Быстрый параллельный прямой метод решения трехмерных краевых задач с разделяющимися переменными

В.П.Ильин, Д.В.Кныш

Рассматривается безытерационный метод решения семиточечных сеточных СЛАУ сверхвысокого порядка, аппроксимирующих на параллелепипедоидальной сетке трехмерные смешанные краевые задачи для эллиптических уравнений с разделяющимися переменными. Вычислительный процесс включает параллельную реализацию двукратного быстрого преобразования Фурье (БПФ), а также двухуровневой редукции (блочной и циклической без обратного хода) для решения большого количества получаемых трехдиагональных систем высокого порядка. Приводятся оценки вычислительной сложности и эффективности распараллеливания предлагаемых методов. Проводится экспериментальное исследование производительности разработанного кода на представительной серии методических задач с использованием в широком диапазоне количества процессоров и порядков решаемых СЛАУ.

1. Введение

В данной работе предлагается и экспериментально исследуется параллельный прямой метод решения сеточных семиточечных систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами сверхвысокого порядка (до миллиарда и более), возникающих при аппроксимации трехмерных эллиптических разделяющихся краевых задач методами конечных разностей, конечных объемов или конечных элементов. Такие матрицы возникают или в самостоятельных актуальных проблемах математического моделирования, или используются в качестве эффективных предобуславливателей при решении более сложных задач. Алгоритмы решения рассматриваемых систем изучались многими авторами, см., например, [1]–[4] и цитируемые там работы.

В качестве характерного примера мы рассматриваем задачу Дирихле для уравнения Пуассона в декартовой системе координат в кубе:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=f(x,y,z),\quad (x,y,z)\in\Omega=[0,1]\times[0,1]\times[0.1],\quad u\big|_{\Gamma}=g. \tag{1}$$

После аппроксимации задачи (1) на кубической сетке с шагом $h=1/(N+1)$ получаемая СЛАУ может быть записана в форме

$$\begin{aligned}&(Au)_{i,j,k}\equiv\left[\left(A^x+A^y+A^z\right)u\right]_{i,j,k}=\\&=6u_{i,j,k}-u_{i-1,j,k}-u_{i,j-1,k}-u_{i,+1j,k}-u_{i,j+1,k}-u_{i,j,k-1}-u_{i,j,k+1}=f_{i,j,k},\end{aligned} \tag{2}$$

где $i,j,k=1,\ldots,N$, каждая из матриц A^x,A^y,A^z представляет “одномерный” оператор, соответствующий дискретизации вдоль одной из осей, а правая часть определяется следующим соотношением:

$$f_{i,j,k}=\begin{cases}f(x_i,y_j,z_k)h^2, & i,j,k=2,\ldots,N-1,\\ f(x_i,y_j,z_k)h^2+g(x_{i'},y_{j'},z_{k'}), & i,j,k\in\{1,N\},\quad i',j',k'\in\{0,N+1\}.\end{cases} \tag{3}$$

Рассматриваемые ниже методы легко обобщаются на другие типы граничных условий, а также на равномерные сетки с различными шагами h^x,h^y,h^z и с разным числом шагов N_x,N_y,N_z по соответствующим осям координат.

2. Описание численных методов

Записывая систему (2) в векторном виде

$$A\bar{u} = \bar{f} \equiv \left\{ f_{i,j,k} \right\}, \quad \bar{u} = \left\{ u_{i,j,k} \right\}, \quad A = A^x + A^y + A^z, \tag{4}$$

отметим, что в данном случае $A, \bar{u}, \bar{f}$ – симметричная матрица и векторы порядка N^3, причем A имеет ортонормальный базис из собственных векторов

$$\varphi^{p,q,r} = \left\{ \varphi_{i,j,k}^{p,q,r} \equiv \varphi_i^{x,p} \cdot \varphi_j^{y,q} \cdot \varphi_k^{z,r} ; \quad p,q,r = 1,\ldots,N \right\},$$
$$\varphi_i^{x,p} = \sqrt{\frac{2}{N+1}} \sin\frac{ip\pi}{N+1}, \varphi_j^{y,q} = \sqrt{\frac{2}{N+1}} \sin\frac{jq\pi}{N+1}, \varphi_k^{z,r} = \sqrt{\frac{2}{N+1}} \sin\frac{kr\pi}{N+1}, \tag{5}$$

а соответствующие собственные числа равны $\lambda^{p,q,r} = \lambda_p^x + \lambda_q^y + \lambda_r^z$, где

$$\lambda_p^x = 4\sin^2\frac{p\pi}{2(N+1)}, \quad \lambda_q^y = 4\sin^2\frac{q\pi}{2(N+1)}, \quad \lambda_r^z = 4\sin^2\frac{r\pi}{2(N+1)}. \tag{6}$$

