

АЛГОРИТМЫ РЕШЕНИЯ СЛАУ НА СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ В ПРИМЕНЕНИИ К ЗАДАЧАМ ЭЛЕКТРОМАГНЕТИЗМА¹

Д.С. Бутюгин

Рассматриваются различные аспекты моделирования гармонических электромагнитных полей на кластерах. Основная вычислительная сложность задачи заключается в решении систем линейных алгебраических уравнений (СЛАУ), возникающих в результате конечно-элементных аппроксимаций соответствующих краевых задач электромагнетизма элементами Неделека различных порядков. Рассмотрены эффективные и экономичные подходы к декомпозиции расчетной области и матрицы системы. Решение распределенных СЛАУ осуществляется итерационными методами в подпространствах Крылова с использованием аддитивного метода Шварца в качестве преобуславливателя. Для повышения эффективности алгоритмов итерации осуществляются в подпространствах следов. Реализованные решатели используют MPI для организации обмена данными. Решение систем в подобластях осуществляется при помощи прямого решателя PARDISO из библиотеки Intel® MKL. Результаты серии численных экспериментов на модельных и практических задачах демонстрируют эффективность предлагаемых алгоритмов.

Ключевые слова: уравнения Максвелла, итерационные алгоритмы, методы декомпозиции подобластей, аддитивный метод Шварца.

Введение

Многие актуальные задачи вычислительной математики требуют решения разреженных СЛАУ высоких порядков. Задача вычисления трехмерного электромагнитного поля в частотной области возникает при расчетах различных волновых устройств, решении задач геоэлектроразведки, таких как электромагнитного каротажа, и других. Требования к точности получаемого решения задач с разномасштабными объектами приводят к необходимости построения сеток с числом конечных элементов до $10^5 \div 10^7$ и порядком получаемых после аппроксимации СЛАУ до $10^7 \div 10^9$. Решение таких СЛАУ невозможно без использования вычислительной мощности кластеров. Это требует подходящих алгоритмов решения систем алгебраических уравнений для машин с распределенной памятью.

В данной работе рассматриваются алгоритмы решения СЛАУ, основанные на аддитивном методе Шварца. Рассмотрено два подхода к декомпозиции задачи. Первый из подходов основан на геометрическом разбиении сетки на связанные подобласти и позволяет построить декомпозицию с пересечениями подобластей. Второй подход основан на переупорядочивании графа матрицы системы. Этот способ представляет из себя вариант одно-направленного разбиения графа и приводит к СЛАУ блочно-трехдиагонального вида. Отличительной особенностью предлагаемых методов является низкая асимптотическая вычислительная сложность алгоритмов, в отличие от вариантов метода вложенных сечений (например, реализованного в библиотеке METIS [1]).

¹Статья рекомендована к публикации программным комитетом международной научной конференции «Параллельные вычислительные технологии 2012»

Решение полученных систем уравнений осуществляется методами в подпространствах Крылова [2], такими как обобщенный метод минимальных невязок (GMRES). В качестве предобуславливателя выступает аддитивный метод Шварца. Использование такого подхода, в отличие от использования метода Шварца напрямую, позволяет решить более широкий класс задач и повысить скорость сходимости итерационного процесса.

Структура данной работы следующая. В разделе 1 описывается математическая постановка задачи, в разделе 2 — дискретная постановка. В разделе 3 описаны предлагаемые методы декомпозиции. Раздел 4 содержит описание используемых итерационных алгоритмов. В разделе 5 приведены результаты численных экспериментов. Наконец, в последнем разделе обсуждаются полученные в работе результаты.

1. Математическая постановка задачи

Электромагнитное поле с гармонической зависимостью от времени в случае линейности задачи при отсутствии внешнего заряда и магнитной проводимости может быть описано следующей формой системы уравнений Максвелла:

$$\begin{aligned} \nabla \times \vec{E} &= -i\omega\mu\vec{H}, & \nabla \cdot (\varepsilon_r\vec{E}) &= \rho/\varepsilon_0, \\ \nabla \times \vec{H} &= i\omega\dot{\varepsilon}\vec{E} + \vec{J}, & \nabla \cdot (\mu_r\vec{H}) &= 0, \end{aligned} \quad (1)$$

где электрическая и магнитная проницаемости имеют вид

$$\dot{\varepsilon} = \varepsilon_0\varepsilon_r - i\sigma^e/\omega, \quad \mu = \mu_0\mu_r - i\sigma^m/\omega.$$

Здесь i — мнимая единица, \vec{E} и \vec{H} — векторы напряженности электрического и магнитного полей соответственно, ω — круговая частота решения, \vec{J} и ρ — плотности внешнего тока и объемного заряда соответственно, ε_0 и μ_0 — диэлектрическая и магнитная проницаемости вакуума, ε_r и μ_r — относительная диэлектрическая и магнитная проницаемости среды, а σ^e и σ^m — электрическая и магнитная проводимости.

