



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М. В. ЛОМОНОСОВА  
Факультет Вычислительной Математики и Кибернетики

Вохмин Максим Эдуардович

608 группа

**Итоговый отчет**

По курсу: «Суперкомпьютерное моделирование и технологии»

Вариант 2

Москва – 2024

## Оглавление

Математическая постановка задачи.....	3
Численный метод решения задачи .....	3
Задание практикума .....	6
Программный метод решения задачи .....	6
OpenMP.....	7
MPI .....	7
Результаты расчетов .....	8
Изображения приближенного решения и ошибок по итерациям .....	11
Вывод.....	14

## Математическая постановка задачи

В области  $D \subset \mathbb{R}^2$ , ограниченной контуром  $\gamma$ , рассматривается дифференциальное уравнение Пуассона

$$-\Delta u = f(x, y), \quad (1)$$

в котором оператор Лапласа

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2},$$

функция  $f(x, y)$  считается известной. Для выделения единственного решения уравнение дополняется граничными условием Дирихле (см. [1]):

$$u(x, y) = 0, \quad (x, y) \in \gamma. \quad (2)$$

Требуется найти функцию  $u(x, y)$ , удовлетворяющую уравнению (1) в области  $D$  и краевому условию (2) на ее границе.

## Численный метод решения задачи

Для приближенного решения задачи (1), (2) предлагается воспользоваться методом фиктивных областей [2].

Пусть область  $D$  принадлежит прямоугольнику  $\Pi = \{(x, y) : A_1 < x < B_1, A_2 < y < B_2\}$ . Обозначим через  $\overline{D}$ ,  $\overline{\Pi}$  замыкание области  $D$  и прямоугольника  $\Pi$  соответственно, через  $\Gamma$  – границу прямоугольника. Разность множеств

$$\hat{D} = \Pi \setminus \overline{D}$$

называется фиктивной областью. Выберем и зафиксируем малое  $\varepsilon > 0$ .

В прямоугольнике  $\Pi$  рассматривается задача Дирихле

$$\begin{aligned} -\frac{\partial}{\partial x} \left( k(x, y) \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y} \left( k(x, y) \frac{\partial v}{\partial y} \right) &= F(x, y), \\ v(x, y) &= 0, \quad (x, y) \in \Gamma \end{aligned} \quad (3)$$

с кусочно-постоянным коэффициентом

$$k(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 1/\varepsilon, & (x, y) \in \hat{D} \end{cases} \quad (4)$$

и правой частью

$$F(x, y) = \begin{cases} f(x, y), & (x, y) \in D, \\ 0, & (x, y) \in \hat{D}. \end{cases} \quad (5)$$

Требуется найти непрерывную в  $\overline{\Pi}$  функцию  $v(x, y)$ , удовлетворяющую дифференциальному уравнению задачи (3) всюду в  $\Pi \setminus \gamma$ , равную нулю на границе  $\Gamma$  прямоугольника, и такую, чтобы вектор потока

$$W(x, y) = -k(x, y) \left( \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \right)$$

имел непрерывную нормальную компоненту на общей части криволинейной границы области  $D$  и прямоугольника  $\Pi$ . Последнее означает, что в каждой точке  $(x_0, y_0) \in \gamma \cap \Pi$  должно выполняться равенство

$$\lim_{\substack{(x,y) \rightarrow (x_0,y_0), \\ (x,y) \in D}} (W(x,y), n(x_0, y_0)) = \lim_{\substack{(x,y) \rightarrow (x_0,y_0), \\ (x,y) \in \hat{D}}} (W(x,y), n(x_0, y_0)), \quad (6)$$

где  $n(x, y)$  – вектор единичной нормали к границе  $\gamma$  в точке  $(x, y)$ , определенный всюду или почти всюду на кривой.

Известно [2], что функция  $v(x, y)$  равномерно приближает решение  $u(x, y)$  задачи (1), (2) в области  $D$ , а именно,

$$\max_{P \in \bar{D}} |v(x, y) - u(x, y)| < C\varepsilon, \quad C > 0. \quad (7)$$

В частности,  $|v(x, y)| < C\varepsilon$  во всех точках кривой  $\gamma$ . Этот результат позволяет получить искомую функцию  $u(x, y)$  с любой наперед заданной точностью  $\varepsilon > 0$ , решая задачу (3), (6) вместо задачи (1), (2). Тем самым, задача Дирихле в криволинейной области приближенно заменяется задачей Дирихле в прямоугольнике с кусочно-постоянным коэффициентом  $k(x, y)$ .

