

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА ПОЛНОЙ РЕДУКЦИИ

В.И. Лаева, М.А. Чижухина

Томский государственный университет

Метод полной редукции является одним из эффективных прямых методов решения разностного уравнения Пуассона в прямоугольнике [1–3]. В данной работе рассмотрен параллельный алгоритм этого метода.

Пусть требуется найти решение задачи Дирихле для уравнения Пуассона

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\varphi(x, y), & (x, y) \in D, \\ u(x, y) = g(x, y), & (x, y) \in \partial D, \\ \bar{D} = \{0 \leq x \leq l_x, 0 \leq y \leq l_y\}. \end{cases} \quad (1)$$

На прямоугольной равномерной сетке

$$\mathfrak{D} = \left\{ (x, y) = (i \cdot h_x, j \cdot h_y) \in \bar{D}, i = 0, \dots, M, j = 0, \dots, N, h_x = \frac{l_x}{M}, h_y = \frac{l_y}{N} \right\}$$

разностный аналог задачи (1) имеет вид [3]:

$$\begin{aligned} -\bar{Y}_{j-1} + C \cdot \bar{Y}_j - \bar{Y}_{j+1} &= \bar{F}_j, 1 \leq j \leq N-1, \\ \bar{Y}_0 &= \bar{F}_0, \bar{Y}_N = \bar{F}_N. \end{aligned} \quad (2)$$

где $\bar{Y}_j = (u_{1,j}, u_{2,j}, \dots, u_{M-1,j})$ – вектор-столбец неизвестных; $\bar{F}_j$ – заданная правая часть; C – трёхдиагональная, симметричная матрица.

Идея метода редукции состоит в последовательном исключении из уравнений (2) неизвестных $\bar{Y}_j$ сначала с нечётными индексами j , затем j , кратными 2, 4 и т. д. Каждый шаг процесса исключения уменьшает число неизвестных, и если $N = 2^q$, $q > 0$, то в результате останется только одно уравнение, из которого можно найти $\bar{Y}_{N/2}$. Обратный ход метода заключается в последовательном нахождении неизвестных $\bar{Y}_j$ сначала с номерами j , кратными $N/4$, затем $N/8$, $N/16$ и т. д. Метод полной редукции можно описать следующими формулами [3]:

$$\begin{aligned}
C^{(0)} &= C, \quad \overline{F_j^{(0)}} = \overline{F_j}, \\
C^{(k)} &= \left(C^{(k-1)}\right)^2 - 2 \cdot E, \\
\overline{F_j^{(k)}} &= \overline{F_{j-2^{k-1}}^{(k-1)}} + C^{(k-1)} \overline{F_j^{(k-1)}} + \overline{F_{j+2^{k-1}}^{(k-1)}}, \\
j &= 2^k, 2 \cdot 2^k, 3 \cdot 2^k, \dots, N - 2^k, \\
k &= 1, 2, \dots, q-1;
\end{aligned} \tag{3}$$

$$\begin{aligned}
C^{(k-1)} \cdot \overline{Y_j} &= \overline{F_j^{(k-1)}} + \overline{Y_{j-2^{k-1}}} + \overline{Y_{j+2^{k-1}}}, \\
\overline{Y_0} &= \overline{F_0}, \quad \overline{Y_N} = \overline{F_N}, \\
j &= 2^{k-1}, 3 \cdot 2^{k-1}, 5 \cdot 2^{k-1}, \dots, N - 2^{k-1}, \\
k &= q, q-1, \dots, 1.
\end{aligned} \tag{4}$$

По формулам (3) преобразуются правые части, а по формулам (4) находится решение.

$C^{(k)}$ – матричный полином степени 2^k относительно матрицы C , представимый в факторизованном виде:

$$C^{(k)} = \prod_{l=1}^{2^k} \left(C - 2 \cdot \cos\left(\frac{(2 \cdot l - 1) \cdot \pi}{2^{k+1}}\right) \cdot E \right), k = 0, 1, \dots, q-1,$$

поэтому для обращения $C^{(k)}$ в (4) достаточно выполнить 2^q прогонок для трёхдиагональных матриц.

