IJMMS 2004:25, 1315–1327
PII. S0161171204202265
http://ijmms.hindawi.com
© Hindawi Publishing Corp.

A PARALLEL CHOLESKY ALGORITHM FOR THE SOLUTION OF SYMMETRIC LINEAR SYSTEMS

R. R. KHAZAL and M. M. CHAWLA

Received 21 February 2002

For the solution of symmetric linear systems, the classical Cholesky method has proved to be difficult to parallelize. In the present paper, we first describe an elimination variant of Cholesky method to produce a lower triangular matrix which reduces the coefficient matrix of the system to an identity matrix. Then, this elimination method is combined with the partitioning method to obtain a parallel Cholesky algorithm. The total serial arithmetical operations count for the parallel algorithm is of the same order as that for the serial Cholesky method. The present parallel algorithm could thus perform with *efficiency* close to 1 if implemented on a multiprocessor machine. We also discuss the *existence* of the parallel algorithm; it is shown that for a symmetric and positive definite system, the presented parallel Cholesky algorithm is well defined and will run to completion.

2000 Mathematics Subject Classification: 65F30, 15A06.

1. Introduction. A very important class of structured systems most frequently occurring is the class of symmetric and positive definite linear systems

$$A\mathbf{x}=\mathbf{d},\quad A=(a_{i,j})_{i,j=1}^{N},\ \mathbf{x}=(x_1,\dots,x_N)^T,\ \mathbf{d}=(d_1,\dots,d_N)^T, \tag{1.1}$$

where $A=A^T$ and $\mathbf{x}^TA\mathbf{x}>0$ for every nonzero $\mathbf{x}\in\mathbb{R}^N$. It is a basic tenet of numerical analysis that the structure should be exploited whenever solving such a problem.

A widely used method for the solution of symmetric and positive definite linear systems (1.1) is based on the following theorem due to Cholesky; see Fox [4, 5]. If A is symmetric and positive definite, then there exists a real nonsingular lower triangular matrix L such that $LL^T=A$; further, if the diagonal elements of L are taken to be positive, then the decomposition is unique.

For *serial* computers, algorithms based on the Cholesky factorization method can be found, for example, in Golub and Van Loan [11]; and an excellent summary of the Cholesky method is given by George and Liu in [10].

There has been a number of the Cholesky factorization implementations reported for various real and assumed *parallel* systems based on several of the so-called “ijk forms of Cholesky”; see, for example, Cosnard et al. [2], Geist and Heath [8], George et al. [9], Heath [12], Ortega and Romine [14]. Details of these implementations of the Cholesky factorization method can also be found in Ortega [15]; reviews of these methods are included in Dongarra et al. [3] and Gallivan et al. [7].

Parallelization of a code of ijk form has been considered in detail by Freeman and Phillips [6]. As noted there, while the two inner loops offer most of the scope for parallelization, there is a limited scope only for parallelism in the outer loop. Note that the outer loop represents the *stages* of the Cholesky factorization.

Parallelization of the outer loop is necessary for an algorithm of this form to be suitable for multiple-instruction multiple-data (MIMD) parallel processing; it is parallelization of the outer loop that concerns us in the present paper.

As has been noted above, for the coefficient matrix in the linear system (1.1), classical the Cholesky method produces a factorization $A = LL^T$, with L being lower triangular. But the Cholesky method has proved to be difficult to parallelize. In the present paper, we first present an elimination variant of the Cholesky method which is equivalent to producing a lower triangular L such that $LAL^T = I$, where I is identity matrix. Then, this elimination variant of the Cholesky method is combined with the partitioning method to obtain a parallel Cholesky algorithm for the solution of symmetric and positive definite linear systems (1.1). For a given system of size N, partitioned into r^2 blocks ($r \ll N$), the total serial arithmetical operations count for the present parallel algorithm is $O((1/3)N^3)$, which is of the same order as that for the serial Cholesky method. The present parallel algorithm could thus perform with *efficiency* (see Kuck [13]) close to 1 if implemented on an r-processor machine. We also address the question of *existence* of the parallel algorithm. It is shown that if A is symmetric and positive definite, then the presented parallel Cholesky algorithm is well defined and will run to completion.

Chawla [1] has described a parallel elimination algorithm for *general dense* linear systems; the present parallel Cholesky algorithm for symmetric and positive definite linear systems is in the same vein.

2. An elimination variant of the serial Cholesky algorithm.

We first describe an elimination variant of the Cholesky factorization method. Suppose that eliminations have been carried out in A in columns 1 to $k-1$ below the main diagonal and in rows 1 to $k-1$ to the right of the main diagonal, with the reduced system assuming the form

$$A^{(k-1)}\mathbf{x}^{(k-1)} = \mathbf{d}^{(k-1)}, \tag{2.1}$$

where

$$A^{(k-1)} = \left[\begin{array}{ccc} 1\cdots k-1 & k & k+1\cdots N \\ I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{k,k}^{(k-1)} & \mathbf{c}^{(k-1)^T} \\ \mathbf{0} & \mathbf{c}^{(k-1)} & B^{(k-1)} \end{array}\right], \tag{2.2}$$

where I denotes the identity matrix.

