



Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра автоматизации систем вычислительных комплексов

**Отчёт по заданию №3 в рамках курса  
«Суперкомпьютерное моделирование и технологии»**

Выполнил: студент 621-ой группы  
Рябченков Владимир Михайлович  
Вариант: 5

Москва 2022

# 1 Описание задачи

Данная работа посвящена решению трехмерного гиперболического уравнения в области, представляющей из себя прямоугольный параллелепипед с помощью численного метода.

Для решения задачи необходимо разработать программу на системе IBM Polus с использованием технологий MPI и OpenMP.

## 2 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

$$\frac{\partial^2 u}{\partial t^2} = \Delta u \quad (2.1)$$

с начальными условиями

$$u|_{t=0} = \phi(x, y, z), \quad (2.2)$$

$$\left. \frac{\partial u}{\partial t} \right|_{t=0} = 0, \quad (2.3)$$

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

$$u(0, y, z, t) = 0, \quad u(L_x, y, z, t) = 0, \quad (2.4)$$

$$u(x, 0, z, t) = 0, \quad u(x, L_y, z, t) = 0, \quad (2.5)$$

$$u(x, y, 0, t) = 0, \quad u(x, y, L_z, t) = 0, \quad (2.6)$$

либо периодические граничные условия

$$u(0, y, z, t) = u(L_x, y, z, t), \quad u_x(0, y, z, t) = u_x(L_x, y, z, t), \quad (2.7)$$

$$u(x, 0, z, t) = u(x, L_y, z, t), \quad u_y(x, 0, z, t) = u_y(x, L_y, z, t), \quad (2.8)$$

$$u(x, y, 0, t) = u(x, y, L_z, t), \quad u_z(x, y, 0, t) = u_z(x, y, L_z, t), \quad (2.9)$$

В данной работе рассматривается вариант №5 с периодическими граничными условиями по переменной «x» и однородными условиями по переменным «y» и «z»

### 3 Численный метод решения задачи

Для численного решения задачи на  $\Omega$  введена сетка  $w_{h\tau} = \bar{w}_h \times w_\tau$ , где

$$T = T_0,$$

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

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

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

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

Для аппроксимации исходного уравнения 2.1 с однородными граничными условиями 2.4–2.6 и начальными условиями 2.2–2.3 воспользуемся следующей системой уравнений:

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

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  $u_{ijk}^0$ ,  $u_{ijk}^1$ ,  $(x_i, y_j, z_k) \in w_h$ . Из условия 2.2 имеем

$$u_{ijk}^0 = \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in w_h. \quad (3.1)$$

Простейшая замена начального условия 2.3 уравнением  $\frac{u_{ijk}^1 - u_{ijk}^0}{\tau} = 0$  имеет лишь первый порядок аппроксимации по  $\tau$ . Аппроксимацию второго порядка по  $\tau$  и  $h$  дает разностное уравнение

$$\frac{u_{ijk}^1 - u_{ijk}^0}{\tau} = \frac{\tau}{2} \Delta_h \phi(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in w_h. \quad (3.2)$$

$$u_{ijk}^1 = u_{ijk}^0 + \frac{\tau^2}{2} \Delta_h \phi(x_i, y_j, z_k) \quad (3.3)$$

Разностная аппроксимация для периодических граничных условий выглядит следующим образом

$$u_{0jk}^{n+1} = u_{Njk}^{n+1}, \quad u_{1jk}^{n+1} = u_{N+1jk}^{n+1},$$

$$u_{i0k}^{n+1} = u_{iNk}^{n+1}, \quad u_{i1k}^{n+1} = u_{iN+1k}^{n+1},$$

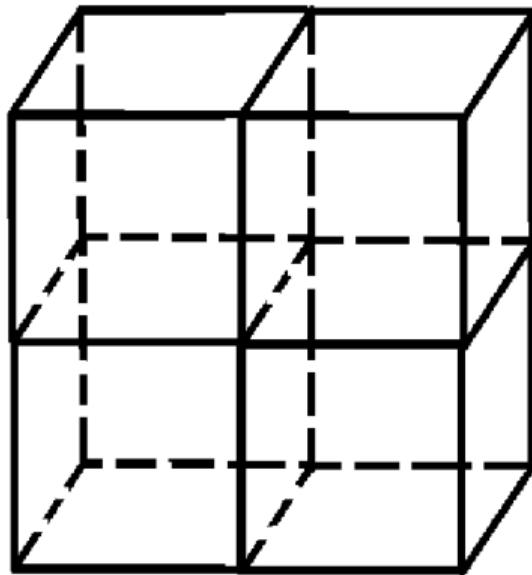
$$u_{ij0}^{n+1} = u_{ijN}^{n+1}, \quad u_{ij1}^{n+1} = u_{ijN+1}^{n+1},$$

$$i, j, k = 0, 1, \dots, N.$$

## 4 Программная реализация

В рамках данной работы для решения поставленной задачи 3 была реализована параллельная программа в двух версиях: MPI и MPI + OpenMP (программа доступна по ссылке <https://github.com/vultar150/parallel-dz3.git>).

Реализовано блочное разбиение области  $\Omega$  между процессами (рис. 1). Разбиение



*Рис. 1. Блоchное разбиение 3d*

выполнялось автоматически, с помощью MPI функции MPI\_Dims\_create.

Программа принимает на вход значения  $L_x, L_y, L_z$  и  $N_x, N_y, N_z$ . После каждой итерации (кроме 0-ой итерации) процессы обмениваются граничными плоскостями со своими соседями. Далее происходит вычисление ошибки для заданной итерации. Ошибка вычисляется как максимальное значение разности по модулю между значениями в сетке, вычисленными предложенным методом, и значениями в сетке аналитической функции.

