

УДК 519.63

**УСКОРЕНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ ТРЕХМЕРНЫХ
КРАЕВЫХ ЗАДАЧ НА КВАЗИСТРУКТУРИРОВАННЫХ СЕТКАХ****И. А. Климонов¹, В. Д. Корнеев², В. М. Свешников³**

Статья посвящена ускорению параллельного решения трехмерных краевых задач методом декомпозиции расчетной области на подобласти, сопрягаемые без наложения. Декомпозиция проводится равномерной параллелепипедальной макросеткой. В каждой подобласти и на границе сопряжения (интерфейсе) строятся свои структурированные подсетки. Объединение этих подсеток образует квазиструктурированную сетку, на которой решается поставленная задача. Распараллеливание решения осуществляется при помощи MPI-технологий. Предложен и экспериментально исследован алгоритм ускорения внешнего итерационного процесса по подобластям для решения системы линейных алгебраических уравнений, аппроксимирующих уравнение Пуанкаре–Стеклова на интерфейсе. Проведены серии численных экспериментов на различных квазиструктурированных сетках и при различных параметрах вычислительных алгоритмов, показывающих ускорение вычислений.

Ключевые слова: краевые задачи, распараллеливание, квазиструктурированные сетки, итерационный процесс, начальное приближение.

1. Введение. В работах [1, 2] предложены и исследованы параллельные алгоритмы и технологии решения трехмерных краевых задач на квазиструктурированных параллелепипедальных сетках методом декомпозиции расчетной области на подобласти, сопрягаемые без наложения. Метод декомпозиции широко применяется при решении краевых задач на многопроцессорных суперЭВМ [3, 4]. В основе нашего подхода к методу декомпозиции лежат следующие принципы. Расчетная область погружается в параллелепипед, в котором строится параллелепипедальная структурированная макросетка. Тем самым проводится декомпозиция расчетной области на подобласти, каждая из которых является макроэлементом макросетки. В подобластях строятся параллелепипедальные структурированные подсетки. На границе сопряжения подобластей (интерфейсе), которая состоит из граней, ребер и макроузлов макросетки, строится своя подсетка. На подсетках в подобластях методом конечных разностей, конечных объемов или конечных элементов [5–7] аппроксимируются и решаются краевые подзадачи, на которые разбивается исходная краевая задача. Заметим сразу, что мы ограничиваемся случаем, когда подсетка на интерфейсе состоит из подмножества узлов подсеток в подобластях. Сшивка решений в подобластях осуществляется путем решения уравнения Пуанкаре–Стеклова на интерфейсе. При этом используется подход, в основе которого лежит прямая (непосредственная) аппроксимация уравнения Пуанкаре–Стеклова, разработанный авторами и изложенный в работах [1, 2]. Система конечно-разностных уравнений, которая получается в результате аппроксимации, решается каким-либо итерационным методом из семейства методов в пространствах Крылова [7].

Важным свойством нашего подхода, позволяющим сократить объем вычислений на каждой итерации, является то, что для реализации итерационного метода не требуется вычисления элементов матрицы конечно-разностных уравнений. Дело в том, что в крыловских методах участвует не сама матрица, а лишь действие матрицы на вектор, которое в нашем случае можно вычислить просто как разность нормальных производных в узлах сетки на интерфейсе. Тогда число действий линейно зависит от числа узлов сетки, а не квадратично, как это требуется в общем случае при вычислении произведения матрицы на вектор. При проведении итерационного процесса решаются краевые подзадачи в подобластях, что составляет наиболее трудоемкую часть решения всей задачи. Поэтому по возможности необходимо уменьшить число итераций данного, назовем его внешним, итерационного процесса. Именно на это направлена наша работа. По сути дела, предлагается использовать многосеточный подход, который заключается в следующем.

¹ Новосибирский государственный университет, механико-математический факультет, ул. Пирогова, д. 2, 630090, Новосибирск; аспирант, e-mail: ilya.klimonov@gmail.com

² Институт вычислительной математики и математической геофизики СО РАН, просп. Лаврентьева, д. 6, 630090, Новосибирск; ст. науч. сотрудник, e-mail: korneev@ssd.sgcc.ru

³ Институт вычислительной математики и математической геофизики СО РАН, просп. Лаврентьева, д. 6, 630090, Новосибирск; зав. лабораторией, e-mail: victor@lapasv.sgcc.ru

Предварительно на макросетке аппроксимируется и решается исходная краевая задача во всей расчетной области. Полученное грубое решение интерполируется в узлы сетки на интерфейсе и используется в качестве начального приближения для внешнего итерационного процесса, при помощи которого находится окончательное решение поставленной задачи.