Используя предположения о линейности задачи (то есть независимости параметров среды от электромагнитного поля), а также отсутствие объемного внешнего заряда ρ и магнитной проводимости $\sigma^m = 0$, уравнения (1) можно привести к форме комплексного векторного уравнения Гельмгольца

$$\nabla \times \left(\mu_r^{-1} \nabla \times \vec{E} \right) - k_0^2 \dot{\varepsilon}_r \vec{E} = -ik_0 Z_0 \vec{J}. \quad (2)$$

Здесь $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$, $Z_0 = \sqrt{\mu_0/\varepsilon_0}$ и $\dot{\varepsilon}_r = \dot{\varepsilon}/\varepsilon_0$. Параметры среды полагаются кусочно-постоянными. Мы будем предполагать, что k_0 не является Максвелловским собственным числом, то есть рабочая частота ω не является резонансной. Стоит отметить, что в случае, если хотя бы в одной подобласти $\sigma^e > 0$, рабочая частота может быть любой, поскольку в системе отсутствуют резонансы [3].

Решение ищется в области Ω с границей $S = S_1 \cup S_2$, на каждой из частей которой поставлено одно из следующих граничных условий:

$$\vec{n} \times \vec{E} \Big|_{S_1} = \vec{n} \times \vec{E}_0, \quad \vec{n} \times \vec{H} \Big|_{S_2} = 0, \quad (3)$$

где \vec{n} — внешняя нормаль к границе. При $\vec{E}_0 = 0$ первое из условий соответствует идеальному электрическому проводнику, при $\vec{E}_0 \neq 0$ — волновому входу, а второе

условие соответствует идеальному магнитному проводнику. Полагая область Ω состоящей из подобластей $\Omega = \bigcup \Omega_k$, в каждой из которых физические параметры среды являются константными, введем на каждой из внутренних границ Γ с нормалью \vec{n} условия сопряжения

$$\begin{aligned} \vec{n} \cdot (\varepsilon_1 \vec{E}_1 - \varepsilon_2 \vec{E}_2) &= 0, & \vec{n} \times (\vec{E}_1 - \vec{E}_2) &= 0, \\ \vec{n} \cdot (\mu_1 \vec{H}_1 - \mu_2 \vec{H}_2) &= 0, & \vec{n} \times (\vec{H}_1 - \vec{H}_2) &= 0. \end{aligned} \quad (4)$$

Вводятся стандартные соболевские пространства и подмножества:

$$\begin{aligned} H^{\text{rot}} &= \left\{ \vec{\psi} \in [L^2(\Omega)]^3 : \nabla \times \vec{\psi} \in [L^2(\Omega)]^3 \right\}, \\ H_0^{\text{rot}} &= \left\{ \vec{\psi} \in H^{\text{rot}} : \vec{n} \times \vec{\psi}|_{S_1} = 0 \right\}, \quad H_S^{\text{rot}} = \left\{ \vec{\psi} \in H^{\text{rot}} : \vec{n} \times \vec{\psi}|_{S_1} = \vec{n} \times \vec{E}_0 \right\}. \end{aligned}$$

Вариационная формулировка уравнения (2) с краевыми условиями (3) в форме Галёркина имеет следующий вид (см. [4]): найти такое $\vec{E} \in H_S^{\text{rot}}$, что для всех $\vec{\psi} \in H_0^{\text{rot}}$ выполнено

$$\int_{\Omega} \mu_r^{-1} (\nabla \times \vec{E}) \cdot (\nabla \times \vec{\psi}) \, d\Omega - k_0^2 \int_{\Omega} \varepsilon_r (\vec{E} \cdot \vec{\psi}) \, d\Omega = -ik_0 Z_0 \int_{\Omega} \vec{J} \cdot \vec{\psi} \, d\Omega. \quad (5)$$

2. Дискретная постановка задачи

В работе используются неструктурированные тетраэдральные сетки. Рассматривается соответствующее разбиение расчетной области Ω на непересекающиеся тетраэдральные элементы $\Omega = \bigcup \Omega_T$. В каждом из тетраэдров вводятся базисные функции, соответствующие его степеням свободы. Пусть \mathcal{W}_l — конечномерное пространство базисных функций порядка не выше l , конформное H^{rot} . В работе рассматриваются иерархические базисные функции, предложенные в [5]:

$$\begin{aligned} \mathcal{W}_l &= \tilde{\mathcal{W}}_1 \oplus \tilde{\mathcal{W}}_2 \oplus \dots \oplus \tilde{\mathcal{W}}_l, \\ \tilde{\mathcal{W}}_1 &= \tilde{\mathcal{A}}_1, \quad \tilde{\mathcal{W}}_i = \tilde{\mathcal{A}}_i \oplus \nabla \tilde{\mathcal{V}}_i, \end{aligned}$$

где $\tilde{\mathcal{W}}_i$ — инкрементальное подпространство с базисными функциями порядка i , $\tilde{\mathcal{V}}_i$ — инкрементальное подпространство со скалярными базисными функциями, конформными H^1 , а $\tilde{\mathcal{A}}_i$ — инкрементальное подпространство с роторными базисными функциями.

Подпространства $\mathcal{V}_{l,0}$ и $\mathcal{W}_{l,0}$ вводятся естественным образом, а подмножество $\mathcal{W}_{l,S}$ — как множество функций, у которых коэффициенты разложения u^S по базисным функциям с ненулевым следом на S_1 принимают фиксированные значения, соответствующие краевому условию (3).

