International Journal of Pure and Applied Mathematics
Volume 1 No. 1 2002, 89-106

A PARALLEL ELIMINATION ALGORITHM FOR PENTADIAGONAL LINEAR SYSTEMS

R.R. Khazal[§], M.M. Chawla
[1]Dept. of Mathematics and Computer Science
Kuwait University
P.O. Box 5969, Safat 13060, KUWAIT

Abstract: Classical elimination procedure is extended to uncouple partitioned pentadiagonal linear systems for parallel processing of their solution. In each block of equations, we need two sets of simultaneous eliminations; each set consists of two usual forward eliminations and two backward from across the succeeding block. While vertical fill-ins in the last two columns of the block on the left pose no difficulty, the purpose of the indicated eliminations is to move fill-ins in the last two rows successively two columns to the right till they reach their destination in the last two columns of each block. At the end of the elimination stage, we reach a relatively small size 2×2 block tridiagonal core system. Once the core system is solved, the blocks of equations uncouple and the uncoupled subsystems are solved in parallel by back substitution. We include arithmetical operations counts for both serial and parallel implementations of the presented algorithm and illustrate the working of the algorithm by an example.

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

Received: January 20, 2002

[§]Correspondence author

1. Introduction

A class of linear systems occurring frequently in applications is that of pentadiagonal systems:

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

where

$$A = \begin{bmatrix} b_1 & c_1 & d_1 & & \\ a_2 & b_2 & c_2 & d_2 & \\ e_3 & a_3 & b_3 & c_3 & d_3 \\ & \cdot & \cdot & \cdot & \cdot \\ & & e_N & a_N & b_N \end{bmatrix}, \ \mathbf{x} = (x_1, ..., x_N)^\top, \ \mathbf{r} = (r_1, ..., r_N)^\top.$$

Pentadiagonal systems occur, e.g. in the numerical solution of partial differential equations. For the diffusion equation, a well known time integration scheme is the Crank-Nicolson scheme which requires the solution of tridiagonal linear systems at each time step of integration (see e.g. Thomas [15]). Tridiagonal linear systems can be said to lie at the heart of scientific computing. Since approximations provided by the Crank-Nicolson scheme can suffer unwanted oscillations in the computed solution, recently Chawla and Al-Zanaidi [3] and Chawla et al [5] have proposed new schemes for the diffusion, convection-diffusion and reaction-diffusion equations which provide the solution more stably. These schemes require the solution of pentadiaognal linear systems at each time step of integration. The extra cost involved in the solution of pentadiagonal systems is more than compensated by the accuracy and stability of the computed solutions provided by these schemes. These time integration schemes have been extended recently by Chawla and Al-Zanaidi [4] and Chawla et al [6] for the diffusion, convection-diffusion and reaction-diffusion equations in two space dimensions. Since the schemes also require the solution of a pentadiagonal system at each time step of integration, it becomes imperative to explore the possibility of parallel algorithms for the solution of pentadiagonal systems. The present paper is intended in this context.

There are few parallel solvers available in the literature for pentadiaognal linear systems; see e.g. Freeman and Phillips [8], Golub and Van Loan [9], Kuck [13] and Ortega [14]. The situation is somewhat different for tridiagonal systems. Parallel algorithms for tridiagonal linear systems have seen some recent advances. Elimination (equivalently, LU-factorization) remains the best known serial method for the solution of tridiagonal systems. The algorithm of Thomas (see Fox [7], Goult et al [10]) is well known and it is widely used (see e.g. Thomas [15]). Wang [16], [17] had suggested an elimination method for partitioned tridiagonal systems. Wang's elimination procedure consisted 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 [11], [12] in his factorization method. More recently, Chawla et al [2] proposed an incomplete factorization method and Chawla and Khazal [1] have proposed one-stage parallel elimination algorithm for partitioned tridiagonal linear systems. The present paper is in the same vein.

In the present paper, classical elimination procedure is extended to uncouple partitioned pentadiagonal systems for parallel processing of their solution. In each block of equations, we need two sets of simultaneous eliminations; each set consists of two usual forward eliminations and two backward from across the succeeding block. While vertical fill-ins in the last two columns of the block on the left pose no difficulty, the purpose of the indicated eliminations is to move the fill-ins in the last two rows successively two columns to the right till they reach their destination in the last two columns of each block. At the end of the elimination stage, we reach a relatively small size 2×2 block tridiagonal core system. Once the core system is solved, the blocks of equations uncouple and the uncoupled subsystems are solved in parallel by back substitution. We include arithmetical operations counts for both serial and parallel implementations of the presented algorithm and illustrate the working of the algorithm by an example.

