

УДК 517.968.78, 519.642.2
doi: 10.21685/2072-3040-2023-3-4

Численное исследование рассеяния электромагнитной волны неоднородным телом и неплоским идеально проводящим экраном

О. С. Скворцов¹, А. А. Цупак²

^{1,2}Пензенский государственный университет, Пенза, Россия

¹ghj.ghh.13@mail.ru, ²altsupak@yandex.ru

Аннотация. *Актуальность и цели.* Цель работы – разработка, программная реализация и апробация проекционного метода и параллельного алгоритма для решения задачи дифракции электромагнитной волны на системе тел и экранов. *Материалы и методы.* Метод Галеркина реализован для векторного интегро-дифференциального уравнения задачи дифракции; определены базисные вектор-функции на объемном теле и параметризованном неплоском экране; реализован параллельный алгоритм решения задачи с использованием библиотеки MSMPI. *Результаты.* Проведено сравнение решений модельной задачи с ранее опубликованными результатами; проведено исследование внутренней сходимости метода Галеркина; исследовано влияние идеально проводящего экрана на решение в области неоднородности. *Выводы.* Предложенный способ аппроксимации решений на неплоском экране является эффективным методом, существенно расширяющим класс задач дифракции, решаемых методом интегральных уравнений. Проведенные тесты подтвердили высокую эффективность параллельного алгоритма.

Ключевые слова: дифракция электромагнитной волны, неоднородное объемное тело, неплоский экран, система интегро-дифференциальных уравнений, базисные функции, метод Галеркина, параллельный алгоритм

Для цитирования: Скворцов О. С., Цупак А. А. Численное исследование рассеяния электромагнитной волны неоднородным телом и неплоским идеально проводящим экраном // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2023. № 3. С. 46–65. doi: 10.21685/2072-3040-2023-3-4

Numerical study of electromagnetic wave scattering from a non-homogeneous solid and curvilinear perfectly conducting screen

O.S. Skvortsov¹, A.A. Tsupak²

^{1,2}Penza State University, Penza, Russia

¹ghj.ghh.13@mail.ru, ²altsupak@yandex.ru

Abstract. *Background.* The purpose of the work is development, software implementation and testing of a projection method and a parallel algorithm for solving the problem of electromagnetic wave diffraction on a system of solids and screens. *Material and methods.* Galerkin method is implemented for the vector integro-differential equation of the diffraction problem; basis vector functions on a three-dimensional body and a parameterized non-planar screen are determined; parallel algorithm for solving the problem is implemented using the MSMPI library. *Results.* approximate solutions of the model problem are compared with the previously published results; the inner convergence of the Galerkin method is investigated; dependence of the solution in the area of inhomogeneity on a perfectly conducting screen is investigated. *Conclusions.* The proposed method of approximation on a curvilinear screen is an effective method that significantly expands the class of diffraction prob-

lems solved by integral equations method; numerical tests confirmed high efficiency of the parallel algorithm.

Keywords: electromagnetic wave diffraction, inhomogeneous solid, curvilinear screen, system of integro-differential equations, basis functions, Galerkin method, parallel algorithm

For citation: Skvortsov O.S., Tsupak A.A. Numerical study of electromagnetic wave scattering from a non-homogeneous solid and curvilinear perfectly conducting screen. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences.* 2023;(3):46–65. (In Russ.). doi: 10.21685/2072-3040-2023-3-4

Введение

В работе описан численный метод и параллельный алгоритм решения векторной задачи рассеяния монохроматической электромагнитной волны препятствием, состоящим из объемного неоднородного тела и непересекающегося с ним неплоского идеально проводящего экрана.

Эта задача является хорошо исследованной теоретически [1–4]. Ее численному решению в случае плоских (или кусочно-плоских) экранов посвящено достаточно большое число работ [5–8]. Однако до настоящего времени не был рассмотрен случай, когда экран представляет собой гладкую криволинейную поверхность. Основной трудностью здесь является определение базисных вектор-функций. Нетривиальны определение и вычисление матричных элементов в методе Галеркина, отвечающих случаю, когда базисные и/или тестовые функции определены на неплоском экране.

Высокая трудоемкость метода Галеркина, необходимость вычисления большого числа многомерных интегралов требуют разработки параллельного алгоритма для решения задач дифракции с приемлемой точностью. Еще одна трудность связана с возможной неравномерностью нагрузки процессов на этапе заполнения матрицы системы алгебраических уравнений (СЛАУ): это обусловлено определением элементов матрицы через интегралы разной кратности (четырёх-, пяти- или шестикратные). В работе описан метод равномерного распределения нагрузки процессов, эффективный при любом числе базисных функций.

1. Интегро-дифференциальные уравнения задачи дифракции

Задача дифракции монохроматической электромагнитной волны на системе тел Q и экранов Ω , расположенных в \mathbb{R}^3 , сводится [2, 4] к системе векторных интегро-дифференциальных уравнений (ИДУ):

$$\begin{aligned} \xi \mathbf{J} - \left(k_0^2 + \text{grad div} \right) \int_Q G(x, y) \mathbf{J}(y) dy - \\ - \left(k_0^2 + \text{grad div} \right) \int_{\Omega} G(x, y) \mathbf{u}(y) ds_y = \mathbf{E}_0(x), \quad x \in Q, \\ \left(- \left(k_0^2 + \text{grad div} \right) \int_Q G(x, y) \mathbf{J}(y) dy - \right. \\ \left. - \left(k_0^2 + \text{grad div} \right) \int_{\Omega} G(x, y) \mathbf{u}(y) ds_y \right)_{\tau} = \mathbf{E}_{0, \tau}(x), \quad x \in \Omega. \end{aligned} \quad (1)$$

Здесь $\mathbf{E}_{0,\tau}$ – касательная компонента падающего электрического поля, $\mathbf{J}(x) = \left[\frac{\hat{\boldsymbol{\varepsilon}}(x)}{\varepsilon_e} - \hat{\mathbf{I}} \right] \mathbf{E}(x)$ и $G(x, y) = \frac{e^{ik_0|x-y|}}{4\pi|x-y|}$, $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$ – волновое число.

Неплоский идеально проводящий экран Ω задан параметрически с помощью вектор-функции

$$\mathbf{x}(t) = (x_1(t_1, t_2), x_2(t_1, t_2), x_3(t_1, t_2))^T, \quad t \in D, \quad (2)$$

$D \subset \mathbb{R}^2$ – область параметров, $x_k \in C^\infty(\bar{D})$, ранг матрицы Якоби $\frac{\partial \mathbf{x}}{\partial t}$ равен 2.

Ограничения, накладываемые на диэлектрические характеристики среды и рассеивателей, и обеспечивающие однозначную разрешимость (1) и сходимость для него метода Галеркина, подробно описаны в работах [1–4]. Формулировка метода Галеркина является стандартной и описана в многочисленных работах (см., например, [9]).

2. Дискретизация задачи

Для решения системы ИДУ (1) методом Галеркина необходимо построить расчетную сетку на системе тел и экранов.

В статье рассматривается случай, когда система рассеивателей состоит из неплоского экрана Ω – части сферической поверхности, и объемного тела Q , являющегося прямоугольным параллелепипедом.

Для параметрически заданного экрана Ω введем в области параметров $D = [c_1, d_1] \times [c_2, d_2]$ двухиндексную нумерацию $i = (i_1, i_2)$ конечных элементов, которые представляют собой прямоугольники

$$D_{i_1, i_2}^1 = [t_{i_1}; t_{i_1+2}] \times [t_{i_2}; t_{i_2+1}] \quad (i_1 = \overline{0, n-2}, i_2 = \overline{0, n-1}),$$

$$D_{i_1, i_2}^2 = [t_{i_1}; t_{i_1+1}] \times [t_{i_2}; t_{i_2+2}] \quad (i_1 = \overline{0, n-1}, i_2 = \overline{0, n-2}),$$

где $t_{i_k} = c_k + i_k h_k$, $h_k = \frac{d_k - c_k}{n}$, $k = 1, 2$.