Приближенное решение \vec{E}^h будем искать в виде

$$\vec{E}^h = \sum_j u_j^S \vec{\psi}_j^S + \sum_i u_i \vec{\psi}_i^0. \quad (6)$$

Вводятся матрицы соответствующих билинейных форм:

$$M_{i,j} = \int_{\Omega} \varepsilon_r (\vec{\psi}_j^0 \cdot \vec{\psi}_i^0) \, d\Omega, \quad S_{i,j} = \int_{\Omega} \mu_r^{-1} (\nabla \times \vec{\psi}_j^0) \cdot (\nabla \times \vec{\psi}_i^0) \, d\Omega,$$

где базисные функции $\vec{\psi}_i^0, \vec{\psi}_j^0 \in \mathcal{W}_{l,0}$. Вектор правой части определяется как

$$f_i = -ik_0 Z_0 \int_{\Omega} \vec{J} \cdot \vec{\psi}_i^0 d\Omega + \sum_j u_j^S \int_{\Omega} \left(\mu_r^{-1} (\nabla \times \vec{\psi}_j^S) \cdot (\nabla \times \vec{\psi}_i^0) - k_0^2 \epsilon_r \vec{\psi}_j^S \cdot \vec{\psi}_i^0 \right) d\Omega.$$

Тогда итоговая система принимает вид

$$[S - k_0^2 M] u = f. \quad (7)$$

Для вычисления элементов матрицы и вектора правой части можно воспользоваться поэлементной технологией сборки [6], заменив интегрирование по расчетной области Ω суммой интегралов по каждому из тетраэдров и вычислением в каждом из тетраэдров локальных матриц и векторов.

3. Методы декомпозиции на подобласти

Для решения задачи на системах с распределенной памятью требуется распределить данные между узлами системы, по возможности минимизируя объем коммуникационных данных. Основной проблемой здесь является то, что алгоритмы, генерирующие разбиения высокого качества, сами требуют больших вычислительных ресурсов. Поэтому, как правило, приходится идти на компромисс и выбирать алгоритмы, позволяющие получить достаточно хорошее разбиение за приемлемое время.

3.1. Геометрическая декомпозиция расчетной области

Предлагаемый в работе алгоритм геометрической декомпозиции расчетной области является упрощенным вариантом алгоритма построения BSP-дерева (binary space partitioning, двоичное разбиение пространства). Метод двоичного разбиения — это метод рекурсивного разбиения пространства на выпуклые множества гиперплоскостями [7]. Получающаяся в результате структура данных находит широкое применение в различных областях компьютерной графики для проверки пересечений, определения прямой видимости объектов и др. Стоит отметить, что построение оптимального BSP дерева представляет из себя очень трудоемкую задачу, поэтому, в большинстве случаев, ограничиваются субоптимальными деревьями.

В рамках данной работы предлагается использовать следующее упрощенное двоичное дерево. Упрощение состоит в том, что секущие плоскости предлагается проводить ортогонально осям координат. При этом мы будем каждый раз разделять имеющееся множество тетраэдров на две приблизительно равные части. Алгоритм построения разбиения будет основан на методе обработки событий. Подготовительная часть алгоритма такая:

- для x , y и z : спроектировать тетраэдры на соответствующую ось; для каждого тетраэдра получаем отрезки, соответствующие его началу и концу;
- для x , y и z : сгенерировать отсортированные по координате события — начало и конец каждого из тетраэдров.

В результате мы получим 3 отсортированных списка событий, для осей Ox , Oy и Oz соответственно.

Далее к этим спискам мы рекурсивно будем применять следующую процедуру. Будем последовательно обрабатывать события для каждой из осей, что соответствует движению секущей плоскости в сторону увеличения соответствующей координаты.

При этом мы будем подсчитывать число тетраэдров в каждом из полупространств. Тетраэдры, пересекающиеся и касающиеся плоскости сечения, будем относить к обеим частям одновременно. Схема алгоритма следующая:

- Для x , y и z :
 - $\text{left_count} \leftarrow 0$, $\text{right_count} \leftarrow T$, где T — число тетраэдров.
 - Для каждой координаты последовательно: обработать события
 - * Обработать начала тетраэдров: для каждого $\text{left_count} \leftarrow \text{left_count} + 1$.
 - * Проверить left_count и right_count на оптимальность.
 - * Обработать концы тетраэдров: $\text{right_count} \leftarrow \text{right_count} - 1$.
 - Запомнить оптимальное разбиение для оси.
- Выбрать наилучшую ось для разбиения.
- Построить разбиение: для x , y и z :
 - Линейным проходом разбить список событий для соответствующей оси на 2 в соответствии с разбиением. При этом упорядоченность сохранится.
- Рекурсивно запустить алгоритм для двух полученных частей.

Алгоритм останавливает рекурсивное разбиение тогда, когда количество тетраэдров в обрабатываемой части становится достаточно мало, либо когда в результате деления количество тетраэдров в одной из частей оказался равен исходному их числу.

