

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

Тан Ин

## **Задача Дирихле для уравнения Пуассона в криволинейной области**

Суперкомпьютерное моделирование и технологии

Группа 616, Вариант 2

Москва, 2024

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>3</b>
1.1	Математическая постановка задачи . . . . .	3
1.2	Метод фиктивных областей . . . . .	3
1.3	Разностная схема решения . . . . .	4
1.4	Метод минимальных невязок . . . . .	6
<b>2</b>	<b>Решение</b>	<b>8</b>
<b>3</b>	<b>Результаты</b>	<b>10</b>

# 1 Постановка задачи

Требуется приближенно решить задачу Дирихле для уравнения Пуассона в криволинейной области. Задание необходимо выполнить на ПВС Московского университета IBM Polus.

## 1.1 Математическая постановка задачи

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

$$-\Delta u = 1 \quad (1.1)$$

с граничным условием Дирихле:

$$u(x, y) = 0, \quad (x, y) \in \gamma. \quad (1.2)$$

Требуется найти функцию  $u(x, y)$ , удовлетворяющую уравнению (1.1) в области  $D$  и краевому условию (1.2) на ее границе.

Область  $D$  - остроугольный треугольник с вершинами в точках  $C(-3, 0)$ ,  $A(3, 0)$ ,  $B(0, 4)$ ,  $D(0, 3)$ .

## 1.2 Метод фиктивных областей

Пусть область  $D$  принадлежит прямоугольнику  $\Pi = \{(x, y) \mid A.x < x < B.x, A.y < y < C.y\}$ . Обозначим границу прямоугольника  $\Pi$  как  $\Gamma$ .

Разность множеств  $\hat{D} = \Pi \setminus \bar{D}$  называется фиктивной областью.

В прямоугольнике  $\Pi$  рассмотрим следующую задачу Дирихле:

$$-\frac{\partial}{\partial x} \left( k(x, y) \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y} \left( k(x, y) \frac{\partial v}{\partial y} \right) = F(x, y) \quad (2.1)$$

Где  $v(x, y) = 0$ ,  $(x, y) \in \Gamma$ ,

$k(x, y)$  - кусочно-постоянный коэффициент:

$$k(x, y) = \begin{cases} 1, & (x, y) \in D, \\ \frac{1}{\varepsilon}, & (x, y) \in \hat{D}. \end{cases} \quad (2.2)$$

и правой частью:

$$F(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 0, & (x, y) \in \hat{D}. \end{cases} \quad (2.3)$$

Требуется найти непрерывную в  $\Pi$  функцию  $v(x, y)$ , удовлетворяющую дифференциальному уравнению (2.1) всюду в  $\Pi \setminus \gamma$ , равную нулю на границе  $\Gamma$  прямоугольника, и такую, чтобы вектор потока:

$$W(x, y) = -k(x, y) \left( \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \right) \quad (2.4)$$

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:

$$\lim_{(x,y) \rightarrow (x_0,y_0), (x,y) \in D} \frac{\partial v}{\partial \nu} = \lim_{(x,y) \rightarrow (x_0,y_0), (x,y) \in \hat{D}} \frac{\partial v}{\partial \nu}. \quad (2.5)$$

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

$$\max_{(x,y) \in \bar{D}} \|v(x, y) - u(x, y)\| \leq C\varepsilon, \quad C > 0 \quad (2.6)$$

Таким образом, решение новой задачи (2.1) позволяет получить решение исходной задачи (1) с любой наперед заданной точностью  $\varepsilon > 0$ , решая при этом задачу Дирихле с кусочно-постоянным коэффициентом  $k(x, y)$ , но в прямоугольнике  $\Pi$ , содержащем исходную область, что существенно упрощает вычисления.

