



Московский государственный университет имени М.В. Ломоносова

Факультет Вычислительной математики и кибернетики

Кафедра Системного программирования

**Здание по курсу**  
**«Суперкомпьютерное моделирование и технологии»**  
**Численный метод решения дифференциальной задачи**  
**Вариант 1**

**Выполнил:**

студент 627 группы

Герасимов Денис Юрьевич

Москва, 2022

## Математическая постановка дифференциальной задачи

В трехмерной замкнутой области

$$\Omega = [0 \leq x \leq L_x] \times [0 \leq y \leq L_y] \times [0 \leq z \leq L_z]$$

для  $(0 < t \leq T]$  требуется найти решение  $u(x, y, z, t)$  уравнения в частных производных

$$\frac{\partial^2 u}{\partial t^2} = \Delta u \quad (1)$$

с начальными условиями

$$u|_{t=0} = \varphi(x, y, z), \quad (2)$$

$$\frac{\partial u}{\partial t} \Big|_{t=0} = 0, \quad (3)$$

при условии, что на границах области заданы однородные граничные условия первого рода

$$u(0, y, z, t) = 0, \quad u(L_x, y, z, t) = 0, \quad (4)$$

$$u(x, 0, z, t) = 0, \quad u(x, L_y, z, t) = 0, \quad (5)$$

$$u(x, y, 0, t) = 0, \quad u(x, y, L_z, t) = 0, \quad (6)$$

## Численный метод решения задачи

Для численного решения задачи введем на  $\Omega$  сетку  $\omega_{h\tau} = \bar{\omega}_h \times \omega_\tau$ , где

$$T = T_0,$$

$$L_x = L_{x_0}, L_y = L_{y_0}, L_z = L_{z_0}$$

$$\bar{\omega}_h = \{(x_i = ih_x, y_j = jh_y, z_k = kh_z), i, j, k = 0, 1, \dots, N, h_x N = L_x, h_y N = L_y, h_z N = L_z\},$$

$$\omega_\tau = \{t_n = n\tau, n = 0, 1, \dots, K, \tau K = T\}.$$

Через  $\omega_h$  обозначим множество внутренних, а через  $\gamma_h$  — множество граничных узлов сетки  $\bar{\omega}_h$ .

Для аппроксимации исходного уравнения (1) с однородными граничными условиями (4)–(6) и начальными условиями (2)–(3) воспользуемся следующей системой уравнений:

$$\frac{u_{ijk}^{n+1} - 2u_{ijk}^n + u_{ijk}^{n-1}}{\tau^2} = \Delta_h u^n, \quad (x_i, y_j, z_k) \in \omega_h, \quad n = 1, 2, \dots, K-1,$$

Здесь  $\Delta_h$  — семиточечный разностный аналог оператора Лапласа:

$$\Delta_h u^n = \frac{u_{i-1,j,k}^n - 2u_{i,j,k}^n + u_{i+1,j,k}^n}{h^2} + \frac{u_{i,j-1,k}^n - 2u_{i,j,k}^n + u_{i,j+1,k}^n}{h^2} + \frac{u_{i,j,k-1}^n - 2u_{i,j,k}^n + u_{i,j,k+1}^n}{h^2}.$$

Приведенная выше разностная схема является явной — значения  $u_{ijk}^{n+1}$  на  $(n+1)$ -м шаге можно явным образом выразить через значения на предыдущих слоях.

Для начала счета (т.е. для нахождения  $u_{ijk}^2$ ) должны быть заданы значения  $u_{ijk}^0, u_{ijk}^1, (x_i, y_j, z_k) \in \omega_h$ . Из условия (2) имеем

$$u_{ijk}^0 = \varphi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \omega_h. \quad (10)$$

Простейшая замена начального условия (3) уравнением  $(u_{ijk}^1 - u_{ijk}^0)/\tau = 0$  имеет лишь первый порядок аппроксимации по  $\tau$ . Аппроксимацию второго порядка по  $\tau$  и  $h$  дает разностное уравнение

$$\frac{u_{ijk}^1 - u_{ijk}^0}{\tau} = \frac{\tau}{2} \Delta_h \varphi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \omega_h. \quad (11)$$

$$u_{ijk}^1 = u_{ijk}^0 + \frac{\tau^2}{2} \Delta_h \varphi(x_i, y_j, z_k) \quad (12)$$

Для вычисления значений  $u^0, u^1 \in \gamma_h$  допускается использование аналитического значения  $u$ , которое задается в программе еще для вычисления погрешности решения задачи.