В соответствии с вариантом задания, аналитическая функция (математическое решение данной задачи) имеет следующий вид:

$$\sin\left(\frac{2\pi}{L_x}x\right) \cdot \sin\left(\frac{\pi}{L_y}y\right) \cdot \sin\left(\frac{\pi}{L_z}z\right) \cdot \cos(a_t \cdot t + 2\pi), \quad a_t = \pi \sqrt{\frac{4}{L_x^2} + \frac{1}{L_y^2} + \frac{1}{L_z^2}}$$

Время работы программы измеряется следующим образом. Каждый MPI-процесс измеряет своё время выполнения, затем среди полученных значений берётся максимум.

Исследования проводились для следующих значений входных параметров:  $L_x = L_y = L_z = L = 1, \pi$ ;  $N_x = N_y = N_z = N = 128, 256, 512$ . Значения  $T = 1.0$ ,  $K = 1000$ ,  $\tau = \frac{T}{K}$ . Для каждого запуска программы выполнялось 80 итераций ( $n = 80$ ). В следующем разделе приведены результаты экспериментов.

## 5 Результаты

Результаты запусков программы на системе Polus, а именно: таблицы результатов расчётов, графики зависимостей времени и ускорения от кол-ва MPI процессов, 4D графики значений аналитической функции (в точках сетки), 4D графики значений сечётной функции (вычисленных итерационном методом), 4D графики значений ошибки в точках сетки — представлены в таблицах 1–4 и на рисунках 2–17 для MPI и MPI+OpenMP версий программы для различных значений параметров  $L$  и  $N$ .

*Таблица 1. Результаты расчётов на Polus при  $L = 1$  (MPI код)*

Число MPI процессов $N_p$	Число точек сетки $N^3$	Время решения Т(с)	Ускорение S	Погрешность $\delta$
1	128 <sup>3</sup>	5.02	1.0	2.6e-5
4		1.33	3.77	2.6e-5
8		0.88	5.7	2.6e-5
16		0.62	8.1	2.6e-5
32		0.47	10.68	2.6e-5
1		37.86	1.0	5.8e-6
4	256 <sup>3</sup>	10.07	3.76	5.8e-6
8		6.26	6.05	5.8e-6
16		4.2	9.01	5.8e-6
32		2.85	13.28	5.8e-6
1		307.7	1.0	8e-7
4	512 <sup>3</sup>	98.03	3.14	8e-7
8		45.83	6.71	8e-7
16		28.49	10.8	8e-7
32		18.77	16.39	8e-7

*Таблица 2. Результаты расчётов на Polus при  $L = \pi$  (MPI код)*

Число MPI процессов $N_p$	Число точек сетки $N^3$	Время решения Т(с)	Ускорение S	Погрешность $\delta$
1	128 <sup>3</sup>	4.84	1.0	2.9e-6
4		1.79	2.7	2.9e-6
8		0.98	4.94	2.9e-6
16		0.67	7.22	2.9e-6
32		0.44	11	2.9e-6
1		38.37	1.0	7.1e-7
4	256 <sup>3</sup>	12.1	3.17	7.1e-7
8		6.01	6.38	7.1e-7
16		3.8	10.1	7.1e-7
32		2.8	13.7	7.1e-7
1		306.68	1.0	1.7e-7
4	512 <sup>3</sup>	100.64	3.05	1.7e-7
8		45.22	6.78	1.7e-7
16		30.18	10.16	1.7e-7
32		19.4	15.81	1.7e-7

Таблица 3. Результаты расчётов на Polus при  $L = 1$  ( $MPI + OpenMP$  код)

Число MPI процессов $N_p$	Число OpenMP нитей в процессе	Число точек сетки $N^3$	Время решения T(с)	Ускорение S	Погрешность $\delta$
1	1	$128^3$	4.64	1.0	2.6e-5
1	4		1.13	4.1	2.6e-5
2			0.73	6.36	2.6e-5
4			0.32	14.5	2.6e-5
8			0.21	22.01	2.6e-5
1	1		35.33	1.0	5.8e-6
1	4	$256^3$	8.9	3.97	5.8e-6
2			4.56	7.75	5.8e-6
4			2.34	15.1	5.8e-6
8			1.32	26.77	5.8e-6
1	1		295.87	1.0	8e-7
1	4	$512^3$	74.36	3.98	8e-7
2			36.3	8.15	8e-7
4			18.29	16.18	8e-7
8			13	22.76	8e-7

Таблица 4. Результаты расчётов на Polus при  $L = \pi$  ( $MPI + OpenMP$  код)

Число MPI процессов $N_p$	Число OpenMP нитей в процессе	Число точек сетки $N^3$	Время решения T(с)	Ускорение S	Погрешность $\delta$
1	1	$128^3$	4.4	1.0	2.9e-6
1	4		1.13	3.89	2.9e-6
2			0.58	7.58	2.9e-6
4			0.32	13.75	2.9e-6
8			0.2	22	2.9e-6
1	1		33.32	1.0	7.1e-7
1	4	$256^3$	8.83	3.77	7.1e-7
2			4.63	7.2	7.1e-7
4			2.3	14.49	7.1e-7
8			1.32	25.24	7.1e-7
1	1		280.4	1.0	1.7e-7
1	4	$512^3$	70.58	3.97	1.7e-7
2			36.51	7.68	1.7e-7
4			18.29	15.33	1.7e-7
8			10.54	26.6	1.7e-7