В настоящей работе проведено экспериментальное исследование указанного итерационного процесса на различных тестовых функциях и при различных способах интерполяции. Показано, что применение предлагаемого подхода до двух раз сокращает число внешних итераций, что означает приблизительно такое же уменьшение времени решения всей задачи в целом.

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, $g_1 = g_1(T)$ и $g_2 = g_2(T)$ — 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} структурированную равномерную 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 $\bar{G}_m, m = \overline{1, M}$, где M — известное целое число. Граница сопряжения подобластей (интерфейс) 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 $G_0 = G \setminus \gamma$.

Построим в подобластях \bar{G}_m структурированные равномерные подсетки $\bar{\Omega}_{h,m}$. Объединение этих подсеток составляет квазиструктурированную сетку $\bar{\Omega}_h = \bigcup_{m=1}^M \bar{\Omega}_{h,m}$.

Исходную краевую задачу (1) переформулируем следующим образом: в замкнутой области \bar{G} требуется найти решение уравнения Пуанкаре–Стеклова

$$Fv(T) = 0, \quad T \in \gamma_f, \quad (2)$$

совместно с решением краевых задач

$$\Delta u(T) = g_1(T), \quad T \in \gamma_{e,m}, \quad (3)$$

$$\Delta u(T) = g_1(T), \quad lu|_{\Gamma} = g_2, \quad u|_{\gamma} = v, \quad T \in G_0 \quad (4)$$

относительно функций u и v . Здесь оператор F определяется как

$$Fv \equiv \left(\frac{\partial u(v)}{\partial \mathbf{n}} \right)_{\gamma_f}^{(+)} - \left(\frac{\partial u(v)}{\partial \mathbf{n}} \right)_{\gamma_f}^{(-)}, \quad (5)$$

где v — след функции u на γ (в том числе на γ_f).

Алгоритм решения задачи (2)–(4) подробно изложен в работах [1, 2], а здесь мы остановимся только на основных его положениях, необходимых для дальнейшего. На квазиструктурированной сетке $\bar{\Omega}_h$ краевая задача (4) методом конечных разностей, конечных элементов или конечных объемов заменяется приближенной задачей

$$\Delta_h u_h = g_1, \quad l_h u_h|_{\Gamma} = g_2, \quad u_h|_{\gamma} = v_h, \quad (6)$$

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 $\bar{\Omega}_{h,m}$, что может быть выполнено параллельно.

Для решения подзадач (2), (3) 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 $\omega_{e,m} = \omega_e \cup \omega_m$.

Заменим в (5) производные приближенными разностными соотношениями и потребуем, чтобы разность приближенных производных в узлах сетки ω_f обращалась в нуль, что дает систему линейных алгебраических уравнений (подробнее см. [1, 2])

$$Av_f + b = 0, \quad (7)$$

где A — квадратная матрица, а b и v_f — векторы.

Вычислять элементы матрицы A нет необходимости, так как для решения системы (7) будем применять итерационные методы в подпространствах Крылова, которые в общем виде можно записать как

$$v_f^{\nu+1} = \Lambda(v_f^\nu, Ap^\nu), \tag{8}$$

где $\nu = 0, 1, \dots$ — номер итерации; Λ — функция, определяющая конкретный метод, например: метод сопряженных градиентов, метод сопряженных невязок или метод обобщенных минимальных невязок (GMRES); p — вспомогательный вектор, определяемый выбором конкретного метода решения. Итерационные методы (8) не требуют знания элементов матрицы, а требуют знания лишь результата действия матрицы на вектор. В нашем подходе произведение матрицы на вектор можно вычислить просто как разность приближенных производных по разные стороны интерфейса в подзадачах вида (6) [1, 2].

На каждом шаге крыловского итерационного процесса необходимо решать краевые подзадачи в подобластях, поэтому его называют итерационным процессом по подобластям. Положительным свойством трудоемкой процедуры итераций по подобластям является то, что она допускает параллельную реализацию.