It might be worth mentioning here that the arithmetical operations count for the best serial algorithm for tridiagonal systems (Thomas' algorithm) is O(8N). All parallel algorithms that have been reported in the literature, including the well known cyclic reduction algorithm, for a general number of processors, have a serial operations count of O(17N). Thus, in terms of efficiency of processor utilization as has been defined by Kuck [13], no parallel algorithm is known so far which has an efficiency exceeding 47%.

A best serial algorithm for pentadiagonal systems (see Section 1.1) has an arithmetical operations count of O(19N). The parallel algorithm presented in this paper has a serial arithmetical operations count of O(49N); see Section 2.2. Thus, our present parallel algorithm can achieve an efficiency of slightly in excess of 38%. Therefore, one of the purposes of this paper is also to initiate further research to investigate if, for a general number of processors, a faster parallel algorithm can be achieved for the solution of pentadiagonal linear systems.

1.1. Serial Elimination Algorithm for Pentadiagonal Systems

In the description of the algorithm, the elements of A and $\mathbf{r}$ are overwritten during updation.

Step 1. Elimination stage.

For $j = 1(1)N-2$:

$$\lambda_{j+1} = a_{j+1}/b_j,\ [b_{j+1}, c_{j+1}, r_{j+1}] := [b_{j+1}, c_{j+1}, r_{j+1}] - \lambda_{j+1}[c_j, d_j, r_j],$$

$$\mu_{j+1} = e_{j+2}/b_j,\ [a_{j+2}, b_{j+2}, r_{j+2}] := [a_{j+2}, b_{j+2}, r_{j+2}] - \mu_{j+2}[c_j, d_j, r_j].$$

For $j = N-1$:

$$\lambda_N = a_N/b_{N-1},\ [b_N, r_N] := [b_N, r_N] - \lambda_N[c_{N-1}, r_{N-1}].$$

Step 2. Solution stage.

$$x_N = r_N/b_N,\ x_{N-1} = (r_{N-1} - c_{N-1}x_N)/b_{N-1},$$

For $j = N-2$ down to 1

$$x_j = (r_j - c_j x_{j+1} - d_j x_{j+2})/b_j.$$

Step 1 of eliminations involves 6N-10 additions, 6N-10 multiplications and 2N-3 divisions. Step 2 of the solution stage involves 2N-3 additions, 2N-3 multiplications and N divisions. Thus, the grand total of arithmetical operations count for the serial algorithm is 19N-29.

2. 2. A Parallel Elimination Algorithm for Pentadiagonal Systems

With $N = pn$, n even, consider the system in (1.1) partitioned into p blocks each consisting of n consecutive equations. In the following we use the notation:

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

For $i \in \{1, ..., p\}$, let $E^{(i)}$ denote the i-th block of equations:

$$\begin{bmatrix} & & & e_1^{(i)} & a_1^{(i)} \\ & & & & e_2^{(i)} \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \mathbf{x}^{(i-1)} + \begin{bmatrix} b_1^{(i)} & c_1^{(i)} & d_1^{(i)} & & & \\ a_2^{(i)} & b_2^{(i)} & c_2^{(i)} & d_2^{(i)} & & \\ & \cdot & \cdot & \cdot & & \\ & e_{n-1}^{(i)} & a_{n-1}^{(i)} & b_{n-1}^{(i)} & c_{n-1}^{(i)} & \\ & & e_n^{(i)} & a_n^{(i)} & b_n^{(i)} & \end{bmatrix} \mathbf{x}^{(i)}$$

$$+ \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ d_{n-1}^{(i)} & & & & \\ c_n^{(i)} & d_n^{(i)} & & & \end{bmatrix} \mathbf{x}^{(i+1)} = \mathbf{r}^{(i)}. \tag{2.1}$$

For $k \in \{1, ..., n\}$, let $E_k^{(i)}$ denote the k-th equation in block $E^{(i)}$.

Assume $j - 1$ eliminations have been carried out; we describe step $j \in \left\{1, ..., \frac{n}{2} - 1\right\}$. In doing so, in the following we overwrite A and $\mathbf{r}$ as their elements get updated during each elimination. Now, we need two sets of eliminations, each set consisting of four *simultaneous* eliminations as defined in the following.

