ISSN 1561-2430 (Print)
ISSN 2524-2415 (Online)
УДК 519.67
https://doi.org/10.29235/1561-2430-2019-55-4-425-434

Поступила в редакцию 23.08.2019
Received 23.08.2019

А. А. Згировский, Н. А. Лиходед

Белорусский государственный университет, Минск, Беларусь

МОДИФИЦИРОВАННЫЙ МЕТОД ПАРАЛЛЕЛЬНОЙ МАТРИЧНОЙ ПРОГОНКИ

Аннотация. Тематика работы относится к области построения параллельных алгоритмов численного решения блочно-трехдиагональных систем линейных алгебраических уравнений. Такие системы часто возникают в приложениях и для ряда задач требуют использования высокопроизводительных многоядерных вычислительных систем. Один из широко применяемых на практике подходов к решению блочно-трехдиагональных систем заключается в использовании оригинальных алгоритмов параллельной матричной прогонки. В настоящей статье рассмотрен метод параллельной матричной прогонки, основанный на разбиении матрицы. Этот метод трехфазный: сначала исходная система разбивается на части и после независимых преобразований каждой из них составляется редуцированная блочно-трехдиагональная система, затем из этой системы находят несколько неизвестных каждой части уравнений, после чего независимо вычисляются остальные неизвестные каждой части. Предложена новая модификация метода; обосновано, что если для исходной системы уравнений справедливы известные (и часто выполненные на практике) условия устойчивости метода матричной прогонки, то вычисления разработанной модификации параллельной матричной прогонки являются устойчивыми.

Ключевые слова: параллельные вычисления, матричная прогонка блочно-трехдиагональных линейных систем, устойчивость параллельной матричной прогонки

Для цитирования. Згировский, А. А. Модифицированный метод параллельной матричной прогонки / А. А. Згировский, Н. А. Лиходед // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2019. – Т. 55, № 4. – С. 425–434. https://doi.org/10.29235/1561-2430-2019-55-4-425-434

A. A. Zgirouski, N. A. Likhoded

Belarusian State University, Minsk, Belarus

MODIFIED METHOD OF PARALLEL MATRIX SWEEP

Abstract. The topic of this paper refers to efficient parallel solvers of block-tridiagonal linear systems of equations. Such systems occur in numerous modeling problems and require usage of high-performance multicore computation systems. One of the widely used methods for solving block-tridiagonal linear systems in parallel is the original block-tridiagonal sweep method. We consider the algorithm based on the partitioning idea. Firstly, the initial matrix is split into parts and transformations are applied to each part independently to obtain equations of a reduced block-tridiagonal system. Secondly, the reduced system is solved sequentially using the classic Thomas algorithm. Finally, all the parts are solved in parallel using the solutions of a reduced system. We propose a modification of this method. It was justified that if known stability conditions for the matrix sweep method are satisfied, then the proposed modification is stable as well.

Keywords: parallel computations, block-tridiagonal linear systems matrix sweep, stability of parallel matrix sweep

For citation. Zgirouski A. A., Likhoded N. A. Modified method of parallel matrix sweep. *Vestsi Natsyianal'nai akademii navuk Belarusi. Seryia fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2019, vol. 55, no. 4, pp. 425–434 (in Russian). https://doi.org/10.29235/1561-2430-2019-55-4-425-434

Введение. Численные методы решения уравнений в частных производных часто приводят к системам линейных алгебраических уравнений (СЛАУ) с блочно-трехдиагональными матрицами. Одним из широко применяемых на практике для решения таких систем является метод матричной прогонки [1]. Размерность систем и их блочных коэффициентов для ряда задач не позволяет эффективно решать эти задачи с использованием последовательных алгоритмов. Актуальной становится разработка параллельных алгоритмов матричной прогонки для реализации на многоядерных вычислительных системах.

Один подход к реализации матричной прогонки на параллельном компьютере – организовать параллельное выполнение операций алгоритмов перемножения матриц и решения СЛАУ (многократно используемых в матричной прогонке). Если матрицы-блоки плотные и достаточно большие, то такой параллелизм можно эффективно использовать.

Другой подход заключается в использовании специально построенных алгоритмов параллельной матричной прогонки [2–6]. Эти алгоритмы обобщают на случай блочно-трехдиагональных матриц идеи скалярных алгоритмов параллельной матричной прогонки, основанных на разбиении матрицы [7, 8] или на циклической редукции [9]. На блочный случай обобщают скалярные алгоритмы параллельной матричной прогонки, хорошо зарекомендовавшие себя на практике.

Целью настоящей работы является обобщение на блочный случай одного из таких параллельных скалярных алгоритмов [10], основанного на разбиении матрицы. Методы, основанные на разбиении матрицы, являются трехфазными. На первой фазе исходная система разбивается на части и после независимых преобразований каждой из них составляется редуцированная система (блочно-трехдиагональная в блочном случае). На второй – из редуцированной системы находят несколько неизвестных каждой части уравнений. На третьей фазе независимо вычисляются остальные неизвестные каждой части. В данной работе предложена новая модификация параллельной матричной прогонки. Обосновано, что если для исходной системы уравнений справедливы известные (и часто выполненные на практике) условия устойчивости метода матричной прогонки, то вычисления разработанной модификации параллельной матричной прогонки являются устойчивыми.

