

Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Вершков Станислав Александрович

Задание по курсу «Суперкомпьютерное  
моделирование и технологии»

Численное решение краевой задачи для  
уравнения Пуассона

ОТЧЕТ

Москва, 2022

# Оглавление

1	Математическая постановка задачи	2
2	Численный метод решения задачи	3
3	Описание MPI программы и гибридной реализации MPI/OpenMP	6
4	Результаты расчетов	7

# 1 Математическая постановка задачи

*Вариант 2.*

Требуется методом конечных разностей приближенно решить краевую задачу для уравнения Пуассона с потенциалом в прямоугольной области.

В прямоугольнике  $\Pi = [0, 4] \times [0, 3]$  рассматривается дифференциальное уравнение Пуассона с потенциалом

$$-\Delta u + q(x, y)u = F(x, y), \quad (1)$$

в котором оператор Лапласа

$$\Delta u = \frac{\partial}{\partial x}(k(x, y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(k(x, y)\frac{\partial u}{\partial y})$$

и функции заданы:

$$k(x, y) = 4 + x + y, \quad q(x, y) = x + y.$$

Для выделения единственного решения уравнение дополняется граничными условиями. На отрезке нижней границы прямоугольника  $\Pi$  задается условие третьего типа:

$$(k\frac{\partial u}{\partial n})(x, y) + \alpha u(x, y) = \psi(x, y),$$

где  $n$  – единичная внешняя нормаль к границе прямоугольника. Заметим, что нормаль  $n$  не определена в угловых точках прямоугольника. Коэффициент  $\alpha$  предлагается взять равным 1. На остальных границах прямоугольника используется граничное условие первого типа:

$$u(x, y) = \phi(x, y).$$

Функции  $F(x, y)$ ,  $\phi(x, y)$ ,  $\psi(x, y)$  получены аналитически.

$$F(x, y) = \frac{(4+x+y)(x^2+y^2)-2(x+y)(4+xy)}{4(4+xy)^{\frac{3}{2}}} + (x+y)\sqrt{4+xy},$$

$$\phi(x, y) = u(x, y),$$

$$\psi(x, y) = \frac{x(4+x+y)}{2\sqrt{4+xy}} + \sqrt{4+xy}.$$

## 2 Численный метод решения задачи

Краевую задачу для уравнения Пуассона с потенциалом 1 предлагается численно решить методом конечных разностей. В расчетной области  $\Pi$  определяется равномерная прямоугольная сетка  $\tilde{\omega}_h = \tilde{\omega}_1 \times \tilde{\omega}_2$ , где

$$\tilde{\omega}_1 = \{x_i = ih_1, i = 0 \dots M\}, \tilde{\omega}_2 = \{y_j = jh_2, j = 0 \dots N\}.$$

Здесь  $h_1 = \frac{4}{M}$ ,  $h_2 = \frac{3}{N}$ . Через  $\omega_h$  обозначим множество внутренних узлов сетки  $\tilde{\omega}_h$ , т.е. множество узлов сетки прямоугольника, не лежащих на границе  $\Gamma$ .

Рассмотрим линейное пространство  $H$  функций, заданных на сетке  $\tilde{\omega}_h$ . Обозначим через  $\omega_{ij}$  значение сеточной функции  $\omega \in H$  в узле сетки. Будем считать, что в пространстве  $H$  задано скалярное произведение и евклидова норма

$$[u, v] = \sum_{i=0}^M h_1 \sum_{j=0}^N h_2 \rho_{ij} u_{ij} v_{ij}, |u| = \sqrt{[u, u]}. \quad (2)$$

Весовая функция  $\rho_{ij} = \rho^{(1)}(x_i) \rho^{(2)}(y_j)$ , где

$$\rho^{(1)}(x_i) = \begin{cases} 1, 1 \leq i \leq M-1 \\ \frac{1}{2}, i=0, i=M \end{cases} \quad \rho^{(2)}(y_j) = \begin{cases} 1, 1 \leq j \leq N-1 \\ \frac{1}{2}, j=0, j=N \end{cases}$$

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

$$A\omega = B, \quad (3)$$

где  $A : H \rightarrow H$  – оператор, действующий в пространстве сеточных функций,  $B \in H$  – известная правая часть. Задача 3 называется разностной схемой. Решение этой задачи считается численным решением исходной задачи.

Уравнение 1 во всех внутренних точках сетки аппроксимируется разностным уравнением

$$-\Delta_h \omega_{ij} + q_{ij} \omega_{ij} = F_{ij}, i = 1 \dots M-1, j = 1 \dots N-1, \quad (4)$$

в котором  $F_{ij} = F(x_i, y_j)$ ,  $q_{ij} = q(x_i, y_j)$ .

