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

Бутиков Илья Иванович, 614 группа

Отчёт по выполнению задания курса «Суперкомпьютерное моделирование и технологии»

Москва, 2023

Содержание

1 Постановка задачи

В области $D \subset R^2$, ограниченной контуром γ, рассматривается дифференциальное уравнение Пуассона:

$$-\Delta u = 1 \tag{1.1}$$

с граничными условием Дирихле:

$$u(x, y) = 0, (x, y) \in \gamma. \tag{1.2}$$

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

Область D - трапеция с вершинами в точках $A(0, 0), B(3, 0), C(2, 3), D(0, 3)$.

2 Численный метод решения

Для решения поставленной задачи используется метод фиктивных областей. Полученная новая краевая задача решается численно методом конечных разностей.

2.1 Метод фиктивных областей

Пусть область D принадлежит прямоугольнику $\Pi = \{(x, y)|\ A.x < x < B.x, A.y < y < C.y\}$. Обозначим границу прямоугольника Π как Γ.
Разность множеств $\hat{D} = \Pi \setminus \bar{D}$, называется фиктивной областью.

В прямоугольнике Π рассмотрим следующую задачу Дирихле :

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

Где $v(x, y) = 0, (x, y) \in \Gamma$,
$k(x, y)$ - кусочно-постоянный коэффициент:

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

и правой частью:

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

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

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

имел непрерывную нормальную компоненту на общей части криволинейной границы области D и прямоугольника Π.

Последнее означает, что в каждой точке $(x0, y0) \in \gamma \cap \Pi$ должно выполняться равенство:

$$\lim_{(x,y)\to(x0,y0),(x,y)\in D,} (x,y) \in D, = \lim_{(x,y)\to(x0,y0),(x,y)\in D,} (x,y)\hat{D}, \tag{2.5}$$

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

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

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 в прямоугольнике П, содержащем исходную область, что существенно упрощает вычисления.

2.2 Разностная схема решения

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

$$\begin{aligned} &\bar{\omega}_1 = \{x_i = A.x + ih_1, i = 0, \dots, M\}, \quad h_1 = (B.x - A.x)/M \\ &\bar{\omega}_2 = \{y_j = A.y + jh_2, j = 0, \dots, N\}, \quad h_2 = (C.y - A.y)/N \end{aligned} \tag{2.7}$$

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

Рассмотрим линейное пространство H функций, заданных на сетке ω_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:

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

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

$$A\omega = B, \tag{2.9}$$

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

$$\begin{aligned} &-\tfrac{1}{h_1}(a_{i+1j}\tfrac{\omega_{i+1j}-\omega_{ij}}{h_1} - a_{ij}\tfrac{\omega_{ij}-\omega_{i-1j}}{h_1}) - \tfrac{1}{h_2}(b_{ij+1}\tfrac{\omega_{ij+1}-\omega_{ij}}{h_2} - b_{ij}\tfrac{\omega_{ij}-\omega_{ij-1}}{h_2}) = F_{ij} \\ &i = 1, \dots, M-1,\ j = 1, \dots, N-1 \end{aligned} \tag{2.10}$$

в котором коэффициенты:

$$\begin{aligned} a_{ij} &= \tfrac{1}{h_2} \textstyle\int_{y_{j-1/2}}^{y_{j+1/2}} k(x_{i-1/2}, t)\, dt \\ b_{ij} &= \tfrac{1}{h_1} \textstyle\int_{x_{i-1/2}}^{x_{i+1/2}} k(t, y_{j-1/2})\, dt \end{aligned} \tag{2.11}$$

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

Правая часть разностного уравнения:

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

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

Краевые условия Дирихле в задаче (2.1) аппроксимируются точно равенством

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

Полуцелые узлы означают $x_{i\pm1/2} = x_i \pm 0.5h_1,\ y_{j\pm1/2} = y_j \pm 0.5h_2$.