Матричная прогонка. Пусть $A_1,\dots,A_N$, $B_0,B_1,\dots, B_{N-1}$, C_0, $C_1,\dots,C_N$ – $M{\times}M$-матрицы, $Y_0,Y_1,\dots,Y_N$, $F_0,F_1,\dots,F_N$ – M-мерные векторы. Рассмотрим блочно-трехдиагональную систему линейных алгебраических уравнений вида

$$\begin{aligned}
C_0Y_0 - B_0Y_1 \qquad\qquad\qquad\qquad &= F_0,\\
-A_1Y_0 + C_1Y_1 - B_1Y_2 \qquad\qquad\qquad &= F_1,\\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots &\\
-A_iY_{i-1} + C_iY_i - B_iY_{i+1} \qquad &= F_i,\\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots &\\
-A_NY_{N-1} + C_NY_N &= F_N,
\end{aligned} \tag{1}$$

или, если записывать отдельно по уравнениям,

$$\begin{gathered}
C_0Y_0 - B_0Y_1 = F_0,\\
-A_iY_{i-1} + C_iY_i - B_iY_{i+1} = F_i, \quad i = 1,\dots, N{-}1,\\
-A_iY_{N-1} + C_NY_N = F_N.
\end{gathered} \tag{1$'$}$$

Приведем формулы матричной прогонки для решения системы (1) (см., напр., [1]):

– прямая прогонка – вычисление матриц α_i и векторов β_i по формулам

$$\begin{gathered}
\alpha_1 = C_0^{-1}B_0, \quad \beta_1 = C_0^{-1}f_0,\\
\alpha_{i+1} = (C_i - A_i\alpha_i)^{-1}B_i,\ \ \beta_{i+1} = (C_i - A_i\alpha_i)^{-1}(F_i + A_i\beta_i),\ i = 1,2,\dots,N-1,\\
\beta_{i+1} = (C_i - A_i\alpha_i)^{-1}(F_i + A_i\beta_i);
\end{gathered} \tag{2}$$

– обратная прогонка – вычисление решения (вычисление векторов Y_i) по формулам

$$Y_i = \beta_{i+1}, \quad Y_i = \alpha_{i+1}Y_{i+1} + \beta_{i+1}, \quad i = N-1,\dots,1,0. \tag{3}$$

Говорят, что алгоритм метода матричной прогонки устойчив, если выполнено условие $\|\alpha_i\| \le 1$ для $i = 1,2,\dots,N$ (предполагается, что в M-мерном пространстве введена какая-либо норма). Имеет место следующее утверждение [1]: если C_i, $i = 0,1,\dots,N,$ – невырожденные матрицы, а $A_1,\dots,A_N$, B_0, $B_1,\dots,B_{N-1}$ – ненулевые матрицы и выполнены условия

$$\left\|C_i^{-1}A_i\right\| + \left\|C_i^{-1}B_i\right\| \le 1, \quad i = 1,2,\dots,N-1, \tag{4}$$

$$\left\| C_0^{-1} B_0 \right\| \le 1, \quad \left\| C_N^{-1} A_N \right\| \le 1, \tag{5}$$

причем хотя бы в одном из условий (4), (5) выполняется строгое неравенство, то алгоритм (2), (3) метода матричной прогонки корректен и устойчив. Под корректностью алгоритма понимается существование обратных матриц в формулах (2).

Параллельная матричная прогонка, основанная на разбиении блочной трехдиагональной матрицы. Пусть дана система – вида (1). Разобьем систему, состоящую из $N + 1$ матрично-векторных уравнений, на K частей, т. е. на K блоков уравнений, где K – параметр алгоритма. Для простоты изложения будем считать, что $N + 1$ нацело делится на K:

$$\frac{N+1}{K} = l,$$

где l – целое число. В k-й блок уравнений, $k = 0,1,\ldots,K - 1$, входят l матрично-векторных уравнений, начиная с уравнения $k \cdot l$ и по уравнение $(k + 1)l - 1$ включительно. Обозначим $s_k = k \cdot l$ через номер первого уравнения в k-м блоке, $f_k = (k + 1)l - 1$ – номер последнего уравнения в k-м блоке. Заметим, что $f_k = s_k + l - 1$, $s_k - 1 = f_{k-1}$, $s_{k+1} - 1 = f_k$. Идея алгоритма заключается в том, чтобы сначала найти граничные неизвестные. Тогда, решая параллельно (независимо друг от друга) блоки уравнений, можно найти оставшиеся неизвестные.