Решение подзадач на ребрах и в макроузлах проводится на каждой ν -й итерации по подобластям при помощи внутреннего двухуровневого итерационного процесса, суть которого заключается в следующем. Предполагая значения в макроузлах известными, решаем подзадачи на сетке ω_e . Затем уточняем значения в макроузлах, используя аппроксимацию исходного уравнения на сетке ω_m , которая строится с привлечением узлов сетки ω_e . К моменту проведения этих вычислений значения функции v_f^ν на гранях вычислены при помощи итерационного процесса (8).

В целом решение задачи (2)–(4) можно представить в виде следующих шагов.

1. Задается начальное приближение $v_f^{(0)}$ на гранях γ_f для итерационного процесса (8).
2. Из уравнения (3) находятся значения функции $u_{e,m}^{(\nu)} = v_{e,m}^{(\nu)}$ на ребрах и в макроузлах $\gamma_{e,m}$.
3. Из решения краевой задачи (4) с граничными условиями Дирихле $v = v^{(\nu)} = v_f^{(\nu)} \cup v_{e,m}^{(\nu)}$ находятся значения искомой функции на ν -ом приближении в подобластях.
4. Рассчитываются производные, входящие в выражение (5).
5. Делается очередной $(\nu + 1)$ -й шаг по решению уравнения Пуанкаре–Стеклова (2) и находятся значения функции $v_f^{(\nu+1)}$ на гранях.
6. Если сходимость итерационного процесса достигнута, то находится окончательное решение на ребрах, в макроузлах и в подобластях, если же это не так, то повторяются пункты 2–5.

Итерационный процесс (8) можно ускорить, используя многосеточный подход. Для этого предварительно на макросетке $\bar{\Omega}_H$ аппроксимируется и решается краевая задача (1) известными сеточными методами. Для определенности ограничимся методами второго порядка точности, обеспечивающими погрешность $O(H^2)$. Полученные значения в макроузлах интерполируются в узлы сеток ω_f , ω_e и берутся в качестве начального приближения для итерационного процесса (8), что ускоряет его сходимость.

В результате решения исходной краевой задачи на макросетке мы имеем предварительные значения искомой функции $v_m^{(0)}$ в макроузлах ω_m .

Требуется проинтерполировать эти значения в узлах сетки ω_f на гранях интерфейса γ_f и тем самым построить начальное приближение для итерационного процесса (8). Интерполяция проводится в два этапа.

На первом этапе интерполируются значения на ребрах в узлы сетки ω_e . Рассматривается два подхода: линейная интерполяция и интерполяция кубическим сплайном.

При линейной интерполяции значения $v_{e,j}^{(0)}$ в узлах на каждом ребре $e \in \gamma_e$, соединяющем макроузлы t и $t + 1$, находятся по формуле

$$v_{e,j}^{(0)} = j \cdot \frac{v_{m,t+1}^{(0)} - v_{m,t}^{(0)}}{N_e - 1},$$

где N_e — количество узлов на ребре, j — текущий номер узла ($j = 1, \dots, N_e - 1$), $v_{m,t}^{(0)}$ — значение в макроузле с номером t .

При интерполяции кубическим сплайном для всех ребер, лежащих на одной прямой, строится одномерный кубический сплайн [8]. Например, для прямой, параллельной оси OX и соединяющей макроузлы

с координатами (X_0, Y_J, Z_K) и (X_{N_x}, Y_J, Z_K) , где $0 < J < N_Y$, $0 < K < N_Z$, а N_Y, N_Z — число интервалов макросетки по координатным осям, строится кубический сплайн $s_{I,K}^x(x)$. Аналогично строятся кубические сплайны $s_{I,K}^y(y)$ и $s_{I,K}^z(z)$ для прямых, параллельных осям OY и OZ .

После построения значений на ребрах интерфейса переходим ко второму этапу по распространению значений с ребер на грани.

Рассмотрим, например, грань $f \in \gamma_f$, перпендикулярную оси OZ . Значения $v_f^{(0)}$ в узлах сетки на грани находятся из решения двумерной краевой задачи

$$\Delta_h^{xy} v_f^{(0)} = g_1(r) - c^z, \quad r \in \gamma_f, \quad v_f^{(0)}|_{\gamma_e} = v_e^{(0)},$$

где Δ_h^{xy} — аппроксимация двумерного оператора Лапласа, а величина c^z определяется следующим образом. Из сплайнов $s_{I,J}^z(z)$ находим вторые производные в четырех макроузлах, лежащих на рассматриваемой грани, и интерполируем их в узлы сетки ω_f с погрешностью $O(H^2)$, что по порядку согласуется с точностью решения задачи на макросетке. Аналогично поступаем для других граней. Для линейной интерполяции $c^z = 0$.