(i) With $E_{2j-1}^{(i)}$, eliminate $x_{2j-1}^{(i)}$ from $E_{2j}^{(i)}$, $E_{2j+1}^{(i)}$, $E_n^{(i-1)}$ and $E_{n-1}^{(i-1)}$; then

(ii) with $E_{2j}^{(i)}$, eliminate $x_{2j}^{(i)}$ from $E_{2j+1}^{(i)}$, $E_{2j+2}^{(i)}$, $E_n^{(i-1)}$ and $E_{n-1}^{(i-1)}$.

We next consider the effect of these eliminations on the system. For the first set of eliminations, consider:

modified

$$E_{2j-1}^{(i)} : v_{2j-1}^{(i)} x_{n-1}^{(i-1)} + u_{2j-1}^{(i)} x_n^{(i-1)} + b_{2j-1}^{(i)} x_{2j-1}^{(i)} + c_{2j-1}^{(i)} x_{2j}^{(i)} + d_{2j-1}^{(i)} x_{2j+1}^{(i)}$$
$$= r_{2j-1}^{(i)};$$

unmodified:

$$E_{2j}^{(i)} : v_{2j}^{(i)} x_{n-1}^{(i-1)} + u_{2j}^{(i)} x_n^{(i-1)} + a_{2j}^{(i)} x_{2j-1}^{(i)} + b_{2j}^{(i)} x_{2j}^{(i)} + c_{2j}^{(i)} x_{2j+1}^{(i)} + d_{2j}^{(i)} x_{2j+2}^{(i)}$$
$$= r_{2j}^{(i)};$$

$$E_{2j+1}^{(i)} : v_{2j+1}^{(i)} x_{n-1}^{(i-1)} + u_{2j+1}^{(i)} x_n^{(i-1)} + e_{2j+1}^{(i)} x_{2j-1}^{(i)} + a_{2j+1}^{(i)} x_{2j}^{(i)} + b_{2j+1}^{(i)} x_{2j+1}^{(i)}$$
$$+ c_{2j+1}^{(i)} x_{2j+2}^{(i)} + d_{2j+1}^{(i)} x_{2j+3}^{(i)} = r_{2j+1}^{(i)};$$

$E_n^{(i-1)} : e_n^{(i-1)} x_{n-2}^{(i-1)} + a_n^{(i-1)} x_{n-1}^{(i-1)} + b_n^{(i-1)} x_n^{(i-1)} + s_{2j-1}^{(i-1)} x_{2j-1}^{(i)} + s_{2j}^{(i-1)} x_{2j}^{(i)}$
$= r_n^{(i-1)};$

$E_{n-1}^{(i-1)} : e_{n-1}^{(i-1)} x_{n-3}^{(i-1)} + a_{n-1}^{(i-1)} x_{n-2}^{(i-1)} + b_{n-1}^{(i-1)} x_{n-1}^{(i-1)} + c_{n-1}^{(i-1)} x_n^{(i-1)}$

$+ t_{2j-1}^{(i-1)} x_{2j-1}^{(i)} + t_{2j}^{(i-1)} x_{2j}^{(i)} = r_{n-1}^{(i-1)}.$

Note that $u_1^{(i)} = a_1^{(i)}$, $u_2^{(i)} = e_2^{(i)}$, $v_1^{(i)} = e_1^{(i)}$, $v_2^{(i)} = 0$; $s_1^{(i)} = c_n^{(i)}$, $s_2^{(i)} = d_n^{(i)}$, $t_1^{(i)} = d_{n-1}^{(i)}$, $t_2^{(i)} = 0$. Also note that there are no u and v for the first block for $i = 1$ and there are no s and t for the last block for $i = p$.

