Math-Net.Ru

Общероссийский математический портал

Л. В. Логанова, Д. Л. Головашкин, О. С. Сягайло, Параллельный алгоритм метода циклической редукции для периодической краевой задачи, *Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки*, 2010, выпуск 1(), 197–204

DOI: <https://doi.org/10.14498/vsgtu740>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.197.130.99

26 апреля 2023 г., 06:00:05

Информатика

УДК 681.3.06

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МЕТОДА ЦИКЛИЧЕСКОЙ РЕДУКЦИИ ДЛЯ ПЕРИОДИЧЕСКОЙ КРАЕВОЙ ЗАДАЧИ

Л. В. Логанова¹, Д. Л. Головашкин², О. С. Сягайло¹

¹ Самарский государственный аэрокосмический университет им. ак. С. П. Королёва, 443086, Самара, Московское ш., 34.

² Институт систем обработки изображений РАН, 443001, Самара, ул. Молодогвардейская, 151.

E-mails: tk@smr.ru, dimitriy@smr.ru

Работа посвящена построению параллельного алгоритма, основанного на методе циклической редукции для периодической краевой задачи. Произведено сравнение с известным алгоритмом, приведены результаты численных экспериментов по исследованию ускорения алгоритма. Показана высокая эффективность его применения.

Ключевые слова: метод циклической редукции, параллельные алгоритмы.

Введение. Математическое моделирование получает все большее распространение в различных областях науки и техники, например, вычислительной теплопередаче [1]. Развитие вычислительных средств приводит к необходимости интеграции методов математического моделирования и современных компьютерных технологий. Моделирование физических процессов зачастую сводится к необходимости решения систем линейных алгебраических уравнений (СЛАУ) вида $Ax = b$, где матрица A имеет ленточную структуру. Решение технических задач в таком случае сопровождается существенным объёмом вычислений. Именно этот фактор определяет растущую потребность в применении параллельных вычислительных систем при решении таких СЛАУ, и, как следствие, необходимость в синтезе алгоритмов, определяемых такой архитектурой.

Для создания параллельных алгоритмов решения СЛАУ с ленточной матрицей традиционно используются следующие прямые методы: прогонки, циклической редукции, декомпозиции области. Метод прогонки является экономичным, но плохо масштабируемым. Так, применение метода правой прогонки из работ [2, 3] непосредственно для одномерной (1D) сеточной области невозможно. Метод встречных прогонок из [4, 5] не может быть применим для решения разностных уравнений с циклическими граничными условиями. Алгоритм, основанный на методе циклической редукции [6, 7], пользуется особой популярностью для решения трёхдиагональных систем как для параллельных, так и для векторных компьютеров, являясь привлекательным в силу

Лилия Владимировна Логанова, старший преподаватель, каф. технической кибернетики. Димитрий Львович Головашкин (д.ф.-м.н., с. н. с.), старший научный сотрудник, лаб. дифракционной оптики. Ольга Сергеевна Сягайло, студент.

присущей ему масштабируемости и небольшой вычислительной сложности. Данные соображения определили выбор авторов в пользу применения метода циклической редукции.

В современной литературе недостаточное внимание уделяется применению указанного метода при решении периодических одномерных задач посредством разностного подхода. Так, в [6] приведён случай непериодической задачи, а в [7] рассмотрена двумерная задача. Настоящая работа призвана восполнить этот пробел.

Развивая подход, применяемый в непериодическом случае, авторы обращаются к синтезу алгоритма для одномерной периодической задачи, его параллельного варианта и подтверждают эффективность результатами вычислительных экспериментов.

1. Циклическая редукция для случая трёхдиагональных систем. Предварим представление параллельного алгоритма описанием метода циклической редукции, приведенного в [6, 8], для случая трёхдиагональных систем вида

$$\begin{cases} dx_1 + fx_2 & = b_1, \\ fx_1 + dx_2 + fx_3 & = b_2, \\ fx_2 + dx_3 + fx_4 & = b_3, \\ & \dots, \\ fx_{n-1} + dx_n & = b_n, \end{cases} \quad (1)$$

где $d, f, b_i \in \mathbb{R}$. Отметим существенное ограничение на размерность системы $n = 2^k - 1$, где $k \in \mathbb{N}$.