3. Технология проведения вычислений. Внешний итерационный процесс по подобластям (8), который занимает подавляющую часть времени решения всей задачи, подлежит MPI-распараллеливанию. Внутренние итерационные процессы поиска решений на ребрах и в макроузлах обладают 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 с целью обеспечения приблизительно равной загрузки процессоров. Для этого в каждое объединение включаются такие подсетки, которые давали бы в сумме приблизительно одинаковое число счетных узлов в подобластях. Устанавливается отображение: одно объединение — один процессор.

В начале работы программы каждый процессор считывает из заранее подготовленного файла информацию двух типов:

- 1) общую для всех объединений;
- 2) специальную 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.

Процессоры на основе полученной информации формируют:

- 1) опорные адреса подобластей в массиве объединения подобластей;
- 2) опорные адреса интерфейсов, принадлежащих подобластям объединения;
- 3) различную информацию, необходимую для обмена интерфейсами, а также данными для решения задач на гранях и узлах.

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) передача вычисленных на гранях подобластей производных другим процессорам, в которых находятся соответствующие смежные грани;
- 4) прием производных и вычисление сумм внутренних нормальных производных на смежных сторонах подобластей;
- 5) вычисление сумм производных для смежных граней подобластей, находящихся в объединении;
- 6) вычисление итерационных параметров;
- 7) проверка завершения счета задачи.

Технология организации вычислений и обменов сумм внутренних производных на гранях следующая. 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.

Технология реализации внешнего итерационного процесса (8) основывается на редуцированной MPI-функции обмена данными — MPI_Allreduce (обмен всех со всеми, схема обменов “взаимодействующие равные”). Необходимо просуммировать вычисленные производные по всем процессорам. В MPI_Allreduce реализуется эффективная схема взаимодействий между всеми процессорами, которая выполняется за время, соответствующее $\log_2 N_p$ обменов (N_p — число процессоров, которое мы предполагаем равным двум в целой степени): 1) на первом шаге процессоры разбиваются на пары, которые затем обмениваются друг с другом данными одновременно, например: процессор номер 1 взаимодействует с процессором 2, процессор 3 с процессором 4, и т.д.; 2) на втором шаге пары первого этапа разбиваются аналогично на пары, процессоры в которых обмениваются друг с другом одновременно. Например, процессор номер 1 взаимодействует с процессором 3, процессор 2 с процессором 4 одновременно, и т.д. Взаимодействия осуществляются на виртуальной MPI-сети вычислительной системы “полный граф”, поэтому и возможны такие взаимодействия. Например, на кластере из 64 процессоров взаимодействие выполняется за время, соответствующее всего 6 обменам, на 128 процессорах — 7 обменам и т.д. Это существенно быстрее, чем реализация обменов по схеме “управляющий–рабочие”. Нулевой процессор — управляющий, все остальные — рабочие. Управляющий принимает от рабочих данные, обрабатывает данные и передает их рабочим. Рабочие процессоры взаимодействуют с управляющим процессором асинхронно. Таким образом, схема “управляющий–рабочие” требует $2N_p$ обменов, что значительно больше, чем реализованные нами обмены с помощью функций MPI_Allreduce.

4. Численные эксперименты. Цель проводимых численных экспериментов — исследование ускорения MPI-распараллеливания решения краевых задач с помощью предложенного выше подхода с построением начального приближения для внешнего итерационного процесса (8) по сравнению с MPI-распараллеливанием при обычно используемом нулевом начальном приближении.

Рассматривалась следующая краевая задача:

$$\Delta u = 0, \quad u|_{\Gamma} = g, \tag{9}$$

где $g = \frac{3}{2} \left(1 - \frac{1}{\rho}\right)$, $\rho = \sqrt{x^2 + y^2 + z^2}$ в кубе $R = \{1 \leq x \leq 2, 0 \leq y \leq 1, 0 \leq z \leq 1\}$.

Функция g взята из аналитического решения более общей задачи Дирихле для уравнения Лапласа в области, образованной двумя концентрическими сферами с радиусами, равными 1 и 3, и граничными условиями $g(1) = 0$, $g(3) = 1$.