При столбцовой упорядоченности узлов и соответствующих компонент векторов $\bar{u} = \left\{ u_l \right\}, \ \bar{f} = \left\{ f_l \right\}, l = k + (i-1)N + (j-1)N^2, k,i,j = 1,\ldots,N$, матрица A^z имеет блочно-диагональный вид $A^z = diag\left\{ \bar{A}^z \right\}$, где каждый блок $\bar{A}^z$ порядка N имеет собственные векторы $\varphi^{z,r} = \left\{ \varphi_k^{z,r} \right\}$ и соответствующие собственные числа λ_r^z из (6). У матрицы A^z – те же собственные числа, но каждое из них имеет кратность N^2, а ее собственные векторы, соответствующие совокупности N неповторяющихся чисел λ_r^z, записываются в блочном виде $\bar{\varphi}_{i,j}^{z,r} = \left\{ \varphi_{i,j}^{z,r}, i, j = 1,\ldots,N \right\}$. Здесь каждый подвектор $\varphi_{i,j}^{z,r} = \left\{ \varphi_k^{z,r} \right\}$ имеет порядок N, а индексы i,j указывают просто номер блока, где расположены ненулевые элементы $\bar{\varphi}_{i,j}^{z,r}$ (которые фактически зависят только от индекса k, но не от i,j). Количество разных собственных векторов $\bar{\varphi}_{i,j}^{z,r}$ и порядок каждого из них равны N^3.

Раскладывая векторы $\bar{f}$ и $\bar{u}$ по базису $\bar{\varphi}_{i,j}^{z,r}$, мы можем записать

$$\bar{f} = \left\{ \sum_{r=1}^{N} f_{i,j}^r \bar{\varphi}_{i,j}^{z,r} \right\}, \quad \bar{u} = \left\{ \sum_{r=1}^{N} v_{i,j}^r \bar{\varphi}_{i,j}^{z,r} \right\}, \tag{7}$$

где $f_{i,j}^r$ и $v_{i,j}^r$ для каждого i,j представляют собой коэффициенты разложения Фурье “одномерных столбцовых” векторов порядка N по соответствующему базису $\varphi_{i,j}^{z,r}$:

$$f_{i,j}^r = \sum_{k=1}^{N} f_{i,j,k} \varphi_k^{z,r}, \quad v_{i,j}^r = \sum_{k=1}^{N} u_{i,j,k} \varphi_k^{z,r}. \tag{8}$$

Формулы (7) и (8) представляют собой реализацию прямого и обратного разложения Фурье соответственно. Вычисление всех коэффициентов Фурье $f_{i,j}^r$ с помощью приведенных выражений требует выполнения $2N^4$ арифметических операций. Однако если здесь применить БПФ, см. [1], объем вычислений сокращается до $2N^3 \log_2 N$.

Схема предлагаемого алгоритма решения поставленной трехмерной задачи подразумевает использование двукратного (по двум направлениям) БПФ и алгоритма редукции для получаемых вспомогательных трехдиагональных СЛАУ.

С этой целью “двумерные” векторы $v^r = \left\{ v_{i,j}^r \right\}$ и $f^r = \left\{ f_{i,j}^r \right\}$ для каждого r разложим в ряды Фурье по второй переменной y:

$$v^r = \left\{ \sum_{q=1}^{N} u_i^{q,r} \varphi_j^{y,q} \right\}, \quad u_i^{q,r} = \sum_{q=1}^{N} v_{i,j}^r \varphi_j^{y,q},$$
$$f^r = \left\{ \sum_{q=1}^{N} f_i^{q,r} \varphi_j^{y,q} \right\}, \quad f_i^{q,r} = \sum_{q=1}^{N} f_{i,j}^r \varphi_j^{y,q}. \tag{9}$$

В результате для векторов $u^{q,r} = \left\{ u_i^{q,r} \right\} \in R^n$ получаем уравнения

$$\left(A^x + \lambda_q^y I + \lambda_r^z I \right) v^{q,r} = f^{q,r} \equiv \left\{ f_i^{q,r} \right\}, \quad q,r = 1,\ldots,N. \tag{10}$$

Решение каждой из этих СЛАУ N-го порядка проводим с помощью хорошо распараллеливаемого блочного метода редукции [2], который кратко представим в следующей интерпретации.