To describe stage $k \in \{1,\ldots,N\}$ of the reduction process, we define

$$L^{(k)} = \left[\begin{array}{ccc} 1\cdots k-1 & k & k+1\cdots N \\ I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda^{(k)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\ell}^{(k)} & I \end{array}\right]. \tag{2.3}$$

Then, the stage k consists of a *symmetric reduction* of (2.1), with the help of $L^{(k)}$, into

$$A^{(k)}\mathbf{x}^{(k)} = \mathbf{d}^{(k)}, \tag{2.4}$$

where

$$A^{(k)} = L^{(k)}A^{(k-1)}L^{(k)^T}, \qquad \mathbf{x}^{(k-1)} = L^{(k)^T}\mathbf{x}^{(k)}, \qquad \mathbf{d}^{(k)} = L^{(k)}\mathbf{d}^{(k-1)}. \tag{2.5}$$

The conditions for the reduction of $A^{(k-1)}$ to $A^{(k)}$ are

$$\begin{aligned} a_{k,k}^{(k-1)}\lambda^{(k)^2} &= 1, \\ a_{k,k}^{(k-1)}\boldsymbol{\ell}^{(k)} + \mathbf{c}^{(k-1)} &= \mathbf{0}, \\ B^{(k)} &= B^{(k-1)} + \mathbf{c}^{(k-1)}\boldsymbol{\ell}^{(k)^T}. \end{aligned} \tag{2.6}$$

The right-hand side is modified as

$$\begin{aligned} d_k^{(k)} &= \lambda^{(k)}d_k^{(k-1)}, \\ d_{k+1\to N}^{(k)} &= d_{k+1\to N}^{(k-1)} + d_k^{(k-1)}\boldsymbol{\ell}^{(k)}, \end{aligned} \tag{2.7}$$

while only one component of the solution vector gets modified as

$$x_k^{(k-1)} = \lambda^{(k)}x_k^{(k)} + \boldsymbol{\ell}^{(k)^T}x_{k+1\to N}^{(k)}. \tag{2.8}$$

The resulting algorithm is described as follows.

2.1. Elimination-variant serial Cholesky algorithm

Step 1 (elimination stage (compact form)). For $k = 1(1)N$, compute

$$\lambda^{(k)} = \sqrt{\frac{1}{a_{k,k}^{(k-1)}}}, \qquad \boldsymbol{\ell}^{(k)} = -\frac{1}{a_{k,k}^{(k-1)}}a_{k+1\to N,k}^{(k-1)}, \tag{2.9}$$

and update

$$A_{k+1\to N,k+1\to N}^{(k)} = A_{k+1\to N,k+1\to N}^{(k-1)} + a_{k+1\to N}^{(k-1)}\boldsymbol{\ell}^{(k)^T}, \tag{2.10}$$

(only the lower triangular part needs to be computed) and (2.7).

Step 2 (solution stage). Set $\mathbf{x}^{(N)} = \mathbf{d}^{(N)}$. For $k = N(-1)1$, update only one component of the solution vector as (2.8).

The arithmetical operations counts for the elimination-variant serial Cholesky algorithm are as stated in Table 2.1.

These counts agree with those for the classical Cholesky method as given in Fox [5].

TABLE 2.1. Operations counts for elimination-variant serial Cholesky.

Steps	$+$	$\times$	$\div$	$\sqrt{\ }$
Step 1	$\frac{1}{6}N^3+\frac{1}{2}N^2-\frac{2}{3}N$	$\frac{1}{6}N^3+N^2-\frac{1}{6}N$	N	N
Step 2	$\frac{1}{2}N^2-\frac{1}{2}N$	$\frac{1}{2}N^2+\frac{1}{2}N$		
Total	$\frac{1}{6}N^3+N^2-\frac{7}{6}N$	$\frac{1}{6}N^3+\frac{3}{2}N^2+\frac{1}{3}N$	N	N

TABLE 2.2. Operation counts for computing L.