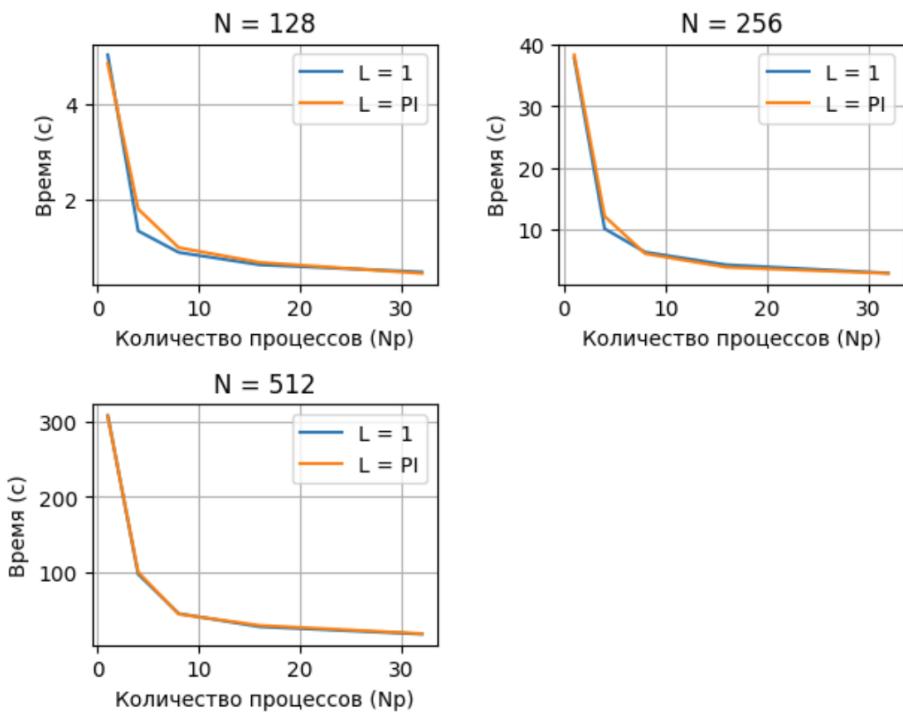


Рис. 2. Зависимость времени от кол-ва процессов, MPI версия

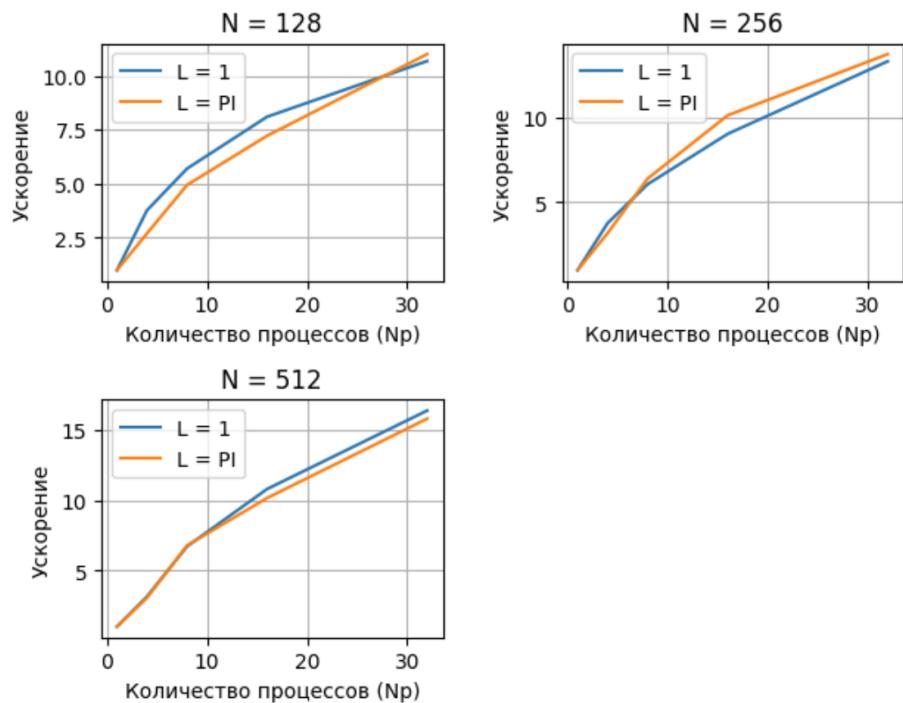


Рис. 3. Зависимость ускорения от кол-ва процессов, MPI версия

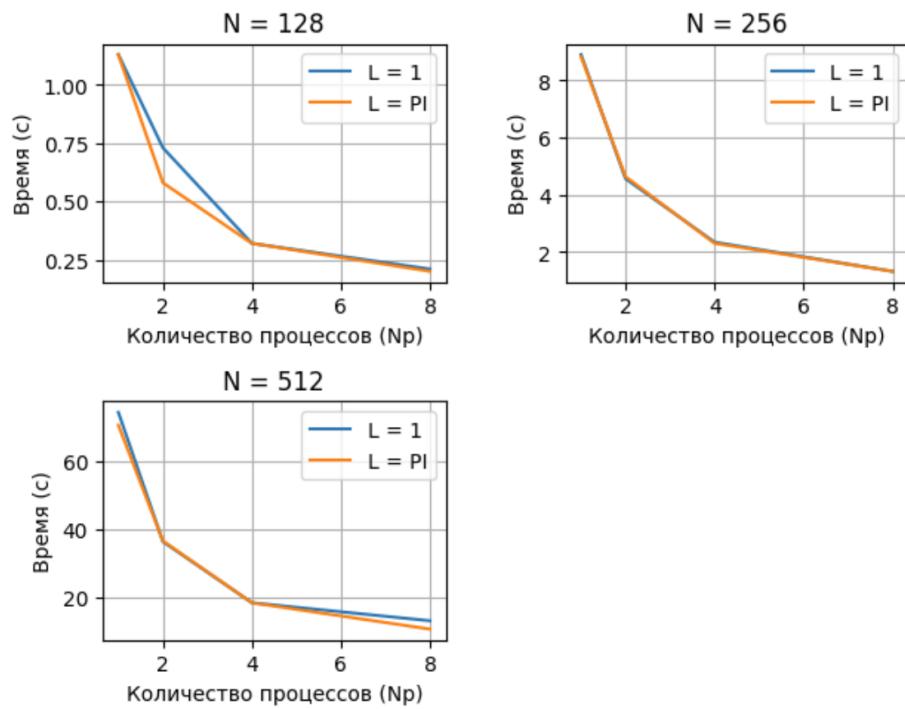


Рис. 4. Зависимость времени от кол-ва процессов, MPI+OpenMP версия (4 потока)

