msu-eps-converted-to.pdf

Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра системного анализа

Отчет по заданию

«Реализация параллельного алгоритма с использованием технологии OpenMP»

Студент 616 группы

М. Н. Преображенский

31 oct 2025

Содержание

1 Введение

Цель работы — решить двумерную задачу Дирихле для уравнения Пуассона в криволинейной области методом фиктивных областей, реализовать вычисления на OpenMP и исследовать масштабируемость на ПВС IBM Polus.

2 Математическая постановка задачи

Рассматривается задача Пуассона в криволинейной области $D \subset \mathbb{R}^2$, ограниченной контуром γ:

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

с граничным условием Дирихле первого рода

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

В данной работе $f(x, y) \equiv 1$. Для **варианта 10** область D задаётся неравенствами

$$D = \{(x, y) : \; x^2 - 4y^2 > 1, \; 1 < x < 3\},$$

то есть область ограничена дугой гиперболы и отрезком прямой $x = 3$.

3 Краткое описание численного метода решения

Далее изложен метод применительно к варианту 10.

3.1. Метод фиктивных областей и переформулировка задачи

Пусть $D \subset \Pi = \{(x, y) : A_1 < x < B_1, \; A_2 < y < B_2\}$ — охватывающий прямоугольник, $\widehat{D} = \Pi \setminus D$ — фиктивная область. В Π решается задача

$$-\frac{\partial}{\partial x}\Big(k\, u_x\Big) - \frac{\partial}{\partial y}\Big(k\, u_y\Big) = F(x, y), \quad (x, y) \in \Pi \setminus \gamma, \qquad u|_{\partial\Pi} = 0, \tag{3}$$

где кусочно-постоянный коэффициент

$$k(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 1/\varepsilon, & (x, y) \in \widehat{D}, \end{cases} \qquad \varepsilon = \max(h_x, h_y)^2.$$

Правая часть берётся как $F \equiv f \equiv 1$ (внутри D) и затухает в $\widehat{D}$ по определению (**??**).

3.2. Сетка и нотация

Покроем П равномерной сеткой ω_h с внутренними узлами $M \times N$, шаги $h_x = \frac{B_1 - A_1}{M+1}$, $h_y = \frac{B_2 - A_2}{N+1}$. Обозначим полуцелые точки $x_{i\pm\frac{1}{2}} = x_i \pm \frac{h_x}{2}$, $y_{j\pm\frac{1}{2}} = y_j \pm \frac{h_y}{2}$.

3.3. Разностная схема (5-точечный шаблон с переменными коэффициентами)

Дифференциальный оператор аппроксимируем дивергентной схемой:

$$-\frac{1}{h_x}\left(a_{i+1,j}\,\frac{w_{i+1,j} - w_{i,j}}{h_x} - a_{i,j}\,\frac{w_{i,j} - w_{i-1,j}}{h_x}\right) - \frac{1}{h_y}\left(b_{i,j+1}\,\frac{w_{i,j+1} - w_{i,j}}{h_y} - b_{i,j}\,\frac{w_{i,j} - w_{i,j-1}}{h_y}\right) = F_{ij}, \tag{4}$$

для $i = 1, \ldots, M$, $j = 1, \ldots, N$, где *гранные коэффициенты* определяются интегралами

$$a_{i,j} = \frac{1}{h_y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} k(x_{i-\frac{1}{2}}, t)\ \mathrm{d}t, \qquad b_{i,j} = \frac{1}{h_x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} k(t, y_{j-\frac{1}{2}})\ \mathrm{d}t. \tag{5}$$

Правая часть ячейки

$$F_{ij} = \frac{1}{h_x h_y}\iint\limits_{\Pi_{ij}} F(x,y)\ \mathrm{d}x\ \mathrm{d}y, \qquad \Pi_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]. \tag{6}$$

Граничные узлы прямоугольника $\partial\Pi$ задаются условием $w_{ij} = 0$ и исключаются из системы. Получаем СЛАУ $Aw = F$ с самосопряжённым положительно-определённым оператором (см. методичку: доказательство SPD через интегральную форму энергии).

