International Journal of Pure and Applied Mathematics

Volume 2 No. 3 2002, 357-371

A PARALLEL ELIMINATION ALGORITHM FOR BANDED LINEAR SYSTEMS

M.M. Chawla[§], R.R. Khazal

Dept. of Mathematics and Computer Science
Kuwait University
P.O. Box 5969, Safat 13060, KUWAIT

Abstract: We present a parallel algorithm for the solution of β-semiband linear systems of size $N \times N$ by partitioning the system into p^2 blocks each of size n $(N = pn)$. In order to uncouple the partitioned blocks, in each diagonal block $B^{(r)}$ we apply two sets of *simultaneous* eliminations; the first set consists of β usual forward eliminations within the block and the second set consists of β eliminations in the block $C^{(r-1)}$ on top of $B^{(r)}$. While the vertical fill-ins in the last β columns of the block on the left of $B^{(r)}$ pose no difficulty, the purpose of the second set of eliminations is to move fill-ins in the last β rows of $C^{(r-1)}$ successively to the right till they reach their destination in the last β columns. At the end of the elimination stage, we reach a core block tridiagonal system with each block of size $\beta \times \beta$. Once the core system is solved, the partitioned blocks of equations uncouple and the uncoupled subsystems can be solved in parallel by back substitution. We include arithmetic complexity for both serial and parallel implementations of the presented algorithm and illustrate the main idea of the presented parallel algorithm by an example.

AMS Subject Classification: 65F30, 15A06
Key Words: banded linear systems, partitioning, elimination, parallel algorithm, efficiency

Received: May 18, 2002

[§]Correspondence author

1. Introduction

We are concerned with parallel solution of β-semiband systems:

$$A\mathbf{x} = \mathbf{d}, \tag{1.1}$$

where $A = (a_{i,j})_{i,j=1}^{N}$, $a_{i,j} = 0$ if $|j-i| > \beta$, $\mathbf{x} = (x_1, ..., x_N)^\top$, $\mathbf{d} = (d_1, ..., d_N)^\top$.

Banded linear systems occur often from discretization of differential equations. Low order time integration schemes invariably lead to tridiagonal linear systems (see, e.g. Thomas [20]) and therefore the solution of a tridiagonal system is central to scientific computing. For reasons of stability and/or accuracy of the computed solution, modifications of the low order schemes or higher order time integration schemes are needed; see, e.g. Chawla et al [4, 5, 6]. For evolution problems, it is desired to resolve the solution profile as quickly as possible. Higher order time integration schemes allow the use of relatively larger time steps to compute the solution. But higher order schemes require the solution of banded systems with bandwidth increasing with the order of the method. Since these systems need to be solved for each time step taken, it becomes imperative to parallel process the solution of these systems.

Elimination (equivalently, LU factorization) method is a best serial method for the solution of banded linear systems. Banded systems can be viewed from two directions: either as general systems with diagonal sparsity, or as special block tridiagonal systems. Taking the first view point, parallel algorithms have been described for banded systems through the so-called "ijk" codes; see, e.g. Golub and Van Loan [11], Ortega [19]. In this approach, however, it is difficult to parallelize the outermost loop whose parallelization is of main interest; see, e.g. Freeman and Phillips [10].

Parallel algorithms for tridiagonal linear systems have seen some recent developments. The best serial algorithm is due to Thomas (see Fox [9], Goutl et al [12]) with a serial count of O(8N), and it is widely used (see, e.g. Thomas [20]). Wang [21], [22] had suggested a parallel method for tridiagonal linear systems based on partitioning of the system and Gaussian elimination. Wang's elimination procedure consists of two stages; first, a forward elimination is applied within each block, which is then followed by a backward elimination after shifting the partitioning lines one row up, similar to the idea of "separator equations" used by Johnsson [13] in his factorization method. The serial operations count for Wang's algorithm is O(17N). Wang's method has been extended for banded systems by Lawrie and Sameh [17], Johnsson [14] and Meier [18].

Taking the second view point, banded systems can be regarded as special block tridiagonal systems. Chawla et al [8] had proposed a fast parallel tridiagonal solver with serial operations count of $(16\frac{1}{2}N)$. Chawla and Passi [3] extended this method to obtain a parallel algorithm for banded linear systems.