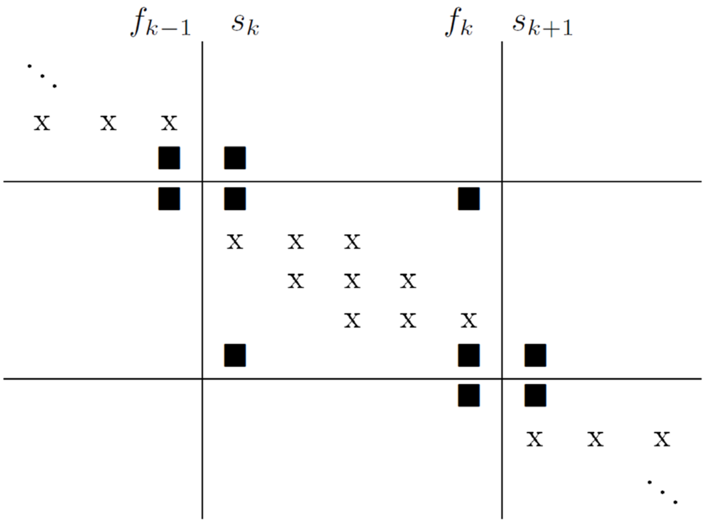
Независимо решить блоки уравнений не представляется возможным, так как первое и последнее уравнения каждого блока содержат переменные, которые входят в соседние блоки (будем называть такие переменные граничными). Известное решение проблемы состоит в том, чтобы считать граничные неизвестные параметрами системы одного блока. А далее в каждом блоке параллельно применить метод матричной прогонки для системы с параметрами. Тогда можно выразить неизвестные исходные системы через значения граничных переменных. Кроме того, можно получить систему блочно-трехдиагональных уравнений относительно граничных неизвестных. Тогда, решив эту систему (последовательно или параллельно), можно параллельно и независимо найти значения неизвестных в каждом блоке.

Таким образом, алгоритм параллельной матричной прогонки состоит из трех фаз. В первой фазе для каждого из K блоков вычисляются коэффициенты матричной прогонки, полагая переменные Y_{s_k} параметрическими. В результате прогонки для каждого неизвестного внутри блока получается выражение относительно граничных неизвестных, а также уравнение, относительно самих граничных неизвестных. Эти уравнения образуют редуцированную блочно-трехдиагональную систему. Во второй фазе указанная система решается последовательным алгоритмом матричной прогонки. После чего в третьей фазе, используя найденные в первой фазе неизвестные и коэффициенты прогонки, независимо вычисляются неизвестные каждого блока.

Модифицированная параллельная матричная прогонка. Отличием предлагаемой модифицированной версии от известного параллельного алгоритма матричной прогонки является отсутствие необходимости хранить после первой фазы для каждого из K блоков новые блочные коэффициенты прогонки для каждого неизвестного.

Пусть произведено разбиение системы вида (1) на K блоков уравнений. В каждом из блоков, кроме нулевого, в первом уравнении есть ненулевые, вообще говоря, коэффициенты при неизвестных с индексами $s_k - 1$ и s_k; эти же неизвестные входят также в уравнения предыдущего блока. Кроме того, в каждом блоке, кроме (K–1)-го, в последнем уравнении есть ненулевые коэффициенты при неизвестных с индексами $s_k + l - 1$ и $s_k + l$, которые входят также в уравнения следующего блока. Это не позволяет решать блоки уравнений независимо.

Неизвестные с индексами $s_k - 1$, s_k и $s_k + l - 1$, $s_k + l$ являются граничными. Всего имеется $2K$ граничных неизвестных. Рассмотрим способ получения редуцированной системы, в которую войдут $2K$ уравнений относительно граничных неизвестных. Для этого в каждом блоке первое уравнение относительно неизвестных с индексами $s_k - 1$, s_k, $s_k + 1$ сведем строчными преобразованиями к уравнению относительно граничных неизвестных с индексами $s_k - 1$, s_k, $s_k + l - 1$,



Схематичное изображение ненулевых блоков матрицы системы уравнений после получения верхних и нижних уравнений; указаны индексы граничных неизвестных k-го блока уравнений

A schematic representation of non-zero blocks of the matrix of the system of equations after obtaining the lower and upper equations; the indexes of the boundary variables of the kth block are indicated

а последнее уравнение относительно неизвестных с индексами $s_k + l - 2$, $s_k + l - 1$, $s_k + l$ сведем к уравнению относительно граничных неизвестных s_k, $s_k + l - 1$, $s_k + l$.

Полученное после преобразований уравнение относительно неизвестных с индексами $s_k - 1$, s_k, $s_k + l - 1$ назовем верхним; уравнение относительно неизвестных s_k, $s_k + l - 1$, $s_k + l$ назовем нижним. На рисунке указаны коэффициенты при граничных неизвестных верхних и нижних уравнений.

Полученная редуцированная система является трехдиагональной. В самом деле, обозначим $Z_{2k-1} = Y_{kl-1}$, $Z_{2k} = Y_{kl}$. Тогда, с учетом $s_k = k \cdot l$, верхние уравнения блоков относительно неизвестных Y_{sk-1}, Y_{s_k}, Y_{s_k+l-1} станут уравнениями относительно Z_{2k-1}, Z_{2k}, Z_{2k+1}, а нижние уравнения относительно Y_{s_k}, Y_{s_k+l-1}, Y_{s_k+l} – уравнениями относительно Z_{2k}, Z_{2k+1}, Z_{2k+2}.

Отметим, что после подстановки граничных неизвестных каждый блок уравнений становится независимой трехдиагональной системой $l - 2$ уравнений относительно $l - 2$ неизвестных.

