Московский государственный университет
имени М.В.Ломоносова

ЗАДАНИЕ ПО КУРСУ

«Суперкомпьютерное моделирование и технологии»

Вариант: 4

Студент: Семеняк Г.А.
Группа: 628

Сентябрь 2025 - Декабрь 2025

Москва 2025

Содержание

1 Математическая постановка дифференциальной задачи

В трехмерной замкнутой области

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

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

$$\frac{\partial^2 u}{\partial t^2} = a^2 \Delta u \tag{1}$$

с начальными условиями

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

$$\left.\frac{\partial u}{\partial t}\right|_{t=0} = 0, \tag{3}$$

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

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

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

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

либо периодические граничные условия

$$u(0, y, z, t) = u(L_x, y, z, t), \quad u_x(0, y, z, t) = u_x(L_x, y, z, t), \tag{7}$$

$$u(x, 0, z, t) = u(x, L_y, z, t), \quad u_y(x, 0, z, t) = u_y(x, L_y, z, t), \tag{8}$$

$$u(x, y, 0, t) = u(x, y, L_z, t), \quad u_z(x, y, 0, t) = u_z(x, y, L_z, t). \tag{9}$$

Конкретная комбинация граничных условий определяется индивидуальным вариантом задания (см. п. 5).

2 Численный метод решения задачи

Содержание данного пункта основано на материале книги [2]. Для численного решения задачи введем на Ω сетку $\omega_{h\tau} = \bar{\omega}_h \times \omega_\tau$, где

$$T = T_0,$$

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

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

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

Через ω_h обозначим множество внутренних, а через γ_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:

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

Здесь Δ_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

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

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

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

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

Разностная аппроксимация для периодических граничных условий выглядит следующим образом

$$
\begin{aligned}
u_{0jk}^{n+1} &= u_{Njk}^{n+1}, \quad u_{1jk}^{n+1} = u_{N+1jk}^{n+1},\\
u_{i0k}^{n+1} &= u_{iNk}^{n+1}, \quad u_{i1k}^{n+1} = u_{iN+1k}^{n+1},\\
u_{ij0}^{n+1} &= u_{ijN}^{n+1}, \quad u_{ij1}^{n+1} = u_{ijN+1}^{n+1},
\end{aligned}
$$

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

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.

3 Программная реализация

Для хранения трёхмерной сеточной функции используется одномерный вектор, а обращение к элементу выполняется через функцию вычисления индекса. Такое хранение экономит память и ускоряет доступ. Размер сетки по каждой координате задаётся переменной `N_PLUS_1 = N+1`.

Аналитическая функция $u(x, y, z, t)$ реализована отдельно и используется как для начальных условий, так и для оценки погрешности. По координате x задаются граничные условия Дирихле ($u = 0$ на границе), а по y и z реализованы периодические условия — значения на противоположных границах совпадают.

Вычисление лапласиана выполняется через центральные разности с учётом соответствующих граничных условий. Эволюция во времени реализована в функции `time_step_evolution`, где используется схема второго порядка:

$$u^{n+1} = 2u^n - u^{n-1} + a^2\tau^2\Delta u^n.$$

Для первого шага применяется модифицированный коэффициент 0.5 для корректного старта по времени.

Параллелизация реализована с помощью OpenMP. Используется директива `#pragma omp parallel for collapse(3)`, что позволяет эффективно распараллелить трёхмерные циклы по узлам сетки. Вычисление максимальной ошибки производится с редукцией `reduction(max:error)`, чтобы избежать гонок при поиске глобального максимума.

В основном цикле по времени сначала выполняется первый шаг, затем на каждом шаге вызывается функция эволюции, после чего массивы переставляются и при необходимости выводится текущая погрешность. В результате программа реализует явную схему второго порядка с периодическими условиями по y, z и Дирихле по x, эффективно распараллеленную средствами OpenMP.

4 Результаты

Вычисления проводились для следующей аналитической функции:

$$u(x,y,z,t) = \sin\left(\frac{3\pi}{L_x}x\right) \cdot \sin\left(\frac{2\pi}{L_y}y\right) \cdot \sin\left(\frac{2\pi}{L_z}z\right) \cdot \cos(a_t\, t + 4\pi),$$

$$a_t = 2\pi, \quad a^2 = \frac{1}{\frac{9}{L_x^2} + \frac{4}{L_y^2} + \frac{4}{L_z^2}}, \quad L_x = L_y = L_z = 1$$

И следующими граничными условиями:

1) Для $L_x = L_y = L_z = 1$

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

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

$$u(x,y,0,t) = u(x,y,Lz,t), \quad u(x,y,0,t) = u(x,y,Lz,t), \tag{6}$$

Вычисления производились с числом временных шагов $K = 20$.

Таблица 1: Результаты OpenMP

Число OpenMP нитей	Число узлов сетки N^3	Время решения T	Ускорение S	Погрешность δ
1	128^3	7.871	1.00	7.6104e-03
2	128^3	4.161	1.89	7.6104e-03
4	128^3	2.262	3.48	7.6104e-03
8	128^3	1.217	6.47	7.6104e-03
16	128^3	0.958	8.22	7.6104e-03
32	128^3	0.704	11.18	7.6104e-03
1	256^3	59.800	1.00	3.8078e-03
2	256^3	30.937	1.93	3.8078e-03
4	256^3	16.866	3.55	3.8078e-03
8	256^3	10.891	5.49	3.8078e-03
16	256^3	6.839	8.74	3.8078e-03
32	256^3	4.526	13.22	3.8078e-03

2) Для $L_x = L_y = L_z = \Pi$

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

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

$$u(x,y,0,t) = u(x,y,Lz,t), \quad u(x,y,0,t) = u(x,y,Lz,t), \tag{6}$$

Вычисления производились с числом временных шагов $K = 20$.

Таблица 2: Результаты OpenMP

Число OpenMP нитей	Число узлов сетки N^3	Время решения T	Ускорение S	Погрешность δ
1	128^3	8.019	1.000	8.3247e-04
2	128^3	4.109	1.952	8.3247e-04
4	128^3	2.250	3.565	8.3247e-04
8	128^3	1.223	6.559	8.3247e-04
16	128^3	0.922	8.701	8.3247e-04
32	128^3	0.709	11.314	8.3247e-04
1	256^3	59.939	1.000	4.1637e-04
2	256^3	31.778	1.887	4.1637e-04
4	256^3	19.505	3.073	4.1637e-04
8	256^3	10.463	5.728	4.1637e-04
16	256^3	7.225	8.295	4.1637e-04
32	256^3	4.674	12.818	4.1637e-04