Chawla et al [7] proposed an incomplete factorization method and Chawla and Khazal [2] have proposed one-stage parallel elimination algorithm for partitioned tridiagonal linear systems. More recently, Khazal and Chawla [15] presented a parallel method for the solution of pentadiagonal linear systems extending their earlier work in Chawla and Khazal [2]. The present paper is in the same vein, extending the work of Chawla and Khazal [2] to general banded linear systems. In terms of operations complexity and efficiency, our present algorithm gives an improved efficiency over the algorithm presented in Chawla and Passi [3].

In the present paper, we present a parallel algorithm for the solution of β-semiband linear systems of size $N \times N$ by partitioning the system into p^2 blocks each of size n ($N = pn$). In order to uncouple the partitioned blocks, in each diagonal block $B^{(r)}$ we apply two sets of *simultaneous* eliminations; the first set consists of β usual forward eliminations within the block and the second set consists of β eliminations in the block $C^{(r-1)}$ on top of $B^{(r)}$. While the vertical fill-ins in the last β columns of the block on the left of $B^{(r)}$ pose no difficulty, the purpose of the second set of eliminations is to move fill-ins in the last β rows of $C^{(r-1)}$ successively to the right till they reach their destination in the last β columns. At the end of the elimination stage, we reach a core block tridiagonal system with each block of size $\beta \times \beta$. Once the core system is solved, the partitioned blocks of equations uncouple and the uncoupled subsystems can be solved in parallel by back substitution. We include arithmetic complexity for both serial and parallel implementations of the presented algorithm. It turns out that the present algorithm has a smaller operations count and better efficiency than the parallel banded system solver in Chawla and Passi [3] for systems with semibandwidth greater than one. Finally, we include an example to illustrate the main idea of the presented algorithm.

1.1. Serial Elimination Algorithm for Banded Systems

A description of the serial LU factorization algorithm for banded systems can be found, for example, in Golub and Van Loan [11]. We include here a compact form of the elimination algorithm together with its precise computational complexity; see Chawla and Evans [1]. In the description of the algorithm, the elements of A and $\mathbf{d}$ are overwritten during updation. In the following, $s_k = \min(k + \beta, N)$.

Elimination stage

for $k = 1$ to $N - 1$

for $j = k + 1$ to s_k

$a_{j,k+1\to s_k} := a_{j,k+1\to s_k} - (a_{j,k} / a_{k,k})\, a_{k,k+1\to s_k}$,

$d_j := d_j - (a_{j,k} / a_{k,k})\, d_k$.

Solution stage

For $k = N$ down to 1

$x_k = (1 / a_{k,k}) \left[d_k - \sum_{s=k+1}^{s_k} a_{k,s} x_s \right]$.

Arithmetic complexity

The elimination stage involves a total of

$$\left(2\beta^2 + 3\beta\right) N - (1/6)\beta\,(\beta + 1)\,(8\beta + 7)$$

operations of additions, multiplications and divisions, while the solution stage involves a total of $(2\beta + 1)\, N - \beta\,(\beta + 1)$. Thus, the grand total (GT) of operations of additions, multiplications and divisions for the serial elimination algorithm for the solution of β-semiband system of size N is given by

$$GT = \left(2\beta^2 + 5\beta + 1\right) N - \frac{1}{6}\beta\,(\beta + 1)\,(8\beta + 13)\,. \tag{1.2}$$

2. Parallel Elimination Method

With $N = pn$, consider the system (1.1) partitioned into p blocks each consisting of n consecutive equations. To facilitate derivation of the method, it is convenient to introduce the following notation. For $r \in \{1, ..., p\}$,

$u_k^{(r)} = u_{(r-1)n+k}$, $k = 1(1)n$, $\mathbf{u}^{(r)} = \left(u_1^{(r)}, ..., u_n^{(r)}\right)^\top$,

$b_{i,j}^{(r)} = a_{(r-1)n+i,(r-1)n+j}$, $j = \max(1, i - \beta)$ to $\max(i + \beta, n)$, $i = 1(1)n$,

$c_{n-\beta+i,j}^{(r)} = a_{rn-\beta+i,rn+j}$, $j = 1(1)i$, $i = 1(1)\beta$,

$a_{i,n+1-j}^{(r)} = a_{(r-1)n+i,(r-1)n+1-j}$, $j = 1(1)\beta + 1 - i$, $i = 1(1)\beta$.