Рассмотрим подробно первую фазу. Сначала исследуем процесс получения нижних уравнений в k-м блоке уравнений. Текущие коэффициенты-блоки будем хранить в массивах L^A, L^C, L^B, L^F. Получим уравнение относительно неизвестных с индексами s_k, $s_k + l - 1$, $s_k + l$. Для этого понадобится $l - 2$ шага: преобразовать потребуется все, начиная с третьего уравнения блока. Пронумеруем эти шаги числами с $s_k + 2$ по f_k (т. е. с $k \cdot l + 2$ по $k \cdot l + l - 1$).

Второе уравнение в блоке имеет ненулевые коэффициенты при неизвестных с индексами s_k, $s_k + 1$, $s_k + 2$. Оно не преобразуется, но в преобразованиях участвует. На шаге r, $r = s_k + 2,\ldots,f_k$, будем преобразовывать уравнение относительно неизвестных с индексами s_k, r, $r + 1$. Запишем коэффициенты-блоки уже преобразованного на предыдущем шаге и этого уравнения:

$$\left.\begin{array}{cccccccc} L^A & 0 & \ldots & 0 & L^C & L^B & 0 & \ldots \\ 0 & 0 & \ldots & 0 & -A_r & C_r & -B_r & \ldots \end{array}\right| \begin{array}{c} L^F \\ F_r \end{array}.$$

При начальном значении $r = s_k + 2$ записываются второе и третье уравнения в блоке, причем

$$L^A = -A_{sk+1},\ \ L^C = C_{sk+1},\ \ L^B = -B_{sk+1},\ \ L^F = F_{sk+1}.$$

Выполним преобразования, приводящие во втором уравнении к неизвестным с индексами s_k, $r + 1$, $r + 2$. Сначала умножим первую строку слева на $A_r(L^C)^{-1}$:

$$\left.\cdots\begin{array}{cccccccc} A_r(L^C)^{-1}L^A & 0 & \dots & 0 & A_r & A_r(L^C)^{-1}L^B & 0 \\ 0 & 0 & \dots & 0 & -A_r & C_r & -B_r \end{array}\right|\begin{array}{c} A_r(L^C)^{-1}L^F \\ F_r \end{array}.$$

Затем прибавим ко второй строке первую строку:

$$\left.\cdots\begin{array}{cccccccc} A_r(L^C)^{-1}L^A & 0 & \dots & 0 & A_r & A_r(L^C)^{-1}L^B & 0 \\ A_r(L^C)^{-1}L^A & 0 & \dots & 0 & 0 & C_r + A_r(L^C)^{-1}L^B & -B_r \end{array}\right|\begin{array}{c} A_r(L^C)^{-1}L^F \\ F_r + A_r(L^C)^{-1}L^F \end{array}.$$

Второе уравнение имеет ненулевые коэффициенты при неизвестных с индексами s_k, $r+1$, $r+2$.

После шага $r = f_k$ последнее уравнение в блоке становится уравнением относительно неизвестных с индексами s_k, $s_k + l - 1$, $s_k + l$.

Теперь рассмотрим процесс получения верхних уравнений в k-м блоке. Текущие коэффициенты-блоки будем хранить в массивах U^A, U^C, U^B, U^F. Для получения уравнения относительно неизвестных с индексами $s_k - 1$, s_k, $s_k + l - 1$ нужно $l - 2$ шага: преобразовать потребуется все, начиная с третьего снизу и заканчивая первым уравнением блока.

Предпоследнее уравнение в блоке имеет ненулевые коэффициенты при неизвестных с индексами $s_k + l - 3$, $s_k + l - 2$, $s_k + l - 1$. Оно не преобразуется, но в преобразованиях участвует. На шаге r, $r = f_k - 2, f_k - 3, \ldots, s_k$, имеется уравнение относительно неизвестных с индексами $r - 1$, r, $s_k + l - 1$. Запишем коэффициенты-блоки уже преобразованного на предыдущем шаге и этого уравнения:

$$\cdots\left.\begin{array}{ccccccc} -A_r & C_r & -B_r & 0 & \dots & 0 & 0 \\ 0 & U^A & U^C & 0 & \dots & 0 & U^B \end{array}\right|\begin{array}{c} F_r \\ U^F \end{array}.$$

При начальном значении $r = f_k - 1$ записываются предпоследнее и третье снизу уравнения в блоке, в которых полагается

$$U^A = -A_{f_k-1},\ \ U^C = C_{f_k-1},\ \ U^B = -B_{f_k-1},\ \ U^F = F_{f_k-1}.$$

Выполним преобразования, приводящие в первом уравнении к неизвестным с индексами $r - 2$, $r - 1$, $s_k + l - 1$. Умножим вторую строку слева на $B_r(U^C)^{-1}$, а затем прибавим к первой строке вторую:

$$\cdots\left.\begin{array}{ccccccc} -A_r & C_r + B_rB_r(U^C)^{-1}U^A & 0 & 0 & \dots & 0 & B_rB_r(U^C)^{-1}U^B \\ 0 & B_rB_r(U^C)^{-1}U^A & B_r & 0 & \dots & 0 & B_rB_r(U^C)^{-1}U^B \end{array}\right|\begin{array}{c} F_r + B_rB_r(U^C)^{-1}U^F \\ B_rB_r(U^C)^{-1}U^F \end{array}.$$

Первое уравнение имеет ненулевые коэффициенты при неизвестных с индексами $r - 2$, $r - 1$, $s_k + l - 1$.

