

Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математической статистики

Губанов Федор Дмитриевич

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

Суперкомпьютерное моделирование и технологии

Группа 616, Вариант 9

Москва, 2023

# 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 Математическую постановку задачи

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

$$-\Delta u = f(x, y), \quad (1)$$

в котором оператор Лапласа

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

функция  $f(x, y)$  считается известной. Для выделения единственного решения уравнение дополняется граничным условием Дирихле:

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

Требуется найти функцию  $u(x, y)$ , удовлетворяющую уравнению (1) в области  $D$  и краевому условию (2) на ее границе.

## 1.2 Метод фиктивных областей

Для приближенного решения задачи (1),(2) предлагается воспользоваться методом фиктивных областей.

Пусть область  $D$  принадлежит прямоугольнику  $\Pi = \{(x, y) : A_1 < x < B_1, A_2 < y < B_2\}$ . Обозначим через  $\bar{D}$ ,  $\bar{\Pi}$  замыкание области  $D$  и прямоугольника  $\Pi$  соответственно, через  $\Gamma$  – границу прямоугольника. Разность множеств

$$\hat{D} = \Pi \setminus \bar{D}$$

называется фиктивной областью. Выберем и зафиксируем малое  $\epsilon > 0$ . В прямоугольнике  $\Pi$  рассматривается задача Дирихле

$$\begin{aligned} -\frac{\partial}{\partial x} \left( k(x, y) \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y} \left( k(x, y) \frac{\partial v}{\partial y} \right) &= F(x, y), \\ v(x, y) &= 0, \quad (x, y) \in \Gamma \end{aligned} \quad (3)$$

с кусочно-постоянным коэффициентом

$$k(x, y) = \begin{cases} 1, & (x, y) \in D, \\ 1/\epsilon, & (x, y) \in \hat{D} \end{cases} \quad (4)$$

и правой частью

$$F(x, y) = \begin{cases} f(x, y), & (x, y) \in D, \\ 0, & (x, y) \in \hat{D} \end{cases} \quad (5)$$

Требуется найти непрерывную в  $\hat{\Pi}$  функцию  $v(x, y)$ , удовлетворяющую дифференциальному уравнению задачи (3) всюду в  $\Pi \setminus \gamma$ , равную нулю на границе  $\Gamma$  прямоугольника, и такую, чтобы вектор потока имел непрерывную нормальную компоненту на общей части криволинейной границы области  $D$  и прямоугольника  $\Pi$ .

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

$$\max_{P \in \hat{D}} |v(x, y) - u(x, y)| < C\epsilon, \quad C > 0. \quad (6)$$

В частности,  $|v(x, y)| < C\epsilon$  во всех точках кривой  $Y$ . Этот результат позволяет получить искомую функцию  $u(x, y)$  с любой наперед заданной точностью  $\epsilon > 0$ , решая задачу (3), (6) вместо задачи (1), (2). Тем самым, задача Дирихле в криволинейной области приближенно заменяется задачей Дирихле в прямоугольнике с кусочно-постоянным коэффициентом  $k(x, y)$ .

### 1.3 Разностная схема решения задачи

Краевые задачу (3), (6) предлагается решать численно методом конечных разностей. В замыкании прямоугольника  $\Pi$  определяется равномерная прямоугольная сетка  $\bar{\omega}_h = \bar{\omega}_1 \times \bar{\omega}_2$ , где

$$\bar{\omega}_1 = \{x_i = A_1 + ih_1, \quad i = \overline{0, M}\}, \quad \bar{\omega}_2 = \{y_j = A_2 + jh_2, \quad j = \overline{0, N}\}$$

Здесь  $h_1 = (B_1 - A_1)/M$ ,  $h_2 = (B_2 - A_2)/N$ . Через  $\omega_h$  обозначим множество внутренних узлов сетки  $\bar{\omega}_h$ , т.е. множество узлов сетки прямоугольника, не лежащих на границе  $\Gamma$ .

Рассмотрим линейное пространство  $H$  функций, заданных на сетке  $\omega_h$ . Обозначим через  $w_{ij}$  значение сеточной функции  $\omega \in H$  в узле сетки  $(x_i, y_j) \in \omega_h$ . Будем считать, что в пространстве  $H$  задано скалярное произведение и евклидова норма

$$(u, v) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} h_1 h_2 u_{ij} v_{ij}, \quad \|u\|_E = \sqrt{(u, u)}. \quad (7)$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вид

$$A\omega = B, \quad (8)$$

где  $A : H \rightarrow H$  – оператор, действующий в пространстве сеточных функций,  $B \in H$  – известная правая часть. Задача (8) называется разностной схемой. Решение этой задачи считается численным решением исходной дифференциальной задачи.