Let $E^{(r)}$ denote the r-th block of equations:

$$A^{(r;0)}\mathbf{x}^{(r-1)} + B^{(r;0)}\mathbf{x}^{(r)} + C^{(r;0)}\mathbf{x}^{(r+1)} = \mathbf{d}^{(r;0)}, \tag{2.1}$$

where $\mathbf{d}^{(r;0)} = \mathbf{d}^{(r)}$, and

$$A^{(r;0)} = \begin{bmatrix} & & a^{(r)}_{1,n-\beta+1} & \cdot & a^{(r)}_{1,n} \\ & & & & \cdot \\ & & & & a^{(r)}_{\beta,n} \\ & & & & \\ & & & & \\ & & & & \end{bmatrix},$$

$$B^{(r;0)} = \begin{bmatrix} b^{(r)}_{1,1} & \cdot & b^{(r)}_{1,1+\beta} & & & & \\ \cdot & \cdot & \cdot & \cdot & & & \\ b^{(r)}_{1+\beta,1} & \cdot & b^{(r)}_{1+\beta,1+\beta} & \cdot & b^{(r)}_{1+\beta,1+2\beta} & & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & b^{(r)}_{n-\beta,n-2\beta} & \cdot & b^{(r)}_{n-\beta,n-\beta} & \cdot & b^{(r)}_{n-\beta,n} \\ & & & \cdot & \cdot & \cdot & \cdot \\ & & & & b^{(r)}_{n,n-\beta} & \cdot & b^{(r)}_{n,n} \end{bmatrix},$$

$$C^{(r;0)} = \begin{bmatrix} & & \\ & & \\ & & \\ c^{(r)}_{n-\beta+1,1} & & \\ \cdot & & \\ c^{(r)}_{n,1} & \cdot & c^{(r)}_{n,\beta} \end{bmatrix}.$$

For $j \in \{1, ..., n\}$ let $E^{(r)}_j$ denote the j-th equation in block $E^{(r)}$.

Assume $k-1$ eliminations have been carried out; we describe step $k \in \{1, ..., n-\beta\}$. In doing so, in the following we overwrite A and $\mathbf{d}$ as their elements get updated during each elimination. Now, we need two sets of *simultaneous* eliminations:

(i) with $E^{(r)}_k$, we eliminate $x^{(r)}_k$ from $E^{(r)}_{k+1\to k+\beta}$:

$E^{(r)}_j := E^{(r)}_j - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k}\right) E^{(r)}_k$, $j = k+1(1)k+\beta$,

and

(ii) with $E^{(r)}_k$, we eliminate $x^{(r)}_k$ from $E^{(r-1)}_{n-\beta+1\to n}$:

$E^{(r-1)}_j := E^{(r-1)}_j - \left(c^{(r-1)}_{j,k} \big/ b^{(r)}_{k,k}\right) E^{(r)}_k$, $j = n-\beta+1(1)n$.

After $n-\beta$ steps of elimination, the r-th block of the reduced system is

$$A^{(r;n-\beta)}\mathbf{x}^{(r-1)} + B^{(r;n-\beta)}\mathbf{x}^{(r)} + C^{(r;n-\beta)}\mathbf{x}^{(r+1)} = \mathbf{d}^{(r;n-\beta)}, \tag{2.2}$$

where

$$A^{(r;n-\beta)} = \left[\begin{array}{cccccc} & & a^{(r)}_{1,n-\beta+1} & \cdot & a^{(r)}_{1,n} \\ & & \cdot & \cdot & \cdot \\ & & & & \\ & a^{(r)}_{n-\beta+1,n-\beta+1} & & \cdot & a^{(r)}_{n-\beta+1,n} \\ & & \cdot & \cdot & \cdot \\ & & a^{(r)}_{n,n-\beta+1} & \cdot & a^{(r)}_{n,n} \end{array} \right],$$

