

Московский государственный университет имени М. В. Ломоносова  
Факультет вычислительной математики и кибернетики

Отчет

По курсу: «Суперкомпьютерное моделирование и технологии»  
«MPI + OpenMP»

Выполнила:  
Казанцева Варвара Денисовна  
Группа: 608

11 декабря 2025 г.

# Содержание

<b>1 Введение</b>	<b>2</b>
<b>2 Математическая постановка задачи</b>	<b>2</b>
<b>3 Метод фиктивных областей</b>	<b>2</b>
<b>4 Разностная схема решения задачи</b>	<b>2</b>
4.1 Определение коэффициентов разностной схемы . . . . .	3
4.2 Определение коэффициентов разностной схемы аналитически . . . . .	3
<b>5 Метод решения системы линейных алгебраических уравнений</b>	<b>3</b>
<b>6 Программная реализация</b>	<b>4</b>
6.1 Ключевые компоненты программы . . . . .	4
6.2 Используемые функции MPI . . . . .	4
6.3 Используемые директивы OpenMP . . . . .	5
6.4 Запуск программы . . . . .	5
<b>7 Результаты расчетов</b>	<b>5</b>
7.1 Параллельная программа с MPI . . . . .	5
7.2 Результаты расчетов на ПВС IBM Polus . . . . .	6
7.3 Графики . . . . .	6
7.4 График приближенного решения . . . . .	6
7.5 Графики ускорения . . . . .	7
7.6 Графики сходимости . . . . .	8
<b>8 Заключение</b>	<b>8</b>

# 1 Введение

В данной работе рассматривается приближенное решение двумерной задачи Дирихле для уравнения Пуассона в криволинейной области. В качестве расчетной области выбран **остроугольный треугольник с вершинами в точках С(3,0), А(3,0), В(0,4)** (вариант 2). Целью работы является распараллеливание последовательного решения данной задачи с использованием средств MPI.

## 2 Математическая постановка задачи

В области  $D \subset \mathbb{R}^2$ , ограниченной кусочно-гладким контуром  $\Gamma$ , рассматривается дифференциальное уравнение Пуассона:

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

где оператор Лапласа:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (2)$$

функция  $f(x, y) = 1$  задана. Для выделения единственного решения уравнение дополняется граничным условием Дирихле:

$$u(x, y) = 0, \quad (x, y) \in \Gamma \quad (3)$$

## 3 Метод фиктивных областей

Для приближенного решения задачи (1)-(3) используется метод фиктивных областей. Область  $D$  принадлежит прямоугольнику  $\Pi = \{(x, y) : A_1 < x < B_1, A_2 < y < B_2\}$ . В прямоугольнике  $\Pi$  рассматривается задача Дирихле с кусочно-постоянным коэффициентом:

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

и правой частью:

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

где  $\hat{D} = \Pi \setminus D$  - фиктивная область,  $\varepsilon > 0$  - малый параметр.

## 4 Разностная схема решения задачи

Краевая задача решается численно методом конечных разностей. В замыкании прямоугольника  $\Pi$  определяется равномерная прямоугольная сетка  $\bar{\omega}_h = \bar{\omega}_1 \times \bar{\omega}_2$ , где:

$$\begin{aligned} \bar{\omega}_1 &= \{x_i = A_1 + ih_1, i = 0, 1, \dots, M\}, \\ \bar{\omega}_2 &= \{y_j = A_2 + jh_2, j = 0, 1, \dots, N\} \end{aligned}$$

с шагами  $h_1 = (B_1 - A_1)/M$ ,  $h_2 = (B_2 - A_2)/N$ .

Дифференциальное уравнение аппроксимируется разностным уравнением:

$$-\frac{1}{h_1}(a_{i+1,j}w_{x,ij} - a_{i,j}w_{\bar{x},ij}) - \frac{1}{h_2}(b_{i,j+1}w_{y,ij} - b_{i,j}w_{\bar{y},ij}) = F_{ij} \quad (6)$$

где коэффициенты  $a_{ij}$ ,  $b_{ij}$  вычисляются через интегралы от функции  $k(x, y)$ .

