строится по узлам сетки $\omega_{e,m}$ и ближайшим узлам сетки ω_f , а аппроксимация в макроузлах по ближайшим “реберным” узлам. Матрицу B можно представить в блочном виде

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

и, соответственно, векторы, входящие в (17), как

$$\hat{v} = \begin{bmatrix} \hat{v}_e \\ \hat{v}_m \end{bmatrix}, \quad f = \begin{bmatrix} f_e \\ f_m \end{bmatrix}.$$

Здесь блоки имеют следующие размерности:

$$B_{11} : N_e \times N_e, \quad B_{12} : N_e \times N_m, \quad B_{21} : N_m \times N_e, \quad B_{22} : N_m \times N_m,$$

где N_e — число “реберных” узлов ω_e , а N_m — число макроузлов ω_m .

Решение системы (17) будем проводить при помощи итерационного процесса, каждый шаг которого состоит из двух полушагов:

$$B_{11}\tilde{v}_e^{\nu+1/2} + B_{12}\tilde{v}_m^\nu = f_e, \quad B_{21}\tilde{v}_e^{\nu+1/2} + B_{22}\tilde{v}_m^{\nu+1} = f_m, \quad \tilde{v}_e^{\nu+1} = \tilde{v}_e^{\nu+1/2}, \quad (18)$$

где $\nu = 0, 1, \dots$ — номер итерации. На первом из них вычисляются значения на ребрах при фиксированных значениях в макроузлах, а на втором корректируются значения в макроузлах при фиксированных значениях на ребрах. На ребрах мы имеем “одномерные” сеточные уравнения, так как известные значения функции $v_f^{(n)}$ на гранях, входящие в аппроксимацию, исключаются из уравнений и переносятся в правую часть f_e .

Матрица B_{11} является блочно-диагональной: $B_{11} = \{B_{11}^{(i)} : N_{e,i} \times N_{e,i}\}$, где $i = \overline{1, N_E}$ — номер блока, совпадающий с номером ребра, N_E — количество ребер. Блоки $B_{11}^{(i)}$ не связаны друг с другом, поэтому первый наиболее трудоемкий полушаг (18) реализуется параллельно.

Второй полушаг, по сути дела, состоит в пересчете значений в отдельных макроузлах, которые также не связаны друг с другом, а связаны с полученными на первом полушаге значениями на ребрах.

4. Технологии распараллеливания

Распараллеливанию подлежит внешний итерационный процесс по подобластям (16), который занимает подавляющую часть времени решения всей задачи, и внутренний итерационный процесс (18) поиска решений на ребрах и в макроузлах. Эти итерационные процессы обладают внутренним естественным параллелизмом и не требуют дополнительных вычислительных затрат. Эффективность их распараллеливания целиком и полностью зависит от технологии проведения расчетов, от организации обменов между процессорами вычислительной сети.

Важным звеном в технологической цепи решений по достижению эффективности распараллеливания является отображение сеточных данных на вычислительную сеть. Обычно принятый способ отображения (одна подобласть — один процессор) в данном случае не эффективен, так как подобласти могут содержать различное число узлов, в которых вычисляются значения искомой функции (в дальнейшем — счетных узлов), что приводит к разбалансировке загрузки процессоров (здесь и далее под словом процессор понимается вычислительное ядро многоядерного компьютера). Поэтому подобласти группируются в объединения U_l , $l = \overline{1, L}$, где L — известное число, с целью обеспечения приблизительно равной загрузки процессоров. Для этого в каждое объединение $U_l = \bigcup_m \bar{\Omega}_{h,m}^{(l)}$ включаются такие подсетки $\bar{\Omega}_{h,m}^{(l)}$, которые давали бы в сумме $N_l = \sum_m N_m^{(l)}$ — число счетных узлов, $N_l \approx N_U$, где N_U — заданная величина, а $N_m^{(l)}$ — число счетных узлов в подобластях. Алгоритм группировки подобластей для идеальной балансировки должен строиться на решении задачи линейного программирования. При этом может оказаться, что в объединение включаются не только соседние подобласти, но и подобласти, разделенные подобластями из других объединений. Последнее обстоятельство может привести к значительному увеличению объема самых медленных операций, а именно операций пересылки в системах с разделенной памятью. Обмены не происходят между соседними подобластями одного объединения. В связи с этим был принят следующий алгоритм построения объединений. Определяется число N_U , которое может быть равно максимальному числу узлов в подобласти или вычисляться как $N_U = N_\Omega/L$, где N_Ω — число счетных узлов во всей квазиструктурированной сетке $\bar{\Omega}_h$. Для каждой свободной, т. е. не включенной ни в одно объединение, подобласти просматриваются свободные соседи. Количество узлов просмотренных подсеток суммируется. Если после просмотра m -й подсетки в l -м объединении окажется $N_l > N_U$, то процесс группировки заканчивается, причем при $N_l - N_U > 0.5 N_l^{(m)}$ последняя подсетка не включается в данное объединение.