$+$	$\times$	$\div$	$\sqrt{\ }$
$\frac{1}{6}N^3-\frac{1}{6}N$	$\frac{1}{6}N^3+\frac{1}{2}N^2-\frac{2}{3}N$	N	N

It is of interest to note here that for the present elimination-variant Cholesky algorithm, at the end of reduction,

$$\begin{aligned} I = A^{(N)} &= L^{(N)}A^{(N-1)}L^{(N)^T} \\ &= L^{(N)}L^{(N-1)}A^{(N-2)}L^{(N-1)^T}L^{(N)^T} \\ &= L^{(N)}L^{(N-1)}\cdots L^{(1)}AL^{(1)^T}\cdots L^{(N-1)^T}L^{(N)^T}. \end{aligned} \tag{2.11}$$

If we now set

$$L = L^{(N)}L^{(N-1)}\cdots L^{(1)}, \tag{2.12}$$

then it is clear that the reduction stage of our present elimination-variant Cholesky algorithm is equivalent to producing a lower triangular L such that

$$LAL^T = I. \tag{2.13}$$

We also note here the arithmetical operations counts for computing only L by the above elimination-variant Cholesky algorithm; these counts are as stated in Table 2.2.

We will need these counts when we do arithmetical operations counts for the parallel Cholesky algorithm in Section 3.

3. Parallel Cholesky algorithm.

We first describe the parallel Cholesky method. With $N = rn$, assume that A is partitioned into r^2 blocks, each of size $n\times n$:

$$A = [A_{(i,j)}]_{i,j=1}^{r}, \quad A_{(i,j)} = a_{(i-1)n+1\to in,(j-1)n+1\to jn}. \tag{3.1}$$

The vectors $\mathbf{x}$ and $\mathbf{d}$ are assumed partitioned conformally. Note that $A^T_{(i,j)} = A_{(j,i)}$.

Assume that $k-1$ reductions have been done, with the reduced system assuming the form $A^{(k-1)}\mathbf{x}^{(k-1)} = \mathbf{d}^{(k-1)}$, where

$$A^{(k-1)}_{(i,j)} = \left[\begin{array}{ccc} 1\cdots k-1 & k & k+1\cdots n \\ \delta_{i,j}I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a^{(k-1)}_{(i,j)} & \mathbf{b}^{(k-1)}_{(i,j)} \\ \mathbf{0} & \mathbf{c}^{(k-1)}_{(i,j)} & B^{(k-1)}_{(i,j)} \end{array}\right], \quad \mathbf{b}^{(k-1)}_{(i,j)} = \mathbf{c}^{(k-1)^T}_{(j,i)}, \tag{3.2}$$

and $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

To describe the stage $k \in \{1,\dots,n\}$ of the reduction, we define the lower triangular elimination transformation

$$L^{(k)} = \left[L^{(k)}_{(i,j)}\right]^{r}_{i,j=1}, \quad L^{(k)}_{(i,j)} = \mathbf{0} \text{ if } j > i, \tag{3.3}$$

where

$$L^{(k)}_{(i,j)} = \begin{bmatrix} \delta_{i,j}I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda^{(k)}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \ell^{(k)}_{(i,j)} & \delta_{i,j}I \end{bmatrix}. \tag{3.4}$$

The stage k now consists of a symmetric reduction of (2.1), with the help of $L^{(k)}$, as in (2.4). The conditions for the intended reduction are

$$\sum_{p,s=1}^{r} \lambda^{(k)}_{(i,p)} a^{(k-1)}_{(p,s)} \lambda^{(k)}_{(j,s)} = \delta_{i,j}, \tag{3.5}$$

$$\sum_{s=1}^{r} \lambda^{(k)}_{(j,s)} \left[\sum_{p=1}^{r} a^{(k-1)}_{(p,s)} \ell^{(k)}_{(i,p)} + \mathbf{c}^{(k-1)}_{(i,s)}\right] = \mathbf{0}, \tag{3.6}$$

$$B^{(k)}_{(i,j)} = B^{(k-1)}_{(i,j)} + \sum_{s=1}^{r} \mathbf{c}^{(k-1)}_{(i,s)} \ell^{(k)^T}_{(j,s)} + \sum_{p=1}^{r} \ell^{(k)}_{(i,p)} \left[\sum_{s=1}^{r} a^{(k-1)}_{(p,s)} \ell^{(k)^T}_{(j,s)} + \mathbf{b}^{(k-1)}_{(p,j)}\right]. \tag{3.7}$$

Let

$$\Lambda^{(k)} = \left[\lambda^{(k)}_{(i,j)}\right]^{r}_{i,j=1}, \qquad P^{(k-1)} = \left[a^{(k)}_{(i,j)}\right]^{r}_{i,j=1}. \tag{3.8}$$

Note that $\Lambda^{(k)}$ is lower triangular and $P^{(k-1)}$ is symmetric and positive definite by induction (see Section 4). Then, the condition in (3.5) can be expressed as

$$\Lambda^{(k)} P^{(k-1)} \Lambda^{(k)^T} = I. \tag{3.9}$$

Note that for a given $P^{(k-1)}$, we can compute $\Lambda^{(k)}$ by the elimination-variant serial Cholesky algorithm described in Section 2.1. With the help of (3.9), it is easy to see that the two conditions in (3.6) and (3.7) can be simplified to give

$$\sum_{p=1}^{r} \ell_{(i,p)}^{(k)} a_{(p,s)}^{(k-1)} + \mathbf{c}_{(i,s)}^{(k-1)} = \mathbf{0}, \tag{3.10}$$

$$B_{(i,j)}^{(k)} = B_{(i,j)}^{(k-1)} + \sum_{s=1}^{r} \mathbf{c}_{(i,s)}^{(k-1)} \ell_{(j,s)}^{(k)^T}. \tag{3.11}$$