Краевые задачу (3), (6) предлагается решать численно методом конечных разностей [3]. В замыкании прямоугольника  $\bar{\Pi}$  определяется равномерная прямоугольная сетка  $\bar{\omega}_h = \bar{\omega}_1 \times \bar{\omega}_2$ , где

$$\bar{\omega}_1 = \{x_i = A_1 + ih_1, i = \overline{0, M}\}, \quad \bar{\omega}_2 = \{y_j = A_2 + jh_2, j = \overline{0, N}\}.$$

Здесь  $h_1 = (B_1 - A_1)/M$ ,  $h_2 = (B_2 - A_2)/N$ . Через  $\omega_h$  обозначим множество внутренних узлов сетки  $\bar{\omega}_h$ , т.е. множество узлов сетки прямоугольника, не лежащих на границе  $\Gamma$ .

Рассмотрим линейное пространство  $H$  функций, заданных на сетке  $\omega_h$ . Обозначим через  $w_{ij}$  значение сеточной функции  $w \in H$  в узле сетки  $(x_i, y_j) \in \omega_h$ . Будем считать, что в пространстве  $H$  задано скалярное произведение и евклидова норма

$$(u, v) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} h_1 h_2 u_{ij} v_{ij}, \quad \|u\|_E = \sqrt{(u, u)}. \quad (8)$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вида

$$Aw = B, \quad (9)$$

где  $A : H \rightarrow H$  – оператор, действующий в пространстве сеточных функций,  $B \in H$  – известная правая часть. Задача (9) называется разностной схемой. Решение этой задачи считается численным решением исходной дифференциальной задачи.

При построении разностной схемы следует аппроксимировать (приблизительно заменить) все уравнения краевой задачи их разностными аналогами – сеточными уравнениями, связывающими значения искомой сеточной функции в узлах сетки. Полученные таким образом уравнения должны быть функционально независимыми, а их общее количество – совпадать с числом неизвестных, т.е. с количеством узлов сетки.

Дифференциальное уравнение задачи (3) во всех внутренних точках сетки аппроксимируется разностным уравнением

$$-\frac{1}{h_1} \left( a_{i+1j} \frac{w_{i+1j} - w_{ij}}{h_1} - a_{ij} \frac{w_{ij} - w_{i-1j}}{h_1} \right) - \frac{1}{h_2} \left( b_{ij+1} \frac{w_{ij+1} - w_{ij}}{h_2} - b_{ij} \frac{w_{ij} - w_{ij-1}}{h_2} \right) = F_{ij},$$

$$i = \overline{1, M-1}, \quad j = \overline{1, N-1}, \quad (10)$$

в котором коэффициенты

$$a_{ij} = \frac{1}{h_2} \int_{y_{j-1/2}}^{y_{j+1/2}} k(x_{i-1/2}, t) dt, \quad b_{ij} = \frac{1}{h_1} \int_{x_{i-1/2}}^{x_{i+1/2}} k(t, y_{j-1/2}) dt \quad (11)$$

при всех  $i = \overline{1, M}$ ,  $j = \overline{1, N}$ . Здесь полуцелые узлы

$$x_{i\pm 1/2} = x_i \pm 0.5h_1, \quad y_{j\pm 1/2} = y_j \pm 0.5h_2.$$

Правая часть разностного уравнения

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(x, y) dx dy, \quad \Pi_{ij} = \{(x, y) : x_{i-1/2} \leq x \leq x_{i+1/2}, y_{j-1/2} \leq y \leq y_{j+1/2}\} \quad (12)$$

при всех  $i = \overline{1, M-1}$ ,  $j = \overline{1, N-1}$ .

Введем обозначения правой и левой разностных производных по переменным  $x$ ,  $y$  соответственно:

$$\begin{aligned} w_{x,ij} &= \frac{w_{i+1,j} - w_{ij}}{h_1}, & w_{\bar{x},ij} &= w_{x,i-1,j} = \frac{w_{ij} - w_{i-1,j}}{h_1}, \\ w_{y,ij} &= \frac{w_{ij+1} - w_{ij}}{h_2}, & w_{\bar{y},ij} &= w_{y,i,j-1} = \frac{w_{ij} - w_{i,j-1}}{h_2}. \end{aligned}$$

С учетом принятых обозначений разностное уравнение (10) можно представить в более компактном и удобном виде:

$$-(aw_{\bar{x}})_{x,ij} - (bw_{\bar{y}})_{y,ij} = F_{ij}, \quad i = \overline{1, M-1}, \quad j = \overline{1, N-1} \quad (13)$$

Краевые условия Дирихле задачи (3), (6) аппроксимируются точно равенством

$$w_{ij} = w(x_i, y_j) = 0, \quad (x_i, y_j) \in \Gamma. \quad (14)$$

Переменные  $w_{ij}$ , заданные равенством (14), исключаются из системы уравнений (13). В результате остаются неизвестными значения  $w_{ij}$  при  $i = \overline{1, M-1}$ ,  $j = \overline{1, N-1}$  и их количество совпадает с числом уравнений. Система является линейной относительно неизвестных величин и может быть представлена в виде (9) с самосопряженным и положительно определенным оператором

$$Aw = -(aw_{\bar{x}})_x - (bw_{\bar{y}})_y$$

и правой частью  $F$ , определенной равенством (12). Таким образом, построенная разностная схема (13), (14) линейна и имеет единственное решение при любой правой части (см. [5]).