Декомпозиция расчетной области проводилась равномерной макросеткой $\bar{\Omega}_H$ с числом интервалов $N_X \times N_Y \times N_Z$. В каждой подобласти строилась своя равномерная подсетка, имеющая число интервалов n_h по каждому направлению, т.е. $n_h \times n_h \times n_h$.

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

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

Краевые подзадачи решались при помощи трехмерного аналога метода Писмана–Рэчфорда, а итерации (8) 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 соседних подсеток отличалось в 2^3 раза и они располагались в шахматном порядке, т.е. в формировании квазиструктурированной сетки участвовало два вида подсеток: одна редкая и одна густая.

Численные эксперименты выполнялись на гибридном кластере НКС-30Т Сибирского суперкомпьютерного центра, имеющего параметры: 576 (2688 ядер) процессоров Intel Xeon E5450/E5540/X5670; 40 серверов SL390s G7 с тремя GPU M2090 каждый. Расчеты проводились при помощи программы, изложенной в работе [2], с привлечением графических ускорителей GPU. Общее количество ядер выбиралось кратным трем, так как в каждом узле имеется три ускорителя GPU. Варьировались следующие параметры: 1) количество ядер — 3, 6, 12 (т.е. 1, 2 и 4 узла); 2) размеры макросетки; 3) размеры подсеток; 4) варианты квазиструктурированных сеток — согласованный и несогласованный; 5) типы интерполяции при построении начального приближения. Рассматривается три варианта: 1) Spline — сплайновая интерполяция, 2) Line — линейная интерполяция, 3) Zero — нулевое начальное приближение. Размеры макросеток равны $4 \times 4 \times 3$ и $8 \times 8 \times 6$. В приведенных ниже таблицах приводится число итераций внешнего итерационного процесса (8) для различных вычислительных параметров. Отметим, что внешний итерационный процесс определяет точность ε_I решения всей задачи.

4.1. Численные эксперименты 1. Количество процессоров — 3, размер макросетки $8 \times 8 \times 6$, все размеры подсеток одинаковые (согласованный вариант) и равны 16^3 , точность итерационного процесса — от 10^{-3} до 10^{-6} . Результаты расчетов представлены в табл. 1.

Таблица 1
Число итераций для решения задачи
на макросетке $8 \times 8 \times 6$ и
подсетках 16^3

ε_I	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Spline	24	45	61	72
Line	83	93	114	133
Zero	84	94	116	135

Таблица 2
Число итераций для решения задачи
на макросетке $8 \times 8 \times 6$ и
подсетках 32^3

ε_I	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Spline	42	74	91	107
Line	120	133	162	185
Zero	125	138	166	189

Из табл. 1 видно, что 1) сплайновая интерполяция при построении начального приближения дает ускорение по отношению к числу итераций при нулевом начальном приближении для точности 10^{-3} , равное $84/24 = 3.50$, а для точности 10^{-6} — $135/72 \approx 1.88$; 2) линейная интерполяция не дает эффекта ускорения, так как число итераций в этом случае приблизительно равно числу итераций при нулевом начальном приближении; 3) разрыв в количестве итераций уменьшается с увеличением точности решения. Приведенные факты объясняются наличием в ошибке решения высокочастотных гармоник, которые должны быть ликвидированы при высокой точности итерационного процесса и которые остаются при построении начального приближения даже при сплайновой интерполяции, а при линейной интерполяции играют ведущую роль.

В табл. 2 приводятся результаты расчетов с параметрами, указанными выше, за исключением размера подсеток, который равен 32^3 . Заметим, что результирующая квазиструктурированная сетка имела более 12.5 миллионов узлов.

В этом эксперименте соотношения числа итераций при нулевом и сплайновом приближениях равны $125/42 \approx 3.00$ для точности 10^{-3} и $189/107 \approx 1.77$ для точности 10^{-6} , т.е. с увеличением размеров подсеток соотношение числа итераций уменьшается, что также можно объяснить увеличением числа непогашенных высокочастотных гармоник при построении начального приближения.

4.2. Численные эксперименты 2. Количество процессоров — 3, размер макросетки $4 \times 4 \times 3$, все размеры подсеток одинаковые (согласованный вариант) и равны 16^3 .