$$B^{(r;n-\beta)} = \left[\begin{array}{ccccccc} b^{(r)}_{1,1} & \cdot & b^{(r)}_{1,1+\beta} & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & b^{(r)}_{1+\beta,1+\beta} & \cdot & b^{(r)}_{1+\beta,1+2\beta} & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & b^{(r)}_{n-\beta+1,n-\beta+1} & \cdot & b^{(r)}_{n-\beta+1,n} \\ & & & & \cdot & \cdot & \cdot \\ & & & & b^{(r)}_{n,n-\beta+1} & \cdot & b^{(r)}_{n,n} \end{array} \right],$$

$$C^{(r;n-\beta)} = \left[\begin{array}{ccc} & & \\ & & \\ & & \\ c^{(r)}_{n-\beta+1,n-\beta+1} & \cdot & c^{(r)}_{n-\beta+1,n} \\ \cdot & \cdot & \cdot \\ c^{(r)}_{n,n-\beta+1} & \cdot & c^{(r)}_{n,n} \end{array} \right].$$

In (2.2), equations $E^{(r)}_{n-\beta+1\to n}$ collected across the blocks, for $r = 1(1)p$, form the core system:

$$\begin{aligned} & \left[\begin{array}{ccc} a^{(r)}_{n-\beta+1,n-\beta+1} & \cdot & a^{(r)}_{n-\beta+1,n} \\ \cdot & \cdot & \cdot \\ a^{(r)}_{n,n-\beta+1} & \cdot & a^{(r)}_{n,n} \end{array} \right] x^{(r-1)}_{n-\beta+1\to n} \\ & + \left[\begin{array}{ccc} {}^{(r)}_{n-\beta+1,n-\beta+1} & \cdot & b^{(r)}_{n-\beta+1,n} \\ \cdot & \cdot & \cdot \\ b^{(r)}_{n,n-\beta+1} & \cdot & b^{(r)}_{n,n} \end{array} \right] x^{(r)}_{n-\beta+1\to n} \\ & + \left[\begin{array}{ccc} c^{(r)}_{n-\beta+1,n-\beta+1} & \cdot & c^{(r)}_{n-\beta+1,n} \\ \cdot & \cdot & \cdot \\ c^{(r)}_{n,n-\beta+1} & \cdot & c^{(r)}_{n,n} \end{array} \right] x^{(r+1)}_{n-\beta+1\to n} = d^{(r)}_{n+1-\beta\to n}. \end{aligned} \qquad (2.3)$$

This is a block tridiagonal system with each block of size $\beta \times \beta$. Once the core system is solved, solution of partitioned subsystems can be obtained, in parallel, from (2.2) by back substitution.

3. The Algorithm

Step 1. Compact Elimination stage.

For $k = 1$ to $n - \beta$

Compute *in parallel* for $r = 1, ..., p$

(i) for $j = k + 1$ to $k + \beta$

$b^{(r)}_{j,k+1\to k+\beta} := b^{(r)}_{j,k+1\to k+\beta} - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k} \right) b^{(r)}_{k,k+1\to k+\beta},$

$a^{(r)}_{j,n-\beta+1\to n} := a^{(r)}_{j,n-\beta+1\to n} - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k} \right) a^{(r)}_{k,n-\beta+1\to n}$ (omit for $r = 1$),

$d^{(r)}_j := d^{(r)}_j - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k} \right) d^{(r)}_k.$

(ii) (omit for $r = 1$) for $j = n - \beta + 1$ to n

$c^{(r-1)}_{j,k+1\to k+\beta} := c^{(r-1)}_{j,k+1\to k+\beta} - \left(c^{(r-1)}_{j,k} \big/ b^{(r)}_{k,k} \right) b^{(r)}_{k,k+1\to k+\beta},$

$b^{(r-1)}_{j,n-\beta+1\to n} := b^{(r-1)}_{j,n-\beta+1\to n} - \left(c^{(r-1)}_{j,k} \big/ b^{(r)}_{k,k} \right) a^{(r)}_{k,n-\beta+1\to n},$

$d^{(r-1)}_j := d^{(r-1)}_j - \left(c^{(r-1)}_{j,k} \big/ b^{(r)}_{k,k} \right) d^{(r)}_k.$

Step 2. Core system

Solve the core system (2.3) for $x^{(r)}_{n-\beta+1\to n}$ for $r = 1, ..., p$.