Введем скалярные финитные кусочно-линейные функции

$$v_{i_1, i_2}^1(t) = \begin{cases} 1 - \frac{|t_1 - t_{1, i_1+1}|}{h_1}, & t \in D_{i_1, i_2}^1, \\ 0, & t \notin D_{i_1, i_2}^1; \end{cases} \quad v_{i_1, i_2}^2(t) = \begin{cases} 1 - \frac{|t_2 - t_{2, i_2+1}|}{h_2}, & t \in D_{i_1, i_2}^2, \\ 0, & t \notin D_{i_1, i_2}^2, \end{cases}$$

и вектор-функции типа гофтор [10] (их функций подробно описаны в [11]):

$$\mathbf{v}_{i_1, i_2}^{0,1}(t) = (v_{i_1, i_2}^1(t), 0), \quad \mathbf{v}_{i_1, i_2}^{0,2}(t) = (0, v_{i_1, i_2}^2(t)).$$

Так как координаты на экране Ω являются гладкими, то дифференциал

$$d\mathbf{x}(t): T_t D \rightarrow T_x \Omega$$

есть биективное отображение касательных пространств [12, 13].

Базисные вектор-функции $\mathbf{v}_{i_1, i_2}^k(x)$ в точках $x = x(t) = x(t_1, t_2)$ неплоского экрана определим по формуле [14]:

$$\mathbf{v}_{i_1, i_2}^k(x(t)) = d\mathbf{x} \circ \mathbf{v}_{i_1, i_2}^{0, k}(t_1, t_2), \quad x(t) \in \Omega. \quad (3)$$

На рис. 1 представлены графики первой и второй компоненты вектор-функции $\mathbf{v}_{1,1}^1(x(t))$ на экране Ω для случая, когда число разбиения в области параметров $n = 2$.

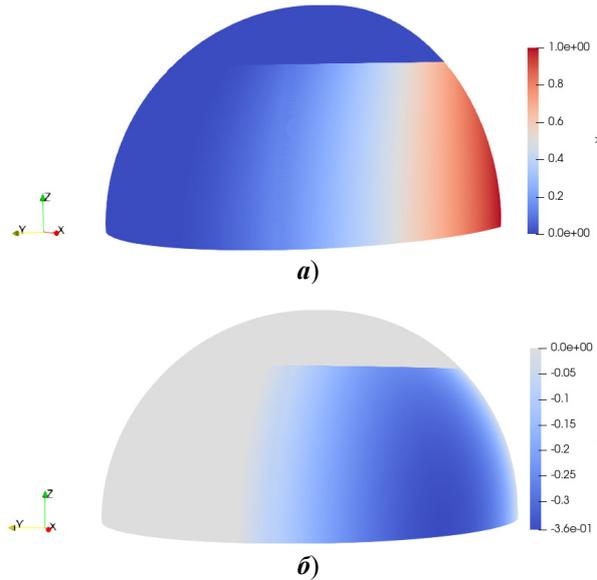


Рис. 1. Первая (а) и вторая (б) компоненты базисной вектор-функции на сферической поверхности

Количество базисных функций на экране при подобном разбиении равно:

$$N_{\Omega} = 2n(n-1).$$

На теле $Q = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ введем трехиндексную нумерацию $j = (j_1, j_2, j_3)$ конечных элементов, представляющих собой параллелепипеды:

$$D_{j_1, j_2, j_3} = [x_{1, j_1}; x_{1, j_1+1}] \times [x_{2, j_2}; x_{2, j_2+1}] \times [x_{3, j_3}; x_{3, j_3+1}],$$

где $j_1 = \overline{0, n-1}$, $j_2 = \overline{0, n-1}$, $j_3 = \overline{0, n-1}$, $x_{k, j_k} = a_k + j_k h_k$, $h_k = \frac{b_k - a_k}{n}$, $k = 1, 2, 3$.

Вводя финитные кусочно-постоянные функции

$$\mu_{j_1, j_2, j_3}(x) = \begin{cases} 1, & x \in D_{j_1, j_2, j_3}, \\ 0, & x \notin D_{j_1, j_2, j_3}, \end{cases}$$

определим базисные вектор-функции на теле по формулам:

$$\mathbf{m}_{j_1, j_2, j_3}^1(x) = (\mu_{j_1, j_2, j_3}(x), 0, 0),$$

$$\mathbf{m}_{j_1, j_2, j_3}^2(x) = (0, \mu_{j_1, j_2, j_3}(x), 0),$$

$$\mathbf{m}_{j_1, j_2, j_3}^3(x) = (0, 0, \mu_{j_1, j_2, j_3}(x)).$$

Количество базисных функций в теле при заданном числе разбиения n равно

$$N_Q = 3n^3.$$

Таким образом, общее количество базисных функций равно

$$N = N_\Omega + N_Q = 3n^3 + 2n(n-1).$$

Представим расширенную матрицу $\widehat{\mathbf{S}} = [\mathbf{S} | \mathbf{f}]$ системы линейных алгебраических уравнений в блочном виде, для удобства введя индексацию базисных функций на экране с помощью верхнего индекса, а на теле – нижнего (рис. 2):

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & | & \mathbf{S}_{1,}^1 & \mathbf{S}_{1,}^2 \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & | & \mathbf{S}_{2,}^1 & \mathbf{S}_{2,}^2 \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & | & \mathbf{S}_{3,}^1 & \mathbf{S}_{3,}^2 \\ - & - & - & - & - & - \\ \mathbf{S}_{,1}^1 & \mathbf{S}_{,2}^1 & \mathbf{S}_{,3}^1 & | & \mathbf{S}^{11} & \mathbf{S}^{12} \\ \mathbf{S}_{,1}^2 & \mathbf{S}_{,2}^2 & \mathbf{S}_{,3}^2 & | & \mathbf{S}^{21} & \mathbf{S}^{22} \end{pmatrix} = \begin{pmatrix} \mathbf{S}_Q & | & \mathbf{S}_{Q,\Omega} \\ - & - & - \\ \mathbf{S}_{\Omega,Q} & | & \mathbf{S}_\Omega \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ - \\ f^1 \\ f^2 \end{pmatrix};$$

Рис. 2. Блочный вид матрицы \mathbf{S}

Элементы правой части системы уравнений, соответствующие телу, вычисляются по формулам:

$$[f_k]_i = \int_Q \mathbf{E}_0(x) \mathbf{m}_{i_1, i_2, i_3}^k(x) dx = \int_{D_{i_1, i_2, i_3}} \mathbf{E}_0(x) \mathbf{m}_{i_1, i_2, i_3}^k(x) dx, \quad k = 1, 2, 3,$$

соответствующие экрану – по формулам:

$$[f^k]_i = \int_\Omega \mathbf{E}_{0,\tau}(x) \mathbf{v}(x) ds_x, \quad k = 1, 2. \tag{4}$$

Элементы матрицы, соответствующие базисным и тестовым функциям на теле (верхний левый блок), получаются путем вычисления трех- и шести-кратных интегралов:

$$[S_{kl}]_{ij} = \int_Q \xi(x) \mathbf{m}_{i_1, i_2, i_3}^k \mathbf{m}_{j_1, j_2, j_3}^l dx -$$

$$\begin{aligned}
 & - \int_Q \left(k_0^2 \int_Q G(x, y) \mathbf{m}_{j_1, j_2, j_3}^l dy + \text{grad div} \int_Q G(x, y) \mathbf{m}_{j_1, j_2, j_3}^l dy \right) \mathbf{m}_{i_1, i_2, i_3}^k(y) dx = \\
 & = \int_Q \xi(x) \mu_{i, j}^{k, l} dx - \int_{D_{i_1, i_2, i_3}} \left(k_0^2 \int_{D_{j_1, j_2, j_3}} G(x, y) \mathbf{m}_{j_1, j_2, j_3}^l(y) dy \right) \mathbf{m}_{i_1, i_2, i_3}^k(x) dx - \\
 & - \int_{D_{i_1, i_2, i_3}} \left(\frac{\partial}{\partial x_k} \int_{D_{j_1, j_2, j_3}} \frac{\partial}{\partial x_l} G(x, y) \mu_{j_1, j_2, j_3}(y) dy \right) \mu_{i_1, i_2, i_3}(x) dx, \quad k, l = 1, 2, 3.
 \end{aligned}$$

Элементы матрицы, соответствующие базисным и тестовым функциям на экране, вычисляются следующим образом:

$$\begin{aligned}
 [S^{kl}]_{ij} = & - \int_{\Omega} \left(k_0^2 \int_{\Omega} G(x, y) \mathbf{v}_{j_1, j_2}^l(y) ds_y + \right. \\
 & \left. + \text{grad div} \int_{\Omega} G(x, y) \mathbf{v}_{j_1, j_2}^l(y) dy \right)_{\tau} \mathbf{v}_{i_1, i_2}^k(x) ds_x, \quad k, l = 1, 2. \quad (5)
 \end{aligned}$$