Множество узлов одномерной сетки $\Omega^h = \{x_i,\ \ i = 1,\ldots,N\}$ разобъем на $2P+1$ подобласти Ω_s, из которых нечетные называются разделителями и состоят из одного узла, а четные – содержат по M_s узлов, причем значения M_s выбираются "почти" одинаковыми и $N = 1 + P + \sum_{s=1}^{P} M_s$.

Соответствующие уравнения могут быть записаны в виде

$$(Au)_s = -A_s u_{s-1} + B_s u_s - C_s u_{s+1} = f_s, \quad s = 1,2,\ldots,2P+1, \quad A_1 = C_{2P+1} = 0, \tag{11}$$

где u_s с нечетными номерами суть скаляры, а с четными – векторы M_s-го порядка.

Отметим также, что B_s для нечетных s суть числа, а для четных – трехдиагональные матрицы.

Исключая из (11) подвекторы $u_{s\pm1}, s = 1,3,\ldots$ с помощью соотношений

$$\begin{aligned} &u_{s+1} = B_{s+1}^{-1}\left(f_{s+1} + A_{s+1}u_s + C_{s+1}u_{s+2}\right) = w_{s+1} + \hat{v}_{s+1}u_s + \breve{v}_{s+1}u_{s+2}, \\ &B_{s+1}\hat{v}_{s+1} = [1,\ldots,0]^T, \quad B_{s+1}\breve{v}_{s+1} = [1,\ldots,0]^T, \quad B_{s+1}w_{s+1} = f_{s+1}, \end{aligned} \tag{12}$$

получаем для нечетных s уравнения, связывающее узлы-разделители:

$$\begin{aligned} &-a_s u_{s-2} + b_s u_s - c_s u_{s+2} = \bar{f}_s = f_s + A_s B_{s-1}^{-1} f_{s-1} + C_s B_{s+1}^{-1} f_{s+1} = f_s + A_s w_{s-1} + C_s w_{s+1}, \\ &a_s = A_s B_{s-1}^{-1} A_{s-1},\ b_s = B_s - A_s B_{s-1}^{-1} C_{s-1} - C_s B_{s-1}^{-1} A_{s+1},\quad c_s = C_s B_{s-1}^{-1} C_{s+1}. \end{aligned} \tag{13}$$

Вычисления всех величин в (12), (13) для разных подобластей могут делаться соответствующими процессорами одновременно, а решение системы из $P+1$ уравнения вида (11) для разделителей – на каком-то одном или на каждом процессоре с помощью экономичного метода прогонки [2].

Однако если таких уравнений решается очень много (напомним, что системы (10) решаются для всех $q, r = 1,\ldots,N$), то памяти одного процессора может не хватить для хранения всех значений $a_s, b_s, c_s, \bar{f}_s$ в (13). Тогда для решения таких систем применим метод четно-нечетной редукции без обратного хода [6], первый этап которого заключается в разбиении системы (13) на две независимые подсистемы вдвое низшего порядка:

$$\begin{aligned} &-a_s^{(1)} u_{s-4} + b_s^{(1)} u_s - c_s^{(1)} u_{s+4} = f_s^{(1)} = \bar{f}_s + b_{s-2}^{-1}\bar{f}_{s-2} + b_{s+2}^{-1}\bar{f}_{s+2}, \\ &a_s^{(1)} = a_s b_{s-2}^{-1} a_{s-2},\ \ b_s^{(1)} = b_s - a_s b_{s-2}^{-1} c_{s-2} - c_s b_{s+2}^{-1} a_{s+2},\quad c_s^{(1)} = c_s b_{s+2}^{-1} c_{s+2}. \end{aligned} \tag{14}$$

Первая из этих подсистем решается для $s = 1,5,9,\ldots$, а вторая – для $s = 3,7,\ldots$ На втором этапе редукции мы аналогично получаем четыре независимые подсистемы меньших порядков и т.д. На последнем этапе имеем независимые подсистемы первого порядка, из которых непосредственно находится само решение. Такой подход будем называть методом двойной (блочной и четно-нечетной) редукции.

3. Распараллеливание вычислительного процесса

Рассмотрим параллельную версию алгоритма на МВС с распределенной памятью, представляющей собой кластер из соединенных общей коммуникационной шиной вычислительных узлов. Резюмируя вышеприведенные описания, решение всей рассматриваемой задачи можем разбить на 5 этапов:

а. выполнение прямого быстрого преобразования Фурье по оси z;

б. выполнение прямого БПФ по оси y;

в. решение N^2 трехдиагональных СЛАУ блочным методом редукции и прогонками по оси x или же методом двойной редукции;

г. выполнение обратного преобразования Фурье по оси y;

д. реализация обратного БПФ по оси z, результатом чего является искомое решение.