## 4.1 Определение коэффициентов разностной схемы

Коэффициенты  $a_{ij}$  вычисляются по формуле:

$$a_{ij} = \frac{1}{h_2} \int_{y_{j-1/2}}^{y_{j+1/2}} k(x_{i-1/2}, t) dt \quad (7)$$

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

Коэффициенты  $b_{ij}$  вычисляются по формуле:

$$b_{ij} = \frac{1}{h_1} \int_{x_{i-1/2}}^{x_{i+1/2}} k(t, y_{j-1/2}) dt \quad (8)$$

Правая часть  $F_{ij}$  вычисляется как:

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(x, y) dx dy \quad (9)$$

где  $\Pi_{ij} = \{(x, y) : x_{i-1/2} \leq x \leq x_{i+1/2}, y_{j-1/2} \leq y \leq y_{j+1/2}\}$ .

## 4.2 Определение коэффициентов разностной схемы аналитически

Для коэффициента  $a_{ij}$ :

$$a_{ij} = \begin{cases} 1, & \text{если отрезок } [P_{ij}, P_{ij+1}] \subset D \\ \frac{1}{\varepsilon}, & \text{если отрезок } [P_{ij}, P_{ij+1}] \subset \hat{D} \\ \frac{l_{ij}}{h_2} + \frac{1}{\varepsilon}(1 - \frac{l_{ij}}{h_2}), & \text{иначе} \end{cases} \quad (10)$$

где  $l_{ij}$  - длина части отрезка  $[P_{ij}, P_{ij+1}]$ , принадлежащей области  $D$ ,  $P_{ij} = (x_{i-1/2}, y_{j-1/2})$ .

Аналогично для коэффициента  $b_{ij}$ :

$$b_{ij} = \begin{cases} 1, & \text{если отрезок } [P_{ij}, P_{i+1j}] \subset D \\ \frac{1}{\varepsilon}, & \text{если отрезок } [P_{ij}, P_{i+1j}] \subset \hat{D} \\ \frac{l_{ij}}{h_1} + \frac{1}{\varepsilon}(1 - \frac{l_{ij}}{h_1}), & \text{иначе} \end{cases} \quad (11)$$

где  $l_{ij}$  - длина части отрезка  $[P_{ij}, P_{i+1j}]$ , принадлежащей области  $D$ .

Для правой части  $F_{ij}$ :

$$F_{ij} = \begin{cases} 0, & \text{если } \Pi_{ij} \subset \hat{D} \\ 1, & \text{если } \Pi_{ij} \subset D \\ \frac{S_{ij}}{h_1 h_2}, & \text{иначе} \end{cases} \quad (12)$$

где  $S_{ij}$  - площадь пересечения  $\Pi_{ij} \cap D$ .

$\varepsilon = h^2$  - малый параметр,  $h = \max(h_1, h_2)$ .

## 5 Метод решения системы линейных алгебраических уравнений

Приближенное решение разностной схемы находится итерационным методом сопряженных градиентов с диагональным предобуславливанием. Алгоритм метода:

1. Выбирается начальное приближение  $w^{(0)}$

2. Вычисляется невязка  $r^{(0)} = B - Aw^{(0)}$

3. Решается система  $Dz^{(0)} = r^{(0)}$

4. Задается направление  $p^{(1)} = z^{(0)}$

5. Для  $k = 0, 1, 2, \dots$  до сходимости:

$$\begin{aligned}\alpha_{k+1} &= \frac{(z^{(k)}, r^{(k)})}{(Ap^{(k+1)}, p^{(k+1)})} \\ w^{(k+1)} &= w^{(k)} + \alpha_{k+1}p^{(k+1)} \\ r^{(k+1)} &= r^{(k)} - \alpha_{k+1}Ap^{(k+1)} \\ Dz^{(k+1)} &= r^{(k+1)} \\ \beta_{k+1} &= \frac{(z^{(k+1)}, r^{(k+1)})}{(z^{(k)}, r^{(k)})} \\ p^{(k+2)} &= z^{(k+1)} + \beta_{k+1}p^{(k+1)}\end{aligned}$$

Условие остановки итерационного процесса:

$$\|w^{(k+1)} - w^{(k)}\|_E < \varepsilon \quad (13)$$

## 6 Программная реализация

Программа реализована на C++ с использованием гибридной технологии MPI+OpenMP. Применяется метод фиктивных областей для решения уравнения Пуассона в треугольной области.