Now, in order to solve (3.10), we first write it as

$$\left[\ell_{(i,1)}^{(k)},\dots,\ell_{(i,r)}^{(k)}\right] P^{(k-1)} = -\left[\mathbf{c}_{(i,1)}^{(k-1)},\dots,\mathbf{c}_{(i,r)}^{(k-1)}\right]. \tag{3.12}$$

For this purpose, (3.9) suggests that we introduce

$$\left[\mathbf{u}_{(i,1)}^{(k)},\dots,\mathbf{u}_{(i,r)}^{(k)}\right] = \left[\mathbf{c}_{(i,1)}^{(k-1)},\dots,\mathbf{c}_{(i,r)}^{(k-1)}\right] \Lambda^{(k)^T}. \tag{3.13}$$

Postmultiplying (3.12) by $\Lambda^{(k)^T}$, in view of (3.9), it follows that

$$\left[\ell_{(i,1)}^{(k)},\dots,\ell_{(i,r)}^{(k)}\right] = -\left[\mathbf{u}_{(i,1)}^{(k)},\dots,\mathbf{u}_{(i,r)}^{(k)}\right] \Lambda^{(k)}. \tag{3.14}$$

We need not compute the $\ell_{(i,j)}^{(k)}$; it is more convenient to express the method in terms of the $\mathbf{u}_{(i,j)}^{(k)}$. These can be computed from (3.13), given by

$$\mathbf{u}_{(i,j)}^{(k)} = \sum_{s=1}^{j} \lambda_{(j,s)}^{(k)} \mathbf{c}_{(i,s)}^{(k-1)}. \tag{3.15}$$

For updates of $B_{(i,j)}^{(k)}$, from (3.11) we have

$$B_{(i,j)}^{(k)} = B_{(i,j)}^{(k-1)} + \left[\mathbf{c}_{(i,1)}^{(k-1)},\dots,\mathbf{c}_{(i,r)}^{(k-1)}\right]\left[\ell_{(j,1)}^{(k)},\dots,\ell_{(j,r)}^{(k)}\right]^T. \tag{3.16}$$

Now, substituting from (3.14) and in view of (3.11), we obtain

$$B_{(i,j)}^{(k)} = B_{(i,j)}^{(k-1)} - \sum_{s=1}^{r} \mathbf{u}_{(i,s)}^{(k)} \mathbf{u}_{(j,s)}^{(k)^T}. \tag{3.17}$$

For a compact reduction, we simultaneously modify the right-hand side vector as follows. Since

$$\mathbf{d}_{(i)}^{(k)} = \sum_{s=1}^{r} L_{(i,s)}^{(k)} \mathbf{d}_{(s)}^{(k-1)}, \tag{3.18}$$

it follows that $\mathbf{d}_{(i)}^{(k)}$ needs to be modified only in its following components:

$$d_{(i-1)n+k}^{(k)} = \sum_{s=1}^{i} \lambda_{(i,s)}^{(k)} d_{(s-1)n+k}^{(k)}, \tag{3.19a}$$

$$d_{(i-1)n+k+1\to in}^{(k)} = d_{(i-1)n+k+1\to in}^{(k-1)} + \sum_{s=1}^{r} d_{(s-1)n+k}^{(k-1)} \ell_{(i,s)}^{(k)}. \tag{3.19b}$$

To express (3.19b) in terms of the $\mathbf{u}^{(k)}_{(i,j)}$, we note that on substituting from (3.14), (3.19b) can be written as

$$d^{(k)}_{(i-1)n+k+1\to in} = d^{(k-1)}_{(i-1)n+k+1\to in} - \left[\mathbf{u}^{(k)}_{(i,1)},\dots,\mathbf{u}^{(k)}_{(i,r)}\right]\Lambda^{(k)}\begin{bmatrix} d^{(k-1)}_{(1-1)n+k} \\ \vdots \\ d^{(k-1)}_{(r-1)n+k} \end{bmatrix}. \tag{3.20}$$