Из табл. 3 видно, что 1) соотношения числа итераций при нулевом и сплайновом начальных приближениях равны $42/18 \approx 2.33$ для точности 10^{-3} и $68/40 = 1.70$ для точности 10^{-6} ; 2) соотношения числа итераций при нулевом и линейном начальных приближениях существенны: для точности 10^{-3} они равны $42/23 \approx 1.83$, а для точности 10^{-6} — $68/49 \approx 1.39$, что объясняется грубой сеткой, на которой существенным является вклад в начальное приближение даже линейной интерполяции; 3) сравнение результатов табл. 1 и 3 показывает уменьшение числа итераций в расчетах на макросетке $4 \times 4 \times 3$ по сравнению с макросеткой $8 \times 8 \times 6$. Последний факт говорит о том, что по сути дела на грубой макросетке увеличивается шаг сетки ω_f на интерфейсе, уменьшается число обусловленности матрицы A в уравнении (7) и, следовательно, уменьшается число итераций.

В табл. 4 приведены результаты расчетов с параметрами, отличающимися от предыдущего варианта только размерами подсеток, которые равны 32^3 .

Таблица 3
Число итераций для решения задачи на макросетке $4 \times 4 \times 3$ и подсетках 16^3

ε_I	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Spline	18	27	34	40
Line	23	31	41	49
Zero	42	48	59	68

Таблица 4
Число итераций для решения задачи на макросетке $4 \times 4 \times 3$ и подсетках 32^3

ε_I	10^{-3}	10^{-4}	10^{-5}	10^{-6}
Spline	30	42	50	59
Line	37	47	59	71
Zero	63	72	86	100

В этом эксперименте соотношения числа итераций при нулевом и сплайновом начальных приближениях равны $63/30 = 2.10$ для точности 10^{-3} и $100/59 \approx 1.70$ для точности 10^{-6} . При линейном начальном приближении эти соотношения равны $63/37 \approx 1.70$ для точности 10^{-3} и $100/71 \approx 1.41$ для точности 10^{-6} . По отношению к этим результатам справедливы рассуждения, приведенные в предыдущих пунктах. Далее отметим следующее.

1. Описанные выше численные эксперименты были повторены на 6 и 12 ядрах. В этих расчетах цифры количества внешних итераций те же самые, но время решения задачи уменьшается с увеличением ядер соответственно в два раза.
2. Такие же расчеты проведены и для несогласованных вариантов. Как было сказано выше, в несогласованном варианте число интервалов соседних подсеток отличалось в 2^3 раз и они располагались в шахматном порядке. Разброс по процессорам подобластей с густыми и редкими сетками одинаковый. Соотношения количества внешних итераций для разных типов интерполяции близки соотношениям для случая согласованных вариантов.
3. Все описанные выше вычислительные эксперименты были проведены для задачи (9) при более простой функции $g = x^2 + y^2 + z^2$. В этом случае, как и следовало ожидать, итерационный процесс (8) сходил за одну итерацию как для случая линейной интерполяции, так и для случая сплайн-интерполяции вне зависимости от точности, а для нулевого начального приближения сходимость достигалась за несколько десятков итераций.

5. Заключение. В настоящей работе предложен и экспериментально исследован многосеточный метод ускорения итерационного процесса по подобластям при решении трехмерных краевых задач методом декомпозиции расчетной области на подобласти, сопрягаемые без наложения, на квазиструктурированных сетках. Суть метода состоит в том, что предварительно аппроксимируется и решается исходная краевая задача на макросетке, с которой проводится декомпозиция, полученное решение интерполируется в узлы сетки на интерфейсе и используется в качестве начального приближения для внешнего итерационного процесса по подобластям. Проведены тестовые расчеты для согласованных и несогласованных квазиструктурированных сеток с различным количеством узлов. Рассмотрены сплайновая и линейная интерполяции. Показано сокращение числа внешних итераций в среднем в два раза.

Теоретическая часть работы поддержана РФФ (проект № 14-11-00485п), экспериментальная часть работы поддержана РФФИ (проект № 16-01-00168) и в рамках Комплексной программы фундаментальных исследований СО РАН “Междисциплинарные исследования” на 2018–2020 г. (проект № 10).