После всех преобразований в k-м блоке получены 2 уравнения для редуцированной системы:

$$U^A Z_{2k-1} + U^C Z_{2k} + U^B Z_{2k+1} = U^F,$$

$$L^A Z_{2k} + L^C Z_{2k+1} + L^B Z_{2k+2} = L^F.$$

Здесь для нулевого блока отсутствует слагаемое с U^A, для $(K–1)$-го блока отсутствует слагаемое с L^B, для всех блоков $L^B = -B_{f_k}$, $U^A = -A_{s_k}$.

Таким образом, получение коэффициентов редуцированной системы уравнений можно представить следующим алгоритмом:

dopar $k = 0, K - 1$

  $s_k = k \cdot l$, $f_k = (k+1)l - 1$

  \\ Получение коэффициентов нижних уравнений:

$L^A = -A_{s_k+1},\ L^C = C_{s_k+1},\ L^B = -B_{s_k+1},\ L^F = F_{s_k+1}$
do $r = s_k + 2, f_k$
$T^A = A_r (L^C)^{-1}$
$L^A = T^A L^A$
$L^C = C_r + T^A L^B$
$L^B = -B_r$ \\ L^B – нулевая матрица при $r = K \cdot l - 1$
$L^F = F_r + T^A L^F$
enddo
$L_k^A = L^A,\ L_k^C = L^C,\ L_k^B = L^B,\ L_k^F = L^F$
\\ Получение коэффициентов верхних уравнений:
$U^A = -A_{f_k-1},\ U^C = C_{f_k-1},\ U^B = -B_{f_k-1},\ U^F = F_{f_k-1}$
do $r = f_k - 2, s_k$ \\ Цикл с шагом –1
$T^B = B_r (U^C)^{-1}$
$U^A = -A_r$ \\ U^A – нулевая матрица при $r = 0$
$U^C = C_r + T^B U_r^A$
$U^B = T^B U^B$
$U^F = F_r + T^B U^F$
enddo
$U_k^A = U^A,\ U_k^B = U^B,\ U_k^C = U^C,\ U_k^F = U^F$
enddopar

Заметим, что в силу блочно-трехдиагональной структуры исходной системы первый блочный коэффициент первого уравнения A_0 и последний блочный коэффициент последнего уравнения B_N отсутствуют. Аналогично для редуцированной системы отсутствуют первый и последний блочные коэффициенты U_0^A и L_{K-1}^B. При программной реализации удобно хранить произвольные, например нулевые, значения.

На этапе второй фазы решается полученная относительно $Z_0, Z_1, \ldots, Z_{2K-1}$ блочно-трехдиагональная система алгоритмом правой матричной прогонки.

На третьей фазе алгоритма для каждого блока уравнений k, $k = 0,1,\ldots,K-1$, после второй фазы найдены граничные неизвестные:

$$Y_{s_k-1} = Z_{2k-1},\ \ Y_{s_k} = Z_{2k},\ \ Y_{s_k+l-1} = Z_{2k+1},\ \ Y_{s_k+l} = Z_{2k+2}.$$

Эти неизвестные входят в первое, второе, предпоследнее и последнее уравнения каждого блока.

Из первого уравнения можно найти неизвестный вектор Y_{s_k+1}, так как уже найдены неизвестные Y_{s_k-1} и Y_{s_k}.

Перенесем во втором уравнении блока (это уравнение с номером $s_k + 1$ исходной системы) известное произведение $(-A_{s_k} Y_{s_k+1})$ в правую часть. Перенесем также в предпоследнем уравнении блока (это уравнение с номером $f_k - 1$) в правую часть известное произведение $(-B_{f_k-1} Y_{f_k})$. Эту систему относительно неизвестных $Y_{s_k+1}, Y_{s_k+2}, \ldots, Y_{s_k+l-2}$ $(Y_{s_k+l-2} = Y_{f_k-1})$ можно решить независимо для каждого из блоков уравнений алгоритмом правой матричной прогонки.

Устойчивость модифицированной параллельной матричной прогонки. На второй и третьей фазах алгоритма выполняется правая матричная прогонка. Если для исходной блочно-трехдиагональной системы выполнены условия устойчивости метода матричной прогонки (4), (5), то они выполняются и для каждой из систем в третьей фазе, так как коэффициенты систем есть коэффициенты исходной системы. Оказывается, аналогичное утверждение справедливо и для второй фазы алгоритма. Предполагается неравенство нулю матриц, встречающихся при выполнении первой фазы алгоритма и, где необходимо, существование обратных матриц.

Т е о р е м а. *Если для исходной блочно-трехдиагональной системы выполнены условия устойчивости метода матричной прогонки* (4), (5), *то они выполняются и для редуцированной системы.*

Д о к а з а т е л ь с т в о. Зафиксируем любой блок уравнений и применим метод математической индукции. Сначала рассмотрим процесс получения коэффициентов нижнего редуцированного уравнения.