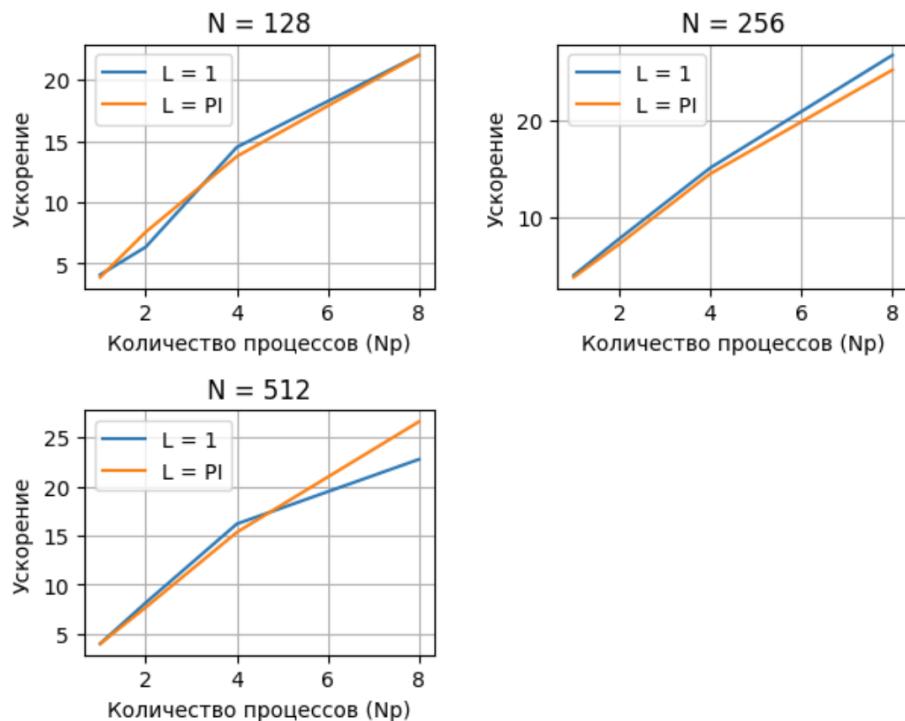


Рис. 5. Зависимость ускорения от кол-ва процессов, MPI+OpenMP версия (4 потока)

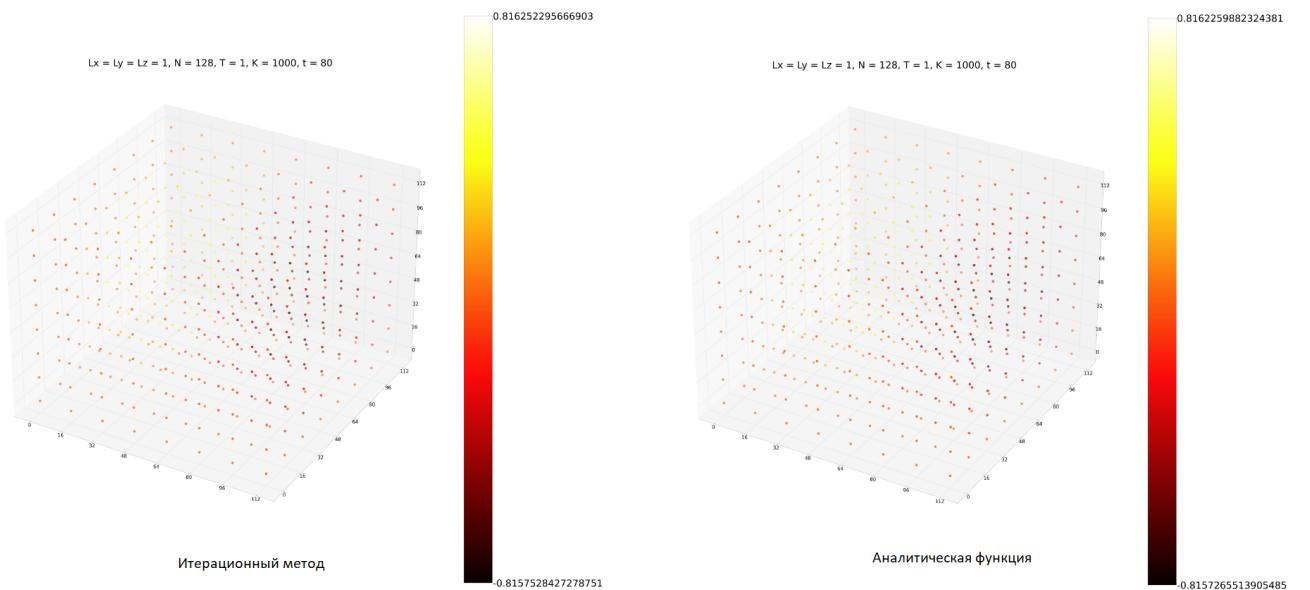


Рис. 6. Значения в сетке при  $L = 1$ ,  $N = 128$  (с шагом 16)