Внедиагональные блоки $\mathbf{S}_{Q, \Omega}$, $\mathbf{S}_{\Omega, Q}$ матрицы отвечают за взаимодействие полей на теле и экране; элементы матрицы в данных блоках получают путем вычисления пятикратных интегралов. Для $\mathbf{S}_{Q, \Omega}$ используем выражения:

$$\begin{aligned}
 [S_l^k]_{ij} = & \int_Q \left(k_0^2 \int_{\Omega} G(x, y) \mathbf{v}_{j_1, j_2}^l(y) dy + \right. \\
 & \left. + \text{grad div} \int_{\Omega} G(x, y) \mathbf{v}_{j_1, j_2}^l(y) ds_y \right) \cdot \mathbf{m}_{i_1, i_2, i_3}^k(x) dx, \quad k = 1, 2, 3, \quad l = 1, 2. \quad (6)
 \end{aligned}$$

Элементы блока $\mathbf{S}_{\Omega, Q}$ вычисляются следующим образом:

$$\begin{aligned}
 [S_k^l]_{ij} = & - \int_{\Omega} \left(k_0^2 \int_Q G(x, y) \mathbf{m}_{j_1, j_2, j_3}^l(y) dy + \right. \\
 & \left. + \text{grad div} \int_Q G(x, y) \mathbf{m}_{j_1, j_2, j_3}^l(y) dy \right)_{\tau} \mathbf{v}_{i_1, i_2}^k(x) ds_x, \quad k = 1, 2, \quad l = 1, 2, 3. \quad (7)
 \end{aligned}$$

Формулы, содержащие интегралы по криволинейному экрану, требуют более детального объяснения. Так, из определения базисных функций по

формуле (3) и выражения для интеграла по поверхности через повторный интеграл по области параметров получим

$$[f^k]_i = \int_{D_{i_1, i_2}^k} \mathbf{E}_{0, \tau}(x(t)) \mathbf{v}_{i_1, i_2}^k(x(t)) \sqrt{g(t)} dt,$$

где $g(t) = \det g_{ij}(t) = g_{11}(t)g_{22}(t) - g_{12}^2(t)$ (g_{ij} – метрический тензор [12]).

Далее операцию div в (5) можно внести под знак интеграла [15], а операцию grad – перебросить на тестовую функцию $\mathbf{v}_{i_1, i_2}^k(x)$, так как ее нормальная равна нулю на границе области интегрирования. Таким образом, формула (5) примет вид

$$\begin{aligned} [S^{kl}]_{ij} &= -k_0^2 \iint_{\Omega \Omega} G(x, y) \mathbf{v}_{j_1, j_2}^l(y) \mathbf{v}_{i_1, i_2}^k(x) ds_y ds_x + \\ &+ \iint_{\Omega \Omega} G(x, y) \text{div} \mathbf{v}_{j_1, j_2}^l(y) \text{div} \mathbf{v}_{i_1, i_2}^k(x) ds_y ds_x = \\ &= \int_{D_{i_1, i_2}^k} \int_{D_{j_1, j_2}^l} G(x(t), y(s)) \sqrt{g(t)} \sqrt{g(s)} \left(-k_0^2 \mathbf{v}_{j_1, j_2}^l(y(s)) \cdot \mathbf{v}_{i_1, i_2}^k(x(t)) + \right. \\ &\left. + g^{ij}(s) \partial_{s_i} \mathbf{v}_{j_1, j_2}^l(y(s)) \partial_{s_j} \mathbf{x}(s) \ g^{ij}(t) \partial_{t_i} \mathbf{v}_{i_1, i_2}^k(x(t)) \partial_{t_j} \mathbf{x}(t) \right) dt ds. \end{aligned}$$

Интегралы по экрану в формулах (6) и (7) преобразуются аналогично.

3. Программная реализация метода Галеркина

Выше вводилась индексация для конечных элементов (и отвечающих им базисных функций) в трехмерном пространстве и двумерной области, параметризующей экран. Ниже представлены формулы, связывающие значения индексов и мультииндексов.

Для сеток на теле Q формулы имеют вид

$$\begin{cases} k = \left[\frac{i}{n^3} \right] + 1, \\ i1 = \left[\frac{(i \bmod(n^3))}{n^2} \right], \\ i2 = \left[\frac{(i \bmod(n^3))}{n} \right] - n \left[\frac{(i \bmod(n^3))}{n^2} \right], \\ i3 = (i \bmod(n^3)) \cdot \text{mod}(n), \end{cases}$$

где $\text{mod}(\cdot)$ – взятие остатка при делении; операция $[\cdot]$ – взятие целой части при делении.

Индексация на экране Ω задается следующим образом:

$$\left\{ \begin{array}{l} k = \left\lceil \frac{(i \bmod(n^3))}{n^2 - n} \right\rceil + 1, \\ i1 = \left\lceil \frac{(i \bmod(n^3))}{n} \right\rceil, \quad (\text{при } k = 1) \\ i2 = (i \bmod(n^3)) \cdot \bmod(n), \quad (\text{при } k = 1) \\ i1 = \left\lceil \frac{(i \bmod(n^3))}{n - 1} \right\rceil - n, \quad (\text{при } k = 2) \\ i2 = (i \bmod(n^3)) \cdot \bmod(n - 1), \quad (\text{при } k = 2) \end{array} \right.$$

Для ускорения работы программы была использована библиотека MPI (msmpi) [16]. Хранение матрицы осуществляется на различных процессах в ленточном виде. Вычисление элементов матрицы осуществлялось согласно схеме, представленной на рис. 3.

$$S = \begin{array}{c} \left(\begin{array}{cccccccccccc|cccc} S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} & S_{1,5} & S_{1,6} & S_{1,7} & S_{1,8} & S_{1,9} & & S_{1,10} & S_{1,11} & S_{1,12} & S_{1,13} \\ S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} & S_{2,5} & S_{2,6} & S_{2,7} & S_{2,8} & S_{2,9} & & S_{2,10} & S_{2,11} & S_{2,12} & S_{2,13} \\ S_{3,1} & S_{3,2} & S_{3,3} & S_{3,4} & S_{3,5} & S_{3,6} & S_{3,7} & S_{3,8} & S_{3,9} & & S_{3,10} & S_{3,11} & S_{3,12} & S_{3,13} \\ S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & S_{4,5} & S_{4,6} & S_{4,7} & S_{4,8} & S_{4,9} & & S_{4,10} & S_{4,11} & S_{4,12} & S_{4,13} \\ S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5} & S_{5,6} & S_{5,7} & S_{5,8} & S_{5,9} & & S_{5,10} & S_{5,11} & S_{5,12} & S_{5,13} \\ S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} & S_{6,5} & S_{6,6} & S_{6,7} & S_{6,8} & S_{6,9} & & S_{6,10} & S_{6,11} & S_{6,12} & S_{6,13} \\ S_{7,1} & S_{7,2} & S_{7,3} & S_{7,4} & S_{7,5} & S_{7,6} & S_{7,7} & S_{7,8} & S_{7,9} & & S_{7,10} & S_{7,11} & S_{7,12} & S_{7,13} \\ S_{8,1} & S_{8,2} & S_{8,3} & S_{8,4} & S_{8,5} & S_{8,6} & S_{8,7} & S_{8,8} & S_{8,9} & & S_{8,10} & S_{8,11} & S_{8,12} & S_{8,13} \\ S_{9,1} & S_{9,2} & S_{9,3} & S_{9,4} & S_{9,5} & S_{9,6} & S_{9,7} & S_{9,8} & S_{9,9} & & S_{9,10} & S_{9,11} & S_{9,12} & S_{9,13} \\ - & - & - & - & - & - & - & - & - & & - & - & - & - \\ S_{10,1} & S_{10,2} & S_{10,3} & S_{10,4} & S_{10,5} & S_{10,6} & S_{10,7} & S_{10,8} & S_{10,9} & & S_{10,10} & S_{10,11} & S_{10,12} & S_{10,13} \\ S_{11,1} & S_{11,2} & S_{11,3} & S_{11,4} & S_{11,5} & S_{11,6} & S_{11,7} & S_{11,8} & S_{11,9} & & S_{11,10} & S_{11,11} & S_{11,12} & S_{11,13} \\ S_{12,1} & S_{12,2} & S_{12,3} & S_{12,4} & S_{12,5} & S_{12,6} & S_{12,7} & S_{12,8} & S_{12,9} & & S_{12,10} & S_{12,11} & S_{12,12} & S_{12,13} \\ S_{13,1} & S_{13,2} & S_{13,3} & S_{13,4} & S_{13,5} & S_{13,6} & S_{13,7} & S_{13,8} & S_{13,9} & & S_{13,10} & S_{13,11} & S_{13,12} & S_{13,13} \end{array} \right) \end{array}$$

Рис. 3. Пример распределения вычисления строк матрицы на четырех процессах

На рис. 3 изображена блочная матрица S порядка 13 (порядки матриц, получившиеся при выполнении реальных вычислений, существенно выше и приведены в табл. 1). Также на рис. 3 опущены (для простоты) индексы k и l и введена сквозная нумерация элементов. Вертикальные и горизонтальные штрихи отделяют блоки на теле Q , экране Ω и т.д.

