



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

Отчёт по заданию в рамках курса

**“Суперкомпьютерное моделирование и технологии”**

Численное решение задачи Дирихле для уравнения Пуассона в  
криволинейной области

Выполнил: Морозов М.Г.

608 группа

Вариант 6

Москва 2023

## Введение

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

Исследуемая область  $D = |x| + |y| < 2, y < 1$

## Математическая постановка задачи

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

$$-\Delta u = f(x, y)$$

в котором оператор Лапласа

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

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

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

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

## Численный метод решения уравнения

Для решения был выбран предложенный метод наименьших невязок. Этот метод позволяет получить последовательность сеточных функций  $w^{(k)} \in H, k = 1, 2, \dots$ , сходящуюся по норме пространства  $H$  к решению разностной схемы, т.е.

$$\|w - w^{(k)}\|_E \rightarrow 0, k \rightarrow \infty$$

Начальное приближение  $w^{(0)}$  можно выбрать любым способом, например, равным нулю во всех точках расчетной сетки. Метод является одношаговым. Итерация  $w^{(k+1)}$  вычисляется по итерации  $w^k$  согласно равенствам:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \tau_{k+1} r_{ij}^{(k)}$$

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

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

В качестве условия остановки итерационного процесса следует использовать неравенство

$$\|w^{(k+1)} - w^{(k)}\|_E < \sigma$$

Где  $\sigma$  – положительное число, определяющее точность итерационного метода.

### Краткое описание проделанной работы по созданию MPI-OpenMP-программы

Для реализации поставленной задачи была использована технология MPI и OpenMP.

При подсчете площадей пересечения данной в 6 варианте фигуры с областью  $\Pi_{ij}$  на каждом узле сетки  $w_1 = \{x_i = A_1 + ih_1, i = \overline{0, M}\}$ ,  $w_2 = \{y_j = A_2 + jh_2, j = \overline{0, N}\}$

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

был использован Метод Монте-Карло для полущелых узлов  $x_{i \pm \frac{1}{2}} = x_i \pm$

$$0.5h_1, y_{j \pm \frac{1}{2}} = y_j \pm 0.5h_2.$$

Количество случайно-сгенерированных точек  $npoints = 1000$ .

Размер сетки:  $\{A_1 = A_2 = -4.0, B_1 = B_2 = 4.0\}$  был увеличен, для корректного подсчета на сетке 160X160.

Для реализации распараллеливания OpenMP использовались директивы:

#pragma omp parallel for для арифметических операций

#pragma omp parallel for reduction(+:res) для скалярного произведения

Для реализации распараллеливания MPI использовались директивы:

MPI\_Init(), MPI\_Finalize()

MPI\_Send, MPI\_Recv - для передачи сообщений

**Результаты расчетов для разных размеров задач и на разном числе процессов и нитей.**

<b>Число процессов MPI</b>	<b>Количество OpenMP нитей</b>	<b>Размер сетки</b>	<b>Время</b>	<b>Ускорение</b>
<b>2</b>	<b>1</b>	<b>80*80</b>	<b>152.01</b>	<b>1</b>
<b>2</b>	<b>2</b>	<b>80*80</b>	<b>117.08</b>	<b>1.3</b>
<b>2</b>	<b>4</b>	<b>80*80</b>	<b>80.005</b>	<b>1.9</b>
<b>2</b>	<b>8</b>	<b>80*80</b>	<b>58.46</b>	<b>2.6</b>
<b>4</b>	<b>1</b>	<b>160*160</b>	<b>621.17</b>	<b>1</b>
<b>4</b>	<b>2</b>	<b>160*160</b>	<b>258.75</b>	<b>2.4</b>
<b>4</b>	<b>4</b>	<b>160*160</b>	<b>144.41</b>	<b>4.3</b>
<b>4</b>	<b>8</b>	<b>160*160</b>	<b>108.94</b>	<b>5.7</b>

Ускорение считалось как отношение времени выполнения последовательной программы к времени выполнения программы на определённой конфигурации программы для заданного числа точек сетки  $M*N$ , числа MPI процессов и числа нитей OpenMP.

Графики результатов для сетки размером 160\*160.