### 6.1 Ключевые компоненты программы

- `Domain` – структура для хранения информации о локальной подобласти процесса
- `initializeCoefficients`, `initializeF` – инициализация коэффициентов и правой части
- `conjugateGradient` – метод сопряженных градиентов с диагональным предобуславливанием
- `applyOperator` – применение разностного оператора с обменом граничными значениями
- `solveDiagonal` – решение диагональной системы

### 6.2 Используемые функции MPI

- `MPI_Init`, `MPI_Finalize` – инициализация/завершение
- `MPI_Dims_create`, `MPI_Cart_create` – создание декартовой топологии
- `MPI_Cart_shift` – определение соседних процессов
- `MPI_Isend`, `MPI_Irecv`, `MPI_Waitall` – неблокирующий обмен данными
- `MPI_Allreduce` – глобальные редукции (скалярные произведения)

### 6.3 Используемые директивы OpenMP

- `#pragma omp parallel for collapse(2)` – параллелизация вложенных циклов вычислений коэффициентов, правой части и оператора
- `reduction(+:variable)` – редукции в скалярных произведениях
- `omp_set_num_threads()` – установка числа потоков OpenMP

OpenMP используется для распараллеливания вычислительно затратных участков:

- Инициализация коэффициентов  $a_{ij}$ ,  $b_{ij}$  и правой части  $F_{ij}$
- Применение разностного оператора
- Решение диагональной системы предобусловливания
- Вычисление скалярных произведений и норм

### 6.4 Запуск программы

Для параллельной версии MPI+OpenMP используется расширенный Makefile. Команды для работы с параллельной версией:

```
make run_1_4      # 1 процесс с 4 потоками
make run_2_4      # 2 процесса по 4 потока
make run_polus_2_all # пакетный запуск на кластере Polus (2 процесса)
make run_polus_4_all # пакетный запуск на кластере Polus (4 процесса)
```

## 7 Результаты расчетов

### 7.1 Параллельная программа с MPI

Проведены расчеты на сетке  $(M, N) = (40 \times 40)$  с различном количестве процессов. Последовательная программа отрабатывает на данной сетке за 0.0296 секунд.

Количество процессов	Количество нитей	Число итераций	Время решения, с
1	4	103	0.055
2	4	103	0.043

Таблица 1: Результаты расчетов MPI+OpenMP-программы на сетке  $40 \times 40$

## 7.2 Результаты расчетов на ПВС IBM Polus

Количество процессов MPI	Количество OpenMP-нитей в процессе	Число точек сетки ( $M \times N$ )	Число итераций	Время решения, с	Ускорение
0	0	$400 \times 600$	1077	112.59	1.0
2	1	$400 \times 600$	1077	6.49	17.36
2	2	$400 \times 600$	1077	3.38	33.31
2	4	$400 \times 600$	1077	1.79	62.90
2	8	$400 \times 600$	1077	0.99	113.73
0	0	$800 \times 1200$	2047	857.14	1.0
4	1	$800 \times 1200$	2047	24.93	34.38
4	2	$800 \times 1200$	2047	13.16	65.13
4	4	$800 \times 1200$	2047	7.21	118.88
4	8	$800 \times 1200$	2047	4.87	176.00

Таблица 2: Таблица с результатами расчетов на ПВС IBM Polus (MPI + OpenMP)

## 7.3 Графики

### 7.4 График приближенного решения

На рисунке 1 представлено приближенное решение задачи Дирихле для уравнения Пуассона в треугольной области, полученное на сетке  $800 \times 1200$ .