Потребность в данном способе заполнения матрицы возникает из-за сильного перевеса по времени в вычислении блока на экране (см. рис. 3).

Заполнение на последнем процессе происходит медленнее всего. Так как в конце заполнения матрицы (до начала решения СЛАУ) отработавшие процессы будут ждать завершения счета на последнем, то заполнение матрицы завершится за 159 с, что, очевидно, не является оптимальным вариантом.

Таблица 1

Время заполнения матрицы на 10 процессах, $n = 10$

| | | | | | | | | | | |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Rank | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| t | 44,33 | 44,33 | 44,33 | 44,33 | 44,33 | 44,33 | 44,33 | 44,33 | 44,33 | 158,7 |

Для оптимизации заполнения матрицы СЛАУ введем понятие веса.

Определение. Под весом элемента $s_{i,j}$ будем понимать время его расчета одним процессом и обозначать $P(s_{i,j})$.

Тогда весом строки с номером i можно назвать время, за которое одним процессом заполняется i -я строка. Этот вес вычисляется по формуле

$$P(s_i) = P\left(\sum_{j=0}^{N-1} s_{i,j}\right) = \sum_{j=0}^{N-1} P(s_{i,j}).$$

Далее, естественно считать, что

$$P(s_i \cup s_i \cup s_i) = P(s_i + s_i + s_i) = P(3 \cdot s_i) = 3 \cdot P(s_i).$$

Из вышесказанного получаем, что функция P линейна.

Возвращаясь к рис. 3, сформулируем полученную проблему иначе:

$$P(S_Q + S_{\Omega,Q}) \ll P(S_{\Omega,Q} + S_{\Omega}). \quad (8)$$

Как видно из рис. 3, зачастую возникает ситуация, когда количество ядер компьютера намного меньше размерности матрицы. Тогда блоки $S_{\Omega,Q} + S_{\Omega}$ вычисляются целиком на одном процессе ($rank$), и в силу неравенства (8) имеем

$$P(rank_{last}) \gg P\left(\bigcup_i^{last-1} rank_i\right).$$

Это приводит к замедлению работы программы.

Исходя из определения матричных элементов через многомерные слабосингулярные интегралы выводим соотношение

$$P(S_Q) < P(S_{\Omega,Q}) < P(S_{\Omega}), \quad (9)$$

также для большинства задач будет выполняться и

$$P(S_Q) + P(S_{\Omega,Q}) < P(S_{\Omega}).$$

Тогда в силу (9), распределяя вычисление блока S_{Ω} приблизительно поровну между всеми процессами, невозможно добиться выполнения условия

$$P(rank_1) \approx P(rank_2) \approx \dots \approx P(rank_{last}). \quad (10)$$

Для выполнения (10) реализован следующий алгоритм действий (отметим, что $P(s_{\Omega})$ не учитывается):

1. Распределим строки матрицы по процессам, не обращая внимания на неравенство (8).

2. Предположим, что блоки $S_{\Omega,Q}$ и S_{Ω} вычисляются на последнем процессе $rank_{last}$ целиком.

3. Вычислим приблизительный вес блока $S_{\Omega,Q}$ по формуле

$$P(S_{\Omega,Q}) \approx 2n(n-1) \cdot 3n^3 \cdot P\left(s_{i,j}^{\Omega,Q}\right),$$

где $s_{i,j}^{\Omega,Q}$ – любой элемент блока $S_{\Omega,Q}$.

4. Выберем произвольный процесс, например первый. Рассчитаем его вес $P(rank_1)$, вычислим сначала вес одной строки на первом процессе по формуле

$$P(s_i) \approx 3n^3 \cdot P\left(s_{i,j}^Q\right) + 2n(n-1) \cdot P\left(s_{i,j}^{Q,\Omega}\right),$$

тогда

$$P(rank_1) \approx P(s_i) \cdot cor_1,$$

где cor_1 – количество строк на первом процессе.

5. Найдём вес последнего процесса:

$$P(rank_{last}) \approx P(S_{\Omega,Q}) + (cor_{last} - 2n(n-1)) \cdot P(s_i),$$

если $(cor_{last} - 2n(n-1)) \geq 0$, cor_{last} – количество строк на последнем процессе.

6. Выясним, какое количество строк не хватает последнему процессу для того, чтобы его вес был приблизительно равным весу первого процесса по формуле, назовем это число Δ :

$$\Delta = \frac{P(rank_1) - P(rank_{last})}{P(s_i)}.$$

7. Равномерно вычтем Δ строк из всех процессов, за исключением последнего, к последнему прибавим Δ строк:

$$\begin{cases} cor_1 = cor_1 - \left\lfloor \frac{\Delta}{size} \right\rfloor - \Delta \bmod(size), \\ cor_2 = cor_2 - \left\lfloor \frac{\Delta}{size} \right\rfloor - \Delta \bmod(size), \\ \vdots \\ cor_{last} = cor_{last} + \Delta. \end{cases}$$

Реализуем алгоритм выше и выведем полученное время заполнения матрицы.

По данным в табл. 2 видим, как и ожидалось, равномерное по времени заполнение матрицы различными процессами. Наибольшая разница в весе

между процессами составляет менее 0,01 с. Очевидно, что время заполнения матрицы на первых 9 процессах увеличилось в сравнении с исходным (см. табл. 1) в 1,41 раза, так как что блок S_{Ω} на экране теперь вычисляется на различных процессах.

Таблица 2

Оптимизированное время заполнения матрицы на 10 процессах после равномерного распределения S_{Ω} по процессам и применения алгоритма, $n = 10$

| Rank | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| t_{opt} | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 | 62,94 |

Также стоит отметить, что хоть вес первых 9 процессов и увеличился в 1,41 раза, вес последнего процесса уменьшился приблизительно в 2,52 раза, что позволяет сказать, что общая скорость заполнения матрицы увеличилась приблизительно в 2,52 раза.

Таким образом, разработан алгоритм эффективного распределения вычислений между процессами на этапе заполнения матрицы СЛАУ. В силу зависимости результата работы алгоритма только от веса вычисляемых элементов данная реализация позволяет равномерно распределять вычислительную нагрузку между процессами при различных значениях порядка матрицы и различном соотношении числа базисных функций на экране и теле – см. табл. 3 и 4 (во второй строке таблиц курсивом выделены наименьшее и наибольшее время нагрузки процесса при неоптимизированном заполнении матрицы).

Таблица 3

Время заполнения матрицы до (t) и после (t_{opt}) оптимизации на 8 процессах, число разбиений для сеток $n = 8$

| Rank | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------|------|------|------|------|------|-------------|------|-------------|
| t | 16,9 | 14,6 | 15,3 | 13 | 16,1 | 10,2 | 15,5 | 60,3 |
| t_{opt} | 22,9 | 22,9 | 22,9 | 22,9 | 22,9 | 22,9 | 22,9 | 22,9 |

Таблица 4

Время заполнения матрицы до (t) и после (t_{opt}) оптимизации на 14 процессах, число разбиений для сеток $n = 14$

| Rank | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|-----|------------|
| t | 198 | 166 | 183 | 184 | 170 | 144 | 193 | 180 | 185 | 121 | 187 | 176 | 183 | 658 |
| t_{opt} | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 | 276 |

Перераспределение вычислений элементов, отвечающих экрану, будет осуществляться согласно схеме на рис. 4. С каждого из четырех процессов строки отправляются на последний процесс (в общем случае может отправляться несколько строк с процесса).