Проверка на оптимальность для каждой из осей может быть следующей: выбирается положение плоскости, при котором размеры левой и правой частей максимально близки друг к другу. Наилучшая из осей выбирается по минимуму $\min(\text{left_count}, \text{right_count})$ для оптимального положения плоскости, что позволяет выбрать минимальный из разрезов подпространства и уменьшить коммуникации между подобластями. Отметим, что «толщину» слоя пересечения легко варьировать. Для этого достаточно осуществить проход по событиям двумя указателями вместо одного. Один из указателей при этом будет соответствовать началу пересечения подобластей, второй — концу. При этом легко поддерживать как нужную геометрическую толщину пересечения подобластей, так и минимальное число тетраэдров в пересечении.

Легко видеть, что асимптотическая сложность алгоритма равна $O(T \log T)$, где T — исходное число тетраэдров, при условии, что каждый раз в пересечении оказывается небольшое число тетраэдров и части делятся примерно поровну. Действительно, сложность сортировки событий в начале алгоритма равна $O(T \log T)$. Далее, если число тетраэдров растет несущественно по сравнению с величиной T (например, конечное число тетраэдров есть $O(T)$), то на каждом уровне дерева требуется линейный проход по всем событиям для каждого из узлов дерева этого уровня, суммарное количество которых есть $O(T)$. При этом число подобластей при переходе на уровень ниже увеличивается в 2 раза, поэтому общее число уровней не превзойдет $O(\log T)$. Отсюда получаем заявленную сложность.

Замечание 1. Алгоритм осуществляет разбиение сетки на подобласти с пересечениями, причем никакие две подобласти не имеют протяженной общей границы кроме, возможно, части внешней границы расчетной области.

Доказательство этого факта следует из следующего тривиального наблюдения: два тетраэдра с общей гранью невозможно разбить плоскостью так, чтобы один тет-

раэдр лежал по одну сторону от нее, второй по другую и плоскость не касалась хотя бы одного из тетраэдров. При этом, как следует из алгоритма, тетраэдры, касающиеся либо пересекающиеся с плоскостью сечения, помещаются в пересечение подобластей. Данный факт оказывается очень важным для дальнейшего решения задачи методом декомпозиции на подобласти.

Рассмотрим теперь постановку задачи нахождения поля для подобластей. Пусть имеется разбиение расчетной области Ω на подобласти $\Omega = \bigcup \Omega_p$. Будем искать решение задачи (5) на области Ω путем решения задачи (5) в каждой из подобластей Ω_p . На внутренней границе Γ каждой из подобластей Ω_p будем ставить условия Дирихле из (3). Эти условия будут определяться из значений поля на внутри подобластей, соседних с Ω_p , и, на конечно-элементном уровне, из коэффициентов разложения поля \vec{E} по базисным функциям, соответствующим граничным ребрам и граням. То есть, требуется найти поля в каждой из подобластей так, чтобы они были согласованы друг с другом по краевым условиям.

Теорема 1. *Решение задачи для подобластей для предлагаемого алгоритма декомпозиции существует. Оно единственно, если частота излучения не совпадает с резонансными частотами полостей, образованных подобластями и их пересечениями.*

Доказательство. Ясно, что решение исходной задачи на всей области Ω удовлетворяет задаче на подобластях. Единственность следует из следующей леммы.

Лемма 1. *Пусть поле \vec{E} является решением задачи для подобластей. Тогда для любых Ω_p и $\Omega_{p'}$, имеющих непустое пересечение, значения \vec{E} в общих точках Ω_p и $\Omega_{p'}$ совпадают.*

Доказательство напрямую следует из того, что для пересечения подобластей мы имеем краевую задачу, которая, если частота не резонансная для данной задачи, имеет единственное решение. Поэтому решение задачи для подобластей однозначно определяет электрическое поле в любой точке исходной области Ω .

Доказательство теоремы 1 завершается замечанием того, что

- для любой внутренней точки области Ω существует замкнутая окрестность, что эта точка вместе с окрестностью целиком лежит внутри какой-либо подобласти Ω_p ;
- для любой базисной функции из $\mathcal{W}_{l,0}$ ее носитель целиком лежит внутри какой-нибудь подобласти Ω_p ;

Отсюда следует, что электрическое поле, являющееся решением задачи для подобластей, удовлетворяет и вариационной формулировке (5) в непрерывном случае, и дискретному аналогу этой формулировке в конечно-элементном случае. Однако решение вариационной формулировки единственно при указанных ограничениях на параметры среды и частоту излучения [3], поэтому единственным оказывается и решение задачи для подобластей. \square

3.2. Алгебраическая декомпозиция

Другим вариантом декомпозиции задачи на подобласти, рассматриваемым в рамках работы, является модифицированный алгоритм одно-направленного разбиения

графа матрицы системы [2]. Алгоритм строит разбиение неизвестных системы на множества — алгебраические подобласти Ω_p^A . В дальнейшем индекс A в Ω_p^A мы будем опускать.

Первым шагом алгоритма является нахождение псевдо-периферийной вершины v . Поиск такой вершины осуществляется следующим образом [2].

1. Выбирается произвольная вершина v . Псевдо-диаметр графа D полагается равным 0.
2. Запускается обход в ширину по графу, начиная от вершины v .
3. Находится любая максимально удаленная вершина v' .
4. Если расстояние $d(v, v')$ больше D , то $v \leftarrow v'$, $D \leftarrow d(v, v')$, переход на шаг 2.