При построении разностной схемы следует аппроксимировать (приблизительно заменить) все уравнения краевой задачи их разностными аналогами – сеточными уравнениями, связывающими значения искомой сеточной функции в узлах сетки. Полученные таким образом уравнения должны быть функционально независимыми, а их общее количество – совпадать с числом неизвестных, т.е. с количеством узлов сетки.

Дифференциальное уравнение задачи (3) во всех внутренних точках сетки аппроксимируется разностным уравнением

$$-\frac{1}{h_1} \left( a_{i+1,j} \frac{\omega_{i+1,j} - \omega_{i,j}}{h_1} - a_{i,j} \frac{\omega_{i,j} - \omega_{i-1,j}}{h_1} \right) - \frac{1}{h_2} \left( b_{i,j+1} \frac{\omega_{i,j+1} - \omega_{i,j}}{h_2} - b_{i,j} \frac{\omega_{i,j} - \omega_{i,j-1}}{h_2} \right) = F_{ij}, \quad i = \overline{1, M-1}, \quad j = \overline{1, N-1}, \quad (9)$$

в котором коэффициент

$$a_{ij} = \frac{1}{h_2} \int_{y_{j-1/2}}^{y_{j+1/2}} k(x_{i-1/2}, t) dt, \quad b_{ij} = \frac{1}{h_1} \int_{x_{i-1/2}}^{x_{i+1/2}} k(t, y_{j-1/2}) dt \quad (10)$$

при всех  $i = \overline{1, M}$ ,  $j = \overline{1, N}$ . Здесь полуцелые узлы

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

Правая часть разностного уравнения

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(x, y) dx dy, \quad \Pi_{ij} = \{(x, y) : x_{i-1/2} \leq x \leq x_{i+1/2}, y_{j-1/2} \leq y \leq y_{j+1/2}\} \quad (11)$$

при всех  $i = \overline{1, M-1}$ ,  $j = \overline{1, N-1}$ .

Краевые условия Дирихле задачи (3) аппроксимируются точно равенств

$$\omega_{ij} = \omega(x_i, y_j) = 0, \quad (x_i, y_j) \in \Gamma. \quad (12)$$

Переменные  $\omega_{ij}$ , заданные равенством (12), исключаются из системы уравнений (11). В результате остаются неизвестными значения  $\omega_{ij}$  при  $i = \overline{1, M-1}$ ,  $j = \overline{1, N-1}$  и их количество совпадает с числом уравнений.

Таким образом, построенная разностная схема линейна и имеет единственное решение при любой правой части.

## 1.4 Метод решения системы линейных алгебраических уравнений

Приближенное решение разностной схемы (9),(14) может быть получено итерационным методом наименьших невязок. Этот метод позволяет получить последовательность сеточных функций  $\omega(k) \in H, k = 1, 2, \dots$ , сходящуюся по норме пространства  $H$  к решению разностной схемы, т.е.

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

Начальное приближение  $\omega^{(0)}$  можно выбрать любым способом, например, равным нулю во всех точках расчетной сетки.

Метод является одношаговым. Итерация  $\omega^{(k+1)}$  вычисляется по итерации  $\omega^{(k)}$  согласно равенствам:

$$\omega_{ij}^{(k+1)} = \omega_{ij}^{(k)} - \tau_{k+1} r_{ij}^{(k)} \quad (13)$$

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

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

В качестве условия останова итерационного процесса следует использовать неравенство

$$\|\omega^{(k+1)} - \omega^{(k)}\|_E < \delta,$$

где  $\delta$  – положительное число, определяющее точность итерационного метода. Константу  $\delta$  для данной задачи предлагается взять равной  $10^{-6}$ .

## 2 Решение

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  $\{(x, y) : x^2 + 4y^2 < 1\}$ .

Схема решения:

- 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
- Выполнение расчетов на сетке  $(M, N) = (40, 40)$  на одном, четырех, восьми и шестнадцати нитях, сравнение с последовательным вариантом алгоритма.
- Выполнение расчетов на сетках  $(M, N) = (80, 80), (160, 160)$  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.

