



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	3
2	Численный метод решения	4
2.1	Метод фиктивных областей	4
2.2	Разностная схема решения	5
2.3	Метод минимальных невязок	6
3	Описание проделанной работы	8
4	Результаты	9

1 Постановка задачи

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

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

с граничными условием Дирихле:

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

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 :

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

Где $v(x, y) = 0, (x, y) \in \Gamma$,

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

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

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 $\bar{\Pi}$ 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:

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

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

Последнее означает, что в каждой точке $(x_0, y_0) \in \gamma \cap \Pi$ должно выполняться равенство:

$$\lim_{(x, y) \rightarrow (x_0, y_0), (x, y) \in D,} (x, y) \in D, = \lim_{(x, y) \rightarrow (x_0, y_0), (x, y) \in \hat{D},} (x, y) \in \hat{D}, \quad (2.5)$$

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

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

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 Π , содержащем исходную область, что существенно упрощает вычисления.

2.2 Разностная схема решения

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

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

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 :

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

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

$$A\omega = B, \quad (2.9)$$

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

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

в котором коэффициенты:

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

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

Правая часть разностного уравнения:

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

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

$i = 1, \dots, M-1, j = 1, \dots, N-1$.

Краевые условия Дирихле в задаче (2.1) аппроксимируются точно равенством

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

Полуцелые 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 $A\omega = B$ 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:

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

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

Аналогично для b_{ij}

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

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

Для вычисления l_{ij} проверяем пересечение соответствующего интервала интегрирования с прямой, проходящей через вершины трапеции CB ,

Правую часть разностной схемы считаем как $F_{ij} = s/(h_1 h_2)$, где s - часть площади прямоугольника с центром (x_i, y_i) и сторонами $h_1 h_2$, принадлежащая области D

2.3 Метод минимальных невязок

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

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

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

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 $\delta = 10^{-6}$

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 `omp: #pragma omp parallel private(i, j, rA, tau)`, в котором так же указал область приватные переменные, объявленные 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 расчёта 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.

Код, отвечающий за вывод промежуточных данных помечен директивами `#ifdef COMMAND`, чтобы не тормозить вычисления, когда вывод не нужен.

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. В презентационных материалах к курсу неправильно выписан скрипт для запуска программы более 8 нитей, что давало неправильный результат расчёта эффективности. Все результаты были пересчитаны заново.

Использование $\epsilon = h^2$ приводит к увеличению времени работы на том же числе процессов при увеличении размера сетки в 2 раза не в 4 раза, а в 20 раз. Возможная причина - достаточно маленькое τ в ходе расчёта, которое сильно зависит от Ar в фиктивной области, что зависит от $1/\epsilon$. Из-за этого часть результатов не удалось получить. Оценка недостающих результатов показывает, что вычисления заняли бы больше 30 минутного лимита.

Несмотря на случайность времени вычисления, связанной с нагрузкой на суперкомпьютер, в среднем использование флага компиляции O2 даёт преимущество в скорости. Но чем больше потоков используется, тем меньше его вклад. Из-за чего ускорение для программы с флагом O2 меньше, так как сама последовательная программа существенно быстрее.

Результат совпадает с последовательным программой. Однако решение на сетке 160 на 160 сильно отличается от меньших сеток. Помимо неправильно написанной программы, возможна причина в слишком большом соотношении ϵ/δ .

Так же дополнительно было рассчитана ошибка в каждой точке сетки. Вычисления показывают, что ошибка распределена равномерно по графику, а график ошибки похож на график результата, но с меньшими значениями.

Результаты приведены в таблице.