Информация с последнего обхода графа в ширину используется для дальнейшего построения разбиения. Для начала все вершины разбиваются во «фронты» F_k — множества вершин, равноудаленных от v на расстояние k . Далее фронты объединяются в алгебраические подобласти по следующему принципу.

С помощью бинарного поиска мы будем максимизировать размер минимальной подобласти. Проверка того, что при данном минимальном размере подобласти можно получить необходимое число подобластей, может быть осуществлена жадным алгоритмом [8]. Пример кода, определяющего максимальное число подобластей, представлен на рис. 1.

```
int get_max_domains(int k, int front_size[], int min_size) {
    int domains = 0, domain_size = 0, i;
    for(i = 0; i < k; i++) {
        domain_size += front_size[i]; // Размер в вершинах
        if(domain_size >= min_size) {
            ++domains;
            domain_size = 0;
        }
    }
    // Оставшиеся вершины отходят к последней подобласти
    return domains;
}
```

Рис. 1. Алгоритм определения максимального числа подобластей

Представленный алгоритм генерирует подобласти без пересечений. Однако получить разбиение с пересечениями довольно легко — достаточно при начале генерации следующей подобласти отступить от границы предыдущей внутрь необходимое число фронтов. Графическое представление объединения фронтов в подобласти с пересечениями изображено на рис. 2, где фронты изображены вертикальными линиями.

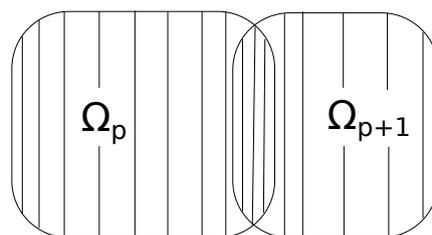


Рис. 2. Объединение фронтов матрицы в подобласти

Обозначив l_p и r_p — границы подобласти Ω_p , мы имеем разбиение на подобласти

$$\Omega_p = \bigcup_{k=l_p}^{r_p} F_k,$$

Необходимо отметить, что ребра в графе матрицы могут соединять вершины либо одного, либо двух соседних фронтов. Это означает, что каждая подобласть Ω_p может граничить одновременно максимум с двумя подобластями — Ω_{p-1} и Ω_{p+1} (при наличии пересечений всегда можно считать, что два граничных с Ω_p фронта $F_{l_{p-1}}$ и $F_{r_{p+1}}$ принадлежат соседним подобластям Ω_{p-1} и Ω_{p+1} соответственно). Тогда последовательно занумеровав переменные в каждой из подобластей, а в каждой из подобластей — последовательно в каждом из фронтов, мы получим следующую блочно-трехдиагональную СЛАУ:

$$\begin{bmatrix} D_1 & U_1 & & & \\ L_1 & D_2 & \ddots & & \\ & \ddots & \ddots & U_{p-1} & \\ & & L_{p-1} & D_p & \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{bmatrix}.$$

При этом если разбиение исходной СЛАУ выполнялось с пересечениями, то в итоговой СЛАУ окажется по несколько переменных, соответствующих вершинам в пересечении.

4. Алгоритмы решения СЛАУ

После декомпозиции задачи на подобласти и построения блоков матрицы на соответствующих узлах кластера, мы имеем распределенную СЛАУ

$$Au = f.$$

Для решения таких СЛАУ широко используются методы Шварца. Одна итерация аддитивного метода Шварца в матричном виде может быть записана как (см. [2])

$$u_{i+1} = u_i + \sum_p R_p^T A_p^{-1} R_p (f - Au_i),$$

где R_p — оператор сужения на подобласть Ω_p . При этом, на самом деле, итерировать достаточно только переменные, соответствующие внутренним границам подобластей, поскольку зная граничные значения, легко определить значения внутри подобластей, решив соответствующую задачу в подобласти. То есть

$$u_p = A_p^{-1} R_p (f - A \hat{R}^T u^\Gamma), \tag{8}$$

где u^Γ — вектор граничных переменных, а \hat{R} — оператор сужения на граничные переменные. Подпространство граничных переменных принято называть подпространством следов. Тогда итерация метода Шварца в подпространстве следов переписывается как

$$u_{i+1}^\Gamma = u_i^\Gamma + \hat{R} \sum_p R_p^T A_p^{-1} R_p (f - A \hat{R}^T u_i^\Gamma).$$

Эту итерацию можно представить в виде $u_{i+1}^\Gamma = S(u_i^\Gamma) = Tu_i^\Gamma + g$, где

$$T = I - \hat{R} \sum_p R_p^T A_p^{-1} R_p A \hat{R}^T, \quad g = \hat{R} \sum_p R_p^T A_p^{-1} R_p f.$$

Сходимость данного метода имеется при спектральном радиусе $\rho(T) < 1$. В то время как для эллиптических уравнений это условие выполнено, для уравнения Гельмгольца это не так. Однако в этом случае можно рассмотреть решение системы

$$[I - T] u^\Gamma = g. \quad (9)$$

Теорема 2. Система (9) является совместной. В случае геометрической декомпозиции при выполнении условий теоремы 1 оно единственно.