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

Интегралы a_{ij}, b_{ij} будем вычислять аналитически:

$$a_{ij} = h_2^{-1} l_{ij} + (1 - h_2^{-1} l_{ij})/\varepsilon, \tag{2.14}$$

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

Аналогично для b_{ij}

$$b_{ij} = h_1^{-1} l_{ij} + (1 - h_1^{-1} l_{ij})/\varepsilon, \tag{2.15}$$

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

Для вычисления l_{ij} проверяем пересечение соотвутсвующего интрервала интергирования с прямой, проходящей через вершины трапеции CB,

Правую часть разностной схемы считаем как $F_{ij} = s/(h1\dot{h}2)$, где s - часть площади прямоугольника с центром (xi, yi) и сторонами $h1h2$, принадлежащяя области D

2.3 Метод минимальных невязок

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

$$\|\omega - \omega^{(k)}\| \to 0,\ k \to \infty \tag{2.16}$$

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:

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

где невязка $r^{(k)} = A\omega^{(k)} - B$, итерационный параметр $\tau_{k+1} = \frac{(Ar^{(k)}, r^{(k)})}{\|Ar^{(k)}\|^2}$

В качестве критерия останова используется условие:

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

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

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

3 Описание проделанной работы

Для выполнения задания был разработан последовательный код, представляющий собой программу на языке C++, реализующий численный метод. Были выполнены расчеты на сгущающихся сетках $(M, N) = (10, 10), (20, 20), (40, 40)$ и построены графики полученных приближенных решений.

Для написания параллельной программы основной цикл вычисления был помечен параллельной областью с помощью директивы opm: #pragma omp parallel default(none) shared(tauNumerator, tauDenominator, deltaSqr, w, a, b, F, k) private(i, j, rA, tau) в котором так же указал область видимости всех переменных, объявленых вне цикла и участвующие в различных этапах вычисления.

Область выделена одна для уменьшения накладных расходов на создание и уничтожение параллельных областей.

Как и в последовательной программе, функция рассчёта Aw была переведена на макроподстановку, что дало большое ускорение.

Двойные циклы помечены директивами shedue(static) и collapse(2). В теле таких циклов нет сильных ветвлений или других операций, которые сильно влияют на время выполнение тела цикла, а так же экспериментальные запуски показали, что это более оптимальные параметры.

4 Результаты

Были проведены расчеты на сетках (40, 40), (80, 80), (160, 160) на разном числе потоков. Время вычисления удалось уменьшить за счет использования распараллеливания программы, а результат такой же, как и для последовательной программы. Однако аписанная программа оказалась немасшабируемой на 16 и более нитей. Либо это отличия в способе запуска.

Результаты приведены в таблице.

Число OpenMP-нитей	Число точек сетки $M \times N$	Время решения	Ускорение
2	40×40	3.72226	1.7273
4	40×40	2.48639	2.5859
8	40×40	1.90195	3.3805
2	80×80	140.406	1.24798
4	80×80	88.4111	1.98190
8	80×80	158.751	1.10377
16	80×80	386.766	0.4530
4	160×160	980.983	3.6698
8	160×160	361.23	9.9659
16	160×160	416.894	8.6352
32	160×160	671.496	5.3611

Таблица 1: Зависимость времени решения от числа нитей для разных сеток, $eps = h^2$