**Замечание.** Интегралы (11) от кусочно-постоянной функции  $k(x, y)$  следует вычислять аналитически. Нетрудно видеть, что если отрезок, соединяющий точки  $P_{ij} = (x_{i-1/2}, y_{j-1/2})$  и  $P_{ij+1} = (x_{i-1/2}, y_{j+1/2})$ , целиком расположен в области  $D$ , то  $a_{ij} = 1$ . Если же указанный отрезок находится в фиктивной области  $\hat{D}$ , то  $a_{ij} = 1/\varepsilon$ . В противном случае

$$a_{ij} = h_2^{-1} l_{ij} + (1 - h_2^{-1} l_{ij})/\varepsilon,$$

где  $l_{ij}$  — длина той части отрезка  $[P_{ij}, P_{ij+1}]$ , которая принадлежит области  $D$ . Аналогичным образом вычисляются коэффициенты  $b_{ij}$ .

Очевидно, правая часть схемы  $F_{ij}$  равна нулю при всех  $(i, j) : \Pi_{ij} \subset \hat{D}$ . Если  $\Pi_{ij} \subset D$ , то правую часть предлагается приближенно заменить значением  $f(x_i, y_j)$ . В противном случае, когда прямоугольник  $\Pi_{ij}$  содержит точки оригинальной области  $D$  и фиктивной области  $\hat{D}$ , величина  $F_{ij}$  может быть вычислена приближенно как произведение

$$(h_1 h_2)^{-1} S_{ij} f(x_i^*, y_j^*),$$

где  $(x_i^*, y_j^*)$  — любая точка пересечения  $\Pi_{ij} \cap D$ ,  $S_{ij} = \text{mes}(\Pi_{ij} \cap D)$  — площадь пересечения областей, при вычислении которой криволинейную часть границы можно заменить отрезком прямой.

Приближенное решение разностной схемы (10), (14) может быть получено итерационным методом скорейшего спуска [4]. Этот метод позволяет получить последовательность сеточных функций  $w^{(k)} \in H$ ,  $k = 1, 2, \dots$ , сходящуюся по норме пространства  $H$  к решению разностной схемы, т.е.

$$\|w - w^{(k)}\|_E \rightarrow 0, \quad k \rightarrow +\infty.$$

Начальное приближение  $w^{(0)}$  можно выбрать любым способом, например, равным нулю во всех точках расчетной сетки.

Метод является одношаговым. Итерация  $w^{(k+1)}$  вычисляется по итерации  $w^{(k)}$  согласно равенствам:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \tau_{k+1} r_{ij}^{(k)}, \quad (15)$$

где невязка  $r^{(k)} = Aw^{(k)} - B$ , итерационный параметр

$$\tau_{k+1} = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})}.$$

В качестве условия остановки итерационного процесса следует использовать неравенство

$$\|w^{(k+1)} - w^{(k)}\|_E < \delta,$$

где  $\delta$  – положительное число, определяющее точность итерационного метода. Оценку точности приближенного решения можно проводить в других нормах пространства сеточных функций, например, в максимум норме

$$\|w\|_C = \max_{x \in \bar{\omega}_h} |w(x)|. \quad (16)$$

Константу  $\delta$  для данной задачи предлагается выбрать так, чтобы итерационный процесс укладывался в отведенное для него время.

## Задание практикума

Требуется приближенно найти решение задачи (1), (2) для случая, когда  $f(x, y) = 1$  при всех  $(x, y) \in D$ .

Область  $D$  задается следующей геометрией: треугольник с вершинами в точках  $C(-3, 0)$ ,  $A(3, 0)$ ,  $B(0, 4)$ .

## Программный метод решения задачи

Файл с программной реализацией содержит функции инициализации сетки, решения  $F$ , оператора  $A$  и метод скорейшего спуска.

В функции **main** рассчитываются размеры ячейки сетки  $h_1$  и  $h_2$  (**calculateH**). А также коэффициент  $\varepsilon$  (**calculateEpsilon**) для расчета коэффициентов (11) разностного уравнения.

Далее происходит создание сетки разбиения, с помощью функции **createGrid**.

Далее, в функции **createF**, инициализируется поэлементное решение задачи Дирихле (5).

В функции **createOperator** формируется оператор  $A$  (9).

Как только все необходимые массивы данных становятся проинициализированными, запускается метод скорейшего спуска, описанный в функции **fastestDescent**. На каждом шаге этого метода рассчитывается ошибка  $e = \|\omega^{(k+1)} - \omega^{(k)}\|_E$ . Выходом из итерационного процесса является неравенство  $e < \delta$ , где  $\delta$  – точность метода, определенная в функции **main**.

## OpenMP

Для распараллеливания программы средствами OpenMP используются директива **#pragma omp parallel for**. Данная директива дополняется различными параметрами:

- **reduction(+:counter)** – для подсчета суммы.
- **collapse(2)** – для распараллеливания вложенных циклов.