Проведем одномерную декомпозицию области на непересекающиеся подобласти $\Omega = \bigcup_{s=1}^{P} \Omega_s$ с помощью горизонтальных плоскостей $x = X_0, \ldots, X_P$, $\Omega_s = \{ X_{s-1} < x < X_s \}$, $s = 1, \ldots, P$, образующих равномерную макросеть в том смысле, что в каждой подобласти находится приблизительно одинаковое число узлов N^3 / P.

Подвергнем соответствующему разбиению массив правых частей f, а полученные подмассивы отправим на обработку соответствующему процессору с номером s.

После выполнения двукратного преобразования Фурье по направлениям z и y переходим к решению трехдиагональных подсистем, правые части которых сформированы в ходе выполнения этапов “а” и “б” алгоритма и располагаются в отдельных процессорах.

Проанализируем число коммуникационных обменов, требуемое в ходе выполнения этапа блочной редукции. Необходимость передачи данных соседнему процессору наступает после вычисления $w, \hat{v}$ и $\check{v}$ для подсчета коэффициентов правых частей уравнений, связывающих узлы-разделители, где задействуются околограничные значения w, расположенные в разных подобластях. Этот обмен требует (P-1) парных операций посылки-получения массива размерности N^2. Следующим шагом вычисляются коэффициенты системы уравнений, связывающих значения решения в узлах-разделителях $a_{i_k}, b_{i_k}, c_{i_k}, f_{i_k}$ $\forall x, \forall y$ и данного i_k, находящемся во взаимнооднозначном соответствии с обрабатывающим подобласть, процессором s. После этого наступает очередь коммуникаций между процессорами с целью сборки всех правых частей системы, связывающей узлы-разделители, в некотором едином процессоре, внутри которого она решается. Это действие требует (P-1) операцию посылки-получения N^2 чисел. При обратной рассылке вычисленных значений в узлах-разделителях на свой процессор, потребуется вдвое больше операций, а именно $2 \cdot (P\text{-}1)$ обменов, т.к. для сборки полного решения трехдиагональных СЛАУ необходимы оба значения в узлах-разделителях, окаймляющих подобласть. Таким образом, всего коммуникационные потери в ходе работы алгоритма составляют

$$T_c = 4(P-1)\left(\tau_0 + \tau_c N^2\right), \tag{15}$$

где τ_c – время пересылки одного числа, а τ_0 – время задержки одной операции обмена.

Время выполнения всех арифметических операций на P процессорах можно оценить величиной

$$T_a = \tau_a \left[8N^4 \left(\log_2 N\right) / P + 12N^3 / P \right],$$

где первое слагаемое в скобках относится ко всем БПФ, второе – к методу блочной редукции решения n^2 трехдиагональная СЛАУ, а τ_a есть среднее время выполнения одной арифметической операции.

Применение алгоритма двойной редукции требует большего количества обменов данными между процессорами, а именно:

$$T_c = 2\left(P \log_2 P + P - 2\right)\left(\tau_0 + \tau_c N^2\right),$$

поскольку на каждом $r = 1, ..., \log_2 P + 1$ шаге четно-нечетной редукции каждый s-ый процессор обменивается со своими процессорами-соседями, имеющими номера ($s \pm 2^{r-1}$) (с учетом границ), а также больших временных затрат на арифметические действия:

$$T_a = \tau_a \left[8N^4 \left(\log_2 N \right) / P + 10N^3 \log_2 P / P \right].$$

Эти затраты компенсируются тем, что алгоритм двойной редукции менее требователен к количеству затрачиваемой памяти одного процессора и позволяет решать очень большие задачи, с которыми не справляется алгоритм блочной редукции с прогонками в силу следующих соображений.

Предположим, что память одного процессора $V = 4$ Гб, тогда для хранения коэффициентов трехдиагональных систем ($a_{i_k}, b_{i_k}, c_{i_k}, f_{i_k}$), связывающих узлы-разделители, требуемая память оценивается величиной $4(P+1)N^2$. Каждый процессор содержит свои коэффициенты, соответствующие индексу i_k, но хотя бы один процессор должен собрать все эти коэффициенты, чтобы решение системы методом прогонки было возможным. Следовательно, для этого процессора должно выполняться соотношение $4(P+1)N^2 < V/8$ (заменатель дроби объясняется размером вещественного числа двойной точности). Упростив приведенное соотношение, мы увидим следующее ограничение на количество процессоров, используемое для решения задачи методом блочной редукции с прогонками – P < 127. С другой стороны, такое количество процессоров оказывается недостаточным для размещения очень больших задач (для решения задачи размером 1025^3 требуется как минимум 128 процессоров).