$$S = \begin{pmatrix}
 \begin{array}{cccccccc|cccc}
 S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} & S_{1,5} & S_{1,6} & S_{1,7} & S_{1,8} & S_{1,9} & S_{1,10} & S_{1,11} & S_{1,12} & S_{1,13} \\
 S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} & S_{2,5} & S_{2,6} & S_{2,7} & S_{2,8} & S_{2,9} & S_{2,10} & S_{2,11} & S_{2,12} & S_{2,13} \\
 S_{3,1} & S_{3,2} & S_{3,3} & S_{3,4} & S_{3,5} & S_{3,6} & S_{3,7} & S_{3,8} & S_{3,9} & S_{3,10} & S_{3,11} & S_{3,12} & S_{3,13} \\
 S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & S_{4,5} & S_{4,6} & S_{4,7} & S_{4,8} & S_{4,9} & S_{4,10} & S_{4,11} & S_{4,12} & S_{4,13} \\
 S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5} & S_{5,6} & S_{5,7} & S_{5,8} & S_{5,9} & S_{5,10} & S_{5,11} & S_{5,12} & S_{5,13} \\
 S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} & S_{6,5} & S_{6,6} & S_{6,7} & S_{6,8} & S_{6,9} & S_{6,10} & S_{6,11} & S_{6,12} & S_{6,13} \\
 S_{7,1} & S_{7,2} & S_{7,3} & S_{7,4} & S_{7,5} & S_{7,6} & S_{7,7} & S_{7,8} & S_{7,9} & S_{7,10} & S_{7,11} & S_{7,12} & S_{7,13} \\
 S_{8,1} & S_{8,2} & S_{8,3} & S_{8,4} & S_{8,5} & S_{8,6} & S_{8,7} & S_{8,8} & S_{8,9} & S_{8,10} & S_{8,11} & S_{8,12} & S_{8,13} \\
 S_{9,1} & S_{9,2} & S_{9,3} & S_{9,4} & S_{9,5} & S_{9,6} & S_{9,7} & S_{9,8} & S_{9,9} & S_{9,10} & S_{9,11} & S_{9,12} & S_{9,13} \\
 - & - & - & - & - & - & - & - & - & - & - & - & - \\
 S_{10,1} & S_{10,2} & S_{10,3} & S_{10,4} & S_{10,5} & S_{10,6} & S_{10,7} & S_{10,8} & S_{10,9} & S_{10,10} & S_{10,11} & S_{10,12} & S_{10,13} \\
 S_{11,1} & S_{11,2} & S_{11,3} & S_{11,4} & S_{11,5} & S_{11,6} & S_{11,7} & S_{11,8} & S_{11,9} & S_{11,10} & S_{11,11} & S_{11,12} & S_{11,13} \\
 S_{12,1} & S_{12,2} & S_{12,3} & S_{12,4} & S_{12,5} & S_{12,6} & S_{12,7} & S_{12,8} & S_{12,9} & S_{12,10} & S_{12,11} & S_{12,12} & S_{12,13} \\
 S_{13,1} & S_{13,2} & S_{13,3} & S_{13,4} & S_{13,5} & S_{13,6} & S_{13,7} & S_{13,8} & S_{13,9} & S_{13,10} & S_{13,11} & S_{13,12} & S_{13,13}
 \end{array}
 \end{pmatrix}$$

 Рис. 4. Направление отправки блоков матрицы S для 4 процессов (экран)

Перейдем к заполнению других блоков. Разобьем блок на теле Q матрицы S на подблоки согласно схеме на рис. 5.

$$S = \begin{pmatrix}
 \begin{array}{ccc|ccc|ccc|cccc}
 S_{1,1} & S_{1,2} & S_{1,3} & S_{1,4} & S_{1,5} & S_{1,6} & S_{1,7} & S_{1,8} & S_{1,9} & S_{1,10} & S_{1,11} & S_{1,12} & S_{1,13} \\
 S_{2,1} & S_{2,2} & S_{2,3} & S_{2,4} & S_{2,5} & S_{2,6} & S_{2,7} & S_{2,8} & S_{2,9} & S_{2,10} & S_{2,11} & S_{2,12} & S_{2,13} \\
 S_{3,1} & S_{3,2} & S_{3,3} & S_{3,4} & S_{3,5} & S_{3,6} & S_{3,7} & S_{3,8} & S_{3,9} & S_{3,10} & S_{3,11} & S_{3,12} & S_{3,13} \\
 S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & S_{4,5} & S_{4,6} & S_{4,7} & S_{4,8} & S_{4,9} & S_{4,10} & S_{4,11} & S_{4,12} & S_{4,13} \\
 S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5} & S_{5,6} & S_{5,7} & S_{5,8} & S_{5,9} & S_{5,10} & S_{5,11} & S_{5,12} & S_{5,13} \\
 S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} & S_{6,5} & S_{6,6} & S_{6,7} & S_{6,8} & S_{6,9} & S_{6,10} & S_{6,11} & S_{6,12} & S_{6,13} \\
 S_{7,1} & S_{7,2} & S_{7,3} & S_{7,4} & S_{7,5} & S_{7,6} & S_{7,7} & S_{7,8} & S_{7,9} & S_{7,10} & S_{7,11} & S_{7,12} & S_{7,13} \\
 S_{8,1} & S_{8,2} & S_{8,3} & S_{8,4} & S_{8,5} & S_{8,6} & S_{8,7} & S_{8,8} & S_{8,9} & S_{8,10} & S_{8,11} & S_{8,12} & S_{8,13} \\
 S_{9,1} & S_{9,2} & S_{9,3} & S_{9,4} & S_{9,5} & S_{9,6} & S_{9,7} & S_{9,8} & S_{9,9} & S_{9,10} & S_{9,11} & S_{9,12} & S_{9,13} \\
 - & - & - & - & - & - & - & - & - & - & - & - & - \\
 S_{10,1} & S_{10,2} & S_{10,3} & S_{10,4} & S_{10,5} & S_{10,6} & S_{10,7} & S_{10,8} & S_{10,9} & S_{10,10} & S_{10,11} & S_{10,12} & S_{10,13} \\
 S_{11,1} & S_{11,2} & S_{11,3} & S_{11,4} & S_{11,5} & S_{11,6} & S_{11,7} & S_{11,8} & S_{11,9} & S_{11,10} & S_{11,11} & S_{11,12} & S_{11,13} \\
 S_{12,1} & S_{12,2} & S_{12,3} & S_{12,4} & S_{12,5} & S_{12,6} & S_{12,7} & S_{12,8} & S_{12,9} & S_{12,10} & S_{12,11} & S_{12,12} & S_{12,13} \\
 S_{13,1} & S_{13,2} & S_{13,3} & S_{13,4} & S_{13,5} & S_{13,6} & S_{13,7} & S_{13,8} & S_{13,9} & S_{13,10} & S_{13,11} & S_{13,12} & S_{13,13}
 \end{array}
 \end{pmatrix}$$

 Рис. 5. Различные блоки матрицы S

Тогда, учитывая формулы для матричных элементов и указанное выше свойство ядра, заключаем, что матрица имеет блочно-теплицеву структуру, и блоки, выделенные одним цветом, совпадают. Следовательно сокращения времени заполнения матрицы можно добиться, вычислив только один из совпадающих блоков (выберем только те блоки, которые находятся под главной диагональю) и отправив их на место блоков над главной диагональю (см. рис. 6). Выбор блоков под главной диагональю для реального счета соответствует случаю, когда перебор индексов j базисных функций идет в порядке возрастания. Это позволяет исключить простаивание процессов и потерю производительности. Если перебор матричных элементов идет в порядке убывания индексов, необходимо заполнить блоки, расположенные над главной диагональю.

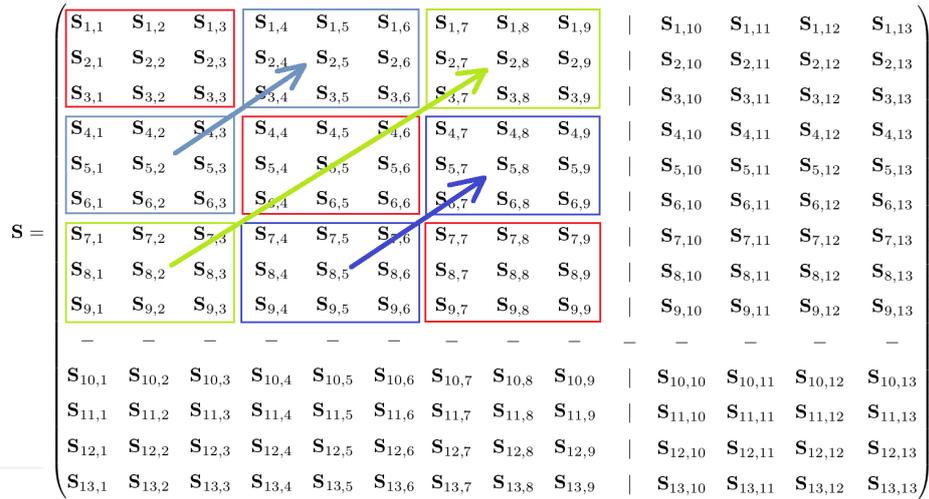


Рис. 6. Направление пересылки блоков матрицы S для 4 процессов

Благодаря легкой геометрии пересылок поиск нужных индексов и, как следствие, реализация алгоритма достаточно просты.