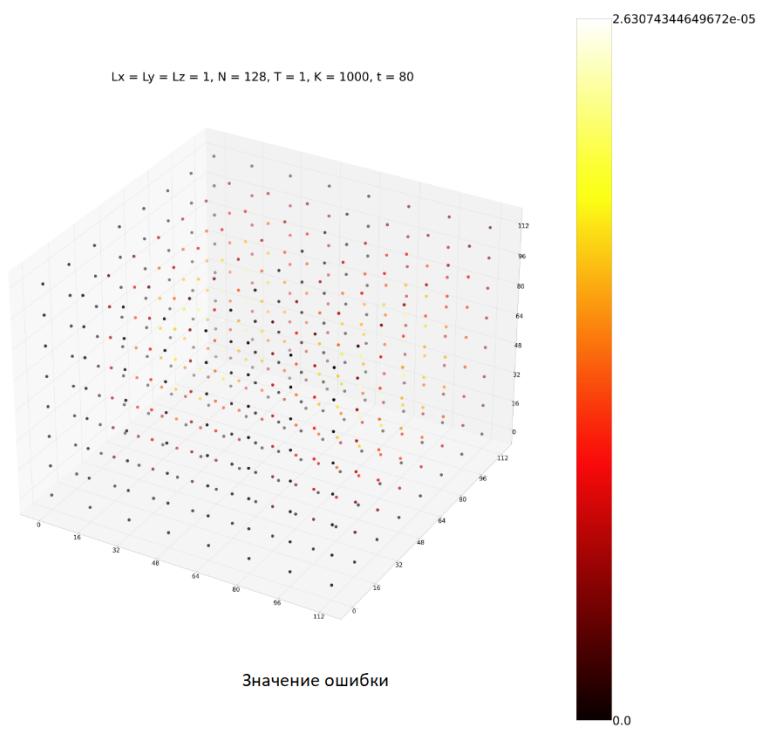


Рис. 7. Значения ошибки при  $L = 1$ ,  $N = 128$  (с шагом 16)

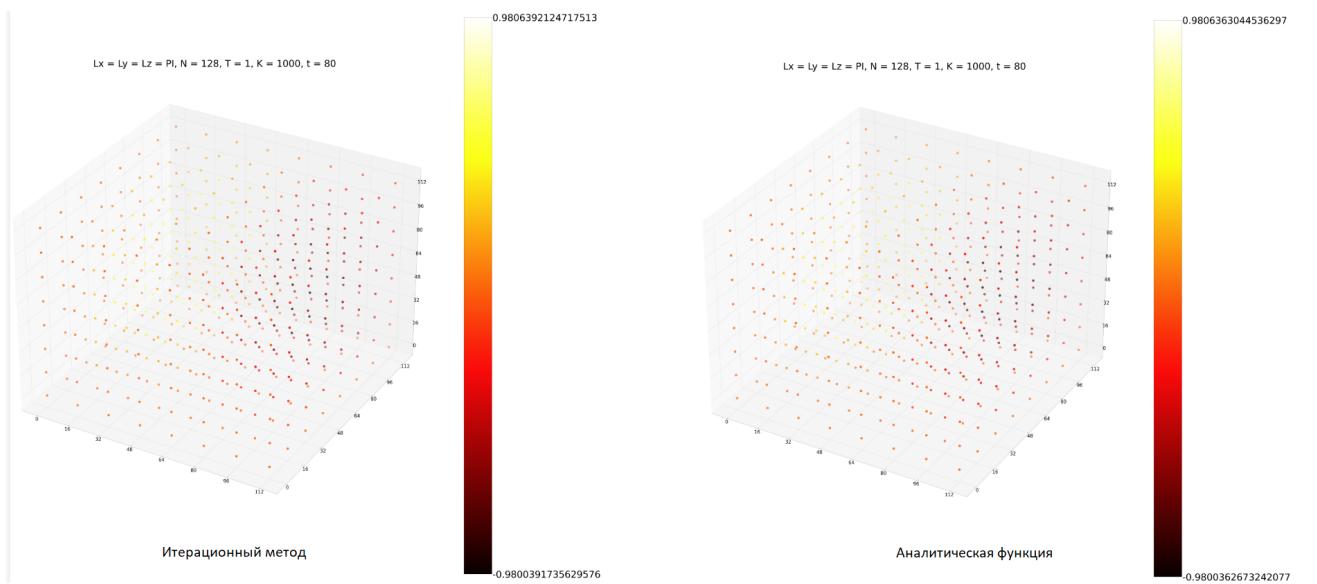


Рис. 8. Значения в сетке при  $L = \pi$ ,  $N = 128$  (с шагом 16)

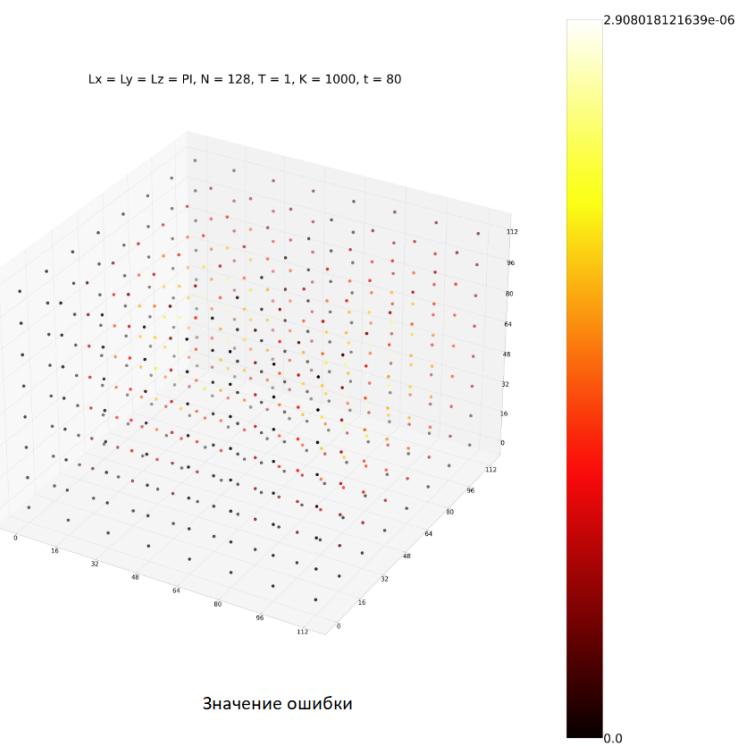


Рис. 9. Значения ошибки при  $L = \pi$ ,  $N = 128$  (с шагом 16)