Число OpenMP-нитей	Число точек сетки $M \times N$	Время решения	Ускорение
2	40×40	6.261	1.666
4	40×40	4.371	2.387
8	40×40	3.197	3.263
16	40×40	3.809	2.739
2	80×80	217.719	1.897
4	80×80	120.175	3.437
8	80×80	72.8317	5.446
16	80×80	59.8206	6.904
4	160×160	>1800, ожидаемо 2644	ожидаемо 2.74
8	160×160	>1800, ожидаемо 2358	ожидаемо 3.072
16	160×160	1177.57	6.151
32	160×160	814.319	8.896

Таблица 1: Зависимость времени решения от числа нитей для разных сеток, использовался флаг компиляции O2

Число OpenMP-нитей	Число точек сетки $M \times N$	Время решения	Ускорение
2	40×40	31.097	1.71
4	40×40	21.619	2.46
8	40×40	18.086	2.941
16	40×40	4.153	12.807
2	80×80	674.737	1.571
4	80×80	375.789	2.821
8	80×80	156.891	6.756
16	80×80	64.1665	16.52
4	160×160	>1800, ожидаемо 2650	3.24
8	160×160	1401.896	6.125
16	160×160	1227.53	6.995
32	160×160	823.882	10.422

Таблица 2: Зависимость времени решения от числа нитей для разных сеток, без использования флага компиляции O2