### 1.3 Разностная схема решения

В замыкании прямоугольника  $\Pi$  определим равномерную прямоугольную сетку  $\bar{\omega}_h = \bar{\omega}_1 \times \bar{\omega}_2$ , где

$$\bar{\omega}_1 = \{x_i = A.x + ih_1, i = 0, \dots, M\}, \quad h_1 = (B.x - A.x)/M, \quad (1)$$

$$\bar{\omega}_2 = \{y_j = A.y + jh_2, j = 0, \dots, N\}, \quad h_2 = (C.y - A.y)/N. \quad (2.7)$$

Множество внутренних узлов сетки  $\bar{\omega}_h$  обозначим  $\omega_h$ .

Рассмотрим линейное пространство  $H$  функций, заданных на сетке  $\omega_h$ . Обозначим через  $w_{ij}$  значение сеточной функции  $H$  в узле сетки  $(x_i, y_j) \in \omega_h$ . Определим скалярное произведение и норму в пространстве сеточных функций  $H$ :

$$(u, v) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} h_1 h_2 u_{ij} v_{ij}, \quad \|u\| = \sqrt{(u, u)}. \quad (2.8)$$

Будем использовать метод конечных разностей, который заключается в замене дифференциальной задачи математической физики на конечно-разностную операторную задачу вида:

$$Aw = B, \quad (2.9)$$

где  $A : H \rightarrow H$ . Дифференциальное уравнение задачи (3) во всех внутренних точках сетки аппроксимируется разностным уравнением:

$$-\frac{1}{h_1} \left( a_{i+1,j} \frac{\omega_{i+1,j} - \omega_{ij}}{h_1} - a_{i,j} \frac{\omega_{ij} - \omega_{i-1,j}}{h_1} \right) - \frac{1}{h_2} \left( b_{i,j+1} \frac{\omega_{i,j+1} - \omega_{ij}}{h_2} - b_{i,j} \frac{\omega_{ij} - \omega_{i,j-1}}{h_2} \right) = F_{ij}, \quad (2.10)$$

при всех  $i = 1, \dots, M, j = 1, \dots, N$ .

Коэффициенты:

$$a_{ij} = \frac{1}{h_2} \int_{y_j - h_2/2}^{y_j + h_2/2} k(x_i, y) dt, \quad (2)$$

$$b_{ij} = \frac{1}{h_1} \int_{x_i - h_1/2}^{x_i + h_1/2} k(t, y_j) dt. \quad (2.11)$$

Правая часть разностного уравнения:

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(x, y) dx dy, \quad (2.12)$$

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

Краевые условия Дирихле в задаче (2.1) аппроксимируются точно равенством

$$w_{ij} = w(x_i, y_j) = 0, \quad (x_i, y_j) \in \Gamma. \quad (2.13)$$

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

Полученная система является линейной относительно неизвестных величин и может быть представлена в виде  $Aw = B$  с самосопряженным и положительно определенным оператором  $A$ . Построенная разностная схема линейна и имеет единственное решение при любой правой части.

Интегралы  $a_{ij}$ ,  $b_{ij}$  будут вычисляться аналитически:

$$a_{ij} = h_2^{-1}l_{ij} + (1 - h_2^{-1}l_{ij})/\varepsilon, \quad (3)$$

$$b_{ij} = h_1^{-1}l_{ij} + (1 - h_1^{-1}l_{ij})/\varepsilon, \quad (2.14)$$

где  $l_{ij}$  - длина части отрезка  $[y_{j-1/2}, y_{j+1/2}]$ , которая принадлежит области  $D$ .

Аналогично для  $b_{ij}$ , где  $l_{ij}$  - длина части отрезка  $[x_{i-1/2}, x_{i+1/2}]$ , которая принадлежит области  $D$ .

Для вычисления  $l_{ij}$  проверяется пересечение соответствующего интеграла интегрирования с прямой, проходящей через вершины трапеции  $CB$ .

Правую часть разностной схемы считаем как  $F_{ij} = s/(h_1h_2)$ , где  $s$  - часть площади прямоугольника с центром  $(x_i, y_j)$  и сторонами  $h_1h_2$ , принадлежащая области  $D$ .