## MPI

Для распараллеливания программы средствами MPI используются следующие функции:

- **MPI\_Init** – инициализация среды MPI.
- **MPI\_Send** и **MPI\_Recv** – для обмена численными переменными между процессами.
- **MPI\_Barrier** – для синхронизации работы процессов.
- **MPI\_Gatherv** – для объединения массивов данных в главный процесс.
- **MPI\_Scatterv** – для создания локальных массивов данных для каждого процесса.
- **MPI\_Reduce** – для суммирования переменной со всех процессов.
- **MPI\_Bcast** – для передачи процессам переменных.
- **MPI\_Finalize()** – освобождение ресурсов и корректное завершение работы MPI.

## Результаты расчетов

Последовательный запуск сетки 80 x 90:

- Ошибка: 0.000001
- Время: 2.74906 мин
- Количество итераций: 47342

Последовательный запуск сетки 160 x 180:

- Ошибка: 0.000003
- Время: 32.3653 мин
- Количество итераций: 98218

Количество OpenMP- нитей	Число точек сетки (M x N)	Число итераций	Время решения, мин	Ускорение
2	80 x 90	50753	0.98318	2.796
4	80 x 90	55448	0.608125	4.521
8	80 x 90	47076	0.301457	9.119
16	80 x 90	52129	0.250903	10.957
4	160 x 180	93866	9.57966	3.379
8	160 x 180	99437	5.07259	6.380
16	160 x 180	109523	3.76768	8.590
32	160 x 180	124366	2.3389	13.838

Таблица 1: Результаты расчётов на ПВС IBM Polus (OpenMP)

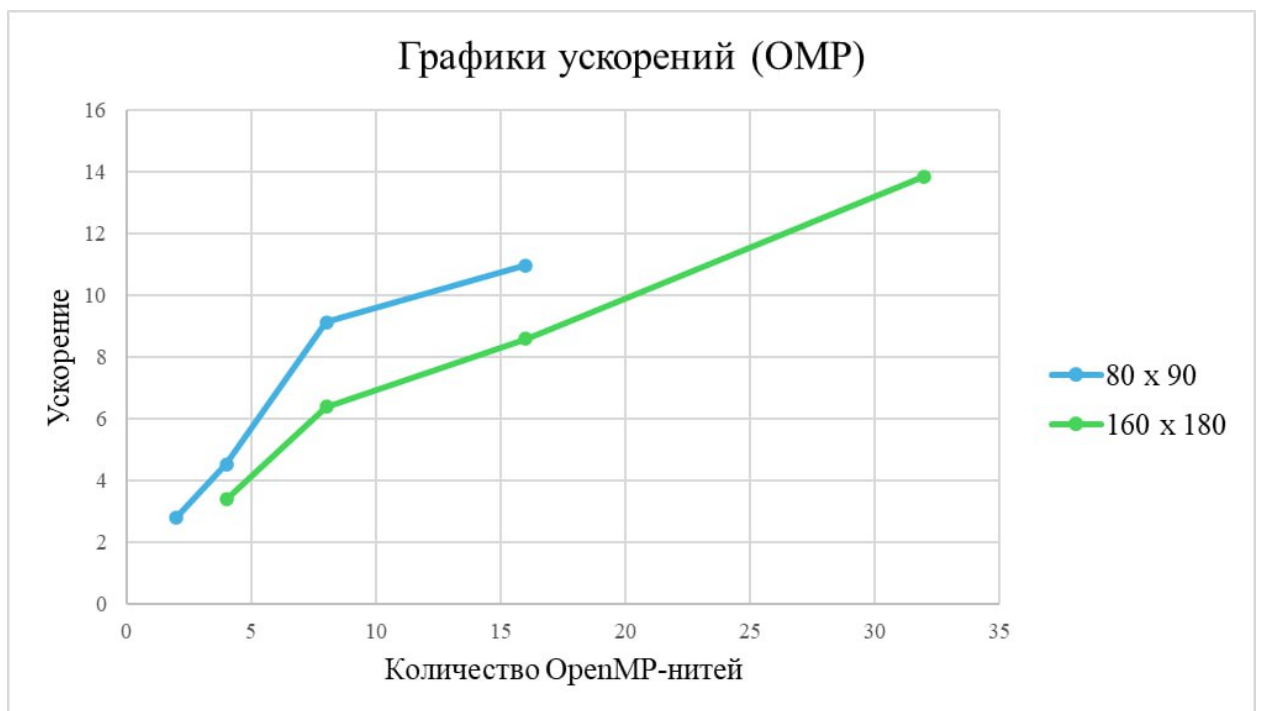


Рисунок 1: Графики ускорений (OpenMP)