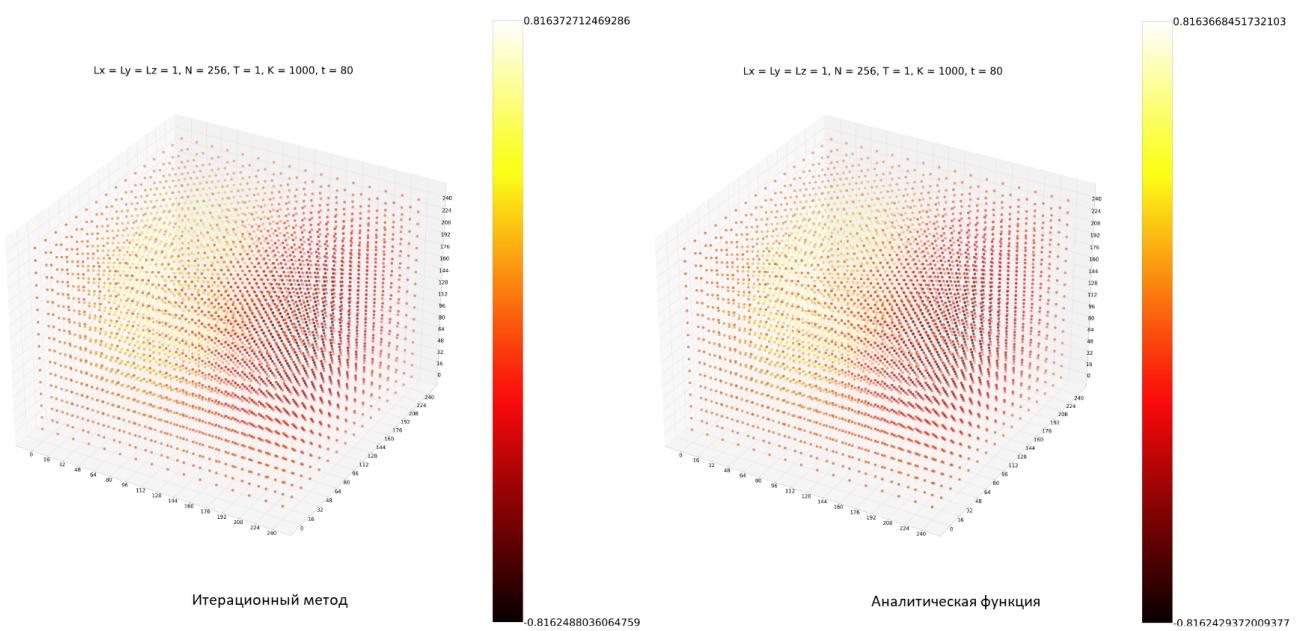


Рис. 10. Значения в сетке при  $L = 1, N = 256$  (с шагом 16)