Теперь рассмотрим подробнее вычисление элементов блоков $S_{Q,\Omega}$ и $S_{\Omega,Q}$. Прежде всего можно увеличить скорость заполнения матрицы, вычислив лишь один из эти блоков, так как $S_{Q,\Omega} = S_{\Omega,Q}^T$.

Специфика структуры основной матрицы определяет наиболее выгодный способ хранения матрицы и заполнения блоков. Заметим, что блок $S_{Q,\Omega}$ содержит $3n^3$ строк, а $S_{\Omega,Q} = 2(n^2 - n)$, поэтому выгоднее заполнять блок $S_{Q,\Omega}$, так как он распределен по большему числу процессов (или даже всем за исключением последнего). Заполнение матрицы завершается серией пересылок после вычисления $S_{Q,\Omega}$; схема рассылки приведена на рис. 7.

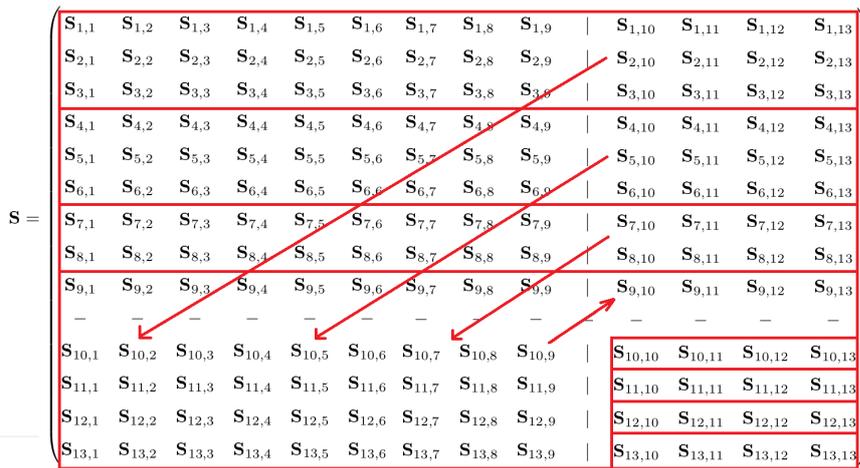


Рис. 7. Пример рассылки элементов блока $S_{Q,\Omega}$ матрицы S для случая 4 процессов

Учитывая пересылку блоков $S_{Q,\Omega}$ и $S_{\Omega,Q}$, скорректируем формулу, определяющую вес блока $P(S_{\Omega,Q})$:

$$P(S_{\Omega,Q}) \approx \frac{P(S_{Q,\Omega})}{\text{size} - 1} \approx \frac{2n(n-1) \cdot 3n^3 \cdot P(s_{i,j}^{Q,\Omega})}{\text{size} - 1}.$$

Замечание. Для решения СЛАУ с расширенной матрицей, хранящейся на нескольких процессах, был реализован параллельный алгоритм метода Гаусса с обнулением элементов только под главной диагональю.

4. Результаты численных экспериментов

Для проверки эффективности применения параллельного алгоритма численного решения системы методом Галеркина был проведен ряд вычислительных экспериментов на персональном компьютере с процессором i9-12900 (16 ядер, 24 потоков), объем доступной для вычислений оперативной памяти – 64 Гб. На рис. 8 приведены сведения о времени решения задачи в зависимости от числа используемых процессов.

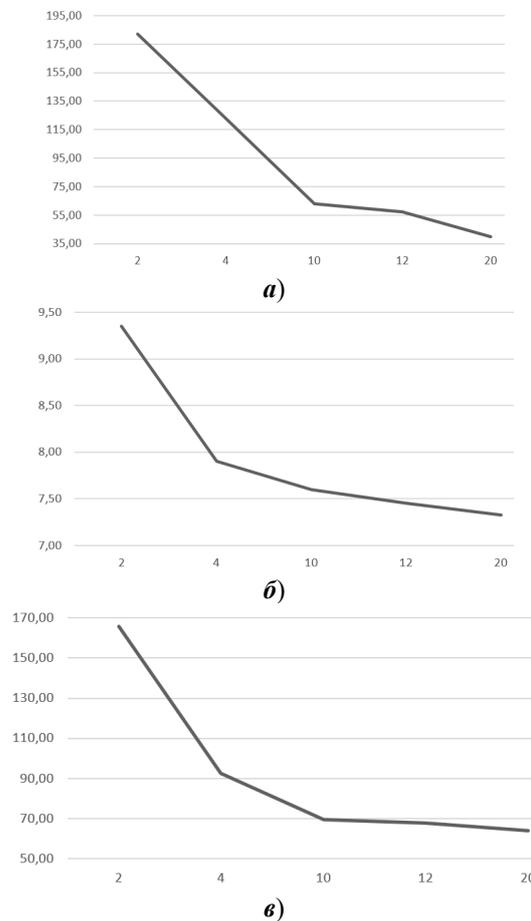


Рис. 8. Зависимость времени заполнения матрицы СЛАУ (a), решения СЛАУ методом Гаусса (b), записи данных в файлы (v) и общего времени решения задачи (z) от числа процессов

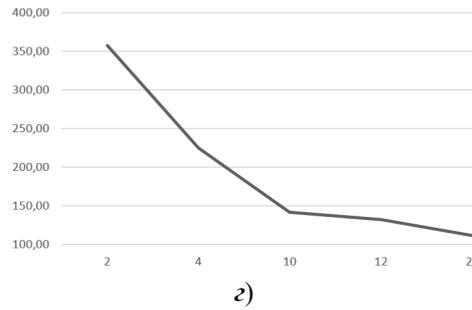


Рис. 8. Окончание

Проведенные эксперименты подтверждают высокую эффективность распараллеливания задачи, как на этапе заполнения матрицы СЛАУ. Это обусловлено прежде всего сравнительно небольшим объемом обмена данными между процессами (особенно при вычислении матричных элементов). Эффективность параллельной реализации метода Гаусса уменьшается с ростом числа процессов в силу все возрастающего числа обменов между процессами.

Следующий тест проводился с целью апробации предложенного в [14] метода аппроксимации решений задачи на криволинейном экране. В качестве экрана был рассмотрен прямоугольник $\Omega = [-0,3575\lambda; 0,3575\lambda]^2$, расположенный в плоскости $x_3 = 0$, падающая волна имеет вид $E_0 = (1, 0, 0)$. Результаты решения задачи дифракции на таком идеально проводящем экране опубликованы в многочисленных статьях (см., например, [17]) и могут считаться референсными. Результаты, представленные на рис. 9, не только подтверждают хорошее совпадение полученных результатов с известным, но и демонстрируют наличие внутренней сходимости.

Перейдем к рассмотрению основного вычислительного эксперимента. В качестве объемного рассеивателя рассмотрен неоднородный куб $Q = \left[-\frac{\lambda}{2}; \frac{\lambda}{2}\right]^3$, а в качестве экрана Ω – четверть сферы $S_\lambda\left(-\frac{\lambda}{2}, 0, 0\right)$, закрывающая верхнюю половину куба. Волновое число $k_0 = \frac{2\pi\nu}{c} + i$, $\nu = 15$ ГГц, c – скорость света в вакууме $\left(\frac{\text{см}}{\text{сек}}\right)$, а диэлектрическая проницаемость куба задается формулой

$$\epsilon_r(x) = \begin{cases} 4 + 2i, & x_2 > 0, \\ 2 + i, & x_2 \leq 0. \end{cases}$$

На рис. 10 приведены графики приближенных решений задачи дифракции для нескольких значений числа разбиений сетки n .

С увеличением числа базисных функций возрастает точность решения, что следует из табл. 5 (здесь $\Delta(\mathbf{u}) = \max |u_{n_1} - u_{n_2}|$).

