

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

Отчет

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

Выполнила:
Казанцева Варвара Денисовна
Группа: 608

28 октября 2025 г.

Содержание

1	Введение	2
2	Математическая постановка задачи	2
3	Метод фиктивных областей	2
4	Разностная схема решения задачи	2
4.1	Определение коэффициентов разностной схемы	3
4.2	Определение коэффициентов разностной схемы аналитически	3
5	Метод решения системы линейных алгебраических уравнений	3
6	Программная реализация	4
6.1	Функции программы	4
6.2	Параллелизация с OpenMP	5
6.3	Запуск программы	5
7	Результаты расчетов	5
7.1	Последовательная программа	5
7.2	Параллельная программа с OpenMP	5
7.3	Результаты расчетов на ПВС IBM Polus	6
7.4	Графики	6
7.5	График приближенного решения	6
7.6	Графики ускорения	6
7.7	Графики сходимости	8
8	Заключение	8

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 C(3,0), A(3,0), B(0,4)** (вариант 2). Целью работы является разработка последовательной программы и параллельной её версии с использованием OpenMP для решения поставленной задачи.

2 Математическая постановка задачи

В области $D \subset \mathbb{R}^2$, ограниченной кусочно-гладким контуром Γ , 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 Π 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 Π 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

6.1 Функции программы

Функции программы включают:

- `get_x_inter()` и `get_y_inter()` - вычисление координат пересечения прямых, ограничивающих треугольную область
- `is_right()` и `is_left()` - проверка положения точки относительно сторон треугольника
- `compute_a()` - вычисление коэффициентов a_{ij}
- `compute_b()` - вычисление коэффициентов b_{ij}
- `compute_F()` - вычисление правой части F_{ij}
- `applyOperator()` - применение разностного оператора к сеточной функции
- `solveDiagonal()` - решение системы с диагональным предобуславливанием
- `dot()` - вычисление скалярного произведения двух сеточных функций
- `normDifference()` - вычисление нормы разности двух решений для критерия останова
- `conjugateGradient()` - основная функция метода сопряженных градиентов

6.2 Параллелизация с OpenMP

Для ускорения вычислений использованы следующие директивы OpenMP:

- `#pragma omp parallel for collapse(2)` - для распараллеливания вложенных циклов по сетке
- `#pragma omp parallel for reduction(+:result)` - для параллельного вычисления скалярных произведений с редуccionной операцией
- `omp_set_num_threads()` - для установки числа потоков выполнения

6.3 Запуск программы

Для сборки последовательной версии программы используется Makefile. Команды для сборки и запуска:

```
make compile
make run
```

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

```
make compile
make run_1    # 1 поток
make run_4    # 4 потока
make run_16   # 16 потоков
make run_polus # пакетный запуск на кластере Polus
```

Число потоков передается через аргумент командной строки и устанавливается с помощью `omp_set_num_threads()`.

7 Результаты расчетов

7.1 Последовательная программа

Размер сетки	Число итераций	Время решения, с
10×10	27	0.0005
20×20	54	0.0038
40×40	103	0.0296

Таблица 1: Результаты расчетов последовательной программы

7.2 Параллельная программа с OpenMP

Проведены расчеты на сетке $(M, N) = (40 \times 40)$ на различном количестве нитей.

Количество нитей	Число итераций	Время решения, с
1	103	0.031
4	103	0.016
16	103	0.077

Таблица 2: Результаты расчетов OpenMP-программы на сетке 40×40

7.3 Результаты расчетов на ПВС IBM Polus

Количество OpenMP-нитей	Число точек сетки ($M \times N$)	Число итераций	Время решения, с	Ускорение
0	400×600	1077	112.59	1.0
2	400×600	1077	60.32	1.87
4	400×600	1077	34.92	3.22
8	400×600	1077	26.93	4.18
16	400×600	1077	16.09	7.00
0	800×1200	2047	857.14	1.0
4	800×1200	2047	250.45	3.42
8	800×1200	2047	169.35	5.06
16	800×1200	2047	122.98	6.97
32	800×1200	2047	51.91	16.51

Таблица 3: Таблица с результатами расчетов на ПВС IBM Polus (OpenMP)