Elimination in the core system

For $r = 1$ to p

For $k = n - \beta + 1$ to n

(i) (omit for $k = n$) for $j = k + 1$ to n

$b^{(r)}_{j,k+1\to n} := b^{(r)}_{j,k+1\to n} - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k} \right) b^{(r)}_{k,k+1\to n},$

$c^{(r)}_{j,n-\beta+1\to n} := c^{(r)}_{j,n-\beta+1\to n} - \left(b^{(r)}_{j,k} \big/ b^{(r)}_{k,k} \right) c^{(r)}_{k,n-\beta+1\to n}$ (omit for $r = p$),

$$d_j^{(r)} := d_j^{(r)} - \left(b_{j,k}^{(r)} \diagup b_{k,k}^{(r)}\right) d_k^{(r)}.$$

(ii) (omit for $r = p$) for $j = n - \beta + 1$ to n

$$a_{j,k+1\to n}^{(r+1)} := a_{j,k+1\to n}^{(r+1)} - \left(a_{j,k}^{(r+1)} \diagup b_{k,k}^{(r)}\right) b_{k,k+1\to n}^{(r)},$$

$$b_{j,n-\beta+1\to n}^{(r+1)} := b_{j,n-\beta+1\to n}^{(r+1)} - \left(a_{j,k}^{(r+1)} \diagup b_{k,k}^{(r)}\right) c_{k,n-\beta+1\to n}^{(r)},$$

$$d_j^{(r+1)} := d_j^{(r+1)} - \left(a_{j,k}^{(r+1)} \diagup b_{k,k}^{(r)}\right) d_k^{(r)}.$$

Solution of the core system

For $r = p$ down to 1

For $k = n$ down to $n - \beta + 1$ (omit second summation for $r = p$)

$$x_k^{(r)} = \left(1 \diagup b_{k,k}^{(r)}\right) \left[d_k^{(r)} - \sum_{s=k+1}^{n} b_{k,s}^{(r)} x_s^{(r)} - \sum_{s=n-\beta+1}^{n} c_{k,s}^{(r)} x_s^{(r+1)}\right].$$

Step 3. Solution stage

Compute *in parallel* for $r = 1, ..., p$

For $k = n - \beta$ down to 1 (omit second summation for $r = 1$)

$$x_k^{(r)} = \left(1 \diagup b_{k,k}^{(r)}\right) \left[d_k^{(r)} - \sum_{s=k+1}^{k+\beta} b_{k,s}^{(r)} x_s^{(r)} - \sum_{s=n-\beta+1}^{n} a_{k,s}^{(r)} x_s^{(r-1)}\right].$$

4. Arithmetic Complexity

We next report on the arithmetic operations complexity of the parallel algorithm presented above. In doing the arithmetical operations counts, in the following we shall denote the number of additions by A, the number of multiplications by M, the number of divisions by D, the total of additions, multiplications and divisions by T. Finally, the grand total of all these operations in the complete algorithm will be denoted by GT. We first report on the operations counts involved in a serial implementation of the presented algorithm.

4.1. Operations Counts for Serial Implementation

The major part of the computational effort is in the elimination stage in Step 1 of the algorithm. First note that in each block for $r = 2(1)p$, $\beta n - (1/2)\beta(\beta+1)$ updates for each of the a's and c's are just fill-ins. Taking this into account, the elimination stage involves

$A = 4\beta^2 N - \beta(3\beta-1)n - \beta(4\beta^2+\beta-1)p + \beta(3\beta^2-1)$, $M = 2\beta(2\beta+1)N - \beta(3\beta+1)n - 2\beta^2(2\beta+1)p + \beta^2(3\beta+1)$,

$D = 2\beta N - \beta n - 2\beta^2 p + \beta^2$.

The total of operations for the elimination stage in Step l is

$$T\left(\text{elimination}\right) = 4\beta\left(2\beta+1\right)N - \beta\left(6\beta+1\right)n - \beta\left(8\beta^2+5\beta-1\right)p + \beta\left(6\beta^2+2\beta-1\right). \quad (4.1)$$

In Step 2 for the solution of the core system, elimination in the core system involves

$A = M = \frac{1}{3}\beta\left(7\beta^2-1\right)p - 2\beta^3$,

$D = \frac{1}{2}\beta\left(3\beta-1\right)p - \beta^2$.