## 1.4 Метод минимальных невязок

Приближенное решение разностной схемы предлагается вычислять методом наименьших невязок. Метод позволяет получить последовательность сеточных функций  $\omega^{(k)} \in H$ ,  $k = 1, 2, \dots$ , сходящуюся по норме пространства  $H$  к решению разностной схемы.

$$\|\omega - \omega^{(k)}\| \rightarrow 0, \quad k \rightarrow \infty \quad (2.16)$$

Начальное приближение  $\omega^{(0)}$  выберем равным нулю во всех точках сетки, кроме одной в центре. В центральной устанавливаем значение  $= 1$ .

Итерация  $\omega^{(k+1)}$  вычисляется по итерации  $\omega^{(k)}$  по формуле:

$$\omega_{ij}^{(k+1)} = \omega_{ij}^{(k)} - \tau_{k+1}r_{ij}^{(k)} \quad (2.17)$$

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

$$\tau_{k+1} = \frac{(Ar^{(k)}, r^{(k)})}{\|Ar^{(k)}\|^2} \quad (4)$$

В качестве критерия останова используется условие:

$$\|r^{(k)}\| < \delta \quad (2.18)$$

с некоторой положительной константой  $\delta > 0$ , задающей точность приближенного решения.

Для вычислений использовалась  $\delta = 10^{-6}$ .

## 2 Решение

[https://github.com/ccying1/supercomputer\\_msu](https://github.com/ccying1/supercomputer_msu)

Требовалось приближенно найти решение задачи (1), (2) для случая, когда  $f(x, y) = 1$  при всех  $(x, y) \in D$ . Где область  $D$  является внутренностью эллипса  $\{(x, y) : |4x| + 3y - 12 \leq 0 \text{ and } y \geq 0\}$ .

- Реализация последовательного кода программы, вычисляющая приближенное решение разностной схемы:
  - вычисление матриц  $F_{ij}$ ,  $a_{ij}$ ,  $b_{ij}$ ;
  - реализация итерационного метода наименьших невязок;
  - выполнение расчетов на сгущающихся сетках  $(M, N) = (10, 10), (20, 20), (40, 40)$ ;
- Разработка параллельного кода программы, вычисляющего приближенное решение разностной схемы, используя средства OpenMP;
- Разработка параллельного кода программы, вычисляющего приближенное решение разностной схемы, используя средства OpenMP.
- Выполнение расчетов на сетке  $(M, N) = (40, 40)$  на одном, четырех, восьми и шестнадцати нитях, сравнение с последовательным вариантом алгоритма.
- Выполнение расчетов на сетках  $(M, N) = (80, 90), (160, 180)$  на 2, 4, 8, 16 и 32 нитях, сравнение с последовательным вариантом алгоритма

Матрицы  $F_{ij}$ ,  $a_{ij}$ ,  $b_{ij}$  вычислялись аналитически:

$$F_{ij} = \begin{cases} 0, & \Pi_{ij} \in \hat{D}, \\ 1, & \Pi_{ij} \in D, \\ \frac{S_{ij}}{h_1 h_2}, & \Pi_{ij} \text{ содержит точки оригинальной и фиктивной областей} \end{cases}$$

где  $S_{ij} = \text{mes}(\Pi_{ij} \cap D)$  - площадь пересечения областей.

$$a_{ij} = \begin{cases} \frac{1}{\epsilon}, & [P_{ij}, P_{i,j+1}] \in \hat{D}, \\ 1, & [P_{ij}, P_{i,j+1}] \in D, \\ \frac{l_{ij}}{h_2} + \left(1 - \frac{l_{ij}}{h_2}\right) / \epsilon, & [P_{ij}, P_{i,j+1}] \text{ содержит точки оригинальной и фиктивной областей} \end{cases}$$