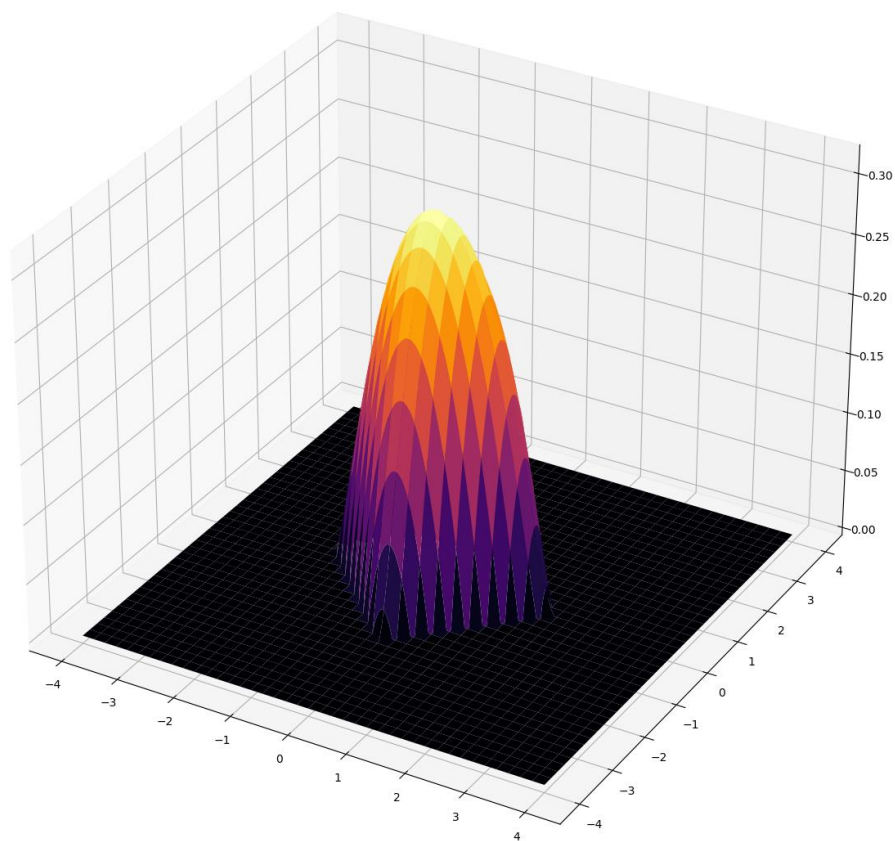
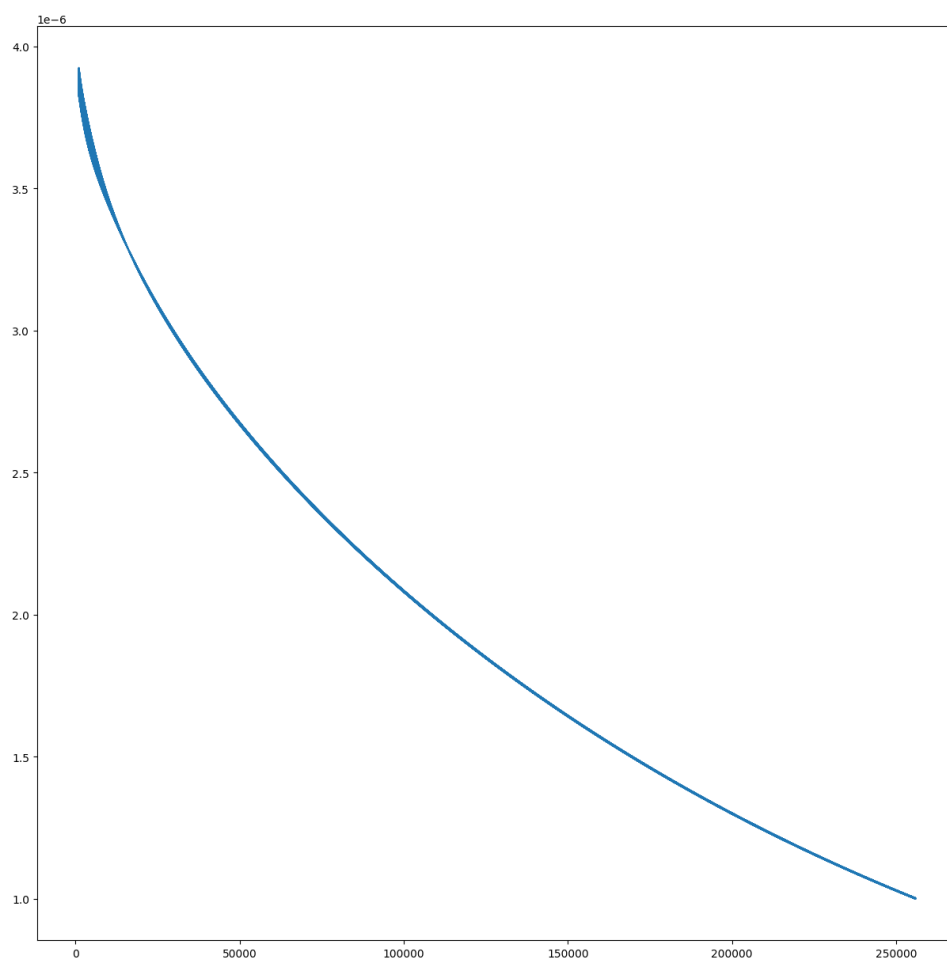