Количество MPI-процессов	Число точек сетки (М x N)	Число итераций	Время решения, мин	Ускорение
2	80 x 90	51863	0.6275	4.9843
4	80 x 90	49625	0.3358	9.3144
8	80 x 90	50326	0.2064	15.1536
16	80 x 90	52104	0.2530	12.3625
2	160 x 180	109528	9.3537	3.4602
4	160 x 180	124478	2.45133	13.2032
8	160 x 180	138613	1.10752	29.2232
16	160 x 180	131596	1.0384	31.1684

Таблица 2: Результаты расчётов на ПВС IBM Polus (MPI)

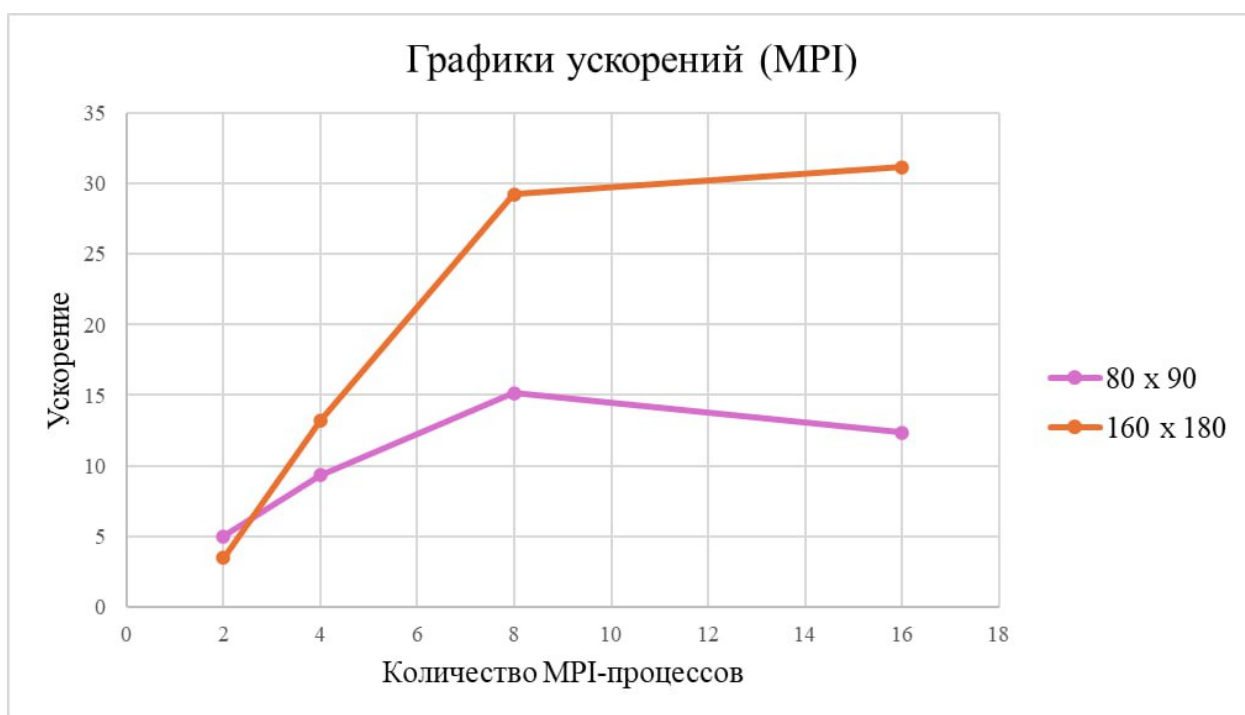


Рисунок 2: Графики ускорений (MPI)

Количество MPI-процессов	Количество OpenMP-нитей	Число точек сетки (M x N)	Число итераций	Время решения, мин	Ускорение
2	1	80 x 90	70844	0.314029	8.7542
2	2	80 x 90	72118	0.269799	10.1893
2	4	80 x 90	74231	0.291382	9.4345
2	8	80 x 90	69281	0.396668	6.9304
4	1	160 x 180	124478	2.45133	13.203
4	2	160 x 180	109855	1.45148	22.298
4	4	160 x 180	95819	1.26113	25.6637
4	8	160 x 180	110383	1.90987	16.9463

Таблица 3: Результаты расчётов на ПВС IBM Polus (MPI+OMP)