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

- Для реализации параллельного кода программы используются директивы OpenMP для распараллеливания циклов и редукций. Использование директив OpenMP позволяет ускорить вычисления на каждом шаге итерационного метода при запуске программы на системах, поддерживающих параллельные вычисления.
- Результаты работы программы демонстрируют эффективность использования параллельного кода для вычисления приближенного решения итерационным методом на системах с многопоточностью по сравнению с использованием последовательного кода.
- Для части с MPI был разработан алгоритм разделения вычислительной области на процессы. С помощью функции `MPI_Dims_create()` создаётся коммунитор с разбиением на двумерную сетку, размены которого вычисляются функцией. В каждом измерении количество элементов равно общему числу элементов, делённому на количество процессов в этом измерении. Чтобы избежать случайного уменьшения количества вычислительных узлов, в последнем измерении количество узлов устанавливается равным оставшимся элементам исходной сетки.
- Кроме того, для обеспечения обмена результатами между процессами в вычислительной сетке каждого процесса добавлены дополнительные строки и столбцы, чтобы хранить данные соседних процессов. Обмен результатами вычислений происходит с помощью функции `MPI_Sendrecv()`.
- Для реализации программы MPI + OpenMP были добавлены директивы OpenMP. В отличие от реализации с чистым OpenMP, гибридная версия программы оказалась более эффективной. В гибридной версии не все вспомогательные действия выполняются параллельно; некоторые действия ограничиваются одним потоком, а для конкретных вычислительных циклов используется директива `#pragma omp parallel`.

### 3 Результаты

[https://github.com/ccying1/supercomputer\\_msu](https://github.com/ccying1/supercomputer_msu)

Число нитей OpenMP	Размер сетки	Время (s)	Ускорение
2	$80 \times 90$	91	1.26
4	$80 \times 90$	74	1.54
8	$80 \times 90$	63	1.80
16	$80 \times 90$	49	2.29
4	$160 \times 180$	811	1.86
8	$160 \times 180$	575	2.63
16	$160 \times 180$	513	2.94
32	$160 \times 180$	433	3.49

Таблица 1: Таблица с результатами расчётов на разном числе нитей и размерах сетки.

Число процессов MPI	Количество OpenMP-нитей	Размер сетки	Время (s)	Ускорение
2	1	$80 \times 90$	257	1.36
2	2	$80 \times 90$	160	2.19
2	4	$80 \times 90$	132	2.65
2	8	$80 \times 90$	125	2.80
4	1	$160 \times 180$	717	3.52
4	2	$160 \times 180$	485	5.21
4	4	$160 \times 180$	385	6.57
4	8	$160 \times 180$	351	7.21

Таблица 2: Таблица с результатами расчётов на OpenMP+MPI.