4. Примеры численных экспериментов

Проиллюстрируем эффективность распараллеливания описанных прямых методов в применении к решению симметричных систем семиточечных уравнений (2), аппроксимирующих задачу (1) на различных кубических сетках. Функции f и g в (1) выбирались из условия, что точное решение $u(x, y, z)$ равно единице. Реализация алгоритмов осуществлялась на Фортране, а все вычисления проводились со стандартной двойной точностью. Эксперименты проводились для трех разных сеток со значениями N=129,257,513. Ниже приведены результаты измерений времени (в сек.) исполнения параллельных версий алгоритмов в системе MPI для 1,2,4,8 и 16 процессоров (на кластере с 2-х процессорными узлами Itanium 2). Каждая клетка таблиц содержит 3 значения, соответствующих времени исполнения соответствующего этапа алгоритма, сверху вниз – общее время двукратного прямого БПФ, решения трехдиагональных СЛАУ с применением прогонки, время выполнения обратного БПФ, и полное время решения задачи ttotal.

N\P	1	2	4	8	16
129	0.17 0.53 0.16	$8.19 \cdot 10^{-2}$ 0.35 $8.19 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$ 0.28 $4.29 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$ 0.18 $1.8 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$ $9.76 \cdot 10^{-2}$ $1.8 \cdot 10^{-2}$
Ttotal	0.86	0.52	0.36	0.22	0.16
257	1.56	0.77	0.38	0.15	0.12

	5.50 1.5	2.83 0.75	1.40 0.36	0.67 0.18	0.37 $8.58 \cdot 10^{-2}$
Ttotal	8.58	4.38	2.17	1.06	0.58
513	- - -	11.65 28.09 11.44	5.55 12.94 5.51	2.86 6.14 2.87	1.46 2.91 1.44
Ttotal	-	51.20	24.03	11.88	5.83

Табл. 1. Временные характеристики прямого метода в MPI
(двукратное БПФ+блочный метод редукции прогонки)

а)

N\P	4	8	16
129	$5.07 \cdot 10^{-2}$ 0.13 $4.29 \cdot 10^{-2}$	$2.34 \cdot 10^{-2}$ 0.10 $2.95 \cdot 10^{-2}$	$7.02 \cdot 10^{-2}$ $6.25 \cdot 10^{-2}$ $11.71 \cdot 10^{-2}$
Ttotal	0.23	0.15	0.10
257	0.37 1.14 0.38	0.18 0.54 0.18	$8.97 \cdot 10^{-2}$ 0.29 $10.14 \cdot 10^{-2}$
Ttotal	1.91	0.93	0.48
513	-	2.69 5.64 2.67	1.42 2.98 1.38
Ttotal	-	11.02	5.80

б)

N\P	128	256
1025	0.51 7.87 0.47	0.23 10.58 0.26
ttotal	8.86	11.10

Табл. 2. Временные характеристики прямого метода в MPI
(двукратное БПФ+двойной метод редукции)

В табл. 2а, 2б приводятся аналогичные данные при решении трехдиагональных СЛАУ алгоритмом двойной редукции.
Анализ полученных экспериментальных данных позволяет сделать следующие выводы:

1. Временные характеристики рассмотренных методов для решения трехмерного уравнения Пуассона, при помощи алгоритма блочной редукции с прогонками и двойной редукции, отличаются незначительно и обеспечивают ускорение, близкое к линейному
2. Использование алгоритма двойной редукции предоставляет существенное преимущество по памяти, что позволило решить задачу на сетке большой размерности (1025^3) при числе процессоров $P \geq 128$ за время $t \approx 9$ сек., что можно считать очень хорошим результатом.
3. Вопрос о возможности масштабируемости распараллеливания при увеличении числа процессоров $P > 256$ для $N > 1025$ требует дальнейших исследований.

Литература

1. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений.–М.: Наука, 1978.
2. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических

уравнений.–Новосибирск: Изд. ИВМ СО РАН, 2000.
3. Вшивков В.А., Снытников В.Н., Снытников Н.В., Моделирование трехмерной динамики вещества в гравитационном поле на многопроцессорных ЭВМ //Вычислительные технологии. Т.11, № 2, 2006., 15-27.
4. Ильин В.П., Кныш Д.В. Параллельные алгоритмы решения разделяющихся краевых задач. –Санкт-Петербург, изд. Политехн. ун-та (СПбПГУ), 2008, 107-118.
5. Богачев К.Ю. Основы параллельного программирования.–М., БИНОМ, Лаборатория знаний, 2003.
6. Ильин В.П., Кузнецов Ю.И. Трехдиагональные матрицы и их приложения.–М.: Наука, 1985.