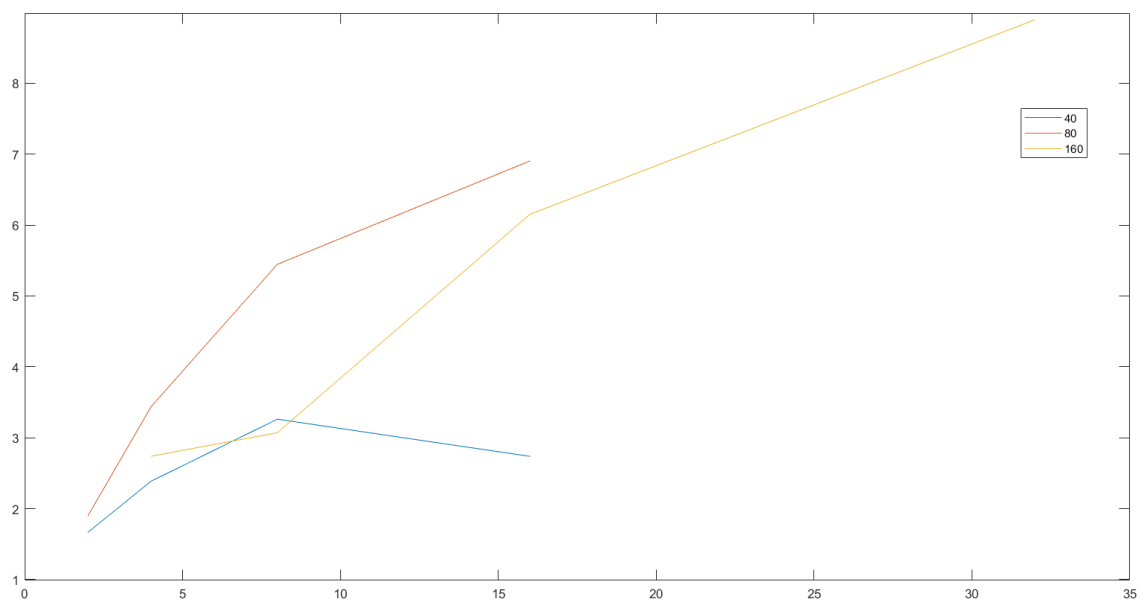


Рис. 1: График ускорения с флагом O2

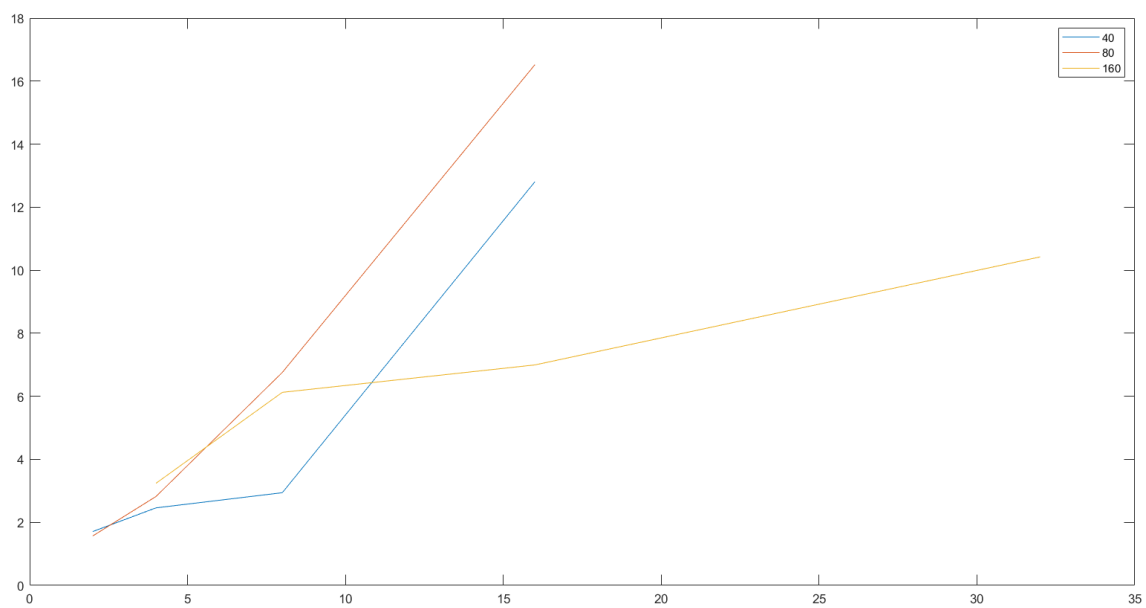


Рис. 2: График ускорения без флага O2

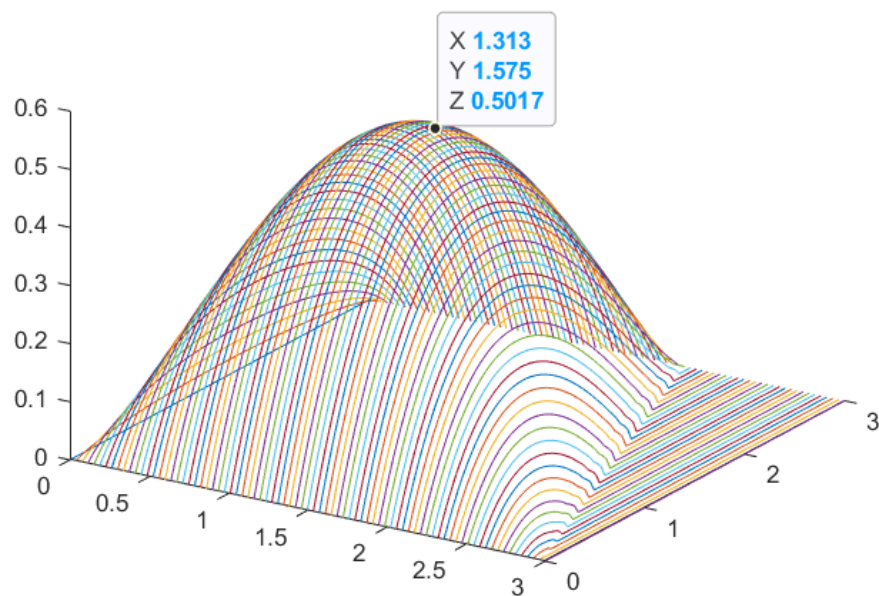