Основная идея метода циклической редукции [7, 8, 10] заключается в исключении неизвестных с номерами не кратными 2^p на каждом шаге p редукции. Алгоритм можно обобщить и сформулировать следующим образом:

```
{прямой ход редукции}
{цикл от начальной системы до последней редуцированной системы}
for  $p = 1 : k - 1$ 
    {поиск коэффициентов новой редуцированной системы}
     $d^{(p)} = 2[f^{(p-1)}]^2 - [d^{(p-1)}]^2$ ,
     $f^{(p)} = [f^{(p-1)}]^2$ 
    { $r$  - расстояние между двумя соседними уравнениями новой системы
    в старой системе}
     $r = 2^p$ 
    {поиск правых частей новой редуцированной системы}
    for  $j = 1 : 2^{(k-p)} - 1$ 
         $b_{jr}^{(p)} = f^{(p-1)}(b_{jr-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)}) - d^{(p-1)}b_{jr}^{(p-1)}$ 
    end
end
{обратный ход редукции}
Решить  $d^{(k-1)}x_{2^{k-1}} = b^{(k-1)}$  относительно  $x_{2^{k-1}}$ 
{цикл по редуцированным системам до начальной системы}
for  $p = k - 2 : 0 : -1$ 
    { $r$  - расстояние между двумя соседними уравнениями новой системы
    в старой системе}
     $r = 2^p$ 
    {цикл по элементам редуцированной системы}
    for  $j = 1 : 2^{(k-p-1)}$ 
        {работаем с первым уравнением}
        if  $j = 1$ 
             $c = b_{(2j-1)r}^{(p)} - f^{(p)}x_{2jr}$ ;
```

```

{работаем с последним уравнением}
  else if  $j = 2^{(k-p-1)}$ 
     $c = b_{(2j-1)r}^{(p)} - f^{(p)} x_{(2j-2)r}$ ;
  {работаем с остальными уравнениями}

    else  $c = b_{(2j-1)r}^{(p)} - f^{(p)} (x_{2jr} + x_{(2j-2)r})$ ;
  end
  Решить  $d^{(p)} x_{(2j-1)r} = c$  относительно  $x_{(2j-1)r}$ 
end
end

```

Представленный алгоритм записан в нотации Дж. Голуба [10], при этом верно упомянутое ограничение на размерность системы и $d^{(0)} = d$, $f^{(0)} = f$, $b^{(0)} = b$.

2. Циклическая редукция для периодической краевой задачи. В практике вычислительного эксперимента нередко встречаются периодические задачи на круговых областях, бесконечных решётках и т.п. Для них система (1) будет модифицирована в следующую:

$$\begin{cases} dx_1 + fx_2 & + & fx_n & = b_1, \\ fx_1 + dx_2 + fx_3 & & & = b_2, \\ & fx_2 + dx_3 + fx_4 & & = b_3, \\ & & \dots & \\ fx_1 & + & fx_{n-1} + dx_n & = b_n. \end{cases}$$

В отличие от (1), здесь $n = 2^k$, где $k \in \mathbb{N}$. Используя подход [7], запишем алгоритм циклической редукции, учитывая ограничение на размерность системы:

```

{прямой ход редукции}
{цикл от начальной системы до предпоследней редуцированной системы}
for  $p = 1 : k - 2$ 
  {поиск коэффициентов новой редуцированной системы}
   $d^{(p)} = 2[f^{(p-1)}]^2 - [d^{(p-1)}]^2$ ,
   $f^{(p)} = [f^{(p-1)}]^2$ 
  { $r$  - расстояние между двумя соседними уравнениями новой системы
  в старой системе}
   $r = 2^p$ 
  {поиск правых частей новой редуцированной системы}
  for  $j = 1 : 2^{(k-p)} - 1$ 
     $b_{jr}^{(p)} = f^{(p-1)} (b_{jr-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)}) - d^{(p-1)} b_{jr}^{(p-1)}$ 
  end
end
{поиск коэффициентов последней редуцированной системы}
 $p = k - 1$ 
 $d^{(p)} = 2[f^{(p-1)}]^2 - [d^{(p-1)}]^2$ ,
 $f^{(p)} = 2[f^{(p-1)}]^2$ 
{ $r$  - расстояние между двумя соседними уравнениями новой системы
в старой системе}
 $r = 2^p$ 
{поиск правых частей последней редуцированной системы}
for  $j = 0 : 2^{(k-p)} - 1$ 
  if  $j=0$ 
     $b_{jr}^{(p)} = f^{(p-1)} (b_{2^k-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)}) - d^{(p-1)} b_{jr}^{(p-1)}$ 
  else