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

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

Для вычисления приближенного решения методом наименьших невязок были реализованы функции вычисляющие оператор  $A$  (9), скалярное произведение и евклидову норму в

пространстве  $H$  (7), а также вспомогательные функции для работы с матрицами (умножение на коэффициент, вычитание).

Разработанный последовательный код программы был протестирован на сетках  $(M, N) = (10, 10), (20, 20), (40, 40)$ . На всех вариантах сетки метод сошелся за конечное число итераций. Сравнение результата применения оператора  $A$  (9) к полученной сеточной функции  $\omega$  с вычисленной ранее матрицей  $F_{ij}$  подтвердил корректность реализованного решения.

**OpenMP вариант:** Для реализации параллельного кода программы использовались директивы OpenMP для распараллеливания циклов и редукции, таким образом можно ускорить вычисление оператора  $A$ , скалярного произведения и других функций.

Использование директив OpenMP позволяет ускорить вычисления на каждом шаге итерационного метода при запуске программы на системах с многопоточностью. Результаты демонстрируют эффективность использования параллельного кода программы для вычисления приближенного решения итерационным методом на ПВС IBM Polus.<sup>1</sup>

**MPI вариант:** Для реализации MPI программы были выполнены следующие шаги:

- реализация двумерного разбиения прямоугольника  $\Pi$  на домены (подобласти);
- вычисление подматриц  $F_{ij}, a_{ij}, b_{ij}$  для каждой подобласти;
- реализация функции обмена граничными данными между процессами с помощью MPI (использовалась MPI-функция - MPI\_Sendrecv\_replace);
- реализация итерационного метода наименьших невязок с обменом граничных данных матриц  $w^{(k)}$  и  $r^{(k)} = Aw^{(k)} - B$  между процессами, суммированием и рассылкой данных со всех процессов после операций скалярного произведения (использовалась MPI-функция - MPI\_Allreduce) для подсчета параметра  $\tau_{k+1}$  и нормы  $\|\omega^{(k+1)} - \omega^{(k)}\|_E$ ;

Использование технологии MPI позволяет параллельно производить вычисления итерационного метода для каждой подобласти и объединять данные с разных процессов посредством передачи сообщений. Такой подход значительно ускоряет вычисление приближенного решения.

**MPI + OpenMP вариант:** Для реализации гибридного кода программы были добавлены директивы OpenMP в функциях вычисления оператора, скалярного произведения, а также простых матричных операциях. Таким образом удалось дополнительно распараллелить вычисление циклов для каждой подобласти.

Результаты демонстрируют эффективность использования гибридного кода программы для вычисления приближенного решения итерационным методом на ПВС IBM Polus.<sup>2</sup>

<sup>1</sup>[https://github.com/Fedor1533/SuperComputers\\_msu/tree/main/task2](https://github.com/Fedor1533/SuperComputers_msu/tree/main/task2)

<sup>2</sup>[https://github.com/Fedor1533/SuperComputers\\_msu/tree/main/task3](https://github.com/Fedor1533/SuperComputers_msu/tree/main/task3)

### 3 Результаты

Таблица 1: Таблица с результатами расчётов на разном числе нитей и размерах сетки (OpenMP код).

Число OpenMP-нитей	Число точек сетки ( $M \times N$ )	Время решения, сек	Ускорение
последовательный код	40x40	11.119	—
1	40x40	11.291	0.98
4	40x40	6.128	1.81
8	40x40	4.883	2.28
16	40x40	5.497	2.02
последовательный код	80x80	158.312	—
2	80x80	92.699	1.71
4	80x80	56.475	2.80
8	80x80	38.612	4.10
16	80x80	37.171	4.27
последовательный код	160x160	509.218	—
4	160x160	150.236	3.39
8	160x160	87.010	5.85
16	160x160	58.775	8.66
32	160x160	64.795	7.87