7.4 Графики

7.5 График приближенного решения

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

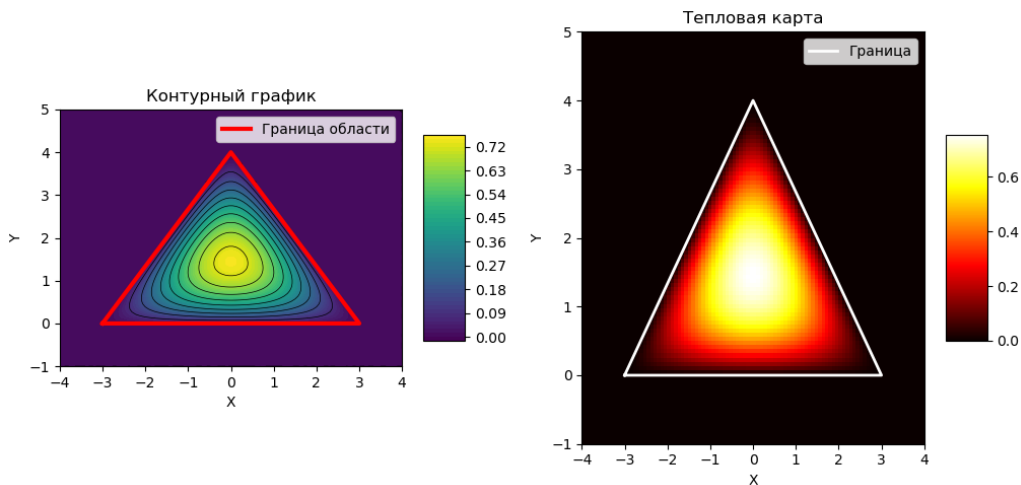


Рис. 1: Приближенное решение уравнения Пуассона в треугольной области на сетке 800×1200

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.

7.6 Графики ускорения

На рисунках 2 и 4 приведены графики ускорения программы при различном числе потоков OpenMP для сеток 400×600 и 800×1200 соответственно.