Устанавливается отображение: одно объединение — один процессор. Какой-либо процессор (нулевой в нашем случае) объявляется управляющим. На управляющем процессоре проводятся следующие действия: 1) предварительная обработка информации; 2) формирование объединений; 3) рассылка информации на все процессоры, участвующие в вычислениях.

На каждый процессор, отвечающий за объединение, поступает информация двух типов: 1) общая для всех объединений; 2) специальная для рассматриваемого объединения. К первому типу относится следующая информация: 1) данные о квазиструктурированной сетке; 2) количество и номера подобластей в объединениях; 3) номера процессоров, отвечающих за каждую подобласть. Второй тип информации — это: 1) данные о подобластях и подсетках, входящих в объединение; 2) значения искомой функции и правой части в подобластях объединения; 3) данные на отрезке интерфейса, принадлежащего объединению. Принята следующая форма хранения информации на каждом процессоре. По возможности вводятся структурированные массивы (например, для хранения информации о квазиструктурированной сетке). Если такое невозможно, то данные хранятся в одномерных массивах, помимо того вводятся массивы опорных адресов, указывающие начало информации для каждой подобласти (например, хранение значений искомой функции в подобластях и на интерфейсе). Такой способ хранения позволяет, во-первых, экономить оперативную память для наиболее затратных вещественных массивов, которые описываются с двойной точностью, и, во-вторых, обрабатывать фрагмент массива для конкретной подобласти как структурированный массив, обращаясь к подпрограмме обработки с указанием опорного адреса.

На текущем процессоре выполняются следующие вычислительные работы: 1) вычисление искомой функции в подобластях; 2) расчет внутренних нормальных производных на гранях подобластей, входящих в объединение; 3) вычисление сумм внутренних нормальных производных на смежных сторонах подобластей; 4) реализация очередного шага итерационного процесса по подобластям (для этого проводятся обмены с управляющим процессором для вычисления итерационных параметров).

Рассмотрим технологию организации вычислений и обменов на примере вычисления сумм внутренних производных на гранях.

Производные, вычисленные на гранях подобластей, граничащих с подобластями других процессоров, необходимо передавать этим процессорам. Для повышения эффективности вначале производные вычисляются, а затем инициируется их передача на фоне расчета производных для граней смежных с подобластями, расположенными в своем процессоре. Процесс вычислений производных для граней подобластей состоит из следующих этапов.

1. Вычисляются производные верхних, дальних и правых граней для подобластей, граничащих с другими процессорами, и асинхронно передаются соответствующим процессорам.
2. Делается асинхронный прием производных от других процессоров со стороны нижней, ближней и левой граней соответственно. Асинхронный прием, на самом деле, это заказ на прием, а пока данные приходят, проводятся вычисления согласно этапам 3, 4.
3. Вычисляются производные верхних, дальних и правых граней, граничащих с подобластями собственного процессора.
4. Вычисляются производные для нижней, ближней и левой грани собственного процессора и суммируются с производными, вычисленными на предыдущем шаге с производными верхней, дальней и правой граней соответственно.
5. Осуществляется блокированная проверка завершенности асинхронных взаимодействий процессоров.
6. Принятые производные суммируются с производными соответствующих граней.