Рис. 1: Ускорение с использованием OpenMP

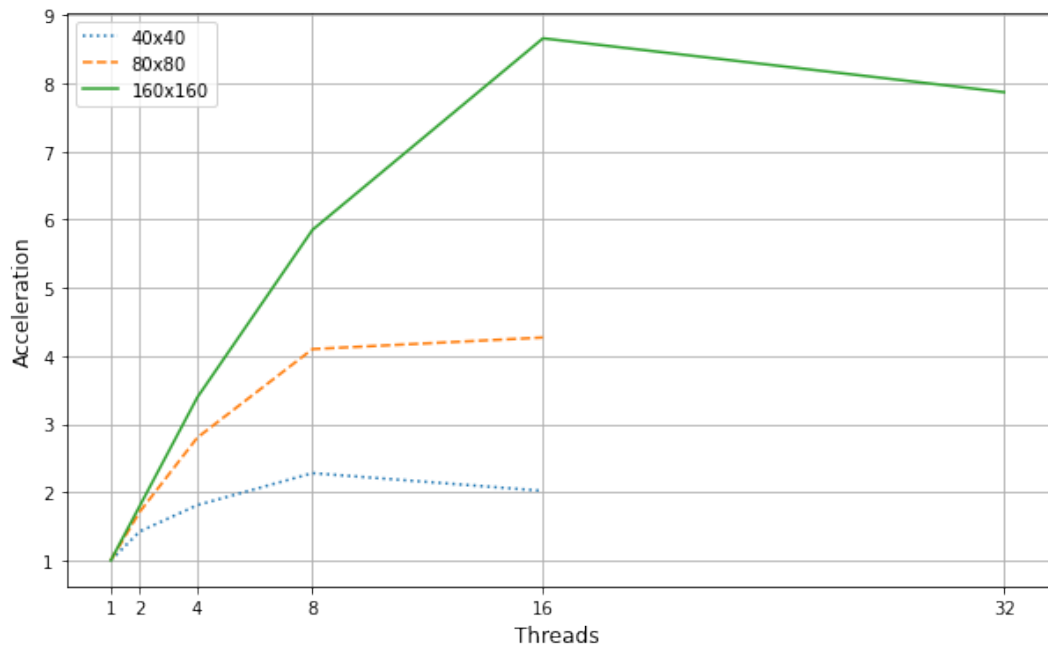




Таблица 2: Таблица с результатами расчётов на разном числе нитей и размерах сетки (MPI+OpenMP код).

Число процессов MPI	Количество OpenMP-нитей в процессе	Число точек сетки ( $M \times N$ )	Время решения, сек	Ускорение
последовательный код	—	40x40	11.119	—
1	—	40x40	11.860	0.94
1	4	40x40	6.524	1.70
2	—	40x40	6.108	1.82
2	4	40x40	5.409	2.05
4	—	40x40	3.948	2.82
последовательный код	—	80x80	158.312	—
2	1	80x80	72.426	2.18
2	2	80x80	39.263	4.03
2	4	80x80	32.814	4.82
2	8	80x80	36.532	4.33
последовательный код	—	160x160	509.218	—
4	1	160x160	141.864	3.58
4	2	160x160	57.608	8.83
4	4	160x160	36.771	13.84
4	8	160x160	45.945	11.08