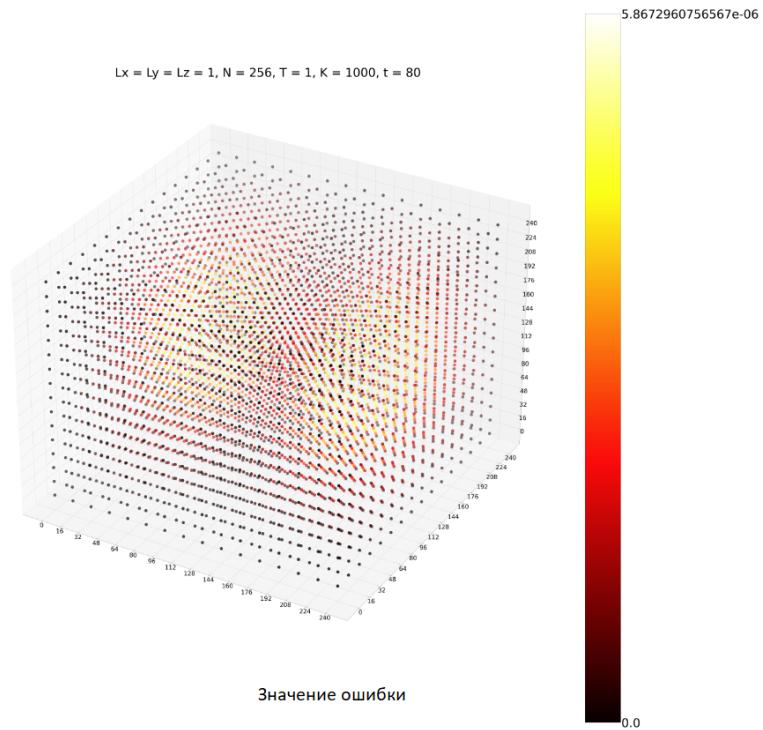
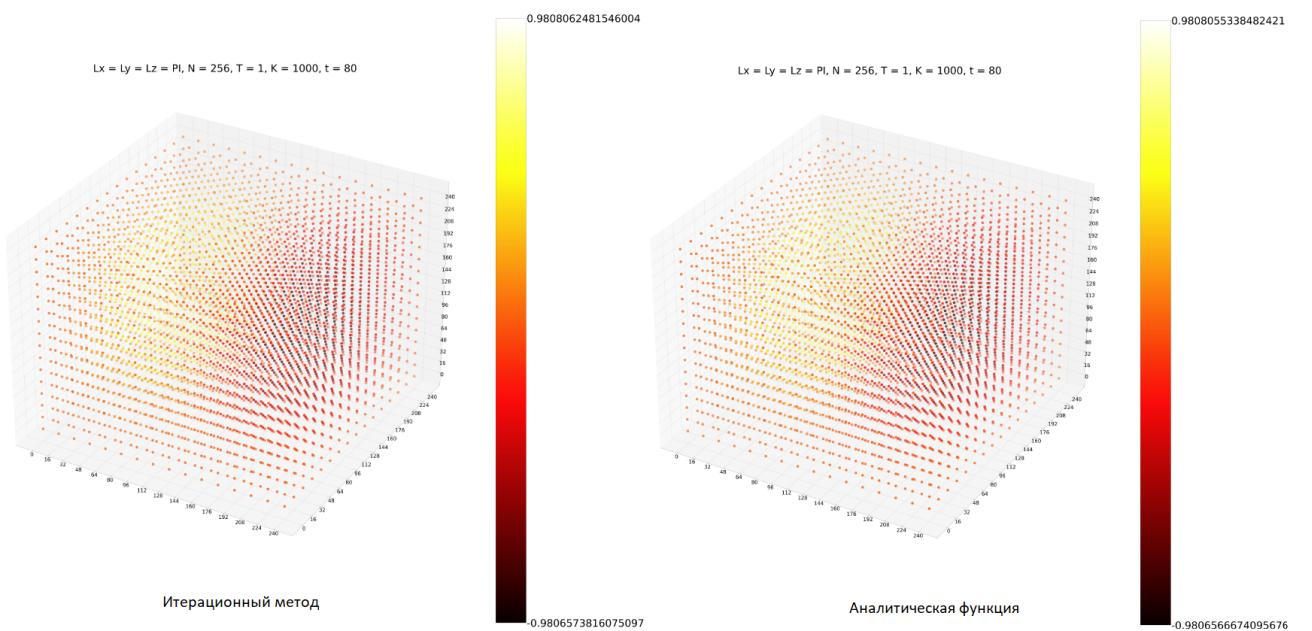
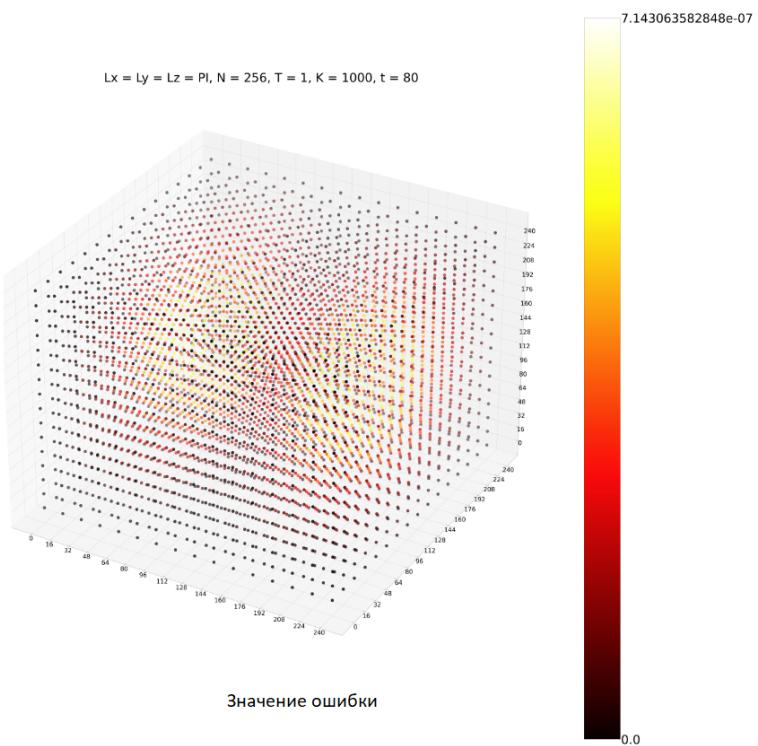


Рис. 11. Значения ошибки при  $L = 1, N = 256$  (с шагом 16)