Рис. 3: Итоговый результат на сетке 80 на 80

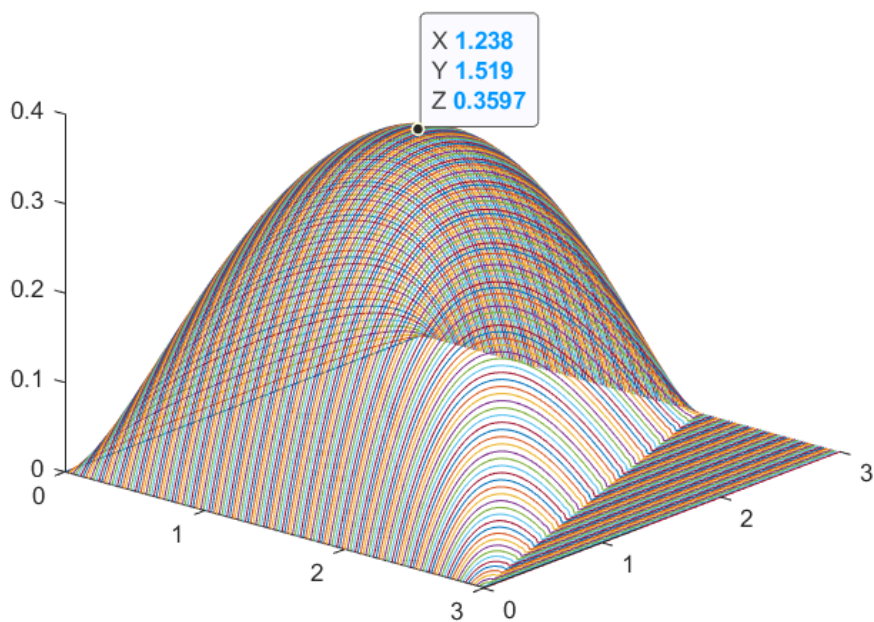


Рис. 4: Итоговый результат на сетке 160 на 160

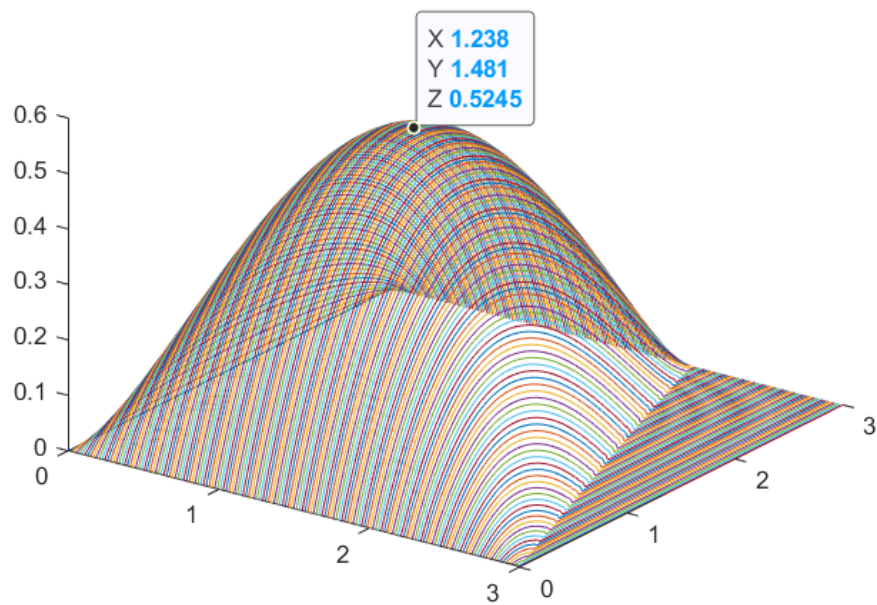