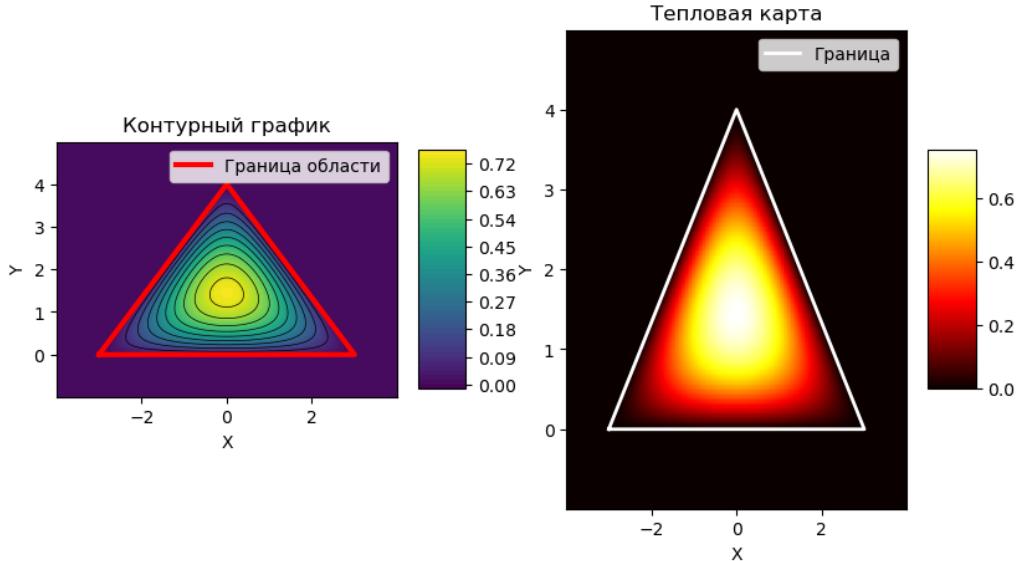


Рис. 1: Приближенное решение уравнения Пуассона в треугольной области на сетке  $800 \times 1200$

Решение демонстрирует ожидаемое поведение - максимальные значения в центре треугольной области с плавным уменьшением к границам, где задано нулевое граничное условие. Форма решения соответствует геометрии расчетной области.

## 7.5 Графики ускорения

На рисунках 2 и 4 приведены графики ускорения гибридной программы (MPI + OpenMP) при различном общем числе потоков для сеток  $400 \times 600$  и  $800 \times 1200$  соответственно.