Solution of the core system involves

$A = M = \frac{1}{2}\beta\left(3\beta-1\right)p - \beta^2$, $D = \beta p$.

The total of operations involved in the solution of the core system is

$$T\left(\text{core system}\right) = \frac{1}{6}\beta\left(28\beta^2+27\beta-7\right)p - \beta^2\left(4\beta+3\right). \quad (4.2)$$

Step 3 of the solution in the partitioned blocks involves

$A = M = 2\beta N - \beta n - 2\beta^2 p + \beta^2$, $D = N - \beta p$.

The total of operations involved in the solution stage is

$$T\left(\text{solution}\right) = \left(4\beta+1\right)N - 2\beta n - \beta\left(4\beta+1\right)p + 2\beta^2. \quad (4.3)$$

Thus, the grand total of operations involved in a serial implementation of the presented algorithm for the solution of a β-semiband linear system employing p partitions is given by

$$GT_{\beta;p}\left(\text{serial}\right) = \left(8\beta^2+8\beta+1\right)N - 3\beta\left(2\beta+1\right)n - \frac{1}{6}\beta\left(20\beta^2+27\beta+7\right)p + \beta\left(2\beta^2+\beta-1\right). \quad (4.4)$$

We note here the following special cases of (4.4). If $p = 1$, then

$$GT_{\beta;1}\,(\text{serial}) = \left(2\beta^2 + 5\beta + 1\right) N - \frac{1}{6}\beta\left(8\beta^2 + 21\beta + 13\right)$$

agrees with the total count of operations for the serial LU factorization method for the solution of β-semiband linear systems as reported in Chawla and Evans [1]. If $\beta = 1$, then

$$GT_{1;p}\,(\text{serial}) = 17N - 9n - 9p + 2$$

agrees with the count for the solution of tridiagonal linear systems using p partitions as reported in Chawla and Khazal [2]. If $\beta = 2$, then

$$GT_{2;p}\,(\text{serial}) = 49N - 30n - 47p + 18$$

agrees with the count for the solution of pentadiagonal linear systems using p partitions as reported in Khazal and Chawla [15]. (It may be pointed out here that there is a small correction in the counts reported in Khazal and Chawla [15]. For the algorithm presented there, in Step 2, eliminations involve $18p-16$ of each of additions and multiplications. This has the effect of subtracting $8\,(p-1)$ operations from the grand total reported therein.)

To provide a comparison with the algorithm in Chawla and Passi [3], we note that the grand total of operations for the algorithm in [3] is

$$\begin{aligned} GT^{C-P}_{\beta;p}\,(\text{serial}) \;=\; & \frac{1}{12}\left(124\beta^2 + 69\beta + 5\right) N - \frac{1}{6}\beta\left(44\beta^2 + 39\beta + 19\right) p \\ & -\frac{1}{3}\beta\left(32\beta^2 - 6\beta - 5\right). \end{aligned}$$

Therefore, for $\beta \ll N$, in comparison with the grand total of operations (3.4) for our present algorithm,

$$GT^{C-P}_{\beta;p}\,(\text{serial}) - GT_{\beta;p}\,(\text{serial}) = \frac{1}{12}\left(28\beta^2 - 27\beta - 7\right) N.$$

It is easy to see now that for tridiagonal linear systems the algorithm of Chawla and Passi [3] has a lesser operations count, while for $\beta > 1$ our present algorithm has a lesser operations count.

4.2. Operations Counts for the Parallel Algorithm

We next report on arithmetical operations involved in a parallel implementation of the presented algorithm.

Again, accounting for the updates in a's and c's which are just fill-ins, Step 1 of elimination involves

$$A = 4\beta^2 n - \beta\left(4\beta^2 + \beta - 1\right),$$

$$M = 2\beta\left(2\beta + 1\right)n - 2\beta^2\left(2\beta + 1\right),$$

$$D = 2\beta n - 2\beta^2.$$

The total number of operations involved in the parallel elimination stage is

$$T\left(\text{parallel elimination}\right) = 4\beta\left(2\beta + 1\right)n - \beta\left(8\beta^2 + 5\beta - 1\right). \tag{4.5}$$