Рис. 2: Ускорение с использованием MPI+OpenMP

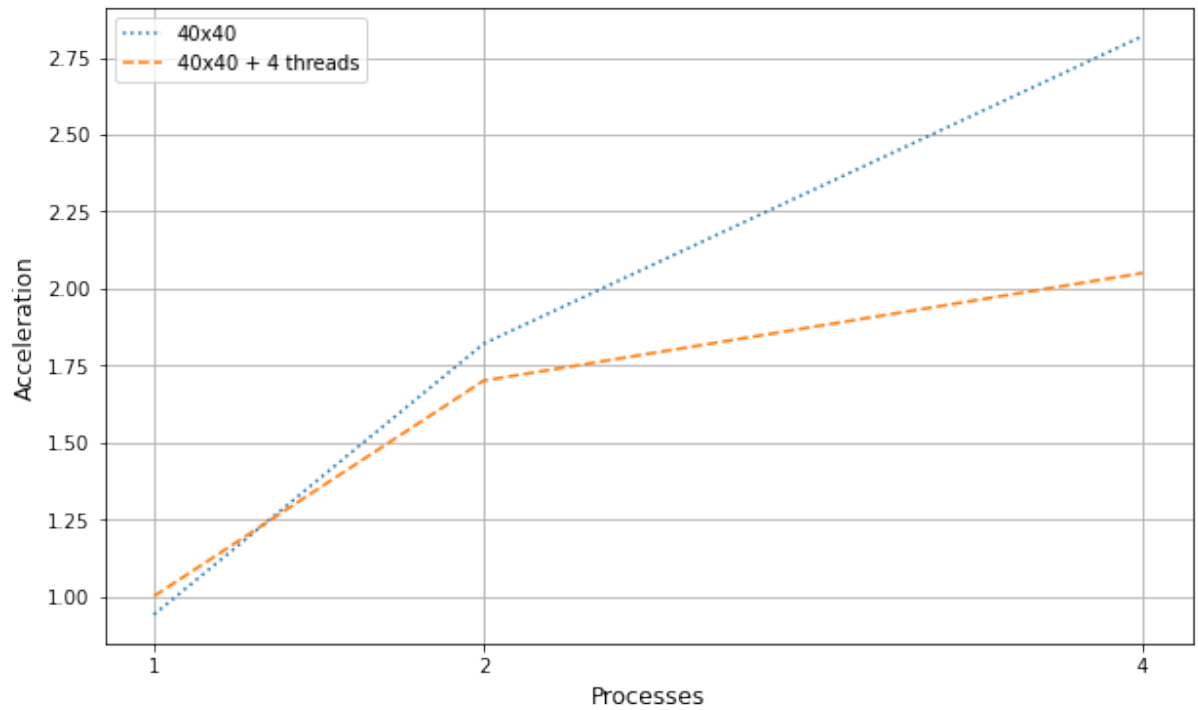
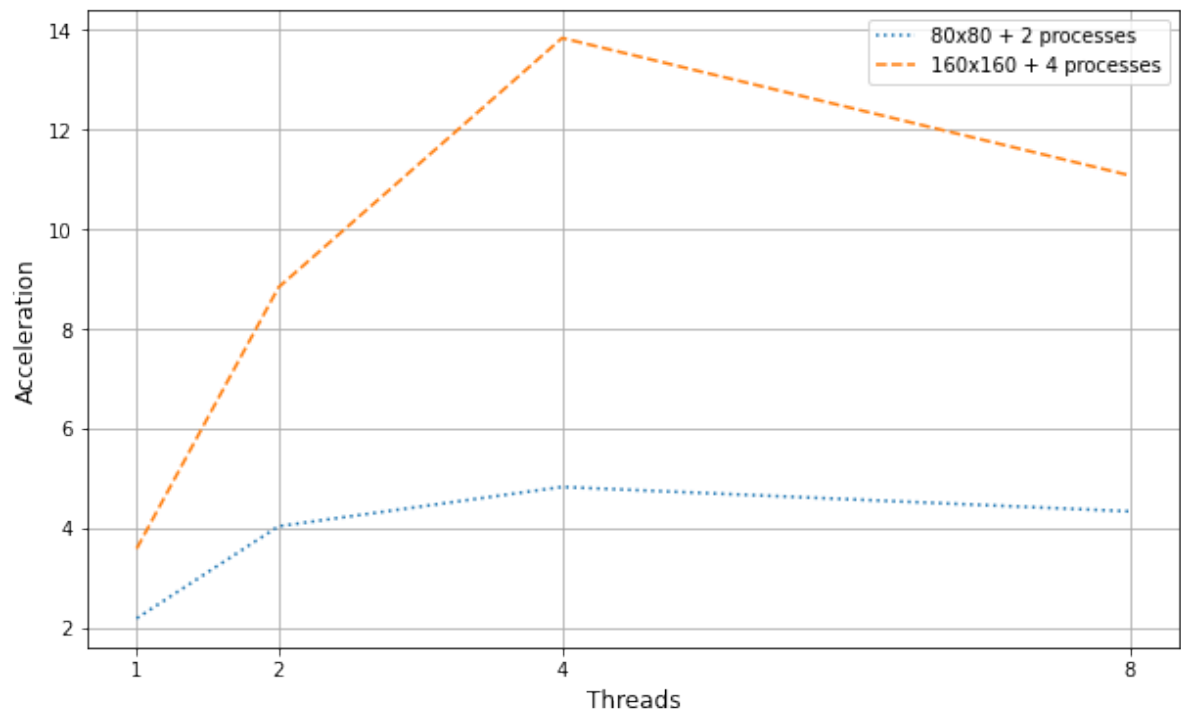
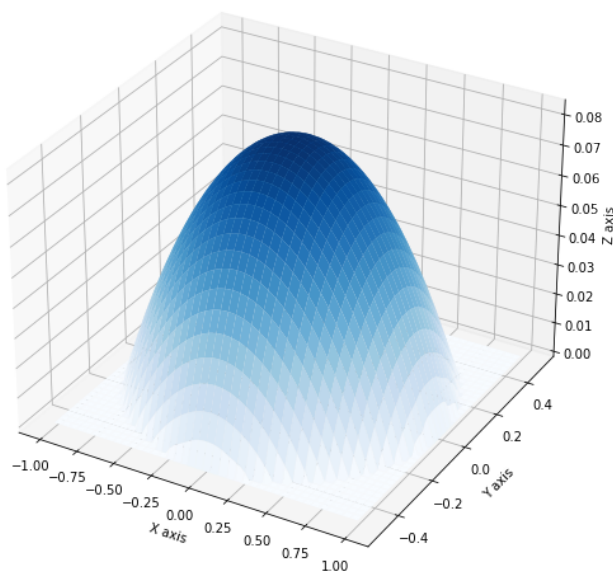
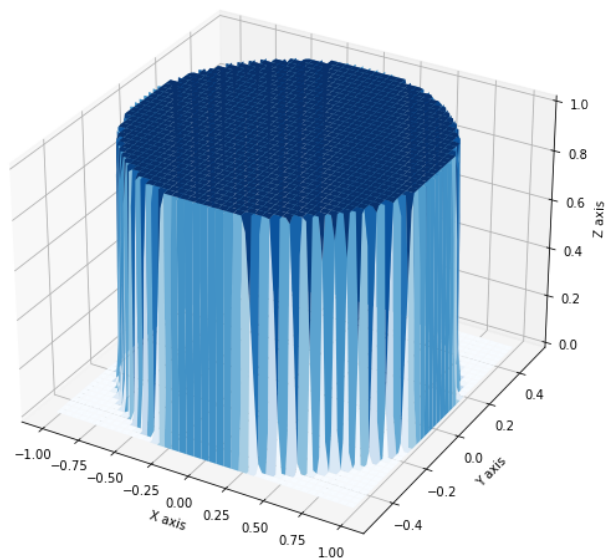