But, from (3.19a) we have

$$\begin{bmatrix} d^{(k)}_{(1-1)n+k} \\ \vdots \\ d^{(k)}_{(r-1)n+k} \end{bmatrix} = \Lambda^{(k)}\begin{bmatrix} d^{(k-1)}_{(1-1)n+k} \\ \vdots \\ d^{(k-1)}_{(r-1)n+k} \end{bmatrix}, \tag{3.21}$$

then from (3.20) it follows that

$$d^{(k)}_{(i-1)n+k+1\to in} = d^{(k-1)}_{(i-1)n+k+1\to in} - \sum_{s=1}^{r}\mathbf{u}^{(k)}_{(i,s)}d^{(k)}_{(s-1)n+k}. \tag{3.22}$$

We next consider the necessary updates in the solution vector $\mathbf{x}$. Since

$$\mathbf{x}^{(k-1)}_{(i)} = \sum_{s=1}^{r}L^{(k)^T}_{(s,i)}\mathbf{x}^{(k)}_{(s)}, \tag{3.23}$$

it is easy to see that only one component needs to be updated:

$$x^{(k-1)}_{(i-1)n+k} = \sum_{s=1}^{r}\left[\lambda^{(k)}_{(s,i)}x^{(k)}_{(s-1)n+k} + \ell^{(k)^T}_{(s,i)}x^{(k)}_{(s-1)n+k+1\to sn}\right]. \tag{3.24}$$

To express it in terms of $\mathbf{u}^{(k)}_{(i,j)}$, substituting for $\ell^{(k)}_{(i,j)}$ from (3.14) and simplifying, we obtain

$$x^{(k-1)}_{(i-1)n+k} = \sum_{s=i}^{r}\lambda^{(k)}_{(s,i)}\left[x^{(k)}_{(s-1)n+k} - \sum_{m=1}^{r}\mathbf{u}^{(k)^T}_{(m,s)}x^{(k)}_{(m-1)n+k+1\to mn}\right]. \tag{3.25}$$

The finally transformed system is

$$\mathbf{x}^{(n)} = \mathbf{d}^{(n)}, \tag{3.26}$$

from which it follows that the solution in each block can be computed starting with

$$x^{(n)}_{(i-1)n+1\to in} = d^{(n)}_{(i-1)n+1\to in}. \tag{3.27}$$

3.1. The algorithm.

Next, we describe the parallel Cholesky algorithm.

Elimination stage (compact form). For $k = 1(1)n-1$, do Steps 1, 2, and 3; and for $k = n$, do only Step 1.

Step 1. Set $P^{(k-1)} = [a_{(i,j)}^{(k-1)}]_{i,j=1}^{r}$ and compute $\Lambda^{(k)}$ as in (3.9) and by the elimination-variant serial Cholesky algorithm previously given in Section 2.1, and update

$$d_{(i-1)n+k}^{(k)} = \sum_{s=1}^{i} \lambda_{(i,s)}^{(k)} d_{(s-1)n+k}^{(k-1)}, \quad i = 1(1)r. \tag{3.28}$$

Step 2. Compute $(in\|)$ for $i = 1(1)r$:

$$\mathbf{u}_{(i,j)}^{(k)} = \sum_{s=1}^{j} \lambda_{(j,s)}^{(k)} A_{(i-1)n+k+1\to in,(s-1)n+k}^{(k-1)}, \quad j = 1(1)r. \tag{3.29}$$

Step 3. Update $(in\|)$ for $i = 1(1)r$:

$$\begin{aligned} A_{(i-1)n+k+1\to in,(j-1)n+k+1\to jn}^{(k)} &= A_{(i-1)n+k+1\to in,(j-1)n+k+1\to jn}^{(k-1)} \\ &\quad - \sum_{s=1}^{r} \mathbf{u}_{(i,s)}^{(k)} \mathbf{u}_{(j,s)}^{(k)^T}, \quad j = 1(1)i, \end{aligned} \tag{3.30}$$

(for $j = i$, only lower triangle needs to be updated),

$$d_{(i-1)n+k+1\to in}^{(k)} = d_{(i-1)n+k+1\to in}^{(k-1)} - \sum_{s=1}^{r} u_{(i,s)}^{(k)} d_{(s-1)n+k}^{(k)}. \tag{3.31}$$

Solution stage

Step 4. Set $\mathbf{x}^{(n)} = \mathbf{d}^{(n)}$. For $k = n$ down to 1, update $(in\|)$ for $i = 1(1)r$ as in (3.25).

3.2. Arithmetical operations counts for parallel Cholesky algorithm.

For the parallel Cholesky algorithm, the arithmetical operations counts are as stated in Table 3.1.

For $r = 1$, these counts agree with the counts for the serial Cholesky given in Section 2.1.

4. Existence of parallel Cholesky algorithm.

Next, we show that if the coefficient matrix of the given system is symmetric and positive definite, then the parallel Cholesky algorithm described above exists and will run to completion.

We first show the following lemma.