Рассмотренные алгоритмы и технологии реализованы в пакете прикладных программ на языке Фортран в системе МРІ. При помощи данного пакета были проведены численные эксперименты, результаты которых даны ниже.

5. Численные эксперименты

Цель проводимых численных экспериментов — исследование эффективности параллелизации решения трехмерных краевых задач на квазиструктурированных сетках.

Рассматривалась следующая модельная задача:

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

в единичном кубе \bar{R} . Проводилась декомпозиция расчетной области при помощи равномерной макросетки

$$\bar{\Omega}_H = \left\{ X_I = IH, Y_J = JH, Z_K = KH, I = \overline{0, N_H}, J = \overline{0, N_H}, K = \overline{0, N_H}, H = \frac{1}{N_H} \right\},$$

где N_H – заданное целое число.

В замкнутых макроэлементах $\bar{R}_m = \bar{R}_{I,J,K}$ строились равномерные параллелепипедальные подсетки

$$\bar{\Omega}_{h,m} = \{ x_{i_m} = X_I + i_m h_{x,m}, y_{j_m} = Y_J + j_m h_{y,m}, z_{k_m} = Z_K + k_m h_{z,m}, \\ i_m = \overline{0, n_{x,m}}, j_m = \overline{0, n_{y,m}}, k_m = \overline{0, n_{z,m}} \}$$

с шагами $h_{x,m} = \frac{X_{I+1} - X_I}{n_{x,m}}, h_{y,m} = \frac{Y_{J+1} - Y_J}{n_{y,m}}, h_{z,m} = \frac{Z_{K+1} - Z_K}{n_{z,m}}$, причем предполагалось без существенного ограничения общности, что $n_{x,k}, n_{y,k}, n_{z,k}$ – двойка в целой степени.

В качестве оператора Δ_h выбирались обычные семиточечные разностные операторы Лапласа. Оператором d_h , аппроксимирующим нормальные производные, служила трехточечная схема второго порядка, которая для соседних узлов, обозначенных цифрами 0, 1, 2, выглядит как

$$(d_h u_h)_0 = \frac{-3(u_h)_0 + 4(u_h)_1 - (u_h)_2}{2h}.$$

Расчеты проводились при различных параметрах сеток $\bar{\Omega}_{h,m}, \bar{\Omega}_H: n_{x,m}, n_{y,m}, n_{z,m} = 2^k, k = 3, 4, 5, 6, N_H = 4, 8$. Краевые подзадачи решались при помощи трехмерного аналога метода Писмана–Рэчфорда [6], а итерации по подобластям проводились при помощи метода сопряженных градиентов, легко поддающегося распараллеливанию. Рассматривалось два варианта квазиструктурированных сеток: согласованный и несогласованный. В согласованном варианте все подсетки равны, а в несогласованном варианте число интервалов смежных подсеток отличалось в 2^3 раз и они располагались в шахматном порядке, т. е. в формировании квазиструктурированной сетки участвовало два вида подсеток: одна редкая и одна густая. В несогласованном варианте подобласти с разными подсетками группируются в объединения, имеющие равное число узлов. Во всех экспериментах проводилось 50 внешних итераций по подобластям. Одинаковое число внешних итераций необходимо при исследовании ускорения на различных сетках. Вычислительный кластер Сибирского суперкомпьютерного центра, на котором проводились расчеты, является 8-ми ядерным. Численные эксперименты проводились на 1, 2, 4, 8, 16, 32 и 64-х процессорах.

В нижеследующих таблицах приведены значения ускорения вычислений $S_P = T_1/T_P$ при распараллеливании на различных сетках для согласованного и несогласованного вариантов при $N_H = 8$ (здесь T_1, T_P – время счета на одном и на P процессорах соответственно).