Введем обозначения правой и левой разностных производных по переменным  $x, y$  соответственно:

$$\omega_{x,ij} = \frac{\omega_{i+1,j} - \omega_{i,j}}{h_1}, \quad \omega_{\tilde{x},ij} = \frac{\omega_{i,j} - \omega_{i-1,j}}{h_1},$$

$$\omega_{y,ij} = \frac{\omega_{i,j+1} - \omega_{i,j}}{h_2}, \quad \omega_{\tilde{y},ij} = \frac{\omega_{i,j} - \omega_{i,j-1}}{h_2},$$

а также определим сеточные коэффициенты

$$a_{ij} = k(x_i - 0.5h_1, y_j), \quad b_{ij} = k(x_i, y_j - 0.5h_2).$$

С учетом принятых обозначений разностный оператор Лапласа представляется как

$$\Delta_h \omega_{ij} = (a\omega_{\tilde{x}})_{x,ij} + (b\omega_{\tilde{y}})_{y,ij}.$$

Краевые условия первого типа аппроксимируются равенством

$$\omega_{ij} = \phi(x_i, y_j).$$

Краевое условие третьего типа для нижней стороны аппроксимируется следующими равенствами

$$-(2/h_2)(b\omega_{\tilde{y}})_{i1} + (q_{i0} + 2/h_2)\omega_{i0} - (a\omega_{\tilde{x}})_{x,i0} = F_{i0} + (2/h_2)\psi_{i0}, \quad i = 1 \dots M - 1.$$

Краевые условия для угловых точек выбраны следующим образом

$$\omega_{00} = \phi(0, 0), \quad \omega_{M0} = \phi(M, 0), \quad \omega_{0N} = \phi(0, N), \quad \omega_{MN} = \phi(M, N).$$

Приближенное решение системы уравнений 3 для сформулированных выше краевых задач получается с помощью итерационного метода 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, то есть

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

Начальное приближение  $\omega^0$  выбирается равным нулю во всех точках.

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

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

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

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

В качестве условия остановки итерационного процесса выбрано неравенство

$$|w^{(k+1)} - w^{(k)}| < \epsilon,$$

где  $\epsilon$  – положительное число, определяющее точность итерационного метода.

### 3 Описание MPI программы и гибридной реализации MPI/OpenMP

Для решения задачи с использованием технологии MPI рассматриваемая область разбивается на подобласти прямоугольной формы, причем число подобластей равно числу процессов. В каждой из них отношение  $\theta$  количества узлов по ширине и длине удовлетворяет неравенствам  $0.5 \leq \theta \leq 2$ . Используется вызов функции *MPI\_Cart\_create*, которая возвращает новый коммуникатор. Далее, каждый процесс, используя функции *MPI\_Cart\_coords* и *MPI\_Cart\_shift* получает свое местоположение в сетке процессов, ранги соседей и рассчитывает диапазон точек матрицы, которые ему необходимо рассчитать.

В процессе работы алгоритма процессам необходимо знать значения граничных областей, рассчитываемых другими процессами. Для этого процесса обмениваются граничными областями с использованием функции *MPI\_Sendrecv*.

Для расчета скалярного произведения необходимо вычислять сумму по всей области. Каждый процесс рассчитывает локальную сумму в своей области, а затем используется операция *MPI\_Allreduce* с функцией агрегации *MPI\_SUM* для того, чтобы все процессы получили общую сумму. Аналогичная операция используется и при расчете условия остановки итерационного процесса, однако там используется агрегации *MPI\_MAX*. В программах в качестве точности выбрано  $\epsilon = 10^{-6}$ .

В гибридной MPI/OpenMP программе дополнительно с помощью технологии OpenMP производилось распараллеливание циклов. Для этого использовались следующие директивы: *#pragma omp parallel for default(shared) private(li, gi, lj, gj) schedule(dynamic)*, которая изменялась в зависимости от существования приватных переменных в цикле. В случае, когда вычисляется сумма в скалярном произведении используется директива: *#pragma omp parallel for default(shared) private(li, gi, lj, gj) schedule(dynamic) reduction(+:local\_sum)*.

## 4 Результаты расчетов

Ускорение рассчитывается по формуле  $S = \frac{Time_{onseq}}{Time_{onparallel}}$ .

Были выбраны размеры сетки  $(M, N) = (160, 160)$  и  $(M, N) = (160, 320)$  в силу того, что мои программы получают time-out на ПВС IBM Polus при увеличении размера сетки.

Последовательная программа на ПВС IBM Polus отработала на сетках  $160 \times 160$  и  $160 \times 320$  соответственно 204.586 (с) и 852.287 (с).