Практическое вычисление $a_{i,j}$ и $b_{i,j}$. Так как k кусочно-постоянна (1 или $1/\varepsilon$), интегралы (**??**) считаются *аналитически* как доля *длины* соответствующей грани внутри D:

$$a_{i,j} = \frac{\ell_{i,j}^{(D)}}{h_y}\cdot 1 + \left(1 - \frac{\ell_{i,j}^{(D)}}{h_y}\right)\cdot\frac{1}{\varepsilon}, \qquad b_{i,j} = \frac{\tilde{\ell}_{i,j}^{(D)}}{h_x}\cdot 1 + \left(1 - \frac{\tilde{\ell}_{i,j}^{(D)}}{h_x}\right)\cdot\frac{1}{\varepsilon}.$$

Здесь $\ell_{i,j}^{(D)}$ — длина пересечения *вертикального* отрезка $[x_{i-\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ с D, а $\tilde{\ell}_{i,j}^{(D)}$ — длина пересечения *горизонтального* отрезка $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}]$ с D. Для варианта 10 граница задаётся $|y| < \frac{1}{2}\sqrt{x^2 - 1}$ при $1 < x < 3$, поэтому:

$$\ell_{i,j}^{(D)} = \max\Big(0, \min(y_{j+\frac{1}{2}}, \tfrac{1}{2}\sqrt{x_{i-\frac{1}{2}}^2 - 1}) - \max(y_{j-\frac{1}{2}}, -\tfrac{1}{2}\sqrt{x_{i-\frac{1}{2}}^2 - 1})\Big),$$

$$\tilde{\ell}_{i,j}^{(D)} = \max\Big(0, \min(x_{i+\frac{1}{2}}, 3) - \max\big(x_{i-\frac{1}{2}}, \sqrt{1 + 4y_{j-\frac{1}{2}}^2}\big)\Big).$$

То есть мы считаем (5) *аналитически*, а не через усреднение по узлам.

Практическое вычисление F_{ij}. Если $\Pi_{ij} \subset D$, то $F_{ij} \approx f(x_i, y_j) = 1$. Если $\Pi_{ij} \subset \widehat{D}$, то $F_{ij} = 0$. В смешанном случае $F_{ij} \approx \dfrac{S_{ij}}{h_x h_y} \cdot 1$, где $S_{ij} = \mathrm{mes}(\Pi_{ij} \cap D)$ — площадь пересечения; криволинейную границу внутри ячейки можно линеаризовать (методичка). На практике удобно оценивать S_{ij} субсемплингом $q \times q$ (например, $q = 4$).

Выбор ε. По заданию и методичке берём $\varepsilon = \max(h_x, h_y)^2$. Это даёт «жёсткое» подавление фиктивной области без ухудшения обусловленности сверх сеточного уровня.

4 Краткое описание реализации OpenMP-решения

Ниже приведена **структура кода** и где именно включён параллелизм. Реализация соответствует схеме (**??**)–(**??**).

4.1. Модули и ключевые функции

- `in_D(x,y)`, `y_cap(x)` — геометрия варианта 10: $|y| < \frac{1}{2}\sqrt{x^2 - 1}$ при $1 < x < 3$.
- `build_faces(ax, by)` — сборка гранных коэффициентов $a_{i,j}$ (`ax`) и $b_{i,j}$ (`by`) по формулам выше: вычисление длин пересечений вертикальных/горизонтальных граней с D и конструирование смеси 1 и $1/\varepsilon$.
- `build_rhs(F)` — правая часть: для каждой ячейки F_{ij} как доля площади $\Pi_{ij} \cap D$ (субсемплинг $q \times q$).
- `build_diag(A_diag)` — диагональ матрицы A по $a_{i\pm1,j}$, $b_{i,j\pm1}$ (нужна & для матвектора, & для Якоби).
- `matvec(v,Av)` — применение 5-точечного оператора с переменными коэффициентами: главный диагональный вклад $\left((a_L + a_R)/h_x^2 + (b_D + b_U)/h_y^2\right) v_{ij}$ плюс четыре соседних со знаками «минус».
- `dot(a,b)` — скалярное произведение.
- `pcg(A,...)` — PCG с диагональным предобуславливанием (Jacobi): $z = D^{-1} r$, обновления u, r, вычисления α, β, критерий остановки $\|r\|/\|b\| \le 10^{-8}$.
- `io::write_csv` — сохранение `solution.csv` с колонками (x, y, u).

4.2. Где стоит OpenMP

- **Сборка граней** (`build_faces`): двойные циклы по (i, j) с `#pragma omp parallel for coll`
— полностью независимые ячейки, нет гонок.
- **Правая часть** (`build_rhs`): двойной цикл + вложенный мини-цикл субсемплинга — тоже `collapse(2)`; мини-цикл оставляем обычным (мелкий).

- **Диагональ** (`build_diag`): двойной цикл, `collapse(2)`.
- **Матвектор** (`matvec`): основной «тяжёлый» участок в PCG; двойной цикл по внутренним узлам с `collapse(2)`.
- **Скалярные произведения** (`dot`): `#pragma omp parallel for reduction(+:s)`.
- **Покомпонентные операции** ($u+=\alpha p$, $r-=\alpha Ap$, $z=D^{-1}r$, $p=z+\beta p$): линейные проходы `#pragma omp parallel for`.