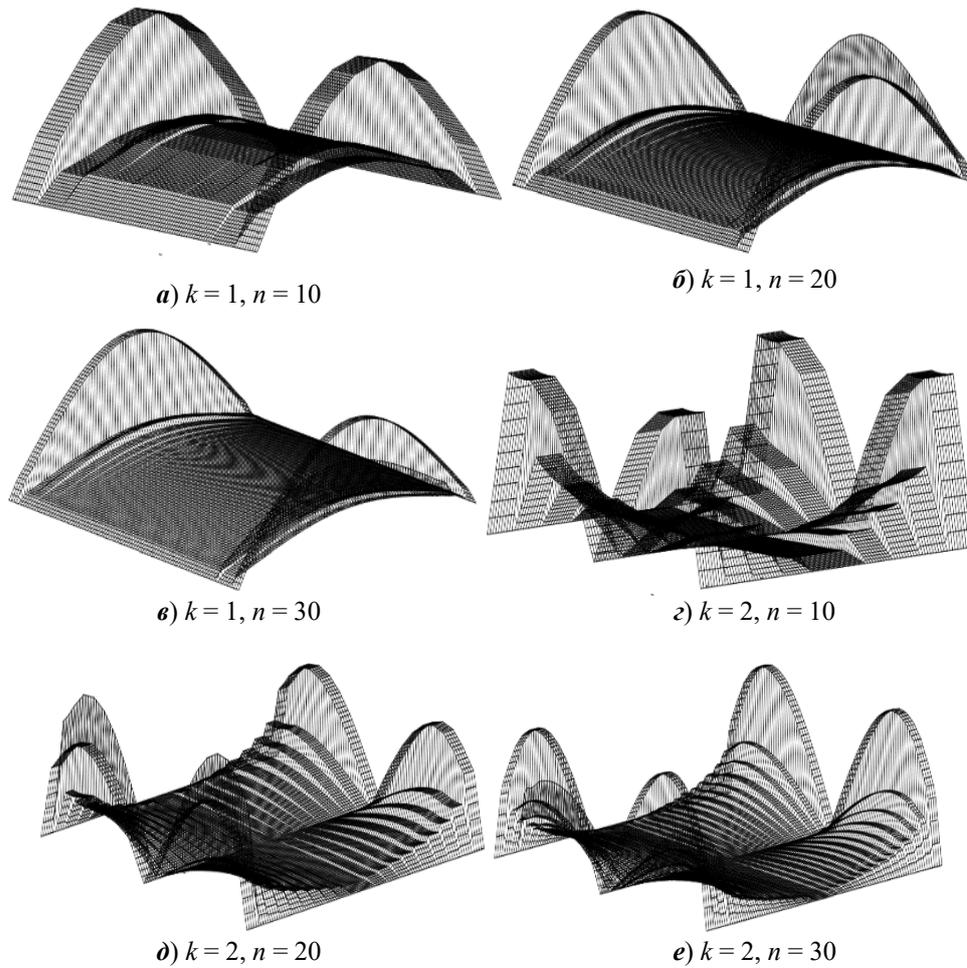


Рис. 9. Графики k -го компонента вектора тока на поверхности плоского экрана при различных значениях числа разбиений сетки n

Таблица 5

Внутренняя сходимость решения на теле

| | $n_1 = 8, n_2 = 10$ | $n_1 = 10, n_2 = 12$ | $n_1 = 12, n_2 = 18$ |
|----------------------|---------------------|----------------------|----------------------|
| $\Delta(\mathbf{u})$ | 0,0200183 | 0,0126882 | 0,0043491 |
| $\Delta(\mathbf{E})$ | 0,135268 | 0,105149 | 0,0764559 |

Положив в основной матрице СЛАУ все элементы блоков $S_{\Omega,Q}$ и $S_{Q,\Omega}$ равными нулю, решим одновременно две задачи дифракции на (отдельно) криволинейном экране и неоднородном кубе. Сравнение результатов (рис. 10 и 11) показывает, что в присутствии идеально проводящего экрана решение в области неоднородности существенно меняется: значение тока поляризации в «экранированной» части куба существенно меньше, чем в «неэкранированной», что полностью согласуется с физической природой моделируемого процесса.

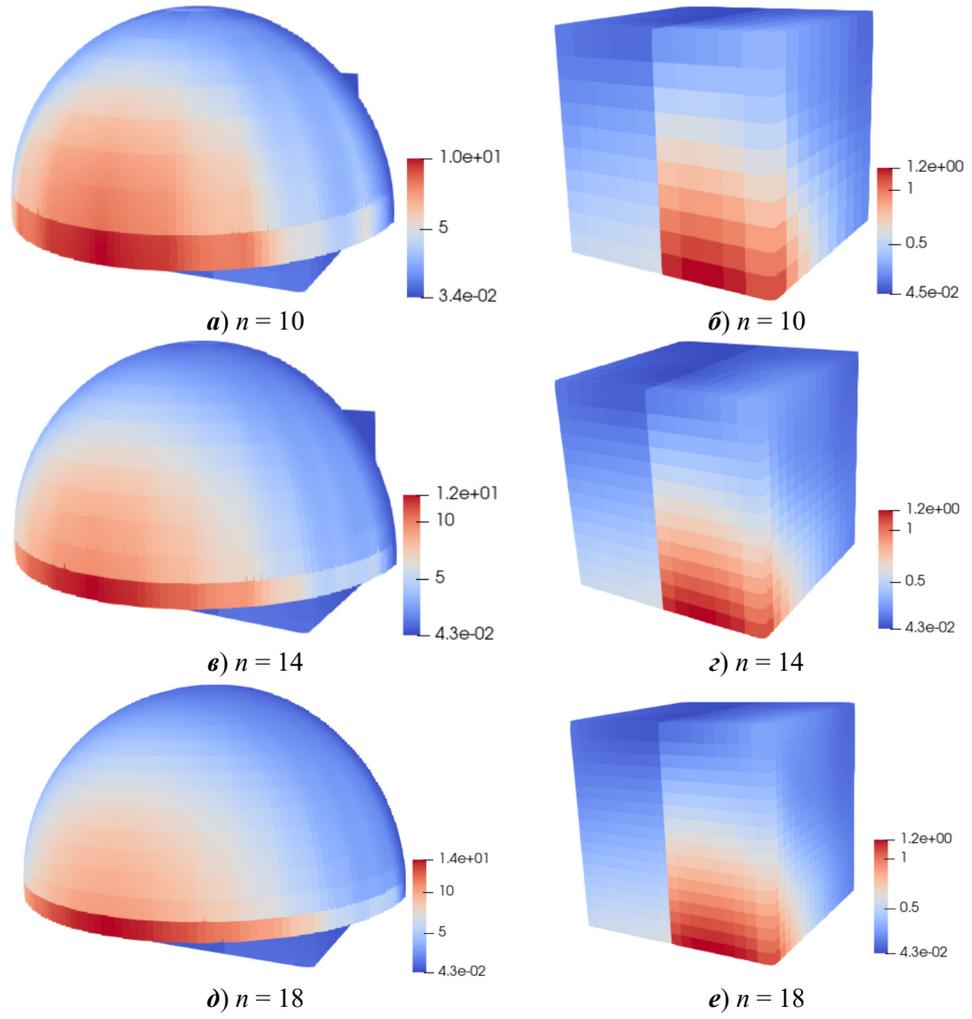


Рис. 10. Плотность поверхностного тока $|\mathbf{u}|$ на экране (a, c, e); модуль вектора поляризации $|\mathbf{J}|$ в области неоднородности (b, d, e)

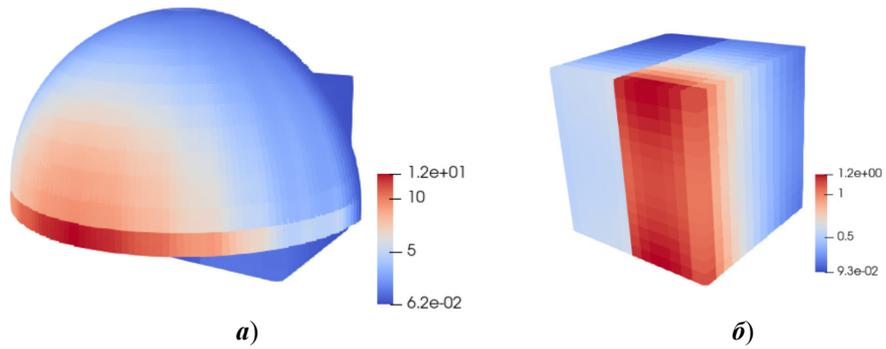


Рис. 11. Решение задачи дифракции на экране (a); решение задачи дифракции на теле в отсутствие экрана (b)

Заключение

В работе описан параллельный алгоритм решения задачи дифракции электромагнитной волны на рассеивателе сложной формы, использующий финитные базисные вектор-функции на неплоском параметрически заданном экране. Проведенные вычислительные эксперименты показали внутреннюю сходимость метода и соответствие полученных авторами результатов с ранее опубликованными.