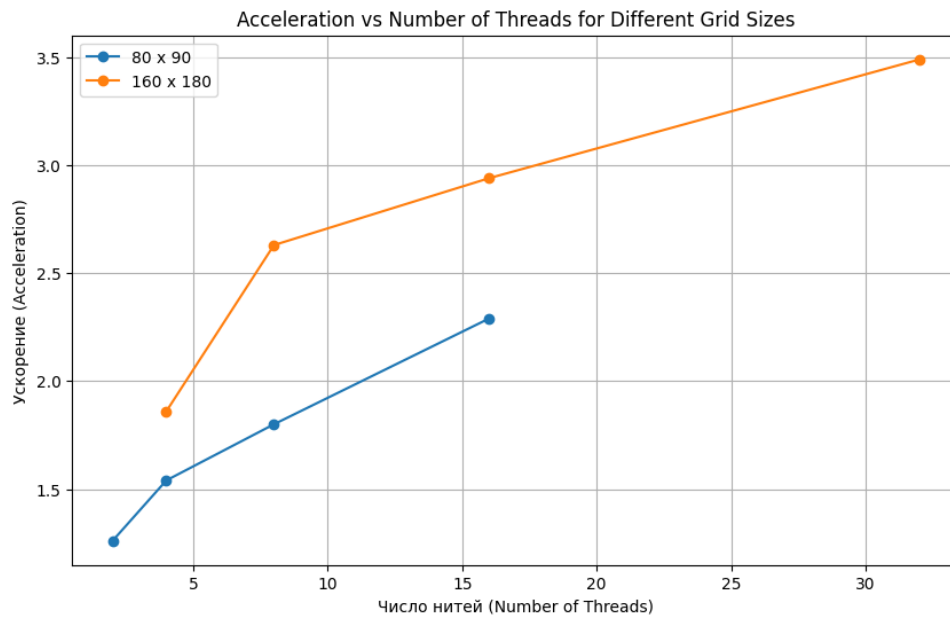


Рис. 1: Ускорение на OPENMP

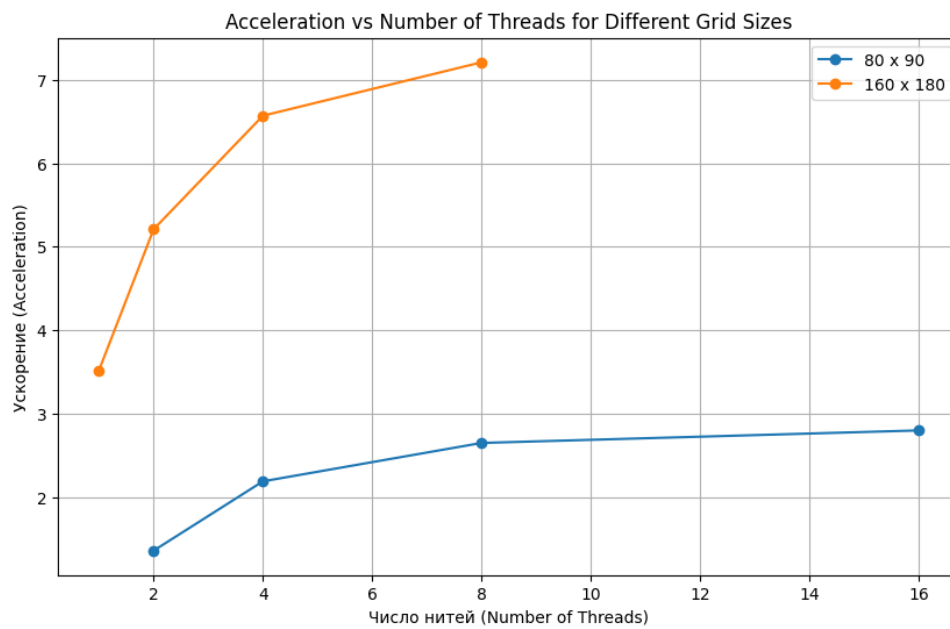


Рис. 2: Ускорение на OPENMP+MPI

Function  $w(x,y)$  - Transposed

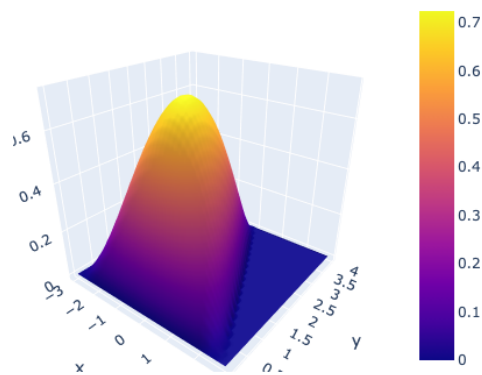


Рис. 3: Итоговый результат на сетке  $40 \times 40$  (ansplot)

Function  $w(x,y)$  - Transposed Heatmap

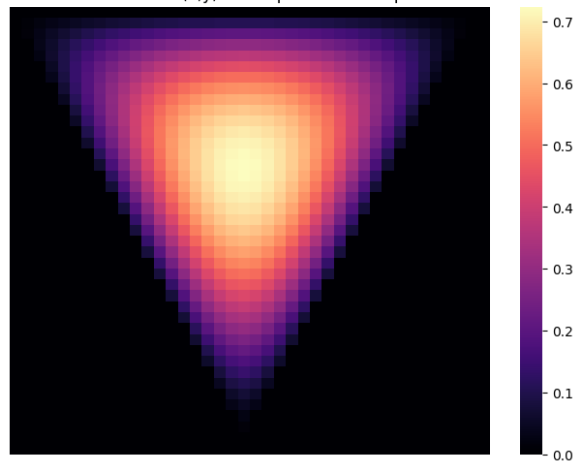


Рис. 4: Итоговый результат на сетке  $40 \times 40$  (ansretu)