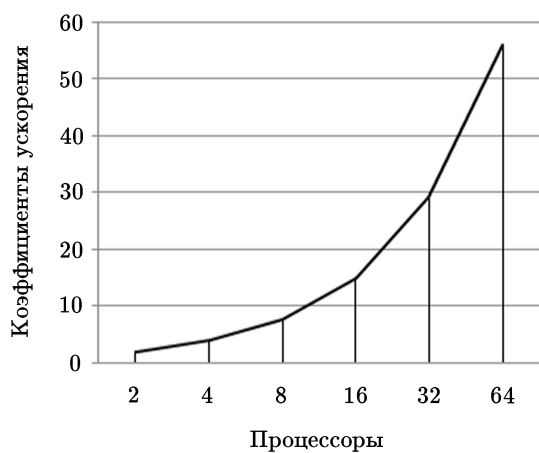
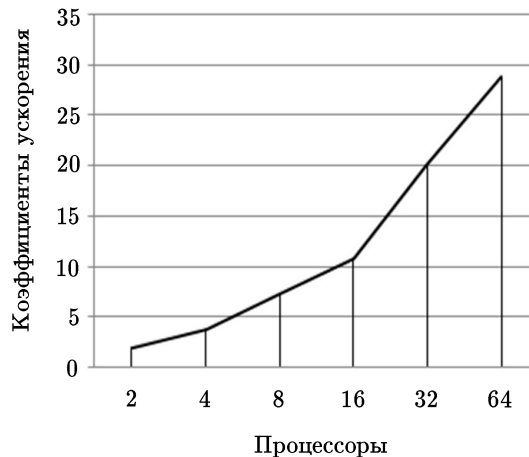
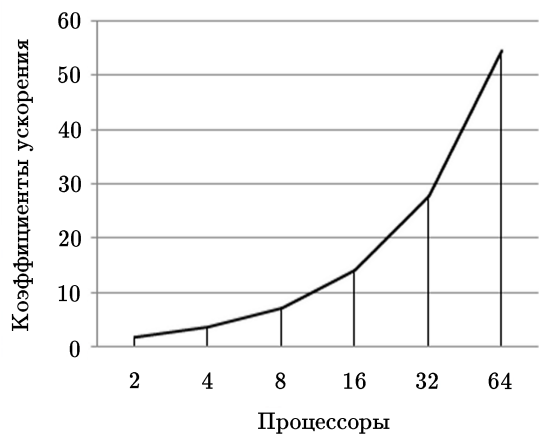
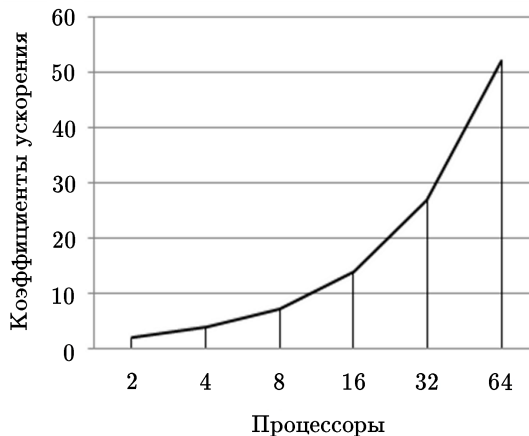
Таблица 1. Коэффициенты ускорения S_P на согласованных подсетках

Подсетки	P					
	2	4	8	16	32	64
32^3	1.984	3.945	7.769	14.94	29.24	56.06
16^3	1.945	3.868	7.685	12.55	24.33	43.74
8^3	1.967	3.779	7.331	10.85	20.08	28.77

Таблица 2. Коэффициенты ускорения S_P на несогласованных подсетках

Подсетки	P					
	2	4	8	16	32	64
32^3-16^3	1.821	3.596	7.132	14.14	27.86	54.53
16^3-8^3	1.896	3.696	7.191	13.91	26.94	52.19

Для наглядности на рисунках 1–4 приведены графики ускорения для согласованного и несогласованного вариантов и для различных подсеток (по горизонтальной оси используется логарифмическая шкала).