5 Результаты тестирования программы

5.1 Сходимость по сетке и корректность

В таблице 1 приведено число итераций и конечная относительная невязка для последовательного запуска (1 поток) на разных сетках.

Таблица 1: Сходимость по сетке (p=1): число итераций, конечная относительная невязка и время решения.

Размер сетки $M\times N$	Итерации	$\|r\|/\|b\|$	Время t, мс
10×10	29	5.091×10^{-9}	0.197
20×20	61	7.575×10^{-9}	0.819
40×40	123	9.802×10^{-9}	5.198

5.2 Сравнение последовательной и параллельной версий

Сетка 40×40. Величина САО считается между решениями, полученными при p=1 (seq) и p>1 (par).

Таблица 2: Сетка 40×40: итерации, конечная невязка и время решения при разном числе потоков.

p	Итерации	$\|r\|/\|b\|$	Время t, мс
1	123	9.801×10^{-9}	5.198ms
4	123	9.801×10^{-9}	3.13ms
16	123	9.801×10^{-9}	2.54ms

5.3 Визуализация решения

На рис.1-4 показаны карты $u(x,y)$ на мелкой и крупной сетках.

sol_40x40.png

Рис. 1: Поле распределения потенциала $u(x, y)$ для размера сетки: (a) 40×40

sol_120x80.png

Рис. 2: Поле распределения потенциала $u(x, y)$ для размера сетки: (b) 120×80

sol_1200x800.png

Рис. 3: Поле распределения потенциала $u(x, y)$ для размера сетки: (c) 1200×800

6 Анализ ускорения OpenMP-реализации

6.1 Strong scaling: сетка 400×600

Таблица 3: Ускорение на 400×600.

p	Итерации	Время t, с	Ускорение s	Эффективность eff
1	1520	9.475	1.000	1.000
2	1520	4.740	2.000	1.000
4	1520	3.544	2.674	0.668
8	1520	2.388	3.970	0.496
16	1520	1.413	6.710	0.419
32	1520	1.270	7.460	0.233

scaling_400x600.png

Рис. 4: График ускорения

6.2 Strong scaling: сетка 800×1200

Таблица 4: Ускорение на 800×1200.

p	Итерации	Время T_p, с	Ускорение S_p	Эффективность E_p
1	3076	87.900	1.000	1.000
4	3076	22.404	3.924	0.981
8	3076	18.915	4.647	0.581
16	3076	16.916	5.195	0.188
32	3076	14.616	6.015	0.188

scaling_800x1200.png

Рис. 5: График ускорения

7 Заключение

В ходе работы была реализована и исследована OpenMP-параллельная версия алгоритма решения уравнения Пуассона на прямоугольной сетке методом сопряжённых градиентов (CG/PCG). Проведено тестирование на наборах различных размеров сеток и числа потоков.

Результаты измерений показывают, что реализованная программа демонстрирует почти идеальное ускорение при переходе с одного до двух потоков и далее — выраженно сублинейный рост. Для сетки 400×600 почти идеальное ускорение достигается при $p = 2$ ($S_2 \approx 2.00$, $E_2 \approx 1.00$), при $p = 8$ получаем $S_8 \approx 3.97$ ($E_8 \approx 0.50$), а максимальное измеренное ускорение составляет $S_{32} \approx 7.46$ ($E_{32} \approx 0.23$). Для более крупной задачи 800×1200 масштабируемость лучше: при $p = 4$ имеем $S_4 \approx 3.92$ ($E_4 \approx 0.98$), при $p = 8$ — $S_8 \approx 4.65$ ($E_8 \approx 0.58$), а при $p = 32$ — $S_{32} \approx 6.01$ ($E_{32} \approx 0.19$), что отражает более выгодное соотношение вычислительной и коммуникационной составляющих для крупной сетки.

При дальнейшем увеличении числа потоков (от 8 к 16–32) ускорение продолжает расти, но прирост становится существенно менее выраженным: эффективность падает ниже 0.5 для обеих сеток. Это объясняется ростом накладных расходов на синхронизацию между потоками, конкуренцией за общие ресурсы памяти (memory bandwidth), а также ограничениями кэш-иерархии. Таким образом, на используемой системе практически целесообразно использовать диапазон примерно от 4 до 8 потоков, обеспечивающий наилучший баланс между производительностью и эффективностью.

Полученные результаты подтверждают корректность параллельной реализации и её эффективность при решении задач с большими размерами сетки. Дальнейшее повышение производительности возможно за счёт оптимизации работы с памятью, использования NUMA-распределения и векторизации вычислений.