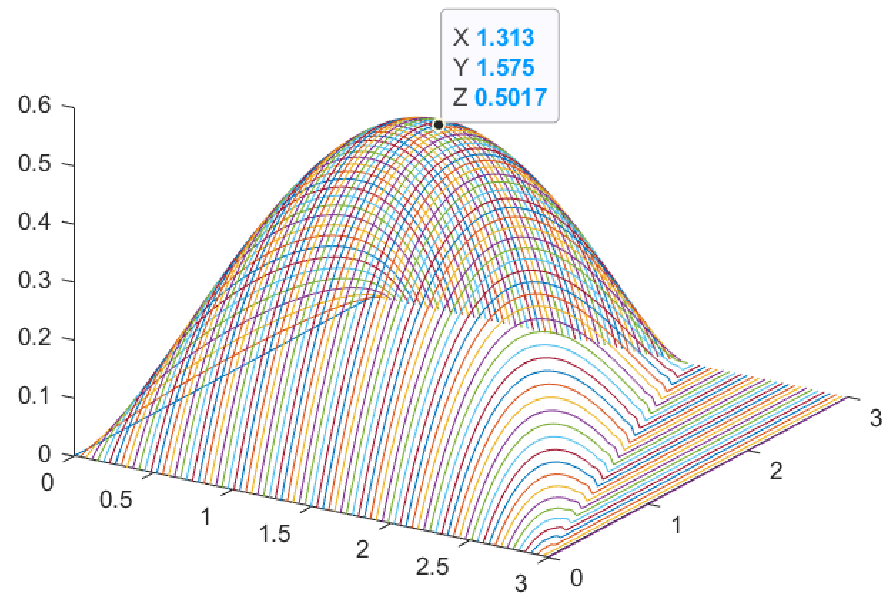

Рис. 1: Итоговый результат на сетке 80 на 80

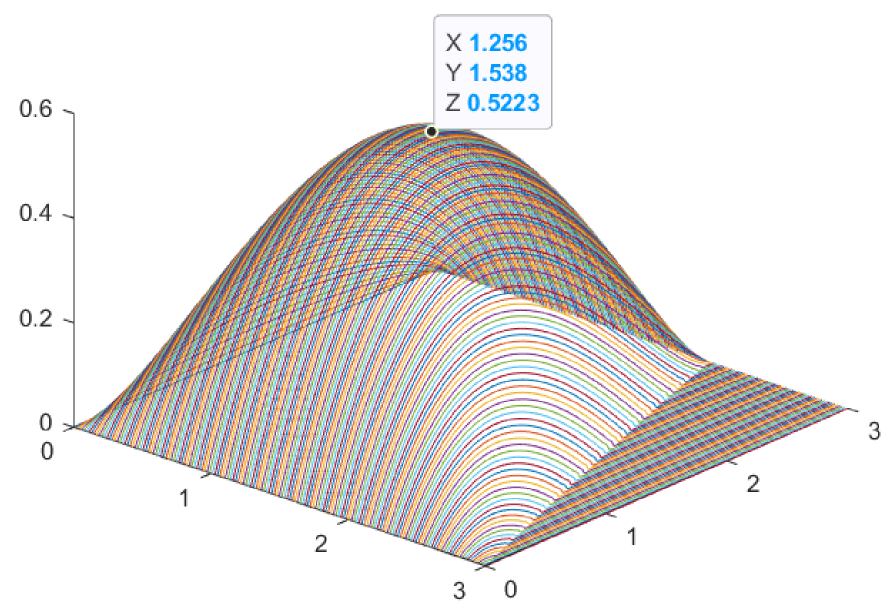

Рис. 2: Итоговый результат на сетке 160 на 160