Доказательство. Действительно, система (9) эквивалентна системе

$$\hat{R} \sum_p R_p^T A_p^{-1} R_p A u = \hat{R} \sum_p R_p^T A_p^{-1} R_p f,$$

поэтому решение $u^\Gamma = \hat{R}u$, где u — решение $Au = f$ удовлетворяет ей. С другой стороны любое решение (9) является неподвижной точкой итерации Шварца. В случае геометрической декомпозиции на подобласти у неподвижной точки итерации Шварца есть простая интерпретация. Это вектор, задающий поле в подобластях, согласованное по граничным условиям. Однако при выполнении условий теоремы 1 последняя утверждает, что такое поле единственно. Значит, единственна и неподвижная точка итерации Шварца. \square

Можно отметить, что решение системы методом Шварца есть решение предобусловленной системы $M^{-1}Au = M^{-1}f$ с M^{-1} , связанным с A и T соотношением $T = I - M^{-1}A$, откуда $M^{-1} = [I - T]A^{-1}$ и предобусловленная система переходит в систему (9). Эту систему можно решать и итерационными методами в подпространствах Крылова, например методом обобщенных минимальных невязок (GMRES) [2]. Для методов в подпространствах Крылова не требуется явный вид матрицы системы, а достаточно только операции умножения матрицы на вектор. Запишем результат умножения $I - T$ на вектор v через результат одной итерации метода аддитивного Шварца $S(v)$:

$$[I - T]v = v - Tv = v - S(v) + g, \quad g = S(0).$$

Таким образом, для решения СЛАУ (9) методом GMRES достаточно реализовать обычный метод аддитивного Шварца. Для решения задач в подобластях $A_p x = b$ можно использовать какой-нибудь прямой решатель для разреженных систем. В работе использовался решатель PARDISO из библиотеки Intel MKL [9]. Такой подход оказывается достаточно эффективным, поскольку на каждой итерации требуется решать системы с одними и теми же матрицами, и LU -разложение матриц A_p можно посчитать только один раз. Кроме того, такой подход оказывается существенно эффективнее простого использования решателя PARDISO для исходной системы, поскольку время, требуемое PARDISO для разложения матрицы, растет практически как $O(N^3)$, где N — порядок системы. Дополнительно эффективность достигается

за счет использования подпространства следов, так как в этом случае существенно уменьшается объем коммуникационных затрат, а также объем памяти, требуемый для хранения базиса подпространства Крылова в методе GMRES, поскольку хранить и пересылать требуется только граничные переменные. После того, как решение системы найдено, искомое поле \vec{E} восстанавливается в каждой из подобластей.

5. Численные эксперименты

5.1. Модельная задача

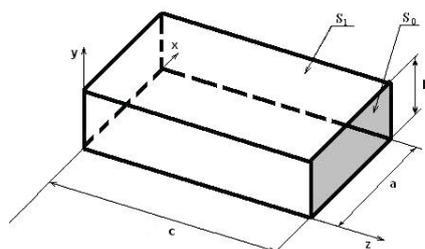


Рис. 3. Расчетная область для модельной задачи

В качестве одного из тестов рассматривалась модельная задача с расчетной областью в виде волновода с линейными размерами $a = 72$, $b = 34$, $c = 200$ мм (см. рис. 3. Параметры задачи: $\mu_r = 1$, $\hat{\epsilon}_r = 1 - 0.1i$, частота $\omega = 6\pi \cdot 10^9$ Гц. Граничные условия: на грани $z = 200$ задавалось условие

$$\vec{E}_0 \times \vec{n} = \vec{e}_y \sin(\pi x/a),$$

на остальных гранях — условие металлической стенки ($\vec{E}_0 = 0$). Аналитическое решение

$$\vec{E} = \vec{e}_y \sin\left(\frac{\pi x}{a}\right) \frac{\sin \gamma z}{\sin \gamma c}.$$

При решении СЛАУ критерием остановки итерационного процесса служило выполнение условия $(r, r) < \varepsilon^2(f, f)$, где f — вектор правой части системы, $r = f - Au$ — вектор невязки. В проведенных экспериментах ε было выбрано равным 10^{-7} .

Для построения сетки расчетная область делилась по каждому из измерений на некоторое число шагов, и каждый из получающихся параллелепипедов делился на 6 тетраэдров. В тестах использовалась квазиравномерная сетка с разбиениями по каждой из координат с некоторым шагом h . Были рассмотрены сетки с разбиениями $4 \times 2 \times 10$, $8 \times 4 \times 20$ и $15 \times 7 \times 40$. Для аппроксимации использовались базисные функции второго порядка.

Тесты проводились на двух системах: системе Intel Core2 Duo E6550 @ 2.33 ГГц, 2 ядра, с оперативной памятью объемом 4 ГБ, и на узлах HP BL2x220 G6 кластера НКС-30Т. Каждый узел — Intel Xeon E5540, 2.53 ГГц, 2x4 ядер, 16 ГБ памяти.