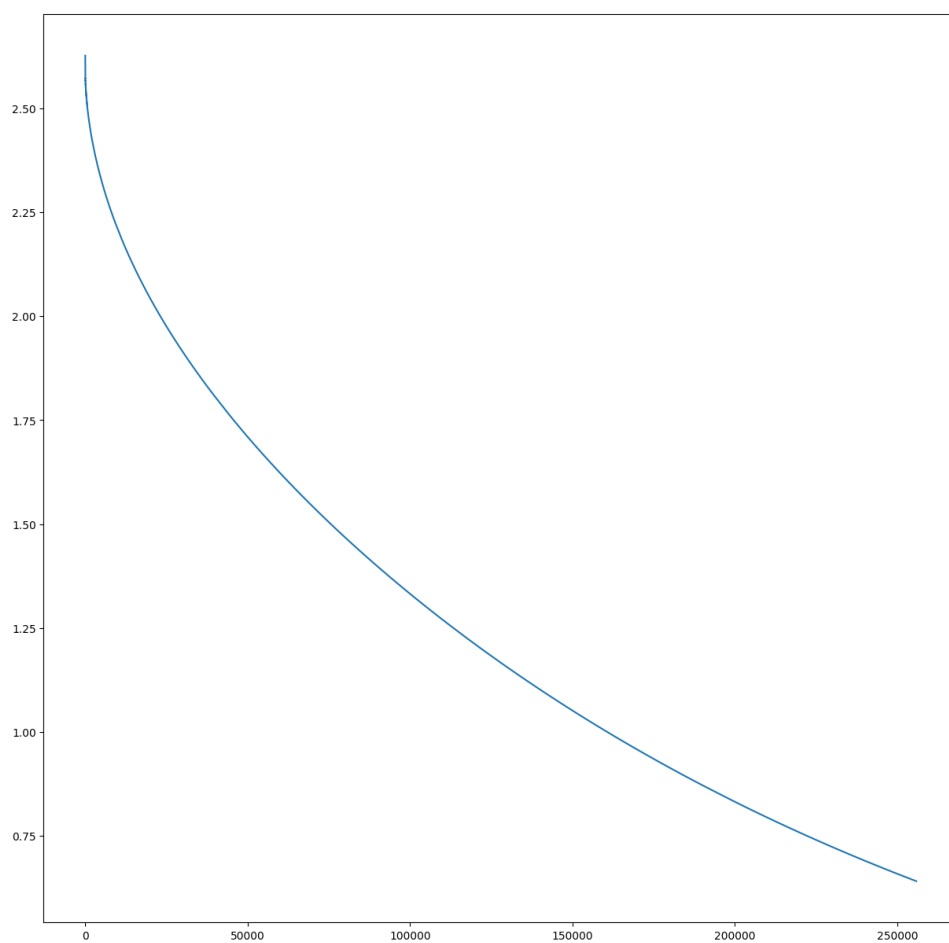