## Краткое описание проделанной работы по созданию гибридной реализации MPI/OpenMP

Иницилируем разбиение сетки, вычисляем всю необходимую информацию о локальных данных и размерах локальной сетки и информацию о соседях.

Каждый процесс выполняет описанный выше алгоритм численного решения уравнения в своей области, при этом пересылки сообщений между процессами выполняются средствами MPI в следующих случаях:

1. Обмен граничными областями между соседними процессами. Данный тип обмена необходим для вычисления оператора Лапласа внутри всех узлов решетки  $\omega_h$ , принадлежащей процессу.
2. Вычисление метрик для оценки качества работы программы. Директивы OpenMP использовались в программе для распараллеливания циклов, так как значения на различных итерациях внутри циклов вычисляются независимо.

## Необходимые графики

Сетка при  $N = 32^5$ , 20 шагов по времени  $\tau = 0,005$

График аналитической функции

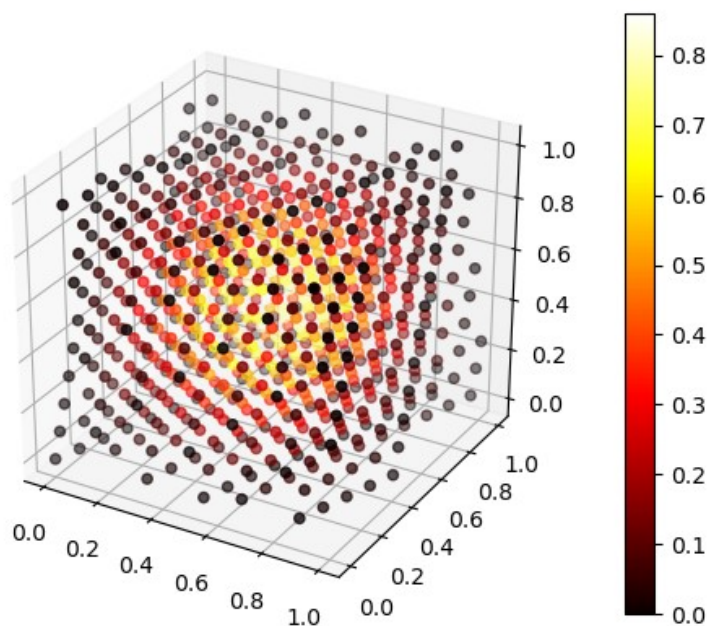
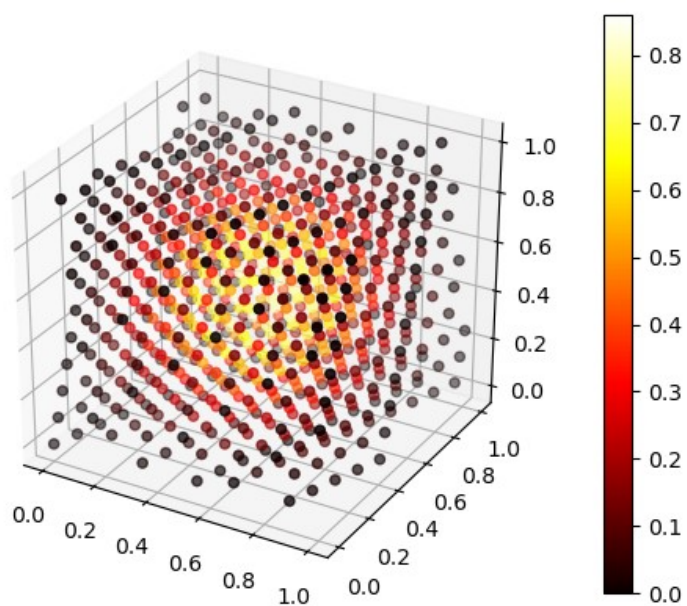
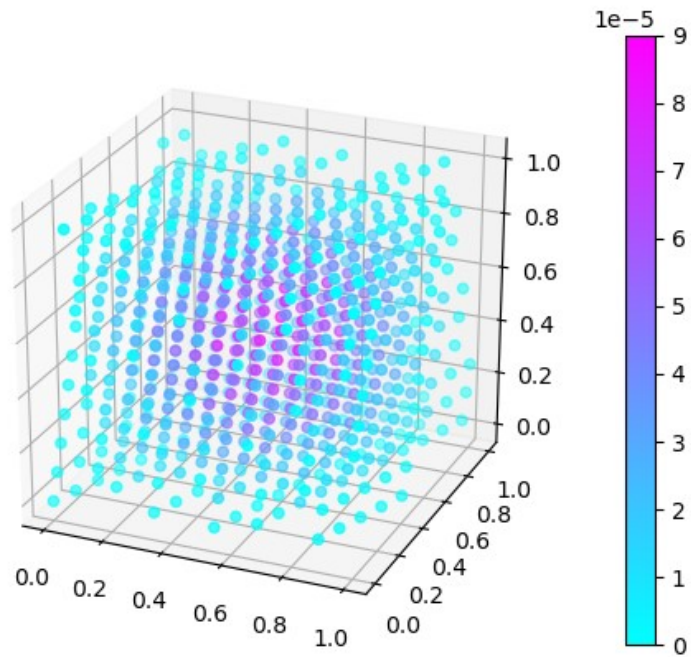