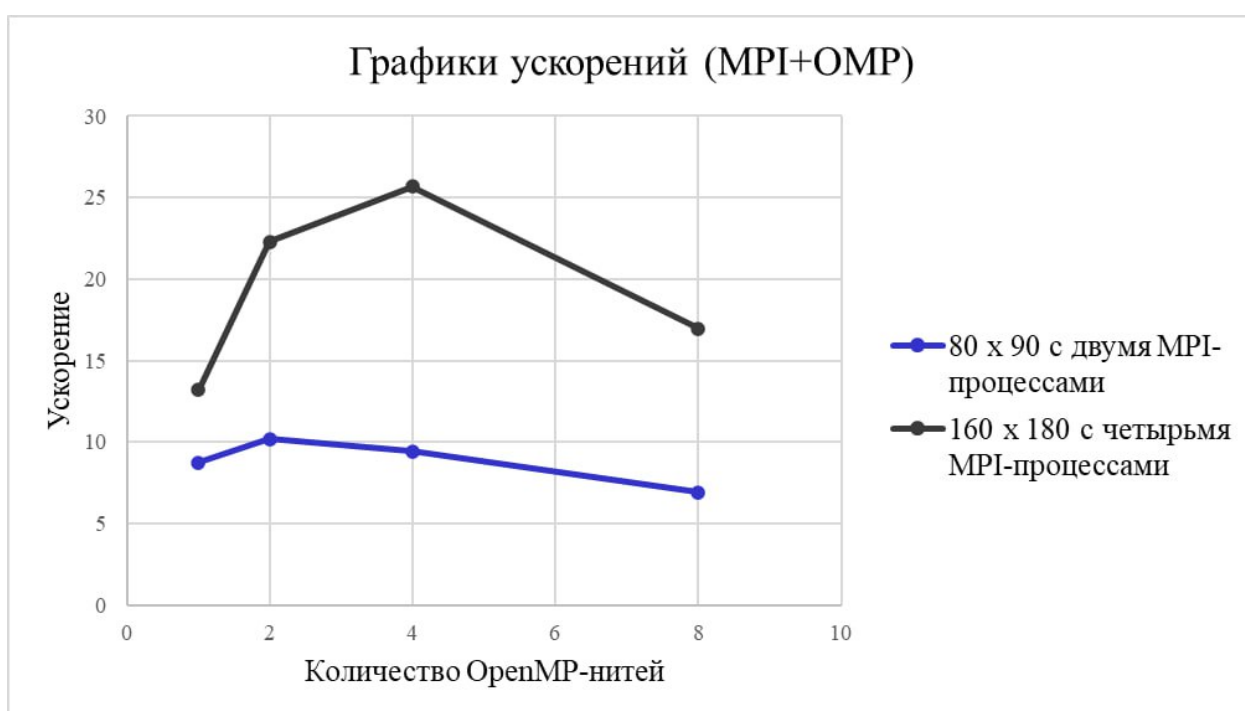


Рисунок 3: Графики ускорений (MPI + OpenMP)