```

```


$$b_{jr}^{(p)} = f^{(p-1)}(b_{jr-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)}) - d^{(p-1)}b_{jr}^{(p-1)}$$

end
end
{обратный ход редукции}
Решить систему относительно  $x_0, x_{2^{k-1}}$ 
 $d^{(k-1)}x_0 + f^{(k-1)}x_{2^{k-1}} = b_0^{(k-1)}$ 
 $f^{(k-1)}x_0 + d^{(k-1)}x_{2^{k-1}} = b_{(2^{k-1})}^{(k-1)}$ 
{цикл по редуцированным системам до начальной системы}
for  $p = k - 2 : 0 : -1$ 
{ $r$  - расстояние между двумя соседними уравнениями новой системы
в старой системе}
 $r = 2^p$ 
{цикл по элементам редуцированной системы}
for  $j = 1 : 2^{(k-p-1)}$ 
{работаем с первым уравнением}
if  $j = 1$ 
 $c = b_{(2j-1)r}^{(p)} - f^{(p)}x_{2jr} - f^{(p)}x_0$ 
{работаем с последним уравнением}
else if  $j = 2^{(k-p-1)}$ 
 $c = b_{(2j-1)r}^{(p)} - f^{(p)}x_{(2j-2)r} - f^{(p)}x_0$ ;
{работаем с остальными уравнениями}
else
 $c = b_{(2j-1)r}^{(p)} - f^{(p)}(x_{2jr} + x_{(2j-2)r})$ ;
end
end
Решить  $d^{(p)}x_{(2j-1)r} = c$  относительно  $x_{(2j-1)r}$ 
end
end

```

Здесь также верно упомянутое ограничение на размерность системы и $d^{(0)} = d$, $f^{(0)} = f$, $b^{(0)} = b$.

3. Параллельный алгоритм метода циклической редукции. Заметим, что операции, ведущие к новым уравнениям на каждом шаге циклической редукции, могут выполняться параллельно. Рассмотрим случай, когда есть два процессора ($L = 2$).

Произведём разбиение на равные части вектора правых частей b : $b^1 = b(0 : \frac{n}{2} - 1)$ и $b^2 = b(\frac{n}{2} : n - 1)$ и вектора неизвестных x : $x^1 = x(0 : \frac{n}{2} - 1)$ и $x^2 = x(\frac{n}{2} : n - 1)$, где верхний индекс — номер процессора, выполняющего действия по алгоритму с данной частью вектора.

Вычисления по алгоритму включают нижеследующие этапы.

1. В течение $k-2$ шагов каждым процессором осуществляется поиск коэффициентов и правых частей новой редуцированной системы. По завершении производится обмен коэффициентами последнего уравнения, пересчитанного на данном шаге (прямой ход редукции).

2. На последнем шаге прямого хода циклической редукции после вычисления коэффициентов редуцированной системы и правых частей выполняется пересылка коэффициентов последнего уравнения от второго процессора первому.

3. Получив данные, первый процессор производит вычисления $x_0, x_{\frac{n}{2}}$ и пересылает результат второму процессору (обратный ход редукции).

4. Каждый процессор выполняет обратный ход редукции для хранящейся у него половины исходной системы без коммуникаций.

Схема работы алгоритма представлена на рис. 1. При прямом ходе (рис. 1, а)

Рис. 1. Схема работы алгоритма для двух процессоров: а — прямой, д — обратный ходы редукции, выполняемые всеми процессорами; б, в, г — пересылки данных

в конце каждого шага редукции все процессоры будут содержать коэффициенты в точности половины уравнений новой редуцированной системы. До начала следующего шага процессорам необходимо обменяться коэффициентами одного уравнения (рис. 1, б). На последнем шаге прямого хода редукции только второй процессор пересылает коэффициент дополнительного уравнения (рис. 1, в). После получения первым процессором этого коэффициента начинается обратный ход редукции. Первый шаг обратного хода выполняется первым процессором в одиночку, результат посылается второму процессору (рис. 1, г). Далее каждый процессор выполняет обратный ход редукции (рис. 1, д) без коммуникаций.