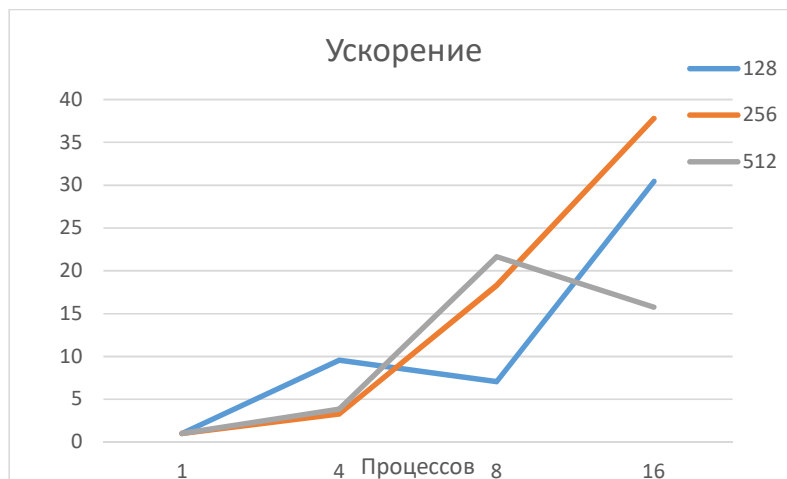
График посчитанной функции

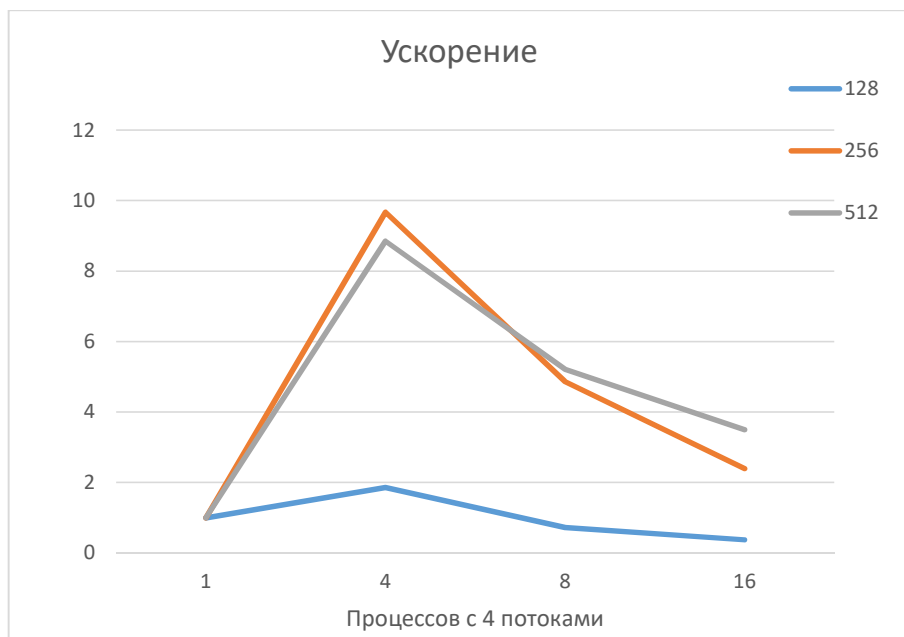


**График разности в каждой точке посчитанной на последнем слое времени**



## **Результаты расчётов**





### Таблички

Procs	N	Time	SpeedUp	Error
1	128	9,08333	1	0,000001
4	128	0,93984	9,664762	0,000001
8	128	1,21193	7,49493	0,000001
16	128	0,22744	39,93726	0,000001
1	256	67,61476	1	0
4	256	7,35779	9,189547	0
8	256	9,60725	7,037889	0
16	256	1,69976	39,779	0
1	512	604,2974	1	0
4	512	157,8204	3,829019	0
8	512	27,52476	21,95468	0
16	512	30,8028	19,61826	0

L = 1

Procs	N	Time	SpeedUp	Error
1	128	8,95314	1	1,00E-07
4	128	0,93331	9,59289	1,00E-07
8	128	1,26836	7,058832	1,00E-07
16	128	0,29402	30,45079	1,00E-07
1	256	65,58322	1	0
4	256	19,99085	3,280662	0
8	256	3,58864	18,27523	0
16	256	1,73574	37,78401	0
1	512	599,2628	1	0
4	512	155,5737	3,851955	0
8	512	27,64557	21,67663	0
16	512	38,0433	15,75212	0

$L = \pi$

Procs	Thread	N	Time	SpeedUp
1	4	128	6,81098	1
4	4	128	3,66111	1,860359
8	4	128	9,4583	0,720106
16	4	128	18,17719	0,374699
1	4	256	50,89094	1
4	4	256	5,26018	9,674753
8	4	256	10,46341	4,863705
16	4	256	21,21117	2,399252
1	4	512	326,5062	1
4	4	512	36,89769	8,848959
8	4	512	62,61475	5,214525
16	4	512	93,28646	3,500038

$L = 1$

Procs	Thread	N	Time	SpeedUp