## Изображения приближенного решения и ошибок по итерациям

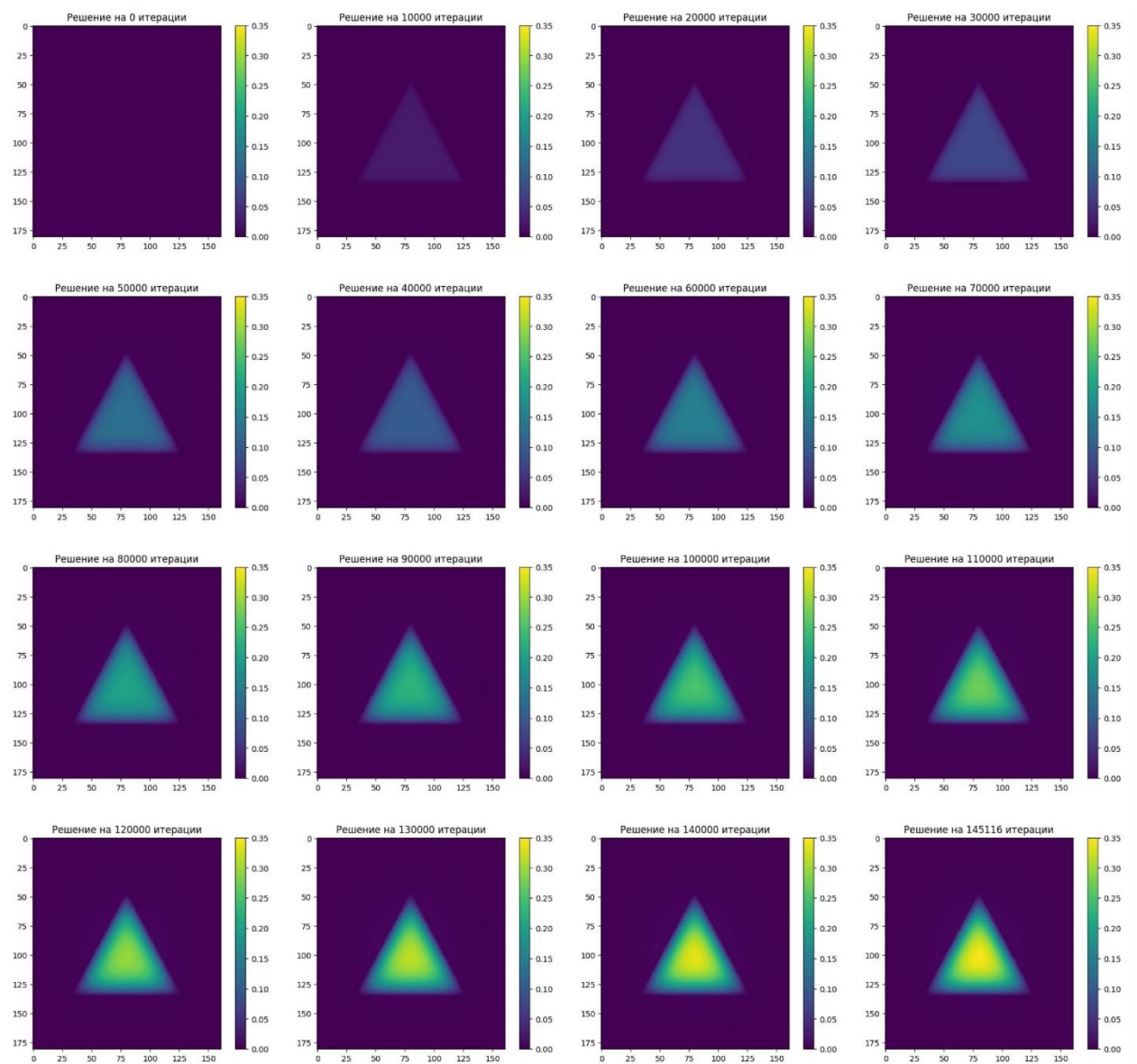


Рисунок 4: Изображения приближенного решения по итерациям

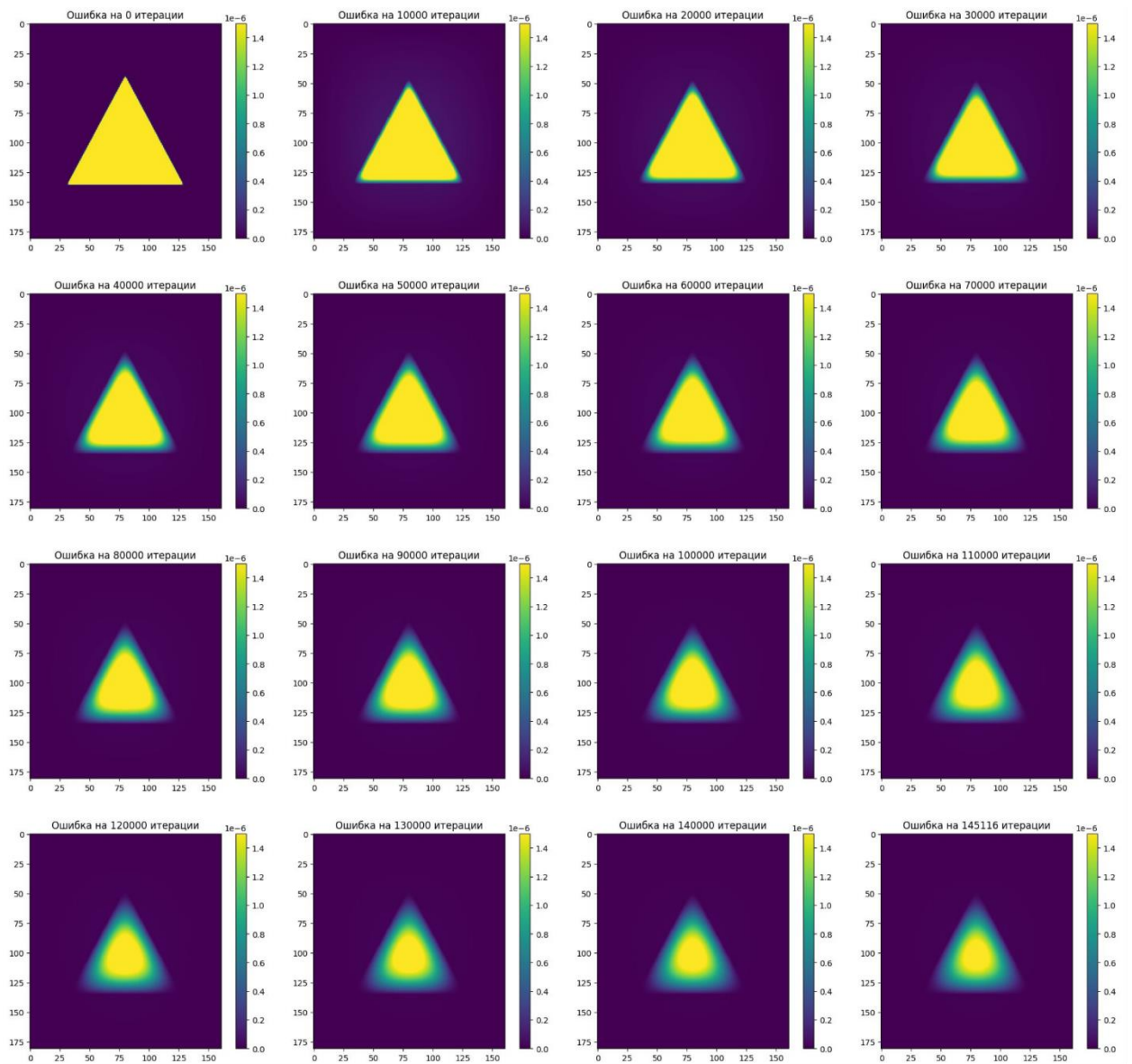


Рисунок 5: Изображения ошибки по итерациям

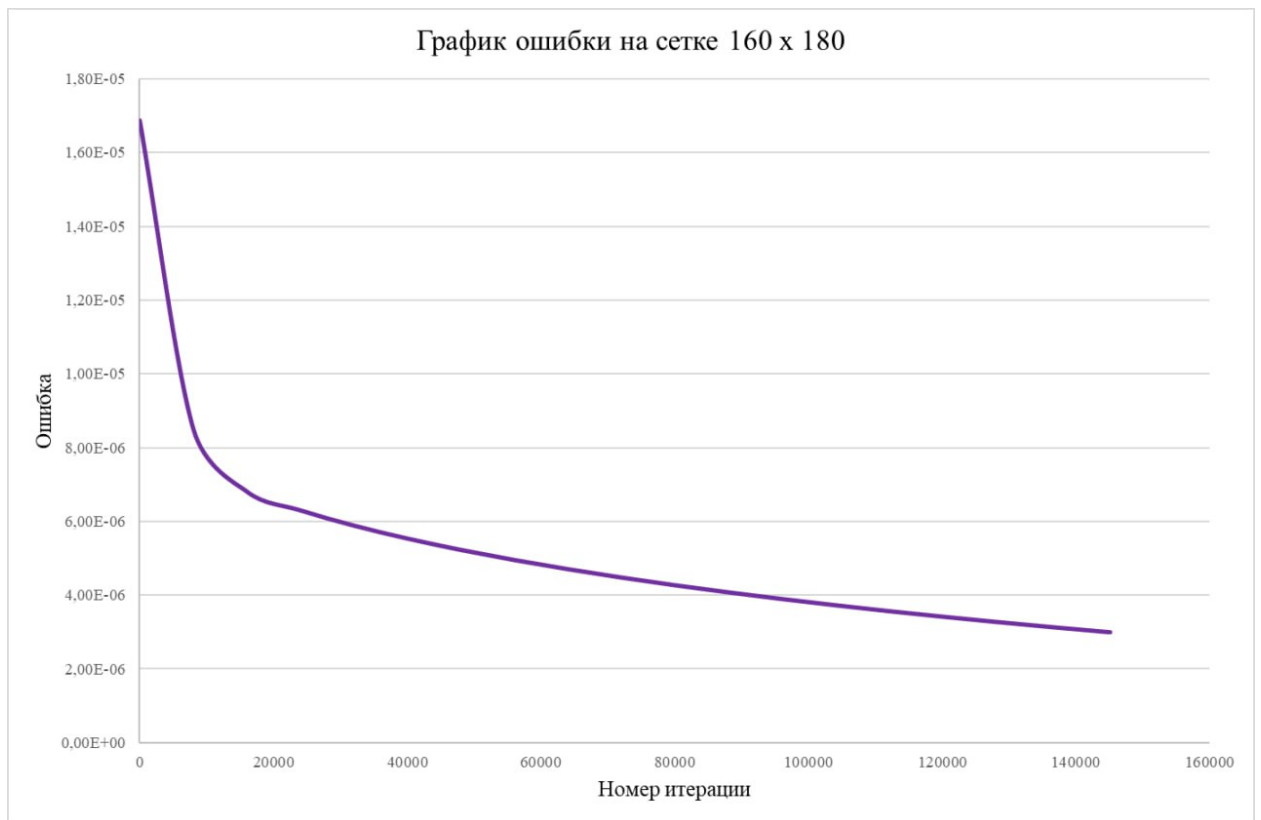


Рисунок 6: График ошибки на сетке 160x180

## Вывод

По полученным результатам видно, что на большой сетке распараллеливание в целом демонстрирует лучшие результаты по сравнению с маленькой сеткой. Это связано с необходимостью пересылки данных: слишком частые передачи небольшого объема информации приводят к перегрузке сети и замедляют работу программы.

При распараллеливании программы методами OpenMP на большой сетке наблюдается практически линейный рост ускорения. В то же время на маленькой сетке рост ускорения замедляется при переходе с 8 до 16 нитей.

Ускорение, достигнутое с помощью средств MPI, показало наилучшие результаты. Однако при переходе от 8 до 16 процессов на маленькой сетке наблюдается спад ускорения, а на большой сетке — уменьшение темпов роста.

Гибридное решение также эффективно ускоряет синхронную программу, но при увеличении количества нитей наблюдается спад ускорения. Может показаться, что гибридное решение обеспечивает худшее ускорение (по сравнению с MPI), это связано с тем, что в данном случае использовалось малое количество процессов: 2 для маленькой сетки и 4 для большой. Тем не менее, как показывает решение задачи методами MPI, лучшее ускорение для большой сетки достигается при использовании 16 процессов.