Lemma 4.1. *$A^{(k)}$ is symmetric if $A^{(k-1)}$ is symmetric.*

Proof. Taking the transpose in (3.17), we have

$$B_{(i,j)}^{(k)^T} = B_{(j,i)}^{(k-1)} - \sum_{s=1}^{r} \mathbf{u}_{(j,s)}^{(k)} \mathbf{u}_{(i,s)}^{(k)^T} = B_{(j,i)}^{(k)}. \tag{4.1}$$

□

Theorem 4.2. *$A^{(k)}$ is positive definite if $A^{(k-1)}$ is positive definite.*

TABLE 3.1. Operations counts for parallel Cholesky.

Step	$+$	$\times$	$\div$	$\sqrt{\ }$
Step 1	$\frac{1}{6}N(r^2+3r-4)$	$\frac{1}{6}N(r^2+6r-1)$	N	N
Step 2	$\frac{1}{4}N(N-r)(r-1)$	$\frac{1}{4}N(N-r)(r+1)$		
Step 3	$\frac{1}{12}N(N-r)(2N-r+9)$	$\frac{1}{12}N(N-r)(2N-r+9)$		
Step 4	$\frac{1}{4}N^2(r+1)-\frac{1}{4}N(r^2-r+2)$	$\frac{1}{4}(r+1)N^2+\frac{1}{4}N(r+1)(2-r)$		
Total	$\frac{1}{6}N^3+\frac{1}{4}N^2(r+3)$ $-\frac{1}{12}N(3r^2-3r+14)$	$\frac{1}{6}N^3+\frac{1}{4}N^2(r+5)$ $-\frac{1}{12}N(3r^2-3r-4)$	N	N

PROOF. For arbitrary $y_{(s-1)n+k+1\to sn}$, let

$$\mathbf{y}_{(s)} = \Big(\underbrace{0,\dots,0}_{1\to k}, y_{(s-1)n+k+1},\dots,y_{sn}\Big)^T, \qquad \mathbf{y} = \left(\mathbf{y}_{(1)},\dots,\mathbf{y}_{(r)}\right)^T. \tag{4.2}$$

Now,

$$\mathbf{y}^T A^{(k)}\mathbf{y} = \sum_{s,p=1}^{r} y^T_{(s-1)n+k+1\to sn} B^{(k)}_{(s,p)} y_{(p-1)n+k+1\to pn}. \tag{4.3}$$

Again, for any $y_{(s-1)n+k}$, let

$$\mathbf{y}^*_{(s)} = \Big(\underbrace{0,\dots,0}_{1\to k-1}, y_{(s-1)n+k},\dots,y_{sn}\Big)^T, \qquad \mathbf{y}^* = \left(\mathbf{y}^*_{(1)},\dots,\mathbf{y}^*_{(r)}\right)^T. \tag{4.4}$$

Then,

$$\begin{aligned}\mathbf{y^*}^{\mathbf{T}} A^{(k-1)}\mathbf{y}^* = \sum_{s,p=1}^{r} \Big[& y_{(s-1)n+k} a^{(k)}_{(s,p)} y_{(p-1)n+k} + 2y^T_{(s-1)n+k+1\to sn}\mathbf{c}^{(k-1)}_{(s,p)} y_{(p-1)n+k} \\ & + y^T_{(s-1)n+k+1\to sn} B^{(k-1)}_{(s,p)} y_{(p-1)n+k+1\to pn}\Big].\end{aligned} \tag{4.5}$$

Subtracting (4.5) from (4.3), in view of (3.17), we obtain

$$\begin{aligned}\mathbf{y}^T A^{(k)}\mathbf{y}-\mathbf{y}^{*\mathbf{T}} A^{(k-1)}\mathbf{y}^* = -\sum_{s,p=1}^{r}\Bigg[& y_{(s-1)n+k}a_{(s,p)}^{(k)}y_{(p-1)n+k}\\ &+2y_{(s-1)n+k+1\to sn}^{T}\mathbf{c}_{(s,p)}^{(k-1)}y_{(p-1)n+k}\\ &+y_{(s-1)n+k+1\to sn}^{T}\left(\sum_{q=1}^{r}\mathbf{u}_{(s,q)}^{(k)}\mathbf{u}_{(p,q)}^{(k)^T}\right)y_{(p-1)n+k+1\to pn}\Bigg].\end{aligned} \tag{4.6}$$

In order to simply express the right-hand side in (4.6), we introduce

$$\begin{gathered} U_{(i)}^{(k)}=\left[\mathbf{u}_{(i,1)}^{(k)},\dots,\mathbf{u}_{(i,r)}^{(k)}\right], \qquad C_{(i)}^{(k)}=\left[\mathbf{c}_{(i,1)}^{(k-1)},\dots,\mathbf{c}_{(i,r)}^{(k-1)}\right],\\ \sum_{s=1}^{r}y_{(s-1)n+k+1\to sn}^{T}\mathbf{c}_{(s,i)}^{(k-1)}=m_i^{(k)},\\ \mathbf{m}^{(k)}=\left(m_1^{(k)},\dots,m_r^{(k)}\right)^T, \qquad \mathbf{w}^{(k)}=\left(y_k,y_{n+k},\dots,y_{(r-1)n+k}\right)^T.\end{gathered} \tag{4.7}$$