The counts for the solution of the core system in Step 2 remain the same as before given in (3.2). Step 3 of parallel solution in partitioned blocks now involves
$A = M = 2\beta n - 2\beta^2$, $D = n - \beta$,
giving a total of

$$T\left(\text{parallel solution}\right) = \left(4\beta + 1\right)n - \beta\left(4\beta + 1\right). \tag{4.6}$$

Thus, the grand total of arithmetic operations involved in the presented parallel algorithm is

$$GT_{\beta;p}\left(\text{parallel}\right) = \left(8\beta^2 + 8\beta + 1\right)n + \frac{1}{6}\beta\left(28\beta^2 + 27\beta - 7\right)p - 12\beta^2\left(\beta + 1\right). \tag{4.7}$$

We note here the following particular cases of (4.7). If $\beta = 1$, then

$$GT_{1;p}\left(\text{parallel}\right) = 17n + 8p - 24$$

agrees with the count for the parallel algorithm for tridiagonal linear systems as reported in Chawla and Khazal [2]. If $\beta = 2$, then

$$GT_{2;p}\left(\text{parallel}\right) = 49n + 53p - 144$$

agrees with the count for the parallel algorithm for pentadiagonal linear systems as reported in Khazal and Chawla [15] (with the correction as indicated above).

4.3. Efficiency

In terms of efficiency of processor utilization as defined by Kuck [16], for a general number of processors, we can define efficiency for the presented parallel algorithm, versus the serial LU algorithm, by

$$e\left(\beta\right)=\frac{2\beta^{2}+5\beta+1}{8\beta^{2}+8\beta+1}. \tag{4.8}$$

It is easy to see that $e\left(\beta\right)$ is monotonic decreasing, $\frac{1}{4} \lessdot e\left(\beta\right) \leq \frac{8}{17}$. In particular, this implies an efficiency of nearly 47% for tridiagonal systems, 38% for pentadiagonal systems and 35% for septadiagonal linear systems. These efficiency indicators may not be surprising since, as has been noted earlier, no parallel algorithm for tridiagonal linear systems is known with efficiency exceeding 48.5%.

5. Illustration of the Algorithm

To illustrate the main idea of the working of the parallel algorithm presented above, we consider an example for $N=9$ and $p=3$ with $A=trid\{-1,2,-1\}$, $\mathbf{d}=(1,0,...,0,1)^{\top}$. The steps of the algorithm are as follows.

Step 1. Elimination stage $k=1$:

$$\left[\begin{array}{ccc|ccc|ccccc}
2 & -1 & & & & & & & & : & 1 \\
0 & \frac{3}{2} & -1 & & & & & & & : & \frac{1}{2} \\
 & -1 & \frac{3}{2} & 0 & -\frac{1}{2} & & & & & : & 0 \\
\hline
 & & -1 & 2 & -1 & & & & & : & 0 \\
 & & -\frac{1}{2} & 0 & \frac{3}{2} & -1 & & & & : & 0 \\
 & & & & -1 & \frac{3}{2} & 0 & -\frac{1}{2} & & : & 0 \\
\hline
 & & & & & -1 & 2 & -1 & & : & 0 \\
 & & & & & -\frac{1}{2} & 0 & \frac{3}{2} & -1 & : & 0 \\
 & & & & & & & -1 & 2 & : & 1
\end{array}\right].$$

$k = 2$:

$$\left[\begin{array}{ccc|ccc|ccccc}
2 & -1 & & & & & & & & : & 1 \\
0 & \frac{3}{2} & -1 & & & & & & & : & \frac{1}{2} \\
 & 0 & \boxed{\frac{2}{3}} & 0 & 0 & \boxed{-\frac{1}{3}} & & & & : & \boxed{\frac{1}{3}} \\
\hline
 & & -1 & 2 & -1 & & & & & : & 0 \\
 & & -\frac{1}{2} & 0 & \frac{3}{2} & -1 & & & & : & 0 \\
 & & \boxed{-\frac{1}{3}} & & 0 & \boxed{\frac{2}{3}} & 0 & 0 & \boxed{-\frac{1}{3}} & : & \boxed{0} \\
\hline
 & & & & & -1 & 2 & -1 & & : & 0 \\
 & & & & & -\frac{1}{2} & 0 & \frac{3}{2} & -1 & : & 0 \\
 & & & & & \boxed{-\frac{1}{3}} & & 0 & \boxed{\frac{4}{3}} & : & \boxed{1}
\end{array}\right].$$