Рис. 3: Ускорение с использованием MPI+OpenMP

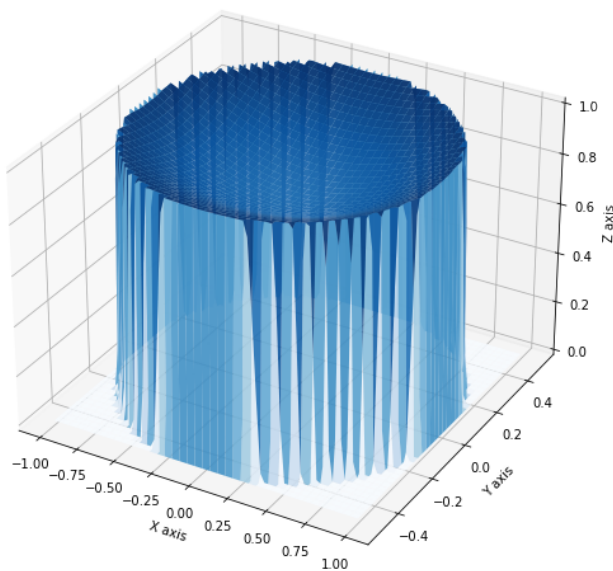




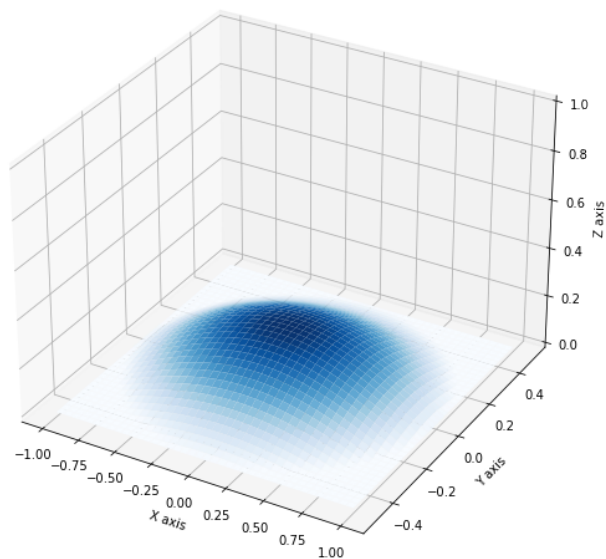
(a) Приближенное решение  $\omega$



(b) Функция  $F$



(c) Функция  $A\omega$



(d) Невязка  $A\omega - F$

Рис. 4: Приближенные решения на сетке 160x160