Note that

$$\sum_{s=1}^{r}y_{(s-1)n+k+1\to sn}^{T}C_{(s)}^{(k)}=\mathbf{m}^{(k)^T}. \tag{4.8}$$

In view of (3.13), $U_{(i)}^{(k)}=C_{(i)}^{(k)}\Lambda^{(k)^T}$, therefore

$$\sum_{q=1}^{r}\mathbf{u}_{(s,q)}^{(k)}\mathbf{u}_{(p,q)}^{(k)^T}=U_{(s)}^{(k)}U_{(p)}^{(k)^T}=C_{(s)}^{(k)}\Lambda^{(k)^T}\Lambda^{(k)}C_{(p)}^{(k)^T}. \tag{4.9}$$

Also, from (3.13), we have $P^{(k-1)}=\Lambda^{(k)^{-1}}(\Lambda^{(k)^T})^{-1}$. With these results, from (4.6) we obtain

$$\begin{aligned}\mathbf{y}^T A^{(k)}\mathbf{y}-\mathbf{y}^{*\mathbf{T}} A^{(k-1)}\mathbf{y}^* &= -\Bigg[\mathbf{w}^{(k)^T}\Lambda^{(k)^{-1}}\left(\Lambda^{(k)^T}\right)^{-1}\mathbf{w}^{(k)}\\ &\qquad +2\mathbf{w}^{(k)^T}\left(\Lambda^{(k)^{-1}}\Lambda^{(k)}\right)\mathbf{m}^{(k)}+\mathbf{m}^{(k)^T}\Lambda^{(k)^T}\Lambda^{(k)}\mathbf{m}^{(k)}\Bigg]\\ &= -\left\|\Lambda^{(k)}\mathbf{m}^{(k)}+\left(\Lambda^{(k)^T}\right)^{-1}\mathbf{w}^{(k)}\right\|^2.\end{aligned} \tag{4.10}$$

Now choose

$$\left(\Lambda^{(k)^T}\right)^{-1}\mathbf{w}^{(k)}=-\Lambda^{(k)}\mathbf{m}^{(k)}=-P^{(k-1)^{-1}}\mathbf{m}^{(k)}, \tag{4.11}$$

then

$$\mathbf{y}^T A^{(k)} \mathbf{y} = \mathbf{y}^{*^{\mathrm{T}}} A^{(k-1)} \mathbf{y}^*, \tag{4.12}$$

and it follows that $A^{(k)}$ is positive definite if $A^{(k-1)}$ is positive definite. □

We now have the following theorem.

Theorem 4.3. *If A is symmetric and positive definite, then the parallel Cholesky algorithm is well defined and will run to completion.*

Proof. By Lemma 4.1 and Theorem 4.2, for any $k \in \{1,\dots,n\}$, $A^{(k)}$ is symmetric and positive definite. In particular, $P^{(k-1)}$ is symmetric and positive definite with diagonal elements $a_{(i,i)}^{(k)} > 0$ for $i = 1,\dots,r$. □

5. An illustration of the algorithm.

To illustrate the working of the presented parallel Cholesky algorithm, we consider the following symmetric linear system ($N = 6$) by taking $r = 2$. We work with the full matrix to highlight that the reduction is symmetric:

$$[A:\mathbf{d}] = \begin{bmatrix} 4 & -1 & 1 & -1 & 1 & -1 & : & 3 \\ -1 & \frac{97}{32} & -1 & 1 & -1 & 1 & : & \frac{65}{32} \\ 1 & -1 & \frac{21}{16} & -1 & 1 & -1 & : & \frac{5}{16} \\ -1 & 1 & -1 & \frac{17}{4} & -1 & 1 & : & \frac{13}{4} \\ 1 & -1 & 1 & -1 & \frac{35}{32} & -1 & : & \frac{3}{32} \\ -1 & 1 & -1 & 1 & -1 & \frac{19}{16} & : & \frac{3}{16} \end{bmatrix}. \tag{5.1}$$