*Рис. 12. Значения в сетке при  $L = \pi$ ,  $N = 256$  (с шагом 16)*



*Рис. 13. Значения ошибки при  $L = \pi$ ,  $N = 256$  (с шагом 16)*

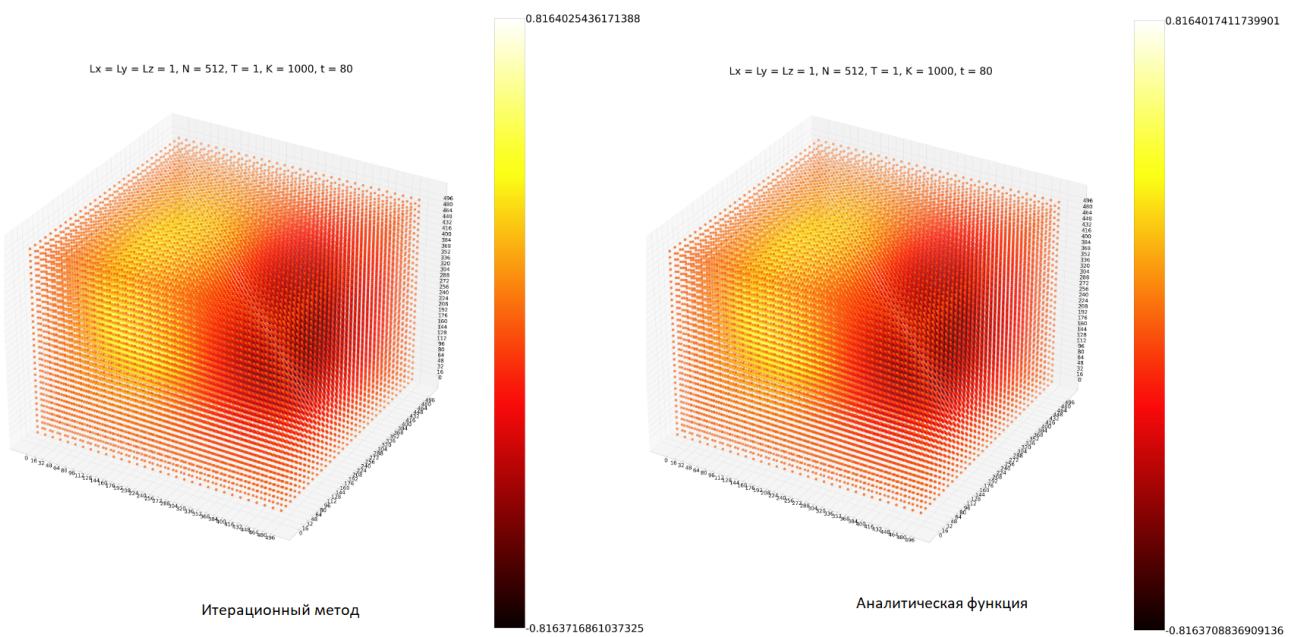


Рис. 14. Значения в сетке при  $L = 1$ ,  $N = 512$  (с шагом 16)

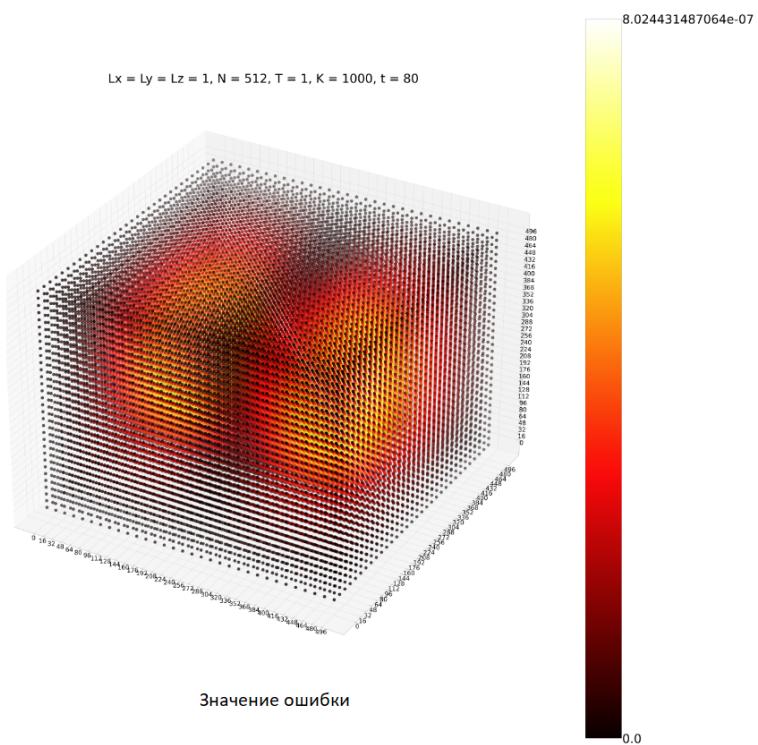


Рис. 15. Значения ошибки при  $L = 1$ ,  $N = 512$  (с шагом 16)

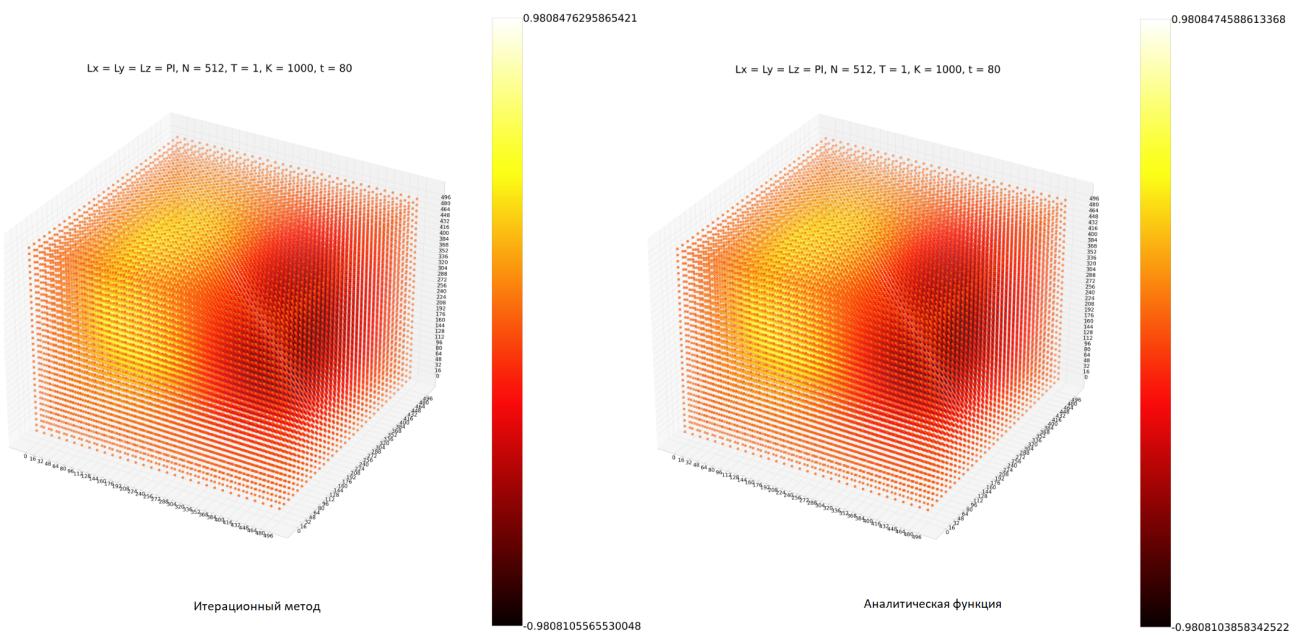


Рис. 16. Значения в сетке при  $L = \pi$ ,  $N = 512$  (с шагом 16)

