

Лабораторная работа № 4

Параллельная реализация метода Якоби в трехмерной области

Решаемая в лабораторной задача – это также численное получение решения уравнения при помощи последовательных приближений. По аналогии с задачей из 1-й лабораторной и по аналогии с моим примером, который я приводил на 1-й паре:

$$x - \cos x - 1 = 0$$

Только здесь искомым является не конкретное значение x , а функция φ , зависящая от трех переменных. Кстати, прошу старост групп написать – были ли у вас уравнения в частных производных? Если нет, то были ли просто дифференциальные уравнения? Заодно узнаю – читают ли мои комментарии хотя бы старосты \varnothing .

Но если никаких дифференциальных уравнений пока не было вообще, то в двух словах скажу, что в этих уравнениях мы ищем некоторую функцию, которая удовлетворяет уравнению/уравнениям. Таких функций часто бывает множество, поэтому для выбора конкретной одной функции ставятся еще начальные и краевые условия («начальные» - значения функции в начальный момент времени, если процесс зависит от времени; «краевые» - значения функции на границе области, внутри которой мы работаем). Здесь процесс не зависит от времени, поэтому только краевые условия. В тексте лабы еще упоминается о том, что это условия 1-го рода, но про это уже говорить здесь излишне – если были/будут уравнения в частных производных, то уже узнали/еще узнаете.

Может возникнуть естественный вопрос – а что это за уравнение и откуда оно взялось? Ответ: например, такому закону подчиняется потенциал электростатического поля, когда задано расположение положительных и отрицательных зарядов. Также распределение температуры в какой-то области, где заданы источники и стоки тепла, и другие статические состояния. То есть данная решаемая задача – абсолютно типичная работа численных программистов, моделирующих какой-то физический процесс или явление. Решение будет тем точнее, чем чаще сетку мы возьмем (больше точек N_x , N_y и N_z) и чем больше сделаем шагов итерационного алгоритма (m). Иными словами – чем больше вычислений, тем точнее решение. Большое количество вычислений требует распараллеливания программы.

Поскольку уравнение решается численно, то получаем мы решение не в виде какой-то явной зависимости значения функции от трех переменных вида $\varphi = \varphi(x, y, z)$, а в виде значения функции на узлах сетки $\varphi(x_i, y_j, z_k)$. На каждой итерации алгоритма эти значения уточняются.

Подобно тому, как в моем элементарном примере мы выражали следующее приближение на основе предыдущего: $x^{(m+1)} = \cos x^{(m)} + 1$, в этой задаче мы выражаем следующее уточненное значение в узле сетки (x_i, y_j, z_k) на основе значений в соседних узлах, полученных на предыдущей итерации:

$$\varphi_{i,j,k}^{m+1} = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2} + \frac{2}{h_z^2} + \alpha} \left[\frac{\varphi_{i+1,j,k}^m + \varphi_{i-1,j,k}^m}{h_x^2} + \frac{\varphi_{i,j+1,k}^m + \varphi_{i,j-1,k}^m}{h_y^2} + \frac{\varphi_{i,j,k+1}^m + \varphi_{i,j,k-1}^m}{h_z^2} - \rho_{i,j,k} \right]$$

Опять-таки для остановки итерационного процесса мы выбираем некоторую точность ε и сравниваем значения функции φ , полученные на предыдущем шаге со значениями, полученными на текущем:

$$\max_{i,j,k} |\varphi_{i,j,k}^{m+1} - \varphi_{i,j,k}^m| < \varepsilon$$

В первой лабе мы брали изначально известный вектор x^* и получали столбец b как произведение $Ax=b$, а потом героически итерационно вычисляли приближенное решение x , которое можно было сравнить на близость к x^* . Здесь аналогично – мы берем изначально известную функцию

$$\varphi(x, y, z) = x^2 + y^2 + z^2,$$

потом вычисляем значения функции в узлах сетки на границе (по условию задачи для запуска итерационного процесса нам нужно знать точные значения функции на границе). Границей, естественно, являются грани куба

$$[-1;1] \times [-1;1] \times [-1;1].$$

То есть перед запуском итераций у нас есть все вычисленные значения $\varphi(x_i, y_j, -1)$, $\varphi(x_i, y_j, 1)$, $\varphi(x_i, -1, z_k)$ и т.д. А в качестве начальных значений внутренних узлов $\varphi(x_i, y_j, z_k)$ мы берем 0, как предписывает задание.

После того, как итерационно пришли к решению, удовлетворяющему условию

$$\max_{i,j,k} |\varphi_{i,j,k}^{m+1} - \varphi_{i,j,k}^m| < \varepsilon$$

, мы проверяем – а действительно ли мы пришли туда, куда надо ϑ , вычисляя

$$\Delta = \max_{i,j,k} |\varphi_{i,j,k}^m - \varphi_{i,j,k}^*|$$

И если окажется, что Δ намного больше, чем ε (например $\Delta > 100 * \varepsilon$), то это повод сделать неутешительный вывод...

О РАСПАРАЛЛЕЛИВАНИИ

Предположим, что нашу область моделирования (по условию задачи - куб) мы разобьем сеткой по 1000 точек в каждом направлении: $Nx=Ny=Nz=1000$. Предположим, что программу мы запускаем на 4 процессах. Нумерацию точек начнем с 1, а нумерацию процессов – с 0. По условию мы должны сделать одномерную декомпозицию. Выполним

её в направлении X . Тогда у каждого из 4-х процессов будет $(Nx/4)*Ny*Nz$ точек. Рассмотрим, к примеру, процесс №2. Он ответственен за точки с координатами (x_i, y_j, z_k) , где $501 \leq i \leq 750, 1 \leq j \leq 1000, 1 \leq k \leq 1000$.

На каждой итерации основного цикла итерационного процесса наш процесс №2:

- 1) Запускает неблокирующую отправку значений точек с координатами (x_{501}, y_j, z_k) процессу №1.
- 2) Запускает неблокирующую отправку значений точек с координатами (x_{750}, y_j, z_k) процессу №3.
- 3) Запускает неблокирующий прием значений точек с координатами (x_{500}, y_j, z_k) от процесса №1.
- 4) Запускает неблокирующий прием значений точек с координатами (x_{751}, y_j, z_k) от процесса №3.
- 5) Производит вычисление следующего приближения $\varphi_{i,j,k}^{m+1}$ в точках своей зоны ответственности, начиная от середины своего слоя, двигаясь к краям. То есть сначала считает $\varphi(x_{625}, y_j, z_k)$, потом - $\varphi(x_{624}, y_j, z_k)$ и $\varphi(x_{626}, y_j, z_k)$ и т.д.
- 6) Добравшись до вычислений граничных точек своей зоны ответственности, ждет завершения неблокирующих коммуникаций приема с шагов 3) и 4).
- 7) Вычисляет $\varphi(x_{500}, y_j, z_k)$ и $\varphi(x_{750}, y_j, z_k)$, используя полученные значения.

После проверки на завершение итерационного процесса ($\max_{i,j,k} |\varphi_{i,j,k}^{m+1} - \varphi_{i,j,k}^m| < \varepsilon$) происходит повторение шагов 1) – 7) на следующей итерации.

ТРЕБОВАНИЕ

- 1) Использование неблокирующих коммуникаций на фоне счета.