

## Параллельный прямой метод решения разделяющихся краевых задач

Д.В. КНЫШ

*Новосибирский государственный университет, Россия*

e-mail: [dasha\\_knysh@gorodok.net](mailto:dasha_knysh@gorodok.net)

A parallel direct method for solution of a separable boundary value problem is considered. The method is on the Fourier transformation (applied twice) along with the method of reduction. Efficiency of parallelization of the proposed method is analyzed for of the case of 3D problems. The programming implementation for distributed and shared memory systems along with series of numerical measurements are performed.

### Введение

Алгоритмы решения систем линейных алгебраических уравнений (СЛАУ) высокого порядка с разреженными матрицами специальной блочной структуры, возникающих при аппроксимации многомерных краевых задач с разделяющимися переменными, изучались многими авторами (см., например, [1–3] и приведенную там библиографию). Такие методы, обладающие рекордным быстродействием, имеют самостоятельное значение для многих приложений, а также могут быть использованы для построения эффективных предобусловленных итерационных процессов, применимых к решению более сложных уравнений математической физики. К числу наиболее экономичных алгоритмов относятся методы быстрого преобразования Фурье (БПФ), циклической редукции, а также итерационные неявные методы переменных направлений с оптимальной последовательностью параметров, использующие особенности спектральных и структурных свойств матриц. В работе [4] рассмотрено применение комбинированных прямых и итерационных методов для решения дискретизированных трехмерных краевых задач с полуразделяющимися переменными.

Целью данной работы является анализ эффективности распараллеливания прямого алгоритма для решения трехмерных смешанных краевых задач. В качестве характерного примера рассматривается задача Дирихле для уравнения Пуассона в кубе:

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

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

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

где каждая из матриц  $A^x, A^y, A^z$  представляет “одномерный” оператор, соответствующий дискретизации вдоль одной из осей, а правая часть определяется следующим соотношением:

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

В настоящей работе описан применяемый алгоритм, рассмотрены вопросы его отображения на архитектуру многопроцессорных вычислительных систем (МВС) с общей и разделенной памятью, приводятся результаты численных экспериментов, демонстрирующие эффективность распараллеливания изучаемого метода с использованием систем MPI и OpenMP [5], дается анализ результатов и обсуждаются возможные направления дальнейших исследований.

## 1. Описание численных алгоритмов

Записывая систему (2) в векторном виде

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

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

$$\begin{aligned} \varphi^{p,q,r} &= \{\varphi_{i,j,k}^{p,q,r} \equiv \varphi_i^{x,p} \varphi_j^{y,q} \varphi_k^{z,r}; p, q, r = 1, \dots, N\}, \\ \varphi_i^{x,p} &= \sqrt{\frac{2}{N+1}} \sin \frac{ip\pi}{N+1}, \quad \varphi_j^{y,q} = \sqrt{\frac{2}{N+1}} \sin \frac{jq\pi}{N+1}, \quad \varphi_k^{z,r} = \sqrt{\frac{2}{N+1}} \sin \frac{kr\pi}{N+1}, \end{aligned} \quad (5)$$

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

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

При столбцовой упорядоченности узлов и соответствующих компонентов векторов  $\bar{u} = \{u_l\}$ ,  $\bar{f} = \{f_l\}$ ,  $l = k + (i-1)N + (j-1)N^2$ ,  $k, i, j = 1, \dots, N$ , матрица  $A^z$  имеет блочно-диагональный вид  $A^z = \text{diag}\{\bar{A}^z\}$ , где каждый блок  $\bar{A}^z$  порядка  $N$  имеет собственные векторы  $\varphi_k^{z,r} = \{\varphi_k^{z,r}\}$  и соответствующие собственные числа  $\lambda_r^z$  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  $\lambda_r^z$ , записываются в блочном виде  $\bar{\varphi}_{i,j}^{z,r} = \{\varphi_{i,j}^{z,r}, i, j = 1, \dots, N\}$ . Здесь каждый подвектор  $\varphi_{i,j}^{z,r} = \{\varphi_k^{z,r}\}$  имеет порядок  $N$ , а индексы  $i, j$  указывают номер блока, где расположены ненулевые элементы  $\varphi_{i,j}^{z,r}$  (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

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

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

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

Формулы (7) и (8) представляют собой реализацию прямого и обратного разложений 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 [1], то 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  $v^r = \{v_{i,j}^r\}$  и  $f^r = \{f_{i,j}^r\}$  для каждого  $r$  разложим в ряды Фурье по второй переменной  $y$ :

$$\begin{aligned} v^r &= \left\{ \sum_{q=1}^N u_i^{q,r} \varphi_j^{y,q} \right\}, & u_i^{q,r} &= \sum_{q=1}^N v_{i,j}^r \varphi_j^{y,q}, \\ f^r &= \left\{ \sum_{q=1}^N f_i^{q,r} \varphi_j^{y,q} \right\}, & f_i^{q,r} &= \sum_{q=1}^N f_{i,j}^r \varphi_j^{y,q}. \end{aligned} \quad (9)$$

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

$$(A^x + \lambda_y^y I + \lambda_z^z I) u^{q,r} = f^{q,r} \equiv \{f_i^{q,r}\}, \quad q, r = 1, \dots, N. \quad (10)$$

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

Множество узлов одномерной сетки  $\Omega = \{l = 1, \dots, N\}$  разобьем на  $2P + 1$  подобласти  $\Omega_p$ , из которых нечетные называются разделителями и состоят из одного узла, а четные области содержат по  $M$  узлов, причем  $1 + (M + 1)P = N$  (см. рисунок).