Список литературы

1. Смирнов Ю. Г., Цупак А. А. О фредгольмовости уравнения электрического поля в векторной задаче дифракции на объемном частично экранированном теле // Дифференциальные уравнения. 2016. Т. 52, № 9. С. 1242–1251.
2. Смирнов Ю. Г., Цупак А. А. Математическая теория дифракции акустических и электромагнитных волн на системе экранов и неоднородных тел. М. : КноРус, 2016.
3. Smirnov Y. G., Tsupak A. A. Existence and uniqueness theorems in electromagnetic diffraction on systems of lossless dielectrics and perfectly conducting screens // *Applicable Analysis*. 2017. Vol. 96, № 8. P. 1326–1341.
4. Smirnov Y. G., Tsupak A. A. Investigation of electromagnetic wave diffraction from an inhomogeneous partially shielded solid // *Applicable Analysis*. 2018. Vol. 97, № 11. P. 1881–1895.
5. Смирнов Ю. Г., Медведик М. Ю., Максимова М. А. Решение задачи электромагнитной волны на экранах сложной формы // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2012. № 4. С. 59–72.
6. Смирнов Ю. Г., Москалева М. А. Сходимость метода Галеркина в задаче дифракции электромагнитных волн на системе произвольно расположенных тел и экранов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2016. № 2. С. 78–86.
7. Смирнов Ю. Г., Медведик М. Ю., Цупак А. А., Максимова М. А. Численное решение задачи дифракции электромагнитных волн на системе тел и экранов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2014. № 3. С. 114–134.
8. Medvedik M. Y., Smirnov Y. G., Tsupak A. A., Valovik D. V. Vector problem of electromagnetic wave diffraction by a system of inhomogeneous volume bodies, thin screens, and wire antennas // *Journal of Electromagnetic Waves and Applications*. 2016. Vol. 30, № 8. P. 1086–1100.
9. Kress R. *Linear integral equations*. Berlin : Springer-Verlag, 1989.
10. Hänninen I., Taskinen M., Sarvas J. Singularity subtraction integral formulae for surface integral equations with RWG, rooftop and hybrid basis functions // *Prog. Electromagn. Res. PIER*. 2006. Vol. 63. P. 243–278.
11. Медведик М. Ю. Применение функций крышек для решения задачи дифракции электромагнитных волн на экранах сложной формы // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2012. № 3. С. 84–98.
12. Дубровин Б. А., Новиков С. П., Фоменко А. Т. *Современная геометрия*. М. : Наука, 1979.
13. Зорич В. А. *Математический анализ : в 2 частях*. М. : ФАЗИС, 1997. Ч. 2.
14. Цупак А. А. Численный метод и параллельный алгоритм решения задачи дифракции электромагнитной волны на неплоском идеально проводящем экране // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2020. № 4. С. 32–41.

15. Ильинский А. С., Смирнов Ю. Г. Дифракция электромагнитных волн на проводящих тонких экранах. М. : ИПРЖ «Радиотехника», 1998.
16. Microsoft MPI. URL: [https://msdn.microsoft.com/en-us/library/bb524831\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/bb524831(v=vs.85).aspx)
17. Peter A., Zwamborn M., van den Berg Peter M. A Weak Form of the Conjugate Gradient FFT Method for Plate Problems // IEEE transactions on antennas and propagation. 1991. Vol. 39. № 2.

References

1. Smirnov Yu.G., Tsupak A.A. On the Fredholm property of the electric field equation in the vector problem of diffraction by a volume partially shielded body. *Differentsial'nye uravneniya = Differential equations*. 2016;52(9):1242–1251. (In Russ.)
2. Smirnov Yu.G., Tsupak A.A. *Matematicheskaya teoriya difraktsii akusticheskikh i elektromagnitnykh voln na sisteme ekranov i neodnorodnykh tel = Mathematical theory of diffraction of acoustic and electromagnetic waves on a system of screens and inhomogeneous bodies*. Moscow: KnoRus, 2016. (In Russ.)
3. Smirnov Y.G., Tsupak A.A. Existence and uniqueness theorems in electromagnetic diffraction on systems of lossless dielectrics and perfectly conducting screens. *Applicable Analysis*. 2017;96(8):1326–1341.
4. Smirnov Y.G., Tsupak A.A. Investigation of electromagnetic wave diffraction from an inhomogeneous partially shielded solid. *Applicable Analysis*. 2018;97(11):1881–1895.
5. Smirnov Yu.G., Medvedik M.Yu., Maksimova M.A. Solving the problem of an electromagnetic wave on screens of complex shape. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences*. 2012;(4):59–72. (In Russ.)
6. Smirnov Yu.G., Moskaleva M.A. Convergence of the Galerkin method in the electromagnetic waves diffraction problem on a system of arbitrary located bodies and screens. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences*. 2016;(2):78–86. (In Russ.)
7. Smirnov Yu.G., Medvedik M.Yu., Tsupak A.A., Maksimova M.A. Numerical solution of the electromagnetic wave diffraction problem on the system of bodies and screens. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences*. 2014;(3):114–134. (In Russ.)
8. Medvedik M.Y., Smirnov Y.G., Tsupak A.A., Valovik D.V. Vector problem of electromagnetic wave diffraction by a system of inhomogeneous volume bodies, thin screens, and wire antennas. *Journal of Electromagnetic Waves and Applications*. 2016;30(8):1086–1100.
9. Kress R. *Linear integral equations*. Berlin: Springer-Verlag, 1989.
10. Hänninen I., Taskinen M., Sarvas J. Singularity subtraction integral formulae for surface integral equations with RWG, rooftop and hybrid basis functions. *Prog. Electromagn. Res. PIER*. 2006;63:243–278.
11. Medvedik M.Yu. Application of lid functions for solving the problem of electromagnetic waves diffraction on screens of complex shape. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences*. 2012;(3):84–98. (In Russ.)
12. Dubrovin B.A., Novikov S.P., Fomenko A.T. *Sovremennaya geometriya = Modern geometry*. Moscow: Nauka, 1979. (In Russ.)
13. Zorich V.A. *Matematicheskii analiz: v 2 chastyakh = Mathematical analysis: in 2 parts*. Moscow: FAZIS, 1997;(part 2). (In Russ.)
14. Tsupak A.A. A numerical method and a parallel algorithm for solving the problem of electromagnetic wave diffraction on a non-planar perfectly conducting screen. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki =*

- University proceedings. Volga region. Physical and mathematical sciences.* 2020;(4):32–41. (In Russ.)
15. Il'inskiy A.S., Smirnov Yu.G. *Difraktsiya elektromagnitnykh voln na provodyashchikh tonkikh ekranakh = Diffraction of electromagnetic waves on thin conductive screens.* Moscow: IPRZh «Radiotekhnika», 1998. (In Russ.)
16. *Microsoft MPI.* Available at: [https://msdn.microsoft.com/en-us/library/bb524831\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/bb524831(v=vs.85).aspx)
17. Peter A., Zwamborn M., van den Berg Peter M. A Weak Form of the Conjugate Gradient FFT Method for Plate Problems. *IEEE transactions on antennas and propagation.* 1991;39(2).

Информация об авторах / Information about the authors

Олег Сергеевич Скворцов

студент, Пензенский
государственный университет (Россия,
г. Пенза, ул. Красная, 40)

E-mail: ghj.ghh.13@mail.ru

Oleg S. Skvortsov

Student, Penza State University
(40 Krasnaya street, Penza, Russia)

Алексей Александрович Цупак

кандидат физико-математических наук,
доцент, доцент кафедры математики
и суперкомпьютерного моделирования,
Пензенский государственный
университет (Россия,
г. Пенза, ул. Красная, 40)

E-mail: altsupak@yandex.ru

Aleksey A. Tsupak

Candidate of physical and mathematical
sciences, associate professor, associate
professor of the sub-department
of mathematics and supercomputer
modeling, Penza State University
(40 Krasnaya street, Penza, Russia)

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.

Поступила в редакцию / Received 15.05.2023

Поступила после рецензирования и доработки / Revised 20.07.2023

Принята к публикации / Accepted 16.08.2023