Рассмотрим теперь четырёхпроцессорный вариант алгоритма ($L = 4$). Как и ранее, процессоры могут параллельно выполнять операции, порождающие редуцированные системы. Произведём разбиение на равные части вектора правых частей b : b^1, b^2, b^3, b^4 и вектора неизвестных x : x^1, x^2, x^3, x^4 , где индекс — номер процессора, выполняющего действия по алгоритму с данной частью вектора. Вычисления по алгоритму, как и предыдущем случае, можно разделить на 4 этапа.

При прямом ходе в конце каждого шага редукции каждый процессор будет содержать коэффициенты в точности четверти от общего числа уравнений новой редуцированной системы. До начала следующего шага процессорам необходимо будет переслать друг другу коэффициенты последнего уравнения, пересчитанные на данном шаге редукции. На предпоследнем шаге прямого хода редукции второй, третий и четвёртый процессоры пересылают первому процессору коэффициенты одного уравнения. Последние два шага ре-

дукции реализуются первым процессором. После этого начинается обратный ход редукции. Первые два шага обратного хода выполняются первым процессором в одиночку, результаты посылаются остальным процессорам. Далее каждый процессор выполняет обратный ход редукции для хранящейся у него части уравнений исходной системы без коммуникаций.

Обобщим данный подход на произвольное число задач L . Произведем разбиение на L равных частей вектора правых частей $b: b^1, b^2, \dots, b^L$ и вектора неизвестных $x: x^1, x^2, \dots, x^L$, где индекс — номер процессора, выполняющего действия по алгоритму с данной частью вектора. Вычисления по алгоритму включают нижеследующие этапы.

1. В течение $k-2$ шагов каждым процессором осуществляется поиск коэффициентов и правых частей новой редуцированной системы. По завершении производится обмен коэффициентами последнего уравнения, пересчитанного на данном шаге (прямой ход редукции).

2. На предпоследнем шаге прямого хода циклической редукции после вычисления коэффициентов и правых частей редуцированной системы выполняется пересылка коэффициентов последнего уравнения первому процессору от процессоров $2, 3, \dots, n$.

3. Получив данные, первый процессор реализует последние шаги редукции.

4. В течении $\log_2 L$ шагов обратного хода первый процессор производит вычисление искомых значений в одиночку и рассылает остальным процессорам.

5. Каждый процессор выполняет обратный ход редукции для хранящейся у него части исходной системы без коммуникаций.

4. Исследование ускорения. Проведём сравнение представленного параллельного алгоритма циклической редукции для периодической краевой задачи и параллельного алгоритма метода встречных циклических прогонок для одной и той же задачи [9]. Несомненным достоинством представленного алгоритма является его масштабируемость. Данное свойство позволяет получить преимущество в скорости вычислений, несмотря на их больший объём ($O(N \log_2 N)$ и $O(N)$ — алгоритм из [9]), а также наличие простоев.

Результаты вычислительных экспериментов подтверждают это. Максимальное ускорение алгоритма из [9] для одномерной сеточной области достигает величины 1,9, в то время как ускорение предложенного алгоритма для 4-х процессоров соответствует 2,5. Недостатком синтезированного алгоритма является привязка к степени двойки, что сужает область его применения.

Вычислительные эксперименты проводились на кластере, состоящем из четырёх ЭВМ AMD Sempron™ Processor 3000+ (1000 MHz) и ОЗУ 1 Гб, связанных локальной сетью Ethernet 100 MB, и работающем под операционной системой GNU Linux (ядро 2.6.17).

На рис. 2 представлены результаты исследования ускорения параллельного алгоритма метода циклической редукции для периодической задачи в случае двух и четырех процессоров.

Двухзадачный алгоритм характеризуется максимальным значением ускорения в 1,85 раза для $(N = 67 \cdot 10^6)$, а четырёхзадачный — в 2,5 раза. В случае двухзадачного алгоритма ускорение по значению близко к максимально возможному по закону Амдала.