Число точек сетки	Число MPI-процессов	Время (с)	Ускорение $S$
$160 \times 160$	4	51.160	3.916
	8	29.768	6.872
	16	18.575	11.014
	32	14.919	13.713
$160 \times 320$	4	229.417	3.715
	8	124.321	6.855
	16	82.307	10.354
	32	52.391	16.267

**Таблица 1.** Таблица с результатами расчетов на ПВС IBM Polus (MPI код).

Всюду в таблице 2 количество OMP-нитей в процессе равно четырем.

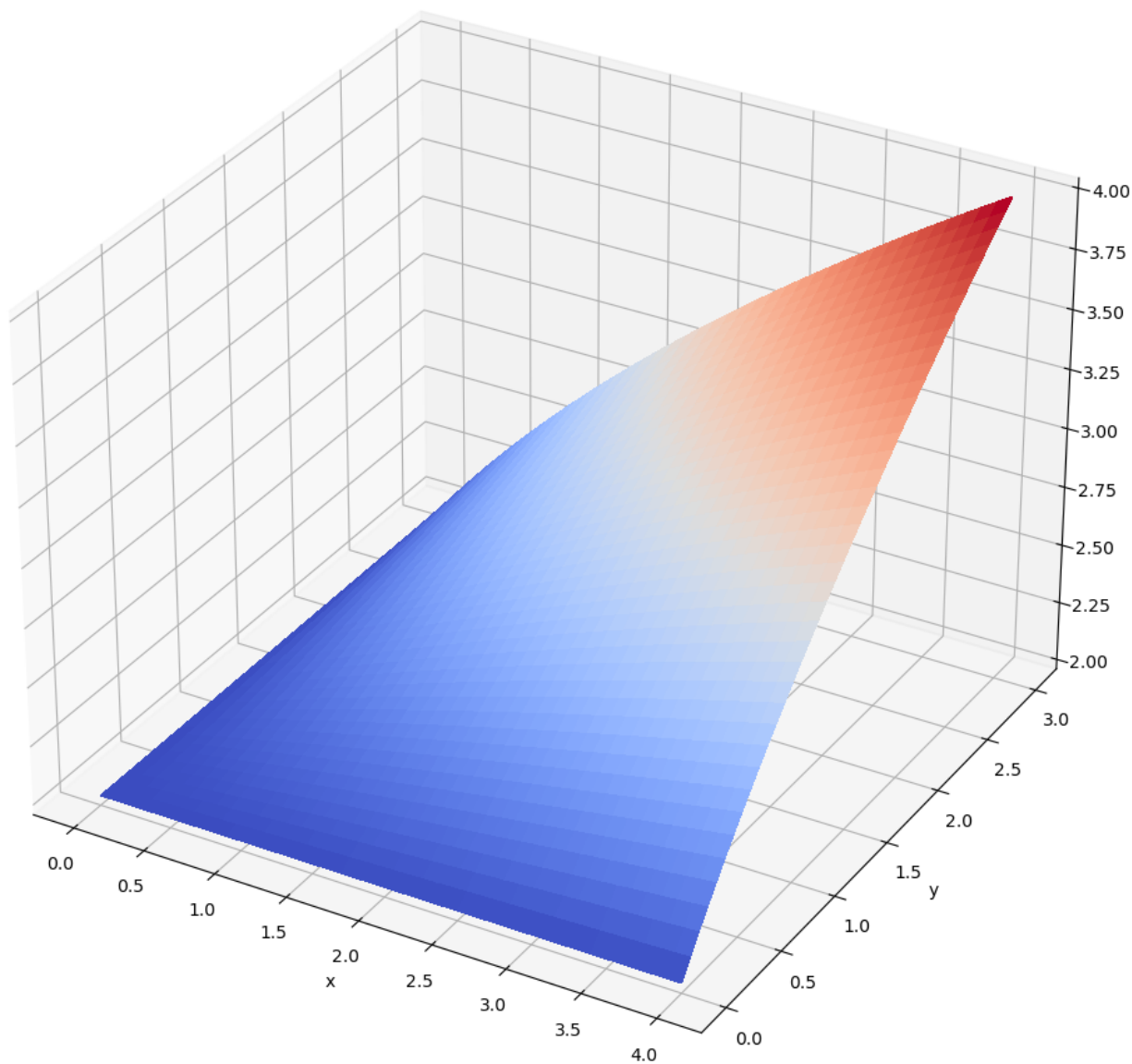
Число точек сетки	Число MPI-процессов	Время (с)	Ускорение $S$
$160 \times 160$	1	54.497	3.754
	2	34.535	5.924
	4	21.126	9.684
	8	17.216	11.878
$160 \times 320$	1	220.571	3.864
	2	149.629	5.696
	4	65.849	12.943
	8	49.425	17.244