СПИСОК ЛИТЕРАТУРЫ

1. Корнеев В.Д., Свешников В.М. Параллельные алгоритмы и технологии декомпозиции расчетной области для решения трехмерных краевых задач на квазиструктурированных сетках // Сиб. журн. вычисл. матем. 2016. **19**, № 2. 183–194.
2. Климонов И.А., Корнеев В.Д., Свешников В.М. Технологии распараллеливания решения трехмерных краевых задач на квазиструктурированных сетках в гибридной вычислительной среде CPU+GPU // Вычислительные методы и программирование: новые вычислительные технологии. 2016. **17**. 65–71.
3. Quarteroni A., Valli A. Domain decomposition methods for partial differential equations. Oxford: Clarendon Press, 1999.
4. Dolean V., Jolivet P., Nataf F. An introduction to domain decomposition methods: algorithms, theory, and parallel implementation. Philadelphia: SIAM Press, 2015.
5. Самарский А.А. Теория разностных схем. М.: Наука, 1977.
6. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: ИВМиМГ СО РАН, 2001.
7. Ильин В.П. Методы и технологии конечных элементов. Новосибирск: ИВМиМГ СО РАН, 2007.
8. Самарский А.А., Гудин А.В. Численные методы математической физики. М.: Научный мир, 2003.

Поступила в редакцию
25.01.2018

Acceleration of Parallel Algorithms for Solving Three-Dimensional Boundary Value Problems on Quasi-Structured Grids

I. A. Klimonov¹, V. D. Korneev², and V. M. Sveshnikov³

¹ *Novosibirsk State University, Faculty of Mechanics and Mathematics; ulitsa Pirogova 2, Novosibirsk, 630090, Russia; Graduate Student, e-mail: ilya.klimonov@gmail.com*

² *Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of Russian Academy of Sciences; prospekt Lavrentyeva 6, Novosibirsk, 630090, Russia; Ph.D., Associate Professor, e-mail: korneev@ssd.sccc.ru*

³ *Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of Russian Academy of Sciences; prospekt Lavrentyeva 6, Novosibirsk, 630090, Russia; Dr. Sci., Professor, Head of Laboratory, e-mail: victor@lapasrv.sccc.ru*

Received January 25, 2018

Abstract: This paper is devoted to the acceleration of the parallel solution of three-dimensional boundary value problems by the computational domain decomposition method into subdomains that are conjugated without overlapping. The decomposition is performed by a uniform parallelepipedal macrogrid. In each subdomain and on the interface, some structured subgrids are constructed. The union of these subgrids forms a quasi-structured grid on which the problem is solved. The parallelization is carried out using the MPI-technology. We propose and experimentally study the acceleration algorithm for an external iterative process on subdomains to solve a system of linear algebraic equations approximating the Poincaré–Steklov equation on the interface. A number of numerical experiments are carried out on various quasi-structured grids and with various parameters of computational algorithms showing the acceleration of computations.

Keywords: boundary value problems, parallelization, quasi-structured grids, iterative process, initial approximation.

References

1. V. D. Korneev and V. M. Sveshnikov, “Parallel Algorithms and Domain Decomposition Techniques for Solving Three-Dimensional Boundary Value Problems on Quasi-Structured Grids,” *Sib. Zh. Vych. Mat.* **19** (2), 183–194 (2016) [*Numer. Anal. Appl.* **9** (2), 141–149 (2016)].
2. I. A. Klimonov, V. D. Korneev, and V. M. Sveshnikov, “Parallelization Technologies for Solving Three-Dimensional Boundary Value Problems on Quasi-Structured Grids Using the CPU+GPU Hybrid Computing Environment,” *Vychisl. Metody Programm.* **17**, 65–71 (2016).

3. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations* (Clarendon Press, Oxford, 1999).
4. V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation* (SIAM Press, Philadelphia, 2015).
5. A. A. Samarskii, *The Theory of Difference Schemes* (Nauka, Moscow, 1977; Marcel Dekker, New York, 2001).
6. V. P. Il'in, *Finite Difference and Finite Volume Methods for Elliptic Equations* (Inst. Comput. Math. Math. Geophys., Novosibirsk, 2000) [in Russian].
7. V. P. Il'in, *Methods and Technologies of Finite Elements* (Inst. Comput. Math. Math. Geophys., Novosibirsk, 2007) [in Russian].
8. A. A. Samarskii and A. V. Gulin, *Numerical Methods of Mathematical Physics* (Nauchnyi Mir, Moscow, 2003) [in Russian].