Пусть на какой-либо итерации первой фазы алгоритма коэффициенты L^A, L^C, L^B (в итоге входящие в редуцированную систему) удовлетворяют условию

$$\left\|(L^C)^{-1}L^A\right\|+\left\|(L^C)^{-1}L^B\right\|\le 1, \tag{6}$$

причем хотя бы для одного из блоков неравенство (6) является строгим. Для (K – 1)-го блока отсутствует слагаемое с L^B. Для начальной итерации неравенство (6) справедливо, так как в этом случае L^A, L^C, L^B есть коэффициенты исходной системы.

На каждой из $f_k - s_k - 1$ итераций r происходит вычисление новых значений коэффициентов L^A, L^C, L^B по формулам

$$\overline{L}^A = A_r(L^C)^{-1}L^A,\ \ \overline{L}^C = C_r + A_r(L^C)^{-1}L^B,\ \ \overline{L}^B = -B_r.$$

Требуется показать справедливость неравенства (строгого хотя бы для одного из блоков)

$$\left\|(\overline{L}^C)^{-1}\overline{L}^A\right\|+\left\|(\overline{L}^C)^{-1}\overline{L}^B\right\|\le 1.$$

Имеем

$$\left\|(\overline{L}^C)^{-1}\overline{L}^A\right\|+\left\|(\overline{L}^C)^{-1}\overline{L}^B\right\|=\left\|\left(C_r + A_r(L^C)^{-1}L^B\right)^{-1}A_r(L^C)^{-1}L^A\right\|+\left\|\left(C_r + A_r(L^C)^{-1}L^B\right)^{-1}B_r\right\|=$$

$$=\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}C_r^{-1}A_r(L^C)^{-1}L^A\right\|+\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}C_r^{-1}B_r\right\|\le$$

$$\le\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\|\left\|C_r^{-1}A_r(L^C)^{-1}L^A\right\|+\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\|\left\|C_r^{-1}B_r\right\|=$$

$$=\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\|\left(\left\|C_r^{-1}A_r(L^C)^{-1}L^A\right\|+\left\|C_r^{-1}B_r\right\|\right),$$

где E – единичная матрица. Таким образом,

$$\left\|(\overline{L}^C)^{-1}\overline{L}^A\right\|+\left\|(\overline{L}^C)^{-1}\overline{L}^B\right\|\le\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\|\left(\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^A\right\|+\left\|C_r^{-1}B_r\right\|\right). \tag{7}$$

Воспользуемся известным утверждением: если для квадратной матрицы S имеет место оценка $\|S\| < 1$, то существует обратная к $E - S$ матрица, причем $\|(E - S)^{-1}\| \le 1/(1 - \|S\|)$. Положим

$$S = -C_r^{-1}A_r(L^C)^{-1}L^B.$$

Тогда с учетом неравенств (4), (6) и неравенства нулю матрицы $(L^C)^{-1}L^A$ получим

$$\|S\| = \left\|-C_r^{-1}A_r(L^C)^{-1}L^B\right\|\le\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^B\right\|\le\left\|C_r^{-1}A_r\right\|\left(1-\left\|(L^C)^{-1}L^A\right\|\right)<1,$$

$$\left\|\left(E-S\right)^{-1}\right\|=\left\|\left(E + C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\|\le\frac{1}{1-\|S\|}\le\frac{1}{1-\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^B\right\|}\le$$

$$\leq \frac{1}{1-\left\|C_r^{-1}A_r\right\|\left(1-\left\|(L^C)^{-1}L^A\right\|\right)},$$

причем последнее неравенство строгое, если строгим является неравенство (6). Следовательно,

$$\left\|\left(E+C_r^{-1}A_r(L^C)^{-1}L^B\right)^{-1}\right\| \leq \frac{1}{1-\left\|C_r^{-1}A_r\right\|+\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^A\right\|} \leq \frac{1}{\left\|C_r^{-1}B_r\right\|+\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^A\right\|}.$$

Вернемся к неравенству (7):

$$\left\|(\overline{L}^C)^{-1}\overline{L}^A\right\|+\left\|(\overline{L}^C)^{-1}\overline{L}^B\right\| \leq \frac{1}{\left\|C_r^{-1}B_r\right\|+\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^A\right\|}\left(\left\|C_r^{-1}A_r\right\|\left\|(L^C)^{-1}L^A\right\|+\left\|C_r^{-1}B_r\right\|\right)=1.$$

Здесь неравенство строгое, если строгим является неравенство (6).