В табл. 1 приведены результаты тестирования решателя при использовании геометрической декомпозиции. В таблице пргос — число MPI процессов (и подобластей), N — порядок системы для пргос=1, N_b — размерность пространства следов, N_p — максимальный порядок СЛАУ для подобластей, n — число итераций решателя GMRES,

δE — относительная ошибка для поля \vec{E} в барицентрах тетраэдров. Тесты для сетки $4 \times 2 \times 10$ при 16, 32 и 64 подобластях не проводились, так как при этом не хватает тетраэдров сетки для построения нужного числа подобластей. Тесты проводились на двухъядерном Core2 Duo, поэтому время работы решателей не приводится.

Таблица 1

Решение модельной задачи с геометрической декомпозицией

Сетка	N	δE		прос					
				2	4	8	16	32	64
$4 \times 2 \times 10$	2392	3,1e-02	n	8	11	22	—	—	—
			N_b	136	408	2656	—	—	—
			N_p	1408	916	670	—	—	—
$8 \times 4 \times 20$	21664	7,3e-03	n	9	14	23	40	53	70
			N_b	592	1776	4556	15764	40396	105200
			N_p	11782	6292	4522	2610	1834	1058
$15 \times 7 \times 40$	149874	2,1e-03	n	10	17	27	50	64	80
			N_b	2012	6036	14084	34560	67872	143256
			N_p	78206	40486	21626	12818	8346	5478

В табл. 2 приведены результаты тестирования для случая алгебраической декомпозиции. Параметр пересечения подобластей был установлен в 4, то есть подобласти пересекались, если это было возможно, по 4 фронтам. Из таблицы видно, что при декомпозиции матрицы на слишком большое число подобластей число итераций начинает быстро расти. В этих случаях на каждую подобласть приходится всего по несколько фронтов.

Таблица 2

Решение модельной задачи с алгебраической декомпозицией

Сетка	N	δE		прос				
				2	4	8	16	32
$4 \times 2 \times 10$	2392	3,1e-02	n	6	12	49	—	—
			N_b	378	980	2261	—	—
			N_p	1542	1334	796	—	—
$8 \times 4 \times 20$	21664	7,3e-03	n	9	13	21	258	—
			N_b	1540	4678	10708	23052	—
			N_p	12420	9232	6306	3200	—
$15 \times 7 \times 40$	149874	2,1e-03	n	13	18	25	38	517
			N_b	5210	16646	37670	78446	162720
			N_p	81132	49618	31168	22678	11282

5.2. Задача электромагнитного каротажа

Вторая из рассматриваемых задач — задача электромагнитного каротажа. Расчетная область представляет из себя куб с центром в начале координат со стороной 10 метров. Куб заполнен средой с $\sigma = 0,1$ См/м, в среде имеется слой с $\sigma = 0,05$

См/м. Координаты слоя по оси Oz : от $-0,425$ м до $-0,275$ м. В среде пробурена вертикальная цилиндрическая скважина радиуса $0,108$ м с центром в начале координат в плоскости Oxy . В скважине имеется цилиндрическая каверна внешнего радиуса $0,118$ м и положением по оси Oz от $-0,0725$ до $0,0725$ м. Проводимость скважины и каверны $\sigma = 5$ См/м. В скважину вставлен полый цилиндрический прибор радиуса $0,043$ м, смещенный по оси x относительно центра скважины на $-0,064$. Проводимость прибора равна нулю. В приборе имеется одна генераторная петля при $z = 0,5$ м и две приемные петли при $z = 0,0$ и $z = 0,1$ м. Радиус петель $-0,0365$ м. По генераторной петле течет ток 1 А с частотой 14 МГц. Искомой является разность фаз ЭДС в приемниках.

Генерация неравномерной сетки проводилась при помощи утилиты NETGEN v4.9.13 [10]. В результате была получена сетка с 71892 узлами и 388836 тетраэдрами.

Результаты решения задачи на кластере с использованием алгебраической декомпозиции представлены в табл. 3. В ней дополнительно указаны времена t_{fact} — максимальное время факторизации матриц D_p в подобластях и t_{tot} — общее время работы решателя. Результаты тестирования для $\text{prroc} < 4$ не приведены, так как в этом случае PARDISO не хватает 16 Гб памяти на узле для хранения LU факторов матрицы, а режим Out-of-Core с хранением факторов на диске является слишком медленным.

Таблица 3

Решение задачи каротажа с алгебраической декомпозицией матрицы

prroc	N	N_b	t_{fact}, c	t_{tot}, c	n
4	2398750	371126	$8,52e+02$	$9,49e+02$	21
8	2398750	833744	$3,30e+02$	$4,26e+02$	28

Результат решения задачи показал хорошее соответствие с результатом группы НГТУ, которая решала эту задачу другим методом — методом выделения поля источника, при этом искомая разность фаз ЭДС в приемниках совпала с ошибкой 1.8% . Стоит отметить также, что для данной задачи обычные итерационные алгоритмы в подпространствах Крылова, примененные к СЛАУ для всей подобласти, показывают отсутствие сходимости.

Заключение

В работе рассматривается задача моделирования электромагнитного поля в частотной области. Предложено два эффективных алгоритма геометрической и алгебраической декомпозиции задачи, причем последний может применяться и для решения других задач. Предлагаемые итерационные алгоритмы в подпространствах следов с предобуславливателем Шварца требуют невысоких коммуникационных затрат и хорошо подходят для вычислительных систем с распределенной памятью. Реализация алгоритмов на языке C++ с использованием средств MPI продемонстрировала хорошую производительность на методических и практических задачах. Предложенные решатели позволили решить задачу электромагнитного каротажа с высокой точностью.