Рис. 5: Итоговый результат на сетке 160 на 160, $\epsilon = 0.01$

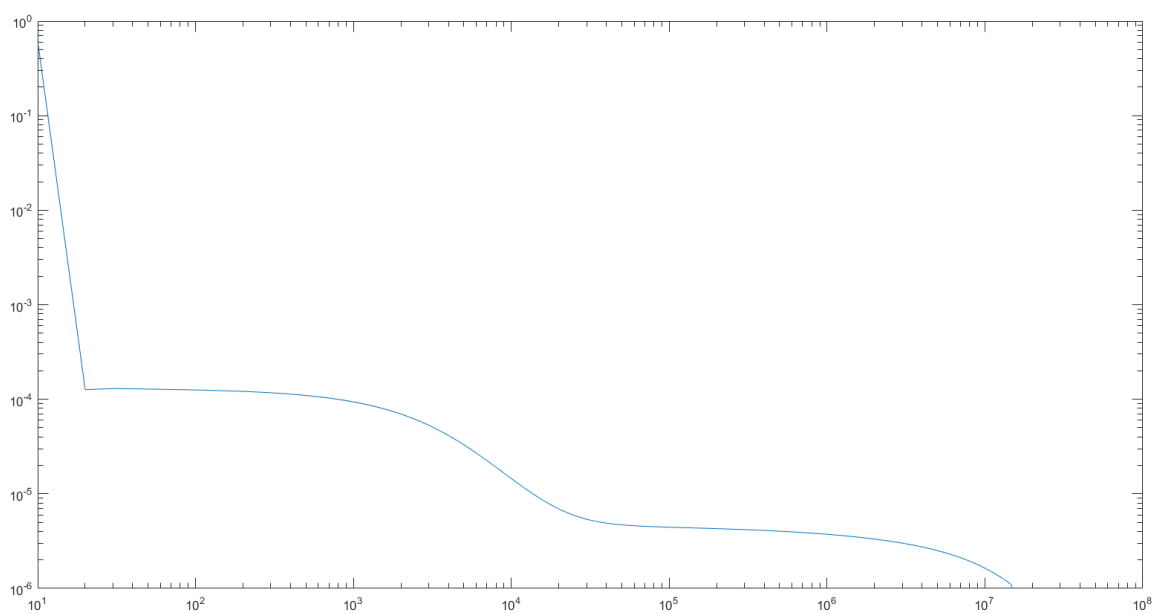


Рис. 6: График погрешности для сетки 160 на 160

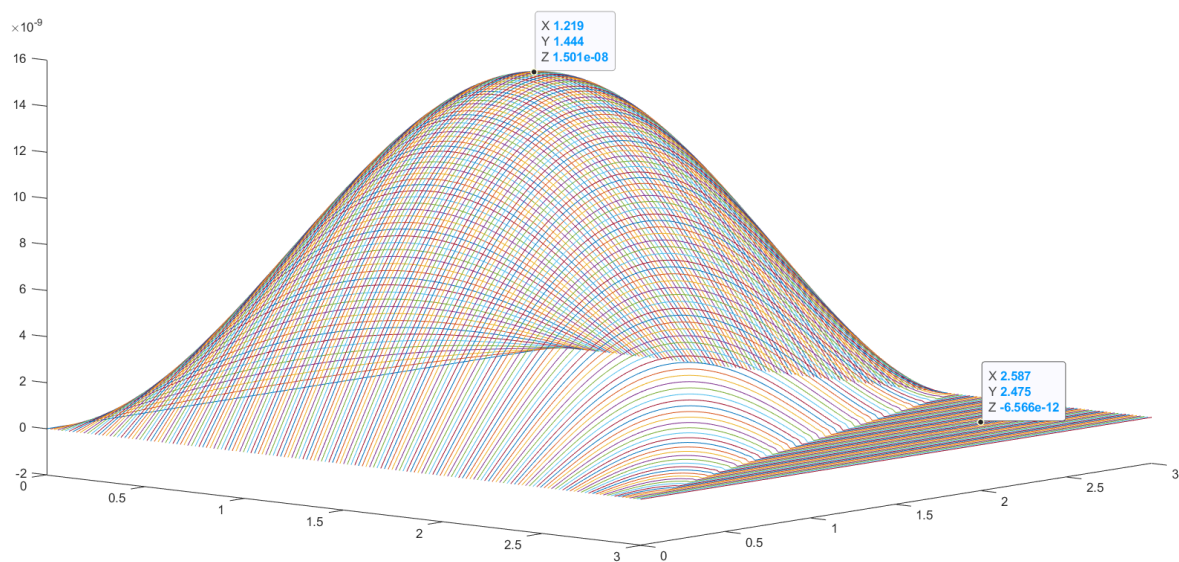