Function  $w(x,y)$  - Transposed

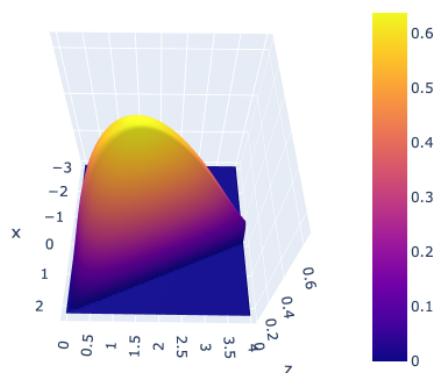


Рис. 5: Итоговый результат на сетке  $40 \times 40$  (ansplot)

Function  $w(x,y)$  - Transposed Heatmap

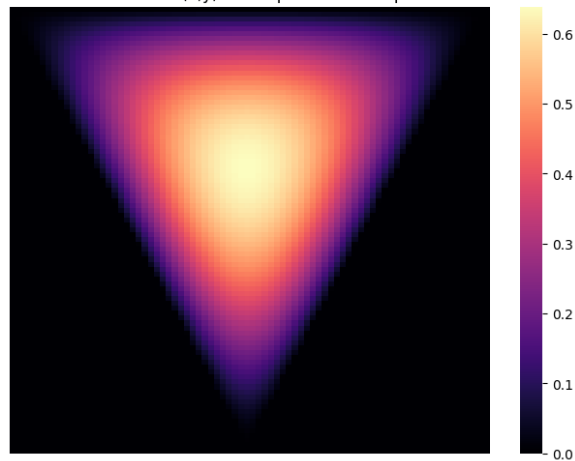


Рис. 6: Итоговый результат на сетке  $80 \times 90$  (ansretu)

Function  $w(x,y)$  - Transposed

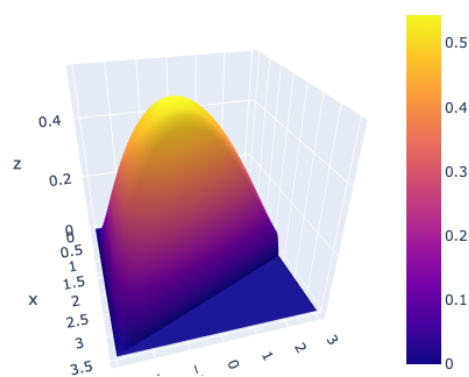


Рис. 7: Итоговый результат на сетке  $40 \times 160$  (ansplot)

Function  $w(x,y)$  - Transposed Heatmap

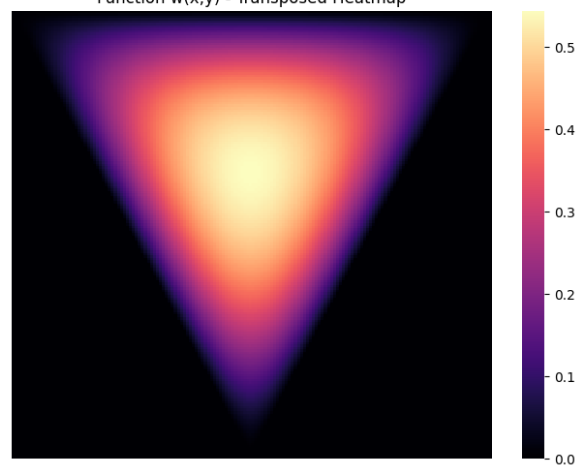


Рис. 8: Итоговый результат на сетке  $40 \times 180$  (ansretu)