Работа выполнена при поддержке РФФИ (грант 11-01-00205).

Литература

1. Karypis, G. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karypis, V. Kumar // *SIAM Journal on Scientific Computing*. – 1999. – Vol. 20, № 1. – P. 359–392.
2. Saad, Y. *Iterative Methods for Sparse Linear Systems, Second Edition*. / Y. Saad – Society for Industrial and Applied Mathematics, 2003.
3. Monk, P. *Finite Element Methods for Maxwell's Equations*. / P. Monk – Oxford University Press, 2003.
4. Соловейчик, Ю.Г. Метод конечных элементов для решения скалярных и векторных задач / Ю.Г. Соловейчик, М.Э. Рояк, М.Г. Персова – Новосибирск: Изд-во НГТУ, 2007.
5. Ingelstrom, P. A new set of H(curl)-conforming hierarchical basis functions for tetrahedral meshes / P. Ingelstrom // *IEEE Transactions on Microwave Theory and Techniques*. – 2006. – Vol. 54, № 1. – P. 160–114.
6. Ильин, В.П. Методы и технологии конечных элементов. / В.П. Ильин – Новосибирск: Изд-во ИВМиМГ СО РАН, 2007.
7. Fuchs, H. On visible surface generation by a priori tree structures / H. Fuchs, Z.M. Kedem, B.F. Naylor // *ACM Computer Graphics*. – 1980. – Vol. 14, № 3. – P. 124–133.
8. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест – М., МЦНМО: БИНОМ. Лаборатория знаний, 2004.
9. Intel. The Flagship High-Performance Computing Math Library for Windows*, Linux*, and Mac OS* X. Intel® Math Kernel Library from Intel. URL: <http://software.intel.com/en-us/articles/intel-mkl/> (дата обращения: 22.01.2012)
10. Schöberl, J. NETGEN — An advancing front 2D/3D-mesh generator based on abstract rules / J. Schöberl // *Computing and Visualization in Science*. – 1997. – Vol. 1, № 1. – P. 41–52.

Дмитрий Сергеевич Бутюгин, младший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН, аспирант, Новосибирский государственный университет, dm.butyugin@gmail.com.

ALGORITHMS OF SLAES SOLUTION FOR THE SYSTEMS WITH DISTRIBUTED MEMORY APPLIED TO THE PROBLEMS OF ELECTROMAGNETISM

D.S. Butyugin, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

Paper presents various aspects of harmonic electromagnetic fields simulation on clusters. The major computational complexity comes from the solution of the systems of linear algebraic equations (SLAEs) arising from the approximations of corresponding electromagnetic boundary value problems by Nedelec elements of various orders. Effective and efficient approaches to the decomposition of the computational domain and the matrix of the system are considered. Distributed SLAEs are solved using iterative Krylov subspace methods preconditioned by additive Schwarz method. In order to increase the effectiveness of the algorithms iterations are performed in the trace space. Implementation of the solvers is based on MPI for data transfers. The solution of the systems in subdomains is performed by PARDISO direct solver from Intel® MKL library. Numerical experiments results on a series of model and real-life problems show the effectiveness of the presented algorithms.

Keywords: Maxwell equations, iterative algorithms, domain decomposition methods, additive Schwarz method.

References

1. Karypis G., Kumar V. A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*. 1999. Vol. 20, № 1. P. 359–392.
2. Saad Y. *Iterative Methods for Sparse Linear Systems*, Second Edition. Society for Industrial and Applied Mathematics, 2003.
3. Monk P. *Finite Element Methods for Maxwell's Equations*. Oxford University Press, 2003.
4. Soloveychick Y.G., Royak M.E., Persova M.G. Metod konechnykh elementov dlya resheniya skalarykh i vektornykh zadach [Finite Element Method for the Solution of Scalar and Vector Problems]. Novosibirsk, NSTU Publ., 2007.
5. Ingelstrom P. A New Set of H(curl)-conforming Hierarchical Basis Functions for Tetrahedral Meshes *IEEE Transactions on Microwave Theory and Techniques*. 2006. Vol. 54, № 1. P. 160–114.
6. Il'in V.P. *Metody i tekhnologii konechnykh elementov* [Methods and Technologies of Finite Elements]. Novosibirsk, ICM&MG SBRAS Publ., 2007.
7. Fuchs H., Kedem Z.M., Naylor B.F. On Visible Surface Generation by A Priori Tree Structures *ACM Computer Graphics*. 1980. Vol. 14, № 3. P. 124–133.
8. Cormen T., Leiserson C., Rivest R. *Introduction to Algorithms*. MIT Press, 2001.
9. Intel (R) Math Kernel Library from Intel:
URL: <http://software.intel.com/en-us/articles/intel-mkl/>
10. Schöberl J. NETGEN — an Advancing Front 2D/3D-mesh Generator Based on Abstract Rules *Computing and Visualization in Science*. 1997. Vol. 1, № 1. P. 41–52.

Поступила в редакцию 14 марта 2012 г.