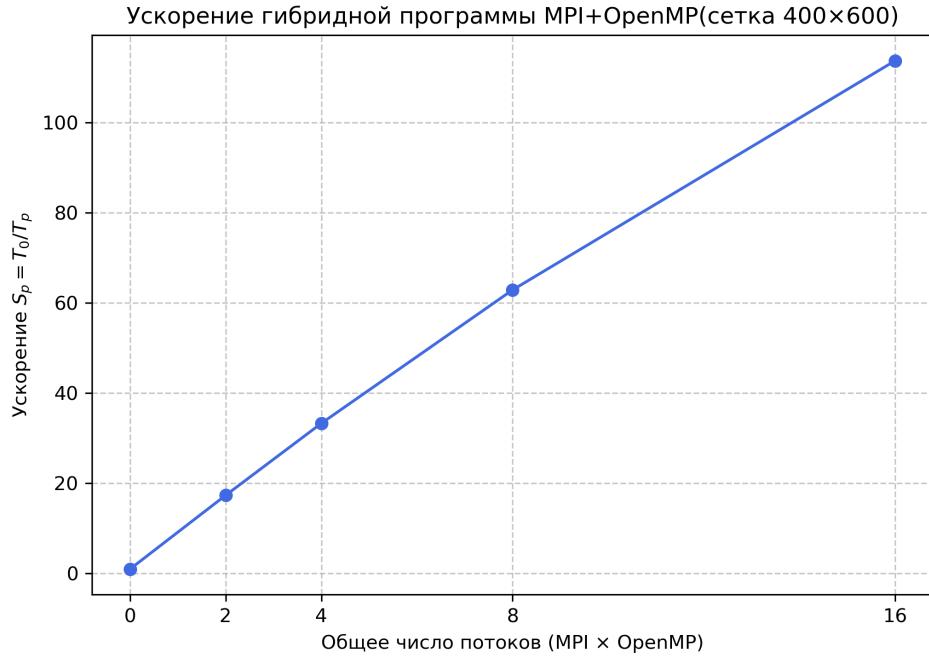


Рис. 2: Ускорение гибридной программы (MPI+OpenMP) для сетки  $400 \times 600$

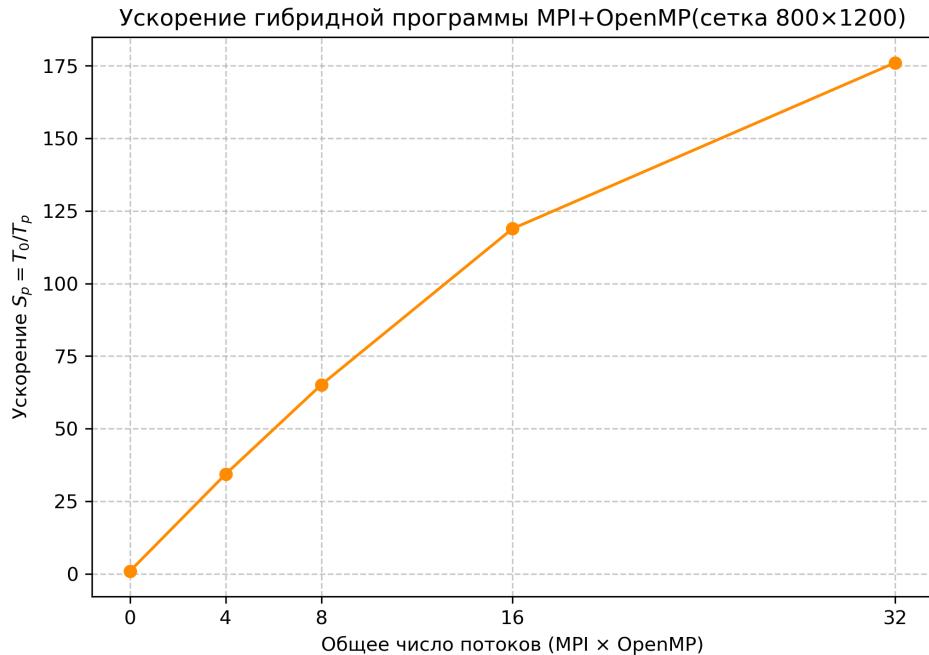


Рис. 3: Ускорение гибридной программы (MPI+OpenMP) для сетки  $800 \times 1200$

Из графиков видно, что ускорение растет с увеличением общего числа потоков (произведения числа MPI-процессов на число OpenMP-нитей в каждом процессе). На сетке  $400 \times 600$  максимальное ускорение составило 113.73 при использовании 16 потоков (2 MPI-

процесса  $\times$  8 OpenMP-нитей), а на сетке  $800 \times 1200$  — 176.00 при 32 потоках (4 MPI-процесса  $\times$  8 OpenMP-нитей).

## 7.6 Графики сходимости

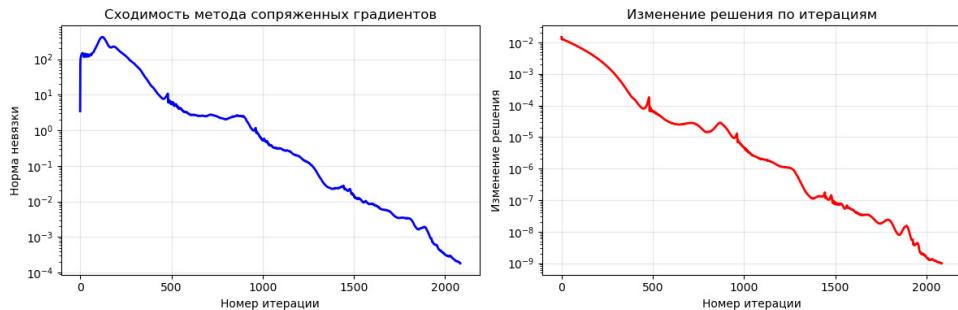


Рис. 4: Сходимость решения для сетки  $800 \times 1200$

## 8 Заключение

В ходе работы была разработана и реализована гибридная параллельная версия программы для решения двумерной задачи Дирихле для уравнения Пуассона в криволинейной области (остроугольном треугольнике) методом фиктивных областей. Программа сочетает использование MPI для распределения вычислений между узлами кластера и OpenMP для параллелизации внутри каждого вычислительного узла.

Основные результаты работы:

1. Разработана гибридная параллельная программа (MPI + OpenMP) для решения уравнения Пуассона в треугольной области с применением метода фиктивных областей и метода сопряженных градиентов с диагональным предобуславливанием.
2. Проведено тестирование программы на различных размерах сеток
3. Достигнуто значительное ускорение вычислений: на сетке  $400 \times 600$  максимальное ускорение составило 113.73 раза при использовании 16 потоков (2 MPI-процесса  $\times$  8 OpenMP-нитей), а на сетке  $800 \times 1200$  — 176.00 раза при использовании 32 потоков (4 MPI-процесса  $\times$  8 OpenMP-нитей).
4. Продемонстрирована эффективность гибридного подхода (MPI + OpenMP), который позволяет оптимально использовать ресурсы вычислительного кластера.

Полученные результаты подтверждают целесообразность использования гибридной параллелизации (MPI + OpenMP) для ускорения итерационных методов решения сеточных уравнений на высокопроизводительных вычислительных системах.