Соответствующие уравнения могут быть записаны в виде

$$(Au)_p = -A_p u_{p-1} + B_p u_p - C_p u_{p+1} = f_p, \quad p = 1, 2, \dots, P, \quad A_1 = C_P = 0, \quad (11)$$

где  $u_p$  с нечетными номерами есть скаляры, а с четными — векторы  $M$ -го порядка. Исключая из (11) подвекторы  $u_{p\pm 1}$  с помощью соотношений

$$\begin{aligned} u_{p+1} &= B_{p+1}^{-1} (f_{p+1} - A_{p+1} u_p - C_{p+1} u_{p+2}) = w_{p+1} - \widehat{v}_{p+1} u_p - \check{v}_{p+1} u_{p+2}, \\ B_{p+1} \widehat{v}_{p+1} &= [1, \dots, 0]^T a, \quad B_{p+1} \check{v}_{p+1} = [0, \dots, 1]^T c, \end{aligned} \quad (12)$$

для четных  $p$  получаем уравнения, связывающее узлы-разделители:

$$\begin{aligned} \bar{a}_p u_{p-2} + \bar{b}_p u_p + \bar{c}_p u_{p+2} &= f_p, \\ \bar{a}_p &= A_p B_{p-1}^{-1} A_{p-1}, \quad \bar{b}_p = B_p - A_p B_{p-1}^{-1} C_{p-1} - C_p B_{p+1}^{-1} A_{p+1}, \quad \bar{c}_p = C_p B_{p+1}^{-1} C_{p+1}. \end{aligned} \quad (13)$$

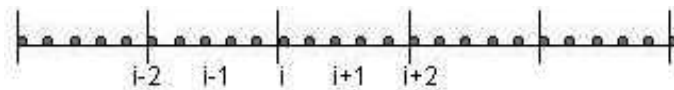


Схема декомпозиции области в методе редукции

Вычисления всех величин в (12), (13) для разных подобластей могут выполняться соответствующими процессорами одновременно, а решение системы из  $P$  уравнений вида (11) для разделителей — на каком-то одном или на каждом процессоре.

## 2. Распараллеливание вычислительного процесса

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

- а) выполнение прямого быстрого преобразования Фурье по оси  $z$ ;
- б) выполнение прямого БПФ по оси  $y$ ;
- в) решение  $N^2$  трехдиагональных СЛАУ методом редукции по оси  $x$ ;
- 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  $\Omega = \bigcup_{t=1}^P \Omega_t$  с помощью горизонтальных плоскостей  $x = X_0, \dots, X_P$ ,  $\Omega_t = \{X_{t-1} < x < X_t\}$ ,  $t = 1, \dots, P$ , образующих равномерную макросеть в том смысле, что в каждой подобласти находится одинаковое число узлов  $N^3/P$ .

Подвергнем соответствующему разбиению массив правых частей  $f$ , а полученные подмассивы отправим на обработку соответствующему процессору  $p$ .

После выполнения двукратного преобразования Фурье по направлениям  $z$  и  $y$  переходим к решению редуцированных 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 “редукция”. 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 околোগраничные значения  $w$ , расположенные в разных подобластях. Этот обмен требует  $(P - 1)$  парных операций отправки-получения массива размерности  $N^2$ . Отметим, что в силу известных соотношений такой обмен работает эффективнее, нежели  $N^2(P - 1)$ -я операция отправки-получения одного числа. Следующим шагом вычисляются  $ik \forall x, \forall y$  и данные  $ik$ , находящиеся во взаимно-однозначном соответствии с обрабатываемой подобластью процессором  $p$ . И 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  $(P - 1)$  операций отправки-получения  $N^2$  чисел. При обратной рассылке вычисленных значений в узлах-разделителях на свой процессор потребуется вдвое больше операций, а именно  $2(P - 1)$  обменов, так как для сборки полного решения трехдиагональных СЛАУ необходимы оба значения в узлах-разделителях, окаймляющих подобласть. Таким образом, всего коммуникационные потери в ходе работы алгоритма составляют

$$T_c = 4(P - 1)(\tau_0 + \tau_c N^2), \quad (14)$$

где  $\tau_c$  — время пересылки одного числа, а  $\tau_0$  — время задержки одной операции обмена.

Время выполнения всех арифметических операций на  $P$  процессорах можно оценить величиной

$$T_a = \tau_a \left( \frac{8N^4 \log_2 N}{P} + \frac{12N^3}{P} \right).$$

где первое слагаемое в скобках относится ко всем БПФ, второе — к методу редукции решения  $n^2$  трехдиагональной СЛАУ, а  $\tau_a$  есть среднее время выполнения одной арифметической операции.