По аналогии со случаем получения коэффициентов нижнего редуцированного уравнения рассмотрим основные этапы доказательства для случая коэффициентов верхнего уравнения. Пусть справедливо неравенство

$$\left\|(U^C)^{-1}U^A\right\|+\left\|(U^C)^{-1}U^B\right\| \leq 1$$

и новые значения коэффициентов вычисляются по формулам

$$\overline{U}^A=-A_r,\ \ \overline{U}^C=C_r+B_r(U^C)^{-1}U^A,\ \ \overline{U}^B=B_r(U^C)^{-1}U^B.$$

Имеем:

$$\left\|(U^C)^{-1}U^A\right\|+\left\|(U^C)^{-1}U^B\right\|=\left\|\left(C_r+B_r(U^C)^{-1}U^A\right)^{-1}A_r\right\|+\left\|\left(C_r+B_r(U^C)^{-1}U^A\right)^{-1}B_r(U^C)^{-1}U^B\right\| \leq$$

$$\leq \left\|\left(E+C_r^{-1}B_r(U^C)^{-1}U^A\right)^{-1}\right\|\left\|C_r^{-1}A_r\right\|+\left\|\left(E+C_r^{-1}B_r(U^C)^{-1}U^A\right)^{-1}\right\|\left\|C_r^{-1}B_r(U^C)^{-1}U^B\right\| \leq$$

$$\leq \left\|\left(E+C_r^{-1}B_r(U^C)^{-1}U^A\right)^{-1}\right\|\left(\left\|C_r^{-1}A_r\right\|+\left\|C_r^{-1}B_r\right\|\left\|(U^C)^{-1}U^B\right\|\right).$$

Положим $S=-C_r^{-1}B_r(U^C)^{-1}U^A,$ тогда

$$\|S\| \leq \left\|C_r^{-1}B_r\right\|\left\|(U^C)^{-1}U^A\right\| \leq \left\|C_r^{-1}B_r\right\|\left(1-\left\|(U^C)^{-1}U^B\right\|\right) \leq 1,$$

$$\left\|(E-S)^{-1}\right\|=\left\|\left(E+C_r^{-1}B_r(U^C)^{-1}U^A\right)^{-1}\right\| \leq \frac{1}{1-\|S\|} \leq \frac{1}{1-\left\|C_r^{-1}B_r\right\|\left(1-\left\|(U^C)^{-1}U^B\right\|\right)} \leq$$

$$\leq \frac{1}{\left\|C_r^{-1}A_r\right\|+\left\|C_r^{-1}B_r\right\|\left\|(U^C)^{-1}U^B\right\|},$$

$$\left\|(U^C)^{-1}U^A\right\|+\left\|\left\|(U^C)^{-1}U^A\right\|^{-1}U^B\right\| \leq \frac{1}{\left\|C_r^{-1}A_r\right\|+\left\|C_r^{-1}B_r\right\|\left\|(U^C)^{-1}U^B\right\|}\times$$

$$\times\left(\left\|C_r^{-1}A_r\right\|+\left\|C_r^{-1}B_r\right\|\left\|(U^C)^{-1}U^B\right\|\right)=1.$$

Теорема доказана.

На первой фазе алгоритма выполняются вычисления типа исключений Гаусса, в которых в качестве «ведущих элементов» берутся матрицы U^C, L^C. Вследствие этого для устойчивости вычислений первой фазы важно, чтобы норма матрицы L^C превосходила нормы матриц L^A и L^B, а норма матрицы U^C превосходила нормы матриц U^A и U^B. Справедливо следующее утверждение.

Следствие. *Если для исходной блочно-трехдиагональной системы выполнены условия устойчивости метода матричной прогонки* (4), (5)*, то на любой итерации r первой фазы имеет место*

$$\left\|U^C\right\| \geq \left\|U^A\right\| + \left\|U^B\right\|, \quad \left\|L^C\right\| \geq \left\|L^A\right\| + \left\|L^B\right\|.$$

Действительно, рассмотрим, например, случай нижних уравнений. Так как

$$\left\|(L^C)^{-1} L^A\right\| + \left\|(L^C)^{-1} L^B\right\| \leq 1,$$

то после умножения левой и правой частей этого неравенства на $\|L^C\|$ и использования свойства мультипликативности матричной нормы ($\|A \times B\| \leq \|A\| \times \|B\|$) получим

$$\left\|L^C\right\|\left\|(L^C)^{-1} L^A\right\| + \left\|L^C\right\|\left\|\left(L^C\right)^{-1} L^B\right\| \leq \left\|L^C\right\|,$$

$$\left\|L^C (L^C)^{-1} L^A\right\| + \left\|L^C (L^C)^{-1} L^B\right\| \leq \left\|L^C\right\|,$$

$$\left\|L^A\right\| + \left\|L^B\right\| \leq \left\|L^C\right\|.$$

Таким образом, нами разработан и исследован с точки зрения устойчивости новый вариант метода параллельной матричной прогонки для решения систем линейных алгебраических уравнений с блочно-трехдиагональными матрицами.

Благодарности. Работа выполнена в рамках государственной программы научных исследований Республики Беларусь «Конвергенция-2020», подпрограмма «Методы математического моделирования сложных систем».

Acknowledgments. The prepared study was sponsored by the Government Program of Scientific Research of the Republic of Belarus "Convergence-2020", the subprogram "Methods of Mathematical Modeling of Complex Systems".

Список использованных источников

1. Самарский, А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. – М.: Наука, 1978. – 592 с.

2. Heller, D. Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems / D. Heller // SIAM J. Numer. Anal. – 1976. – Vol. 13, №. 4. – P. 484–496. https://doi.org/10.1137/0713042

3. Акимова, Е. Н. Распараллеливание алгоритма матричной прогонки / Е. Н. Акимова // Математическое моделирование. – 1994. – Т. 6, № 9. – С. 61–67.