**Таблица 2.** Таблица с результатами расчетов на ПВС IBM Polus (MPI+OpenMP код).

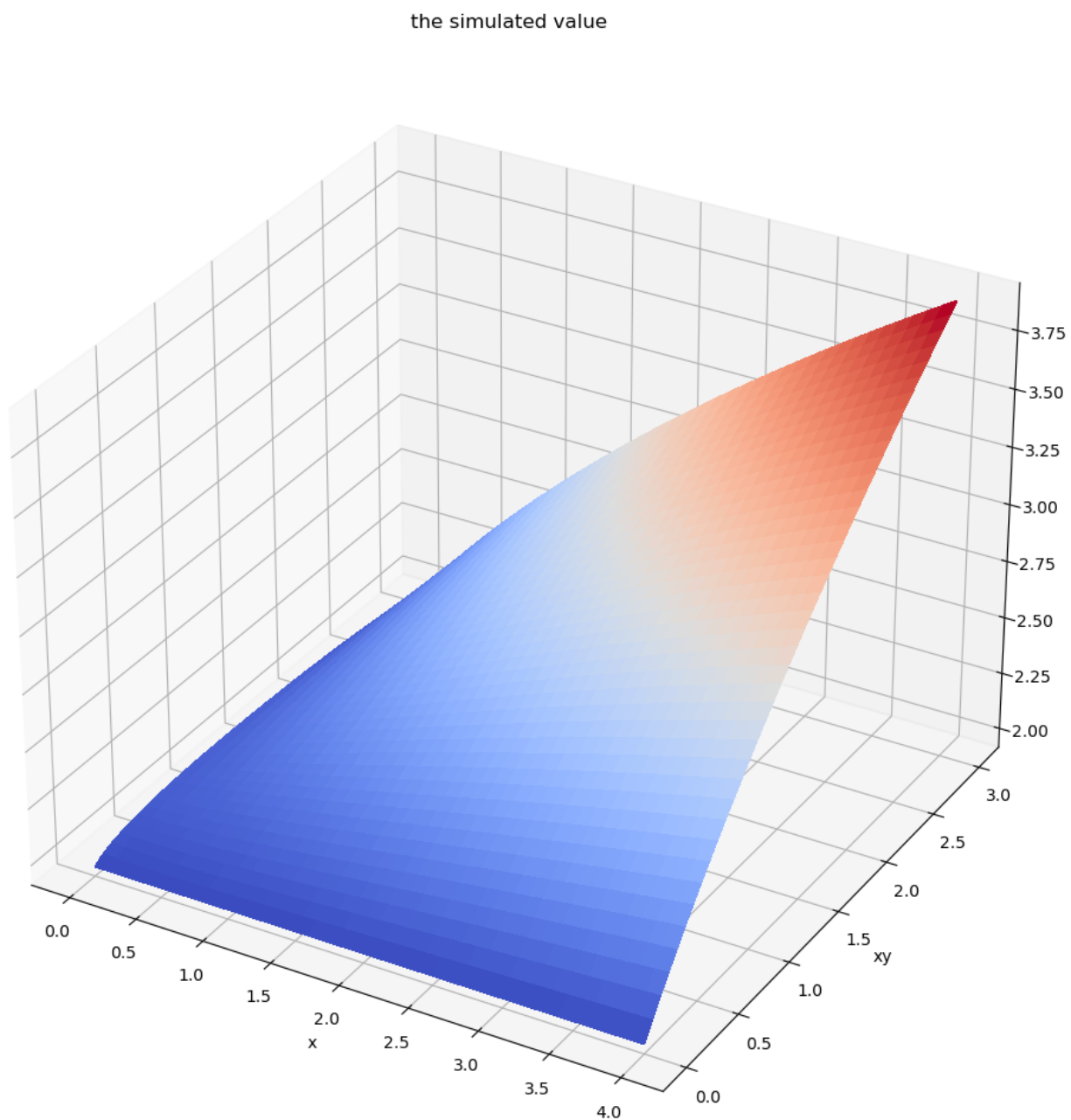
Погрешность приближенного решения на сетке  $160 \times 160$   $\delta = 0.047324$ .  
Погрешность приближенного решения на сетке  $160 \times 320$   $\delta = 0.131496$ .

Выбраны графики решений на сетке  $160 \times 160$ , так как они уже визуально достаточно точны и неотличимы.

the real value



**Рисунок 1.** График точного решения на сетке  $160 \times 160$ .



**Рисунок 2.** График приближенного решения на сетке  $160 \times 160$ .