Рис. 2. Графики зависимости ускорений от размерности задачи ($n = 2^k$): сплошная линия — для двухмерной, штриховая — для четырёхмерных задач

Сложность процесса взаимодействия процессоров между собой привела к потерям во времени работы четырёхпроцессорного алгоритма. Для него использовалась топология коммуникаций в виде кольца, хотя на самом деле все пересылки осуществлялись через сервер.

Заключение. Применение параллельного алгоритма, основанного на методе циклической редукции, при исследовании периодических краевых задач разностным методом характеризуется высоким ускорением и эффективностью. Представляется целесообразным исследование зависимости ускорения от схемы коммуникаций и развитие данного подхода для случая двумерной сеточной области.

Работа выполнена при поддержке российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE), гранта Президента РФ поддержки ведущих научных школ (код проекта НШ-3086.2008.9) и гранта РФФИ (код проекта 07-07-00210-а).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Самарский А. А., Вабищевич П. Н. Вычислительная теплопередача. — М.: Едиториал УРСС, 2003. — 784 с.
2. Миренков Н. Н. Параллельные алгоритмы для решения задач на однородных вычислительных системах // *Вычислительные системы*, 1973. — № 57. — С. 3–32.
3. Бирюкова Л. Н., Четверушкин Б. Н. О возможности реализации квазигидродинамической модели полупроводниковой плазмы на многопроцессорных вычислительных системах // *Матем. моделирование*, 1991. — Т. 3, № 6. — С. 61–71.
4. Милукова О. Ю. Параллельный вариант обобщенного попеременно-треугольного метода для решения эллиптических уравнений // *Ж. вычисл. матем. и матем. физ.*, 1998. — Т. 38, № 12. — С. 2002–2012; англ. пер.: Milyukova O. Yu. A parallel variant of the generalized alternating triangular method for elliptic equations // *Comput. Math. Math. Phys.*, 1998. — Vol. 38, No. 12. — P. 1922–1932.
5. Головашкин Д. Л. Параллельные алгоритмы решения сеточных уравнений трехдиагонального вида, основанного на методе встречных прогонок // *Матем. моделирование*, 2005. — Т. 17, № 11. — С. 118–128.
6. Ярмушкин С. В., Головашкин Д. Л. Исследование параллельных алгоритмов решения трёхдиагональных систем линейных алгебраических уравнений // *Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки*, 2004. — № 26. — С. 78–82.
7. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений. — М.: Наука, 1978. — 561 с.
8. Ortega J. M. Introduction to Parallel and Vector Solution of Linear Systems. — New York: Plenum Press, 1988. — 318 p.; русск. пер.: Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. — М.: Мир, 1991. — 367 с.

9. Логанова Л. В. Параллельный алгоритм метода циклических прогонок // *Вестн. Сам. гос. аэрокосм. ун-та*, 2008. — № 2(15). — С. 167–174.
10. Golub G., van Loan Ch. *Matrix Computations*: 3rd edit., Johns Hopkins University Press, 1996. — 694 p.; русск. пер.: Голуб Дж., Ван Лоун Ч. Матричные вычисления. — М.: Мир, 1999. — 548 с.

Поступила в редакцию 03/X/2009;
в окончательном варианте — 03/III/2010.

MSC: 68W10

A PARALLEL ALGORITHM OF THE CYCLIC REDUCTION METHOD IN THE PERIODIC BOUNDARY-VALUE PROBLEM

L. V. Loganova, D. L. Golovashkin, O. S. Syagailo

¹ S. P. Korolyov Samara State Aerospace University,
34, Moskovskoe sh., Samara, 443086.

² Image Processing Systems Institute,
151, Molodogvardeyskaya str., Samara, 443001.

E-mails: tk@smr.ru, dimitriy@smr.ru

We report constructing a parallel algorithm based on the cyclic reduction method in the boundary-value problem. Comparison with the familiar algorithms has been made. Results of the studies into the acceleration of the algorithm are discussed. The algorithm is shown to be highly efficient.

Key words: *cyclic reduction method, parallel algorithm.*

Original article submitted 03/X/2009;
revision submitted 03/III/2010.

Liliya V. Loganova, Lecturer, Dept. of Technical Cybernetics. *Dimitrii L. Golovashkin* (Dr. Sci. (Phys. & Math.)), Leading Research Scientist, Lab. of Diffractive Optics. *Ol'ga S. Syagailo*, Student.