4. Акимова, Е. Н. Параллельные алгоритмы решения СЛАУ с блочно-трехдиагональными матрицами на многопроцессорных вычислителях / Е. Н. Акимова, Д. В. Белоусов // Вестн. УГАТУ. – 2011. – Т. 15, № 5. – С. 87–93.

5. BCYCLIC: A parallel block tridiagonal matrix cyclic solver / S. P. Hirshman [et al.] // J. Comput. Phys. – 2010. – Vol. 229, №. 18. – P. 6392–6404. https://doi.org/10.1016/j.jcp.2010.04.049

6. Davina, A. Lamas. MPI-CUDA parallel linear solvers for block-tridiagonal matrices in the context of SLEPc'seigensolvers/ A. Lamas Davina, J. E. Roman // Parallel Comput. – 2018. – Vol. 74. – P. 118–135. https://doi.org/10.1016/j.parco.2017.11.006

7. Об организации параллельных вычислений и «распараллеливании» прогонки / Н. Н. Яненко [и др.] // Численные методы механики сплошной среды. – 1978. – Т. 9, № 7. – С. 139–146.

8. Wang, H. H. A parallel method for tridiagonal equations / H. H. Wang / ACM Trans. Math.Software. – 1981. – Vol. 7, № 2. – P. 170–183. https://doi.org/10.1145/355945.355947

9. Buzbee, B. L. On direct methods for solving Poisson's equations / B. L. Buzbee, G. H. Golub, C. W. Nielson // SIAM J. Numer. Anal. – 1970. – Vol. 7, №. 4. – P. 627–656. https://doi.org/10.1137/0707049

10. Austin, T. M. A memory efficient parallel tridiagonal solver: Preprint LA-VR-03-4149 / T. M. Austin, M. Berndt, J. D. Moulton. – 2004. – 13 p.

References

1. Samarskii A. A., Nikolaev E. S. *Numerical Methods for Grid Equations. Vol. 1. Direct Methods*. Birkhauser Verlag, 1989. 242 p.
2. Heller D. Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems. *SIAM Journal on Numerical Analysis*. 1976, vol. 13, no. 4, pp. 484–496. https://doi.org/10.1137/0713042
3. Akimova E. N. Parallelization of the matrix sweep algorithm. *Matematicheskoe modelirovanie = Mathematical Models*, 1994, vol. 6, no. 9, pp. 61–67 (in Russian).
4. Akimova E. N, Belousov D. V. Parallel algorithms for solving the systems of equations with block-three-diagonal matrices on multiprocessors computer systems. *Vestnik UGATU* [Scientific Journal of Ufa State Aviation Technical University], 2011 vol. 15, no. 5, pp. 87–93 (in Russian).
5. Hirshman S. P., Perumalla K. S., Lynch V. E., Sanchez R. BCYCLIC: A parallel block tridiagonal matrix cyclic solver. *Journal of Computational Physics*, 2010, vol. 229, no. 18, pp. 6392–6404. https://doi.org/10.1016/j.jcp.2010.04.049
6. Davina A. Lamas, Roman J. E. MPI-CUDA parallel linear solvers for block-tridiagonal matrices in the context of SLEPc'seigensolvers. *Parallel Computing*, 2018, vol. 74, pp. 118–135. https://doi.org/10.1016/j.parco.2017.11.006
7. Yanenko N. N., Konovalov A. N., Bugrov A. N., Shustov G. V. Organization of Parallel Computing and the Thomas Algorithm Parallelization. *Chislennye metody mekhaniki sploshnoi sredy* [Numerical Methods in Continuum Mechanics], 1978, vol. 9, no. 7, pp. 139–146 (in Russian).
8. Wang H. H. A parallel method for tridiagonal equations. *ACM Transactions on Mathematical Software*, 1981, vol. 7, no. 2, pp. 170–183. https://doi.org/10.1145/355945.355947
9. Buzbee B. L., Golub G. H., Nielson C. W. On direct methods for solving Poisson's equations. *SIAM Journal on Numerical Analysis*, 1970, vol. 7, no. 4, pp. 627–656. https://doi.org/10.1137/0707049
10. Austin T. M., Berndt M., Moulton J. D. *A memory efficient parallel tridiagonal solver.* Preprint LA-VR-03-4149, 2004. 13 p.

Информация об авторах

Згировский Андрей Александрович – магистрант факультета прикладной математики и информатики, Белорусский государственный университет (пр. Независимости, 4, 220030, г. Минск, Республика Беларусь). E-mail: zgirovskya@gmail.com

Лиходед Николай Александрович – доктор физико-математических наук, профессор кафедры вычислительной математики факультета прикладной математики и информатики, Белорусский государственный университет (пр. Независимости, 4, 220030, г. Минск, Республика Беларусь). E-mail: likhoded@bsu.by

Information about the authors

Andrei A. Zgirouski – Undergraduate, Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus). E-mail: zgirovskya@gmail.com

Nikolai A. Likhoded – Dr. Sc. (Physics and Mathematics), Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus). E-mail: likhoded@bsu.by