Содержание

1 Постановка задачи **2**

2 Численный метод решения **3**

3 Программная реализация **4**

1 Постановка задачи

В трёхмерной замкнутой области

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

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

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

с начальными условиями

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

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

при условии, что на границах заданы периодические граничные условия для переменной x:

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

и однородные граничные условия первого рода для переменных y и z:

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

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

Требуется программно реализовать численный метод решения поставленной задачи с использованием технологий MPI и OpenMP, а также провести исследование масштабируемости разработанной программы.

Аналитическое решение считается заданным и равно:

$$u_{analytical} = \sin(\frac{2\pi}{L_x}x)\cdot\sin(\frac{\pi}{L_y}y)\cdot\sin(\frac{\pi}{L_z}z)\cdot\cos(\alpha_t\cdot t+2\pi), \alpha_t = \pi\sqrt{\frac{4}{L_x^2}+\frac{1}{L_y^2}+\frac{1}{L_z^2}}$$

Начальное условие выражается в виде:

$$\phi(x, y, z) = u_{analytical}(x, y, z, 0)$$

Значения константы T выбрано равным 1000.

2 Численный метод решения

Для численного решения задачи введём на Ω сетку $\omega_{h\tau} = \overline{\omega_h} \times \omega_\tau$, где

$$T = T_0,\ L_x = L_{x_0},\ L_y = L_{y_0},\ L_z = L_{z_0},$$

$$\overline{\omega_h} = \{(x_i = ih_x,\ y_j = jh_y,\ z_k = kh_z),\ i,j,k = 0,1,\dots,N,\ h_xN = L_x,\ h_yN = L_y,\ h_zN = L_z\},$$

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

Через ω_h обозначим множество внутренних, а через γ_h – множество внешних граничных узлов сетки $\overline{\omega_h}$.

Для аппроксимации исходного уравнения 1 воспользуемся следующей системой уравнений:

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

Здесь Δ_h – семиточечный разностный аналог оператора Лапласа:

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

Приведённая выше схема является явной – значения u^{n+1}_{ijk} можно явным образом выразить через значения на предыдущих слоях.

Для начала счёта (вычисления u^2_{ijk}) должны быть заданы значения u^0_{ijk} и u^1_{ijk}, $(x_i, y_j, z_k) \in \omega_h$. Для начальных условий имеем:

$$u^0_{ijk} = \phi(x_i, y_j, z_k),\ (x_i, y_j, z_k) \in \omega_h.$$

$$u^1_{ijk} = u^0_{ijk} + \frac{\tau^2}{2}\Delta_h \phi(x_i, y_j, z_k),\ (x_i, y_j, z_k) \in \omega_h.$$

Для граничных условий по x имеем:

$$u^{n+1}_{0jk} = u^{n+1}_{Njk},\ i, f, k = 0,1,\dots,N.$$

Для граничных условий по y и z имеем:

$$u^{n+1}_{i0k} = 0,\ u^{n+1}_{iNk} = 0,\ u^{n+1}_{ij0} = 0,\ u^{n+1}_{ijN} = 0.$$

3 Программная реализация

Для решения поставленной задачи была написана программа на языке C++ с использованием технологий MPI и OpenMP.

Программа представляет из себя четыре файла:

- equation.cpp – основная вызывающая программа.
- solver.cpp – последовательное решение задачи.
- mpiSolver.cpp – решение задачи с использованием технологии MPI.
- ompMpiSolver.cpp – решение задачи с использованием технологий MPI+OpenMP.

Алгоритм программы:

1. Разбиваем область между процессами (в параллельной версии).
2. Используя формулы из Секции 2 находим значения u^0 и u^1.
3. Передаём необходимые значения последнего слоя соседям.
4. Считаем значения краевых точек по краевым условиям.
5. Высчитываем значения нового слоя во всех внутренних точках процесса.
6. Определяем максимальную погрешность на сетке между посчитанным и аналитическим решением.
7. Повторяем шаги 3-6 для необходимого количества слоёв по времени.

Особенности решения задачи с использованием технолии MPI

1. Разбиение области между процессами происходит с помощью функций библиотеки mpi MPI_Dims_create и MPI_Cart_create.
2. Соседние процессы (с которыми будут производиться обмены) определяются с помощью фукнции библиотеки mpi MPI_Cart_shift.
3. Первые и последние элементы слоя, хранящиеся на каждом процессе, если это необходимо, заполняются полученными от соседей элементами. Это сделано для сохранения основных вычисляющих функций.
4. Пересылки необходимых данных производятся в асинхронном режиме.
5. Максимальная погрешность вычисляется с помощью функции редукции MPI_Reduce.

Особенности решения задачи с использованием технолии OpenMP

Технология OpenMP используется для распараллеливания вычислений краевых условий и внутренних точек на каждом временном слое. Это возможно, потому что циклы в этих вычислениях не имеют зависимостей по данным.