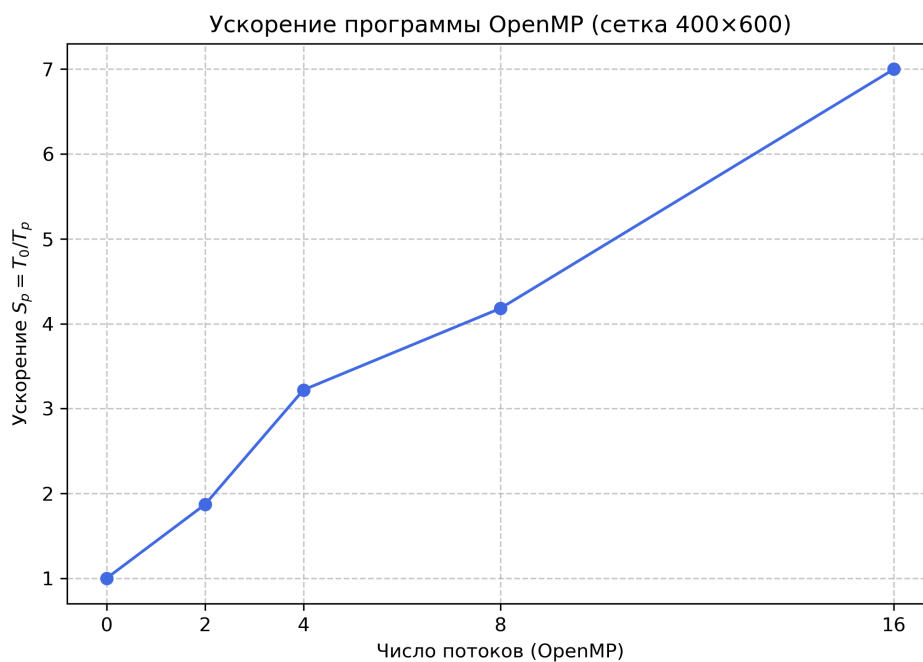


Рис. 2: Ускорение программы OpenMP для сетки 400×600

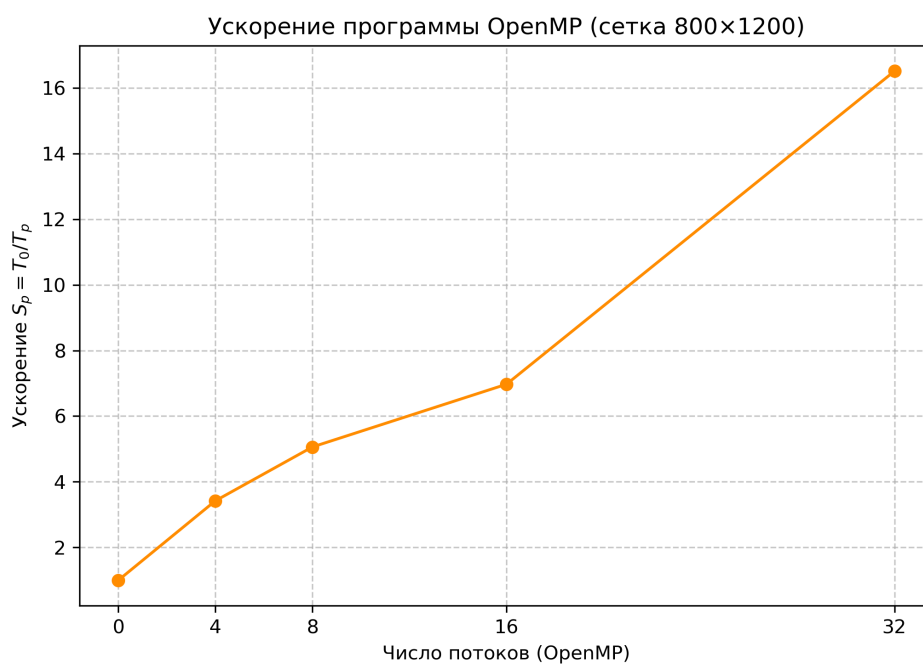


Рис. 3: Ускорение программы OpenMP для сетки 800×1200

Из графиков видно, что с увеличением числа потоков ускорение возрастает. Максимальное ускорение составило 16.51 при 32 потоках на сетке 800×1200 .

7.7 Графики сходимости

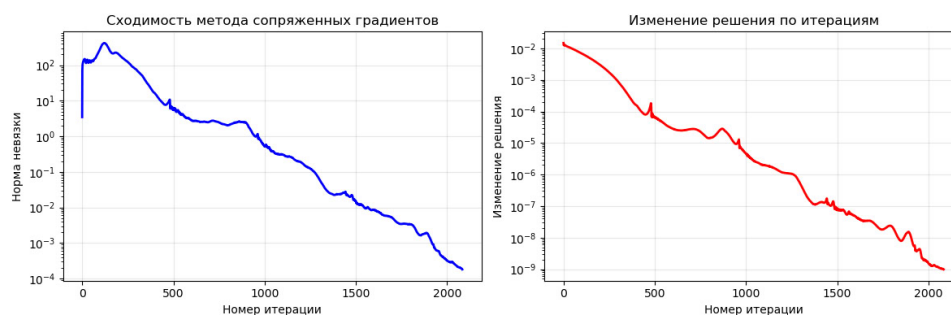


Рис. 4: Сходимость решения для сетки 800×1200

8 Заключение

В ходе работы была разработана и реализована последовательная и параллельная версии программы для решения двумерной задачи Дирихле для уравнения Пуассона в криволинейной области (остроугольном треугольнике) методом фиктивных областей.

Основные результаты работы:

1. Реализован численный алгоритм на основе метода конечных разностей и метода сопряженных градиентов с диагональным предобуславливанием
2. Разработана эффективная параллельная версия программы с использованием технологии OpenMP
3. Проведено тестирование программы на различных размерах сеток и количестве потоков
4. Получено ускорение вычислений до 16.51 раз на сетке 800×1200 при использовании 32 потоков OpenMP
5. Продемонстрирована зависимость эффективности параллелизма от размера решаемой задачи

Наибольшая эффективность параллельных вычислений достигается на крупных сетках, где вычислительная нагрузка на каждый поток достаточна для компенсации накладных расходов на организацию параллелизма. Полученные результаты подтверждают целесообразность использования OpenMP для ускорения итерационных методов решения сеточных уравнений.