Реализация алгоритма на архитектурах с общей памятью (МВС-ОП) отличается от описанного способа распараллеливания на МВС-РП в первую очередь оригинальным применением композиции методов решения трехдиагональных СЛАУ, где решение каждой редуцированной подсистемы этапа 3 алгоритма осуществляется с помощью алгоритма встречных прогонок (см., например [2]), в основе которого лежит идея использования для подобластей разных вариантов методов прогонки, осуществляющих вычисления в противоположных направлениях и встречающихся лишь однажды посередине отрезка после прямого хода. Таким образом задача расщепляется на две части, обработка которых может проводиться параллельно отдельными процессорами. Вообще говоря, использование общей памяти здесь не принципиально, обозначенное разделение может быть применено и на МВС-РП, но привлекательно за счет отсутствия коммуникационных потерь и перспективы применения в дальнейшем этого подхода для реализации на гибридных архитектурах, где каждая редуцированная подсистема располагается и обрабатывается на каком-нибудь узле своим MPI-процессом, а внутри узла части встречной прогонки обрабатываются каждая своим потоком.

### 3. Примеры численных экспериментов

Проиллюстрируем эффективность распараллеливания описанного прямого метода на МВС-РП и МВС-ОП в применении к решению симметричных систем семиточечных уравнений, аппроксимирующих задачу Дирихле для уравнения Пуассона в единичном кубе 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

Т а б л и ц а 1. Временные характеристики прямого метода в MPI (двукратное БПФ + метод редукции)

$N \setminus p$	1	2	4	8	16
129	0.17	$8.19 \cdot 10^{-2}$	$3.9 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$
	0.53	0.35	0.28	0.18	$9.76 \cdot 10^{-2}$
	0.16	$8.19 \cdot 10^{-2}$	$4.29 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$	$1.8 \cdot 10^{-2}$
ttotal	0.86	0.52	0.36	0.22	0.16
257	1.56	0.77	0.38	0.15	0.12
	5.50	2.83	1.40	0.67	0.37
	1.5	0.75	0.36	0.18	$8.58 \cdot 10^{-2}$
ttotal	8.58	4.38	2.17	1.06	0.58
513	-	11.65	5.55	2.86	1.46
	-	28.09	12.94	6.14	2.91
	-	11.44	5.51	2.87	1.44
ttotal	-	51.20	24.03	11.88	5.83

Т а б л и ц а 2. Временные характеристики прямого метода в OpenMP (двукратное БПФ + метод редукции \* метод встречной прогонки)

$N \setminus th$	0	1	2
129	0.17	0.17	0.1
	0.35	0.39	0.38
	0.17	0.17	0.1
ttotal	0.68	0.72	0.59
257	1.56	1.56	0.83
	3.32	3.35	2.98
	1.51	1.50	0.80
ttotal	6.41	6.45	4.64
513	22.89	22.84	11.84
	33.21	31.10	25.28
	22.49	22.47	11.64
ttotal	78.61	76.43	48.78

на Фортране, а все вычисления проводились со стандартной двойной точностью. Эксперименты проводились для трех разных сеток со значениями  $N = 129, 257, 513$ . В табл. 1 и 2 приведены результаты измерений времени (в секундах) исполнения параллельных версий алгоритма в системе MPI для 1, 2, 4, 8 и 16 процессоров (на кластере с двух процессорными узлами Itanium 2) и OpenMP для двух потоков (на четырех-процессорном Itanium 2) ради анализа эффективности использования параллельной версии метода встречных прогонок в последовательной версии рассматриваемого метода. Каждая клетка обеих таблиц содержит три значения, соответствующих времени исполнения соответствующего этапа алгоритма, сверху вниз — общее время ППФ( $z$ ) и ППФ( $y$ ), редукция, общее время ОПФ( $y$ ) и ОПФ( $z$ ) и полное время решения задачи ttotal.

## Заключение

Использование предложенного параллельного прямого метода показывает высокую эффективность распараллеливания на МВС-РП в случае применения технологии распараллеливания MPI, а получаемое ускорение практически во всех случаях линейно. Метод встречных прогонок перспективен для использования при решении редуцированных подсистем. Рост производительности имеет место при его применении для решения задач на МВС-ОП, что дает основания предположить полезность его применения в случае гибридных систем (что будет проанализировано в последующих работах автора).

## Список литературы

- [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: ИВМ СО РАН, 2000.

- [3] Вшивков В.А., Снытников В.Н., Снытников Н.В. Моделирование трехмерной динамики вещества в гравитационном поле на многопроцессорных ЭВМ // Вычисл. технологии. 2006. Т. 11, № 2. С. 15–27.
- [4] Ильин В.П., Кныш Д.В. Параллельные алгоритмы решения разделяющихся краевых задач. СПб.: Изд. СПбГУ, 2008. С. 107–118.
- [5] Богачев К.Ю. Основы параллельного программирования. М.: БИНОМ, Лаб. знаний, 2003.

*Поступила в редакцию 20 февраля 2007 г.,  
в переработанном виде — 13 мая 2008 г.*