The resulting core system is indicated by boxes in the above reduced augmented matrix. Solving this reduced system, we obtain

$$x_3^{(1)} = 1,\ x_3^{(2)} = 1,\ x_3^{(3)} = 1.$$

Now, the three partitioning blocks uncouple and the remaining solution can be obtained in parallel from each block by back substitution, giving

$$\begin{aligned}
x_2^{(1)} &= 1,\ x_2^{(2)} = 1,\ x_2^{(3)} = 1, \\
x_1^{(1)} &= 1,\ x_1^{(2)} = 1,\ x_1^{(3)} = 1.
\end{aligned}$$

References

[1] M.M. Chawla and D.J. Evans, Arithmetic complexity of direct solvers for banded systems, *Intern. J. Comput. Math.*, To appear.

[2] M.M. Chawla and R.R. Khazal, Parallel elimination in partitioned tridiagonal systems, *Intern. J. Computer Math.*, **58** (1995), 249-252.

[3] M.M. Chawla and K. Passi, A parallel elimination method for the solution of banded linear systems, *Intern.J. Computer Math.*, **50** (1994), 197-201.

[4] M.M. Chawla, M.A. Al-Zanaidi and A.Z. Al-Shammeri, High-accuracy finite-difference schemes for the diffusion equation, *Neural, Parallel & Sci. Computations*, **6** (1998), 523-535.

[5] M.M. Chawla, M.A. Al-Zanaidi and D.J. Evans, Generalized trapezoidal formulas for parabolic equations, *Intern. J. Computer Math.*, **70** (1999), 429-443.

[6] M.M. Chawla, M.A. Al-Zanaidi and D.J. Evans, Lod generalized trapezoidal formula schemes for parabolic differential equations in two space dimensions, *Intern. J. Computer Math.*, **78** (2001), 269-287.

[7] M.M. Chawla, R.R. Khazal and H.F. Al-Fahd, A direct incomplete factorization method for parallel solution of tridiagonal linear systems, *Intern. J. Computer Math.*, **54** (1994), 249-259.

[8] M.M. Chawla, K. Passi and P.N. Shivakumar, A fast parallel algorithm for the solution of tridiagonal linear systems, *Intern. J. Computer Math.*, **45** (1992), 113-121.

[9] L. Fox, *An Introduction to Numerical Linear Algebra*, Oxford Univ. Press, Oxford (1964).

[10] T.L Freeman and C. Phillips, *Parallel Numerical Algorithms*, Prentice Hall International (UK) Ltd., Hemel Hemstead, Hertfordshire (1992).

[11] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2-nd Edn., The Johns Hopkins Univ. Press, Baltimore (1989).

[12] R.J. Goutl, R.F. Hoskins, J.A. Milner and M.J. Pratt, *Computational Methods in Linear Algebra*, John Wiley (Halstead Press), New York (1974).

[13] S.L. Johnsson, Solving narrow banded systems on ensemble architectures, *ACM Trans. Math. Software*, **11** (1985), 271-288.

[14] S.L. Johnsson, Solving tridiagonal systems on ensemble architectures, *SIAM Sci. Stat. Comput.*, **8** (1987), 354-392.

[15] R.R. Khazal and M.M. Chawla, A parallel elimination algorithm for pentadiagonal linear systems, *Intern. J. Pure Appl. Math.*, **1**, No. 1 (2002), 95-112.

[16] D.J. Kuck, *The Structure of Computers and Computations*, Vol. **1**, John Wiley, New York (1978).

[17] D.H. Lawrie and A.H. Sameh, The computational and communication complexity of a parallel banded system solver, *ACM Trans. Math. Software*, **10** (1984), 185-195.

[18] U. Meier, A parallel partition method for solving banded linear systems, *Parallel Comput.*, **2** (1985), 33-43.

[19] J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York (1988).

[20] J.W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Springer-Verlag, New York (1995).

[21] H.H. Wang, A parallel method for tridiagonal equations, *ACM Trans. Math. Software*, **7** (1981), 170-183.

[22] H.H. Wang, The partition method for solving tridiagonal equations on multiprocessor computers, Document No. 320-3499, *IBM Palo Alto Scientific Center*, Palo Alto, CA 94304 (May 1987).