**Рис. 1.** Согласованные подсетки 32^3 **Рис. 2.** Согласованные подсетки 8^3 **Рис. 3.** Несогласованные подсетки 32^3-16^3 **Рис. 4.** Несогласованные подсетки 16^3-8^3

Из полученных результатов можно сделать вывод о том, что эффективность распараллеливания растет с увеличением числа счетных узлов в подобластях. Например, ускорение на 64 процессорах для подсетки 8^3 значительно меньше ускорения для подсетки 32^3 ($S_P = \{28, 56\}$ соответственно). Объясняется это тем, что при асинхронном

обмене с увеличением числа узлов растет величина τ_a/τ_c , где τ_a — время выполнения арифметических операций, а τ_c — время обменов. Отсюда следует, что не рекомендуется строить объединения квазиструктурированной сетки с малым числом узлов.

Заключение

В работе предложены параллельные алгоритмы и технологии решения трехмерных краевых задач методом декомпозиции расчетной области, основанные на прямой конечно-разностной аппроксимации уравнения Пуанкаре–Стеклова на интерфейсе. Решение ищется на адаптивных квазиструктурированных сетках, которые допускают локальную регулировку плотности узлов в зависимости от физических особенностей конкретной задачи и просты в использовании, при помощи двухуровневого итерационного процесса. Для качественного отображения расчетов на вычислительную сеть подобласти группируются в объединения, что способствует балансировке загрузки процессоров. Проведены численные эксперименты на согласованных и несогласованных сетках, демонстрирующие ускорение вычислений при распараллеливании.

Литература

1. **Agoshkov V., Ovtchinnikov E.** Projection decomposition method // Math. Models Methods Appl. Sci. — 1994. — Vol. 4. — P. 773–794.
2. **Gervasio P., Ovtchinnikov E., and Quarteroni A.** The spectral projection decomposition method for elliptic equations in two dimensions // SIAM J. in Numer. Analysis. — 1997. — Vol. 34, № 4. — P. 1616–1639.
3. **Свешников В.М.** Построение прямых и итерационных методов декомпозиции // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. — Новосибирск, 2009. — Т. 12, № 3. — С. 99–109.
4. **Свешников В.М., Беляев Д.О.** Построение квазиструктурированных локально-модифицированных сеток для решения задач сильноточной электроники // Вестник ЮУрГУ, серия ММП. — 2012. — № 40(299), вып. 14. — С. 130–140.
5. **Свешников В.М., Рыбдылов Б.Д.** О распараллеливании решения краевых задач на квазиструктурированных сетках // Вестник ЮУрГУ, серия ВМИ. — 2013. — Т. 2, № 3. — С. 63–72.
6. **Ильин В.П.** Методы конечных разностей и конечных объемов для эллиптических уравнений. — Новосибирск: Изд-во ИВМиМГ СО РАН, 2001.

Поступила в редакцию 8 апреля 2015 г.

Литература в транслитерации

1. **Agoshkov V., Ovtchinnikov E.** Projection decomposition method // Math. Models Methods Appl. Sci. — 1994. — Vol. 4. — P. 773–794.
2. **Gervasio P., Ovtchinnikov E., and Quarteroni A.** The spectral projection decomposition method for elliptic equations in two dimensions // SIAM J. in Numer. Analysis. — 1997. — Vol. 34, № 4. — P. 1616–1639.
3. **Sveshnikov V.M.** Postroenie pryamyh i iteratsionnyh metodov dekompozitsii // Sib. zhurn. vychisl. matematiki / RAN. Sib. otd-nie. — Novosibirsk, 2009. — T. 12, № 3. — S. 99–109.
4. **Sveshnikov V.M., Belyaev D.O.** Postroenie kvazistrukturirovannyh lokal'no-modifitsirovannyh setok dlya resheniya zadach sil'notochnoy elektroniki // Vestnik YuUrGU, seriya MMP. — 2012. — № 40(299), vyp. 14. — S. 130–140.

5. **Sveshnikov V.M., Rybdylov B.D.** O rasparallelivanii resheniya kraevykh zadach na kvazi-strukturirovannykh setkakh // Vestnik YuUrGU, seriya VMI. — 2013. — Т. 2, № 3. — S. 63–72.
6. **П'ин В.Р.** Metody konechnykh raznostey i konechnykh ob"emov dlya ellipticheskikh uravneniy. — Novosibirsk: Izd-vo IVMiMG SO RAN, 2001