Рассматриваемый метод хорошо распараллеливается в силу того, что на каждом уровне редукции производятся независимые вычисления, которые могут выполняться одновременно. Параллельная реализация алгоритма проводилась методом декомпозиции исходной расчётной области по координате y . Таким образом, на каждом процессоре рассчитывается $z = 1 + (N+1)/p$ уравнений, где $p = 2^r$ – количество процессоров ($0 < r < q$). При прямом ходе максимальная степень параллелизма наблюдается до $(q-r)$ -го шага, затем она начинает падать. Это связано с остановкой процессоров, на которых новые значения не пересчитываются, и на $q-l$ шаге работает только один. Пересылки проводятся на всех уровнях редукции, кроме первого и последнего, каждый процессор выполняет 4 двусторонних обмена по M чисел. При обратном ходе сначала работает только один процессор, затем за r шагов подключаются остальные. С $(r+1)$ -го шага все процессоры начинают работать автономно, и пересылки заканчиваются. Для приёма и передачи данных использовалась блокирующая процедура двустороннего обмена из библиотеки MPI (MPI_SendRecv).

Ускорение предложенного алгоритма метода полной редукции было вычислено по формуле

$$S_p = \frac{Q}{P + \chi \cdot O}, \tag{5}$$

где

$Q = (7 \cdot N \cdot \log_2 N + 6 \cdot \log_2 N + 1) \cdot M$ – количество арифметических операций последовательного алгоритма;

$P = \left(\frac{21}{2} \cdot N + 7 \cdot \frac{N}{p} \cdot (\log_2 N - \log_2 p - 1) + 3 \cdot (3 \cdot \log_2 p - 5) \right) \cdot M$ – количество арифметических операций параллельного алгоритма;

$O = (4 \cdot (\log_2 N - 2) + 2 \cdot \log_2 p) \cdot M$ – количество обменов, требуемых для параллельной реализации метода;

χ – количество арифметических операций с плавающей точкой, которые можно выполнить за время, необходимое для одной пересылки; M, N – размерности сетки по переменным x и y соответственно. Из формулы (5) видно, что ускорение параллельного алгоритма метода полной редукции не зависит от размерности сетки по x . Идеальное ускорение при больших N , $\chi = 0$ и $p \rightarrow N$ равно $\frac{2 \cdot \log_2 N}{3}$.

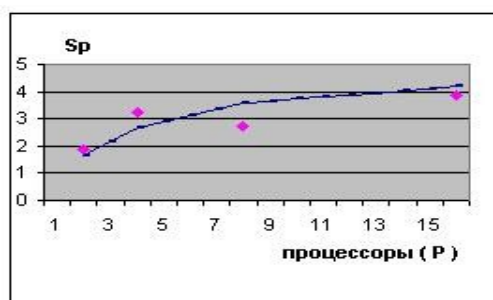


Рис. 1 Ускорение алгоритма (N=1024)

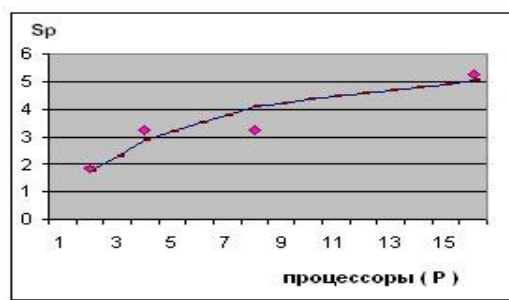


Рис. 2 Ускорение алгоритма (N=2048)

Рассматривалось численное решение задачи Дирихле для двумерного уравнения Лапласа в квадратной области с граничными условиями, равными 1. На рис. 1, 2 показано ускорение параллельного алгоритма для расчётной сетки 1024×1024 и 2048×2048 узлов. Сплошная линия соответствует аналитической зависимости, рассчитанной по формуле (5), с параметром $\chi = 100$, а ромбики – экспериментальным данным.

В таблице приведено время решения задачи на сетках с числом узлов 512×512 , 1024×1024 и 2048×2048 в зависимости от используемых процессоров.

Время счета программы для различной размерности и числа используемых процессоров

N	p				
	1	2	4	8	16
512	0,7677	0,427	0,2505	0,328	0,412
1024	3,409	1,843	1,049	1,238	0,886
2048	17,83	9,58	5,49	6,39	3,31

На сетке 512×512 наименьшее время достигается на 4 процессорах и максимальное ускорение равно 3. При $N=1024$ время расчёта на одном процессоре больше времени расчёта на 16 процессорах в 4 раза, а при $N=2048$ – почти в 5,5 раз. Полученные результаты позволяют сделать вывод, что метод полной редукции хорошо распараллеливается и ускорение параллельного алгоритма возрастает с увеличением размерности расчётной сетки.

Тестирование программы проводилось на кластере ТГУ (9 двухпроцессорных узлов с локальной памятью на базе процессоров Intel Pentium-3).

Литература

1. Хокни Р., Джессхоуп К. Параллельные ЭВМ. – М.: Радио и связь, 1986. – 392 с.
2. Марчук Г.И. Методы вычислительной математики. – М.: Наука, 1980. – 536 с.
3. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.: Наука, 1978. – 592 с.