Рис 1. Полученное решение



**Рис 2. Макс. Отклонение от решения**



**Рис 3. График модуля невязки**

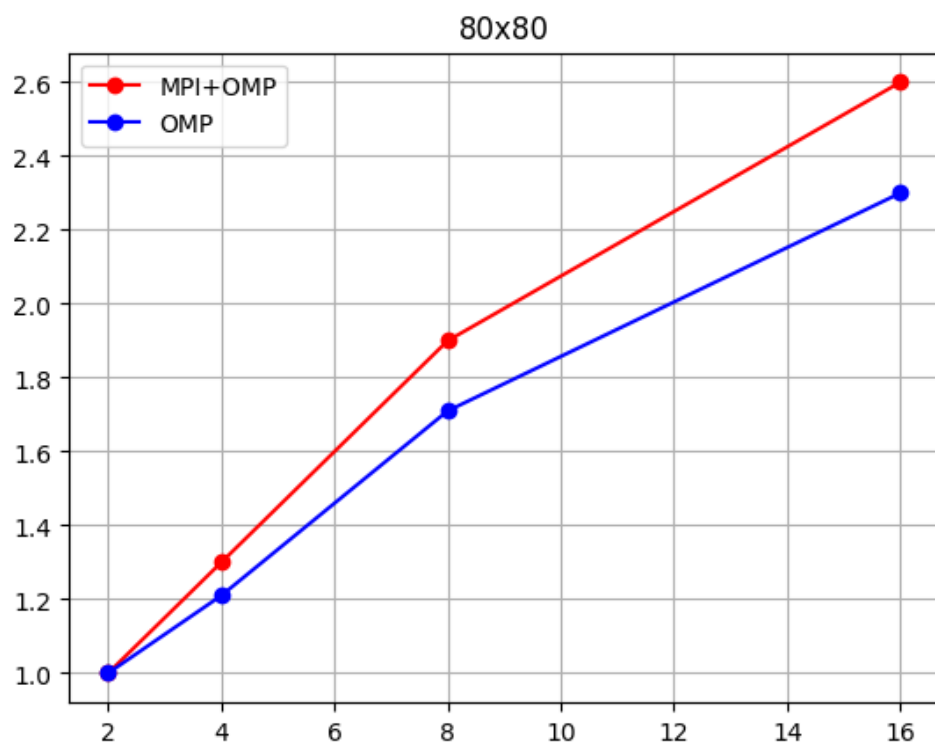


Рис 4. Сравнение ускорения OpenMP+ MPI и OpenMP для параметров M, N = 80.



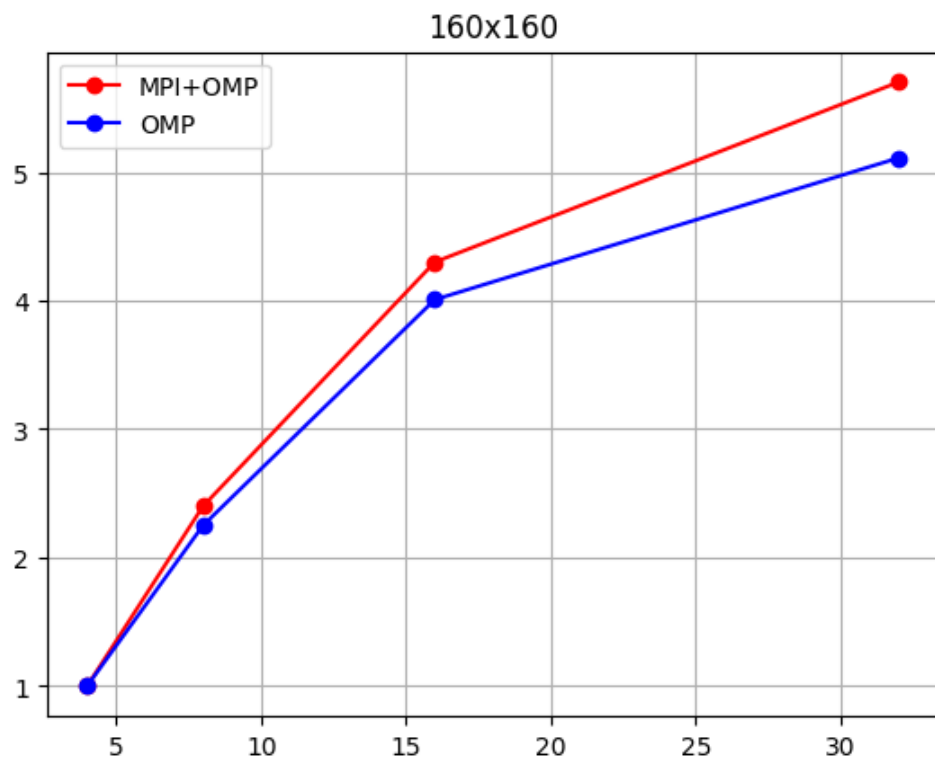


Рис 5. Сравнение ускорения OpenMP+ MPI и OpenMP для параметров M, N = 160.