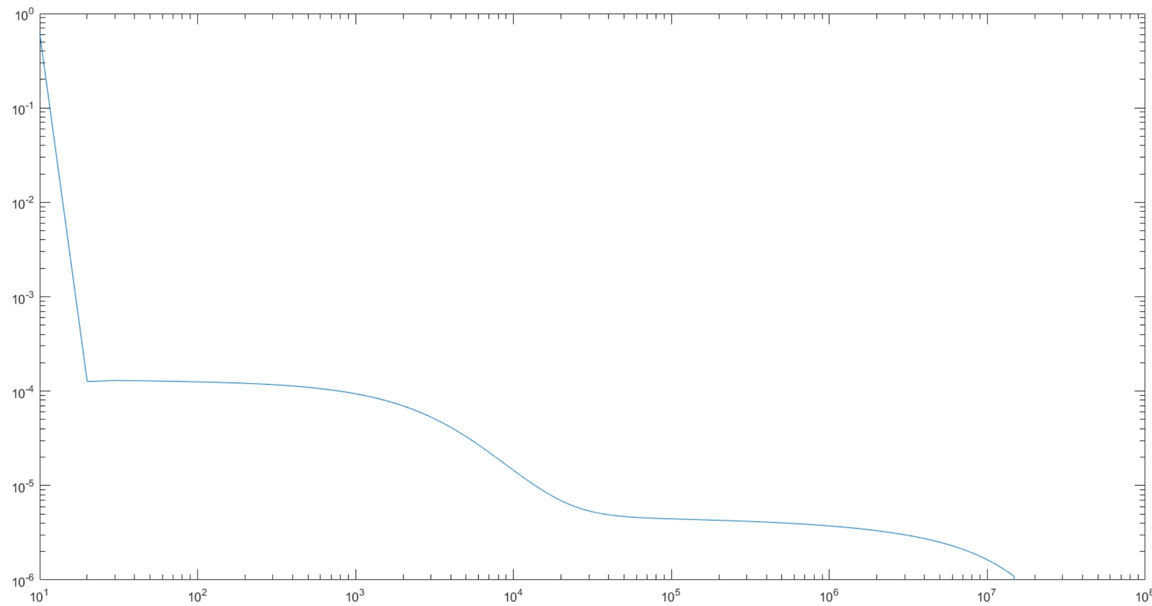

Рис. 3: График погрешности для сетки 160 на 160

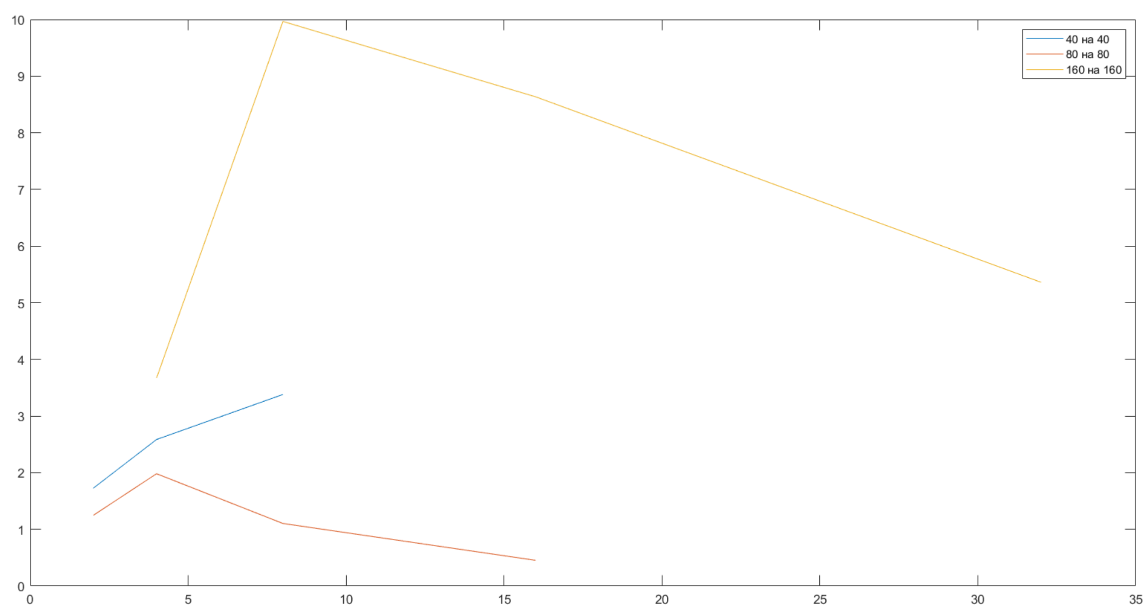

Рис. 4: График ускорения