The steps of reduction are as follows:

$$[A^{(1)}:\mathbf{d}^{(1)}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & : & \frac{3}{2} \\ 0 & \frac{169}{64} & -\frac{39}{64} & 0 & -\frac{39}{64} & \frac{39}{64} & : & \frac{65}{32} \\ 0 & -\frac{39}{64} & \frac{59}{64} & 0 & \frac{39}{64} & -\frac{39}{64} & : & \frac{5}{16} \\ 0 & 0 & 0 & 1 & 0 & 0 & : & 2 \\ 0 & -\frac{39}{64} & \frac{39}{64} & 0 & \frac{45}{64} & -\frac{39}{64} & : & \frac{3}{32} \\ 0 & \frac{39}{64} & -\frac{39}{64} & 0 & -\frac{39}{64} & \frac{51}{64} & : & \frac{3}{16} \end{bmatrix},$$

$$[A^{(2)} : \mathbf{d}^{(2)}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & : & \frac{3}{2} \\ 0 & 1 & 0 & 0 & 0 & 0 & : & \frac{5}{4} \\ 0 & 0 & \frac{25}{64} & 0 & 0 & -\frac{5}{64} & : & \frac{5}{16} \\ 0 & 0 & 0 & 1 & 0 & 0 & : & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & : & \frac{3}{4} \\ 0 & 0 & -\frac{5}{64} & 0 & 0 & \frac{17}{64} & : & \frac{3}{16} \end{bmatrix},$$

$$[A^{(3)} : \mathbf{d}^{(3)}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & : & \frac{3}{2} \\ 0 & 1 & 0 & 0 & 0 & 0 & : & \frac{5}{4} \\ 0 & 0 & 1 & 0 & 0 & 0 & : & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & : & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & : & \frac{3}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 & : & \frac{1}{2} \end{bmatrix}. \tag{5.2}$$

The recovery of the solution proceeds as follows:

$$\mathbf{x}^{(3)} = \mathbf{d}^{(3)} = \begin{bmatrix} \frac{3}{2} \\ \frac{5}{4} \\ \frac{1}{2} \\ 2 \\ \frac{3}{4} \\ \frac{1}{2} \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} \frac{3}{2} \\ \frac{5}{4} \\ \frac{1}{2} \\ \frac{3}{4} \\ 1 \end{bmatrix}, \quad \mathbf{x}^{(1)} = \begin{bmatrix} \frac{3}{2} \\ 1 \\ \frac{1}{2} \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ \frac{1}{1} \\ 1 \\ 1 \end{bmatrix} = \mathbf{x}, \tag{5.3}$$

giving the solution of the system.

References

[1] M. M. Chawla, *A parallel elimination algorithm for the solution of dense linear systems*, Int. J. Comput. Math. **47** (1993), no. 1-2, 97–107.

[2] M. Cosnard, M. Marrakchi, Y. Robert, and D. Trystram, *Parallel Gaussian elimination on an MIMD computer*, Parallel Comput. **6** (1988), no. 3, 275–296.

[3] J. J. Dongarra, F. G. Gustavson, and A. Karp, *Implementing linear algebra algorithms for dense matrices on a vector pipeline machine*, SIAM Rev. **26** (1984), no. 1, 91–112.

[4] L. Fox, *Practical solution of linear equations and inversion of matrices*, Contributions to the Solution of Systems of Linear Equations and the Determination of Eigenvalues, National Bureau of Standards Applied Mathematics Series, no. 39, U. S. Government Printing Office, District of Columbia, 1954, pp. 1–54.

[5] ______, *An Introduction to Numerical Linear Algebra*, Monographs on Numerical Analysis, Clarendon Press, Oxford, 1964.

[6] T. L. Freeman and C. Phillips, *Parallel Numerical Algorithms*, Prentice Hall International Series in Computer Science, Prentice Hall, New York, 1992.

[7] K. A. Gallivan, R. J. Plemmons, and A. H. Sameh, *Parallel algorithms for dense linear algebra computations*, SIAM Rev. **32** (1990), no. 1, 54–135.

[8] A. Geist and M. Heath, *Matrix factorization on a hypercube multiprocessor*, Hypercube Multiprocessors (M. Heath, ed.), Soc. Industr. Appl. Math., Pennsylvania, 1986, pp. 161–180.

[9] A. George, M. Heath, and J. Liu, *Parallel Cholesky factorization on a shared memory multiprocessor*, Linear Algebra Appl. **77** (1986), 165–187.

[10] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Series in the Mathematical Sciences, vol. 3, Johns Hopkins University Press, Maryland, 1983.

[12] M. Heath, *Parallel Cholesky factorization in message-passing multiprocessor environments*, Tech. Report ORNL-6150, Oak Ridge National Laboratory, May 1985.

[13] D. J. Kuck, *The Structure of Computers and Computations*, vol. 1, John Wiley & Sons, New York, 1978.

[14] J. Ortega and C. Romine, *The ijk forms of factorization methods II. Parallel systems*, Parallel Comput. **7** (1988), no. 2, 149–162.

[15] J. M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1989.

R. R. Khazal: Department of Mathematics and Computer Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait
E-mail address: khazal@mcs.sci.kuniv.edu.kw

M. M. Chawla: Department of Mathematics and Computer Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait
E-mail address: chawla@mcs.sci.kuniv.edu.kw