1	4	128	6,88289	1
4	4	128	3,46815	1,9846
8	4	128	9,2508	0,744032
16	4	128	18,6167	0,369716
1	4	256	54,15837	1
4	4	256	4,88128	11,09512
8	4	256	13,12916	4,125045
16	4	256	21,1317	2,562897
1	4	512	301,5025	1
4	4	512	125,154	2,409052
8	4	512	58,99885	5,110311
16	4	512	172,5705	1,747126

$L = \pi$

## **Выводы**

Судя по таблице время работы программы во многом зависит от погоды, давления и расположения звёзд.

Из адекватных соображений: при увеличении числа процессов можно наблюдать сверхлинейное ускорение, которое скорее всего связано с более оптимальной работой с памятью и локальностью распределенных данных. При этом при дальнейшем увеличении можно видеть и обратный эффект, когда кластер захлёбывается коммуникациями между процессами.

При включении OpenMP зачастую при большом числе процессов и потоков связанных с ним мы видим значительное ухудшение, которое объясняется скорее неоптимальным распределением процессов и потоков по NUMA узлам кластера.

Кроме того в силу загруженности кластера замеры следовало бы проводить СОТНИ раз для установления адекватных цифр. Что, к сожалению, невозможно и растратно.

Комментарий по вычисляемой ошибке:

При большом размере сетки, действительно, ошибка была близкой к нулю (при форматном выводе имеем 0.0000000)