Рис. 7: График ошибки для сетки 160 на 160

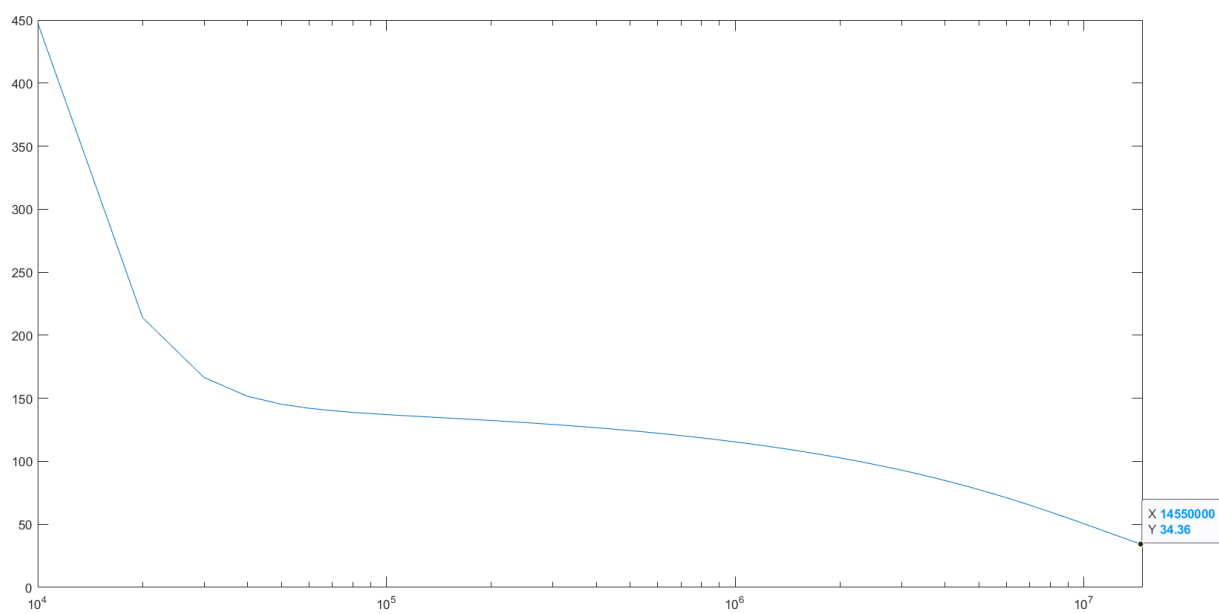


Рис. 8: График нормы невязки для сетки 160 на 160

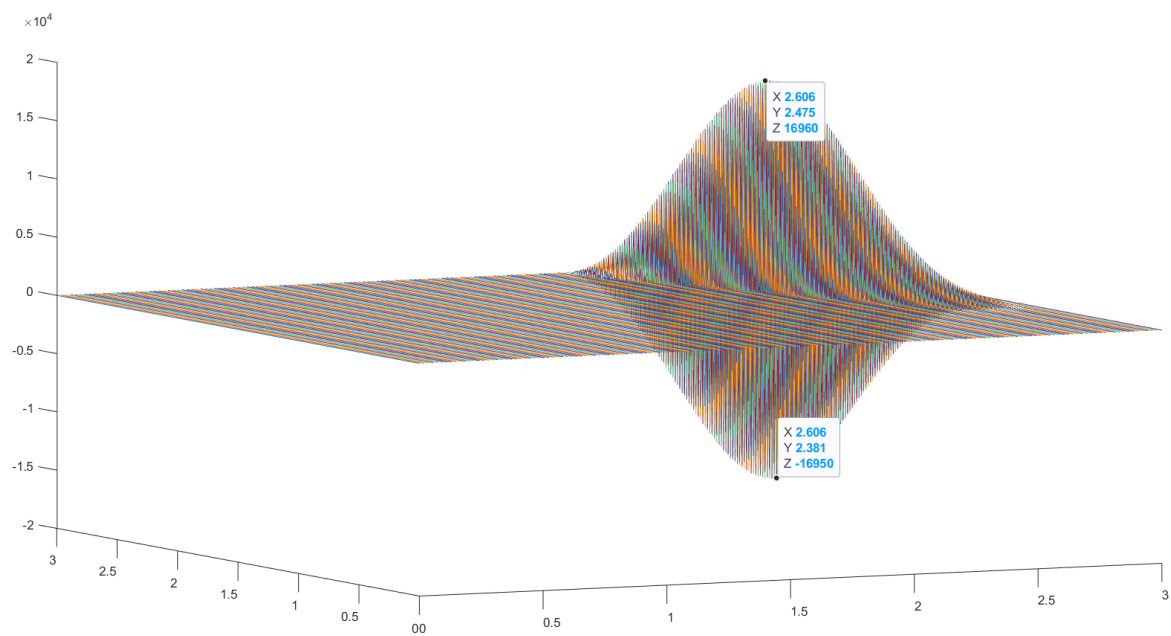


Рис. 9: График Ar для сетки 160 на 160