Now, for the desired eliminations in (i):

modified

$E_{2j}^{(i)} := E_{2j}^{(i)} - \alpha_{2j}^{(i)} E_{2j-1}^{(i)}$, $\alpha_{2j}^{(i)} = a_{2j}^{(i)} / b_{2j-1}^{(i)} \Rightarrow$

$v_{2j}^{(i)} x_{n-1}^{(i-1)} + u_{2j}^{(i)} x_n^{(i-1)} + b_{2j}^{(i)} x_{2j}^{(i)} + c_{2j}^{(i)} x_{2j+1}^{(i)} + d_{2j}^{(i)} x_{2j+2}^{(i)} = r_{2j}^{(i)},$

where

$$\left[v_{2j}^{(i)}, u_{2j}^{(i)}, b_{2j}^{(i)}, c_{2j}^{(i)}; r_{2j}^{(i)}\right] := \left[v_{2j}^{(i)}, u_{2j}^{(i)}, b_{2j}^{(i)}, c_{2j}^{(i)}; r_{2j}^{(i)}\right]$$
$$-\alpha_{2j}^{(i)} \left[v_{2j-1}^{(i)}, u_{2j-1}^{(i)}, c_{2j-1}^{(i)}, d_{2j-1}^{(i)}; r_{2j-1}^{(i)}\right].$$

$E_{2j+1}^{(i)} := E_{2j+1}^{(i)} - \beta_{2j+1}^{(i)} E_{2j-1}^{(i)}$, $\beta_{2j+1}^{(i)} = e_{2j+1}^{(i)} / b_{2j-1}^{(i)} \Rightarrow$

$v_{2j+1}^{(i)} x_{n-1}^{(i-1)} + u_{2j+1}^{(i)} x_n^{(i-1)} + a_{2j+1}^{(i)} x_{2j}^{(i)} + b_{2j+1}^{(i)} x_{2j+1}^{(i)} + c_{2j+1}^{(i)} x_{2j+2}^{(i)}$
$+ d_{2j+1}^{(i)} x_{2j+3}^{(i)} = r_{2j+1}^{(i)},$

where

$$\left[v_{2j+1}^{(i)}, u_{2j+1}^{(i)}, a_{2j+1}^{(i)}, b_{2j+1}^{(i)}; r_{2j+1}^{(i)}\right] := \left[v_{2j+1}^{(i)}, u_{2j+1}^{(i)}, a_{2j+1}^{(i)}, b_{2j+1}^{(i)}; r_{2j+1}^{(i)}\right]$$
$$-\beta_{2j+1}^{(i)} \left[v_{2j-1}^{(i)}, u_{2j-1}^{(i)}, c_{2j-1}^{(i)}, d_{2j-1}^{(i)}; r_{2j-1}^{(i)}\right].$$

$E_n^{(i-1)} := E_n^{(i-1)} - \gamma_{2j-1}^{(i-1)} E_{2j-1}^{(i)}$, $\gamma_{2j-1}^{(i-1)} = s_{2j-1}^{(i-1)} / b_{2j-1}^{(i)} \Rightarrow$

$e_n^{(i-1)} x_{n-2}^{(i-1)} + a_n^{(i-1)} x_{n-1}^{(i-1)} + b_n^{(i-1)} x_n^{(i-1)} + s_{2j}^{(i-1)} x_{2j}^{(i)} + s_{2j+1}^{(i-1)} x_{2j+1}^{(i)}$

$= r_n^{(i-1)},$

where

$$\left[a_n^{(i-1)}, b_n^{(i-1)}, s_{2j}^{(i-1)}, s_{2j+1}^{(i-1)}; r_n^{(i-1)}\right] := \left[a_n^{(i-1)}, b_n^{(i-1)}, s_{2j}^{(i-1)}, 0; r_n^{(i-1)}\right]$$
$$-\gamma_{2j-1}^{(i-1)}\left[v_{2j-1}^{(i)}, u_{2j-1}^{(i)}, c_{2j-1}^{(i)}, d_{2j-1}^{(i)}; r_{2j-1}^{(i)}\right].$$

$$E_{n-1}^{(i-1)} := E_{n-1}^{(i-1)} - \delta_{2j-1}^{(i-1)} E_{2j-1}^{(i)},\ \delta_{2j-1}^{(i-1)} = t_{2j-1}^{(i-1)}/b_{2j-1}^{(i)} \Rightarrow$$

$$e_{n-1}^{(i-1)} x_{n-3}^{(i-1)} + a_{n-1}^{(i-1)} x_{n-2}^{(i-1)} + b_{n-1}^{(i-1)} x_{n-1}^{(i-1)} + c_{n-1}^{(i-1)} x_n^{(i-1)} + t_{2j}^{(i-1)} x_{2j}^{(i)}$$
$$+t_{2j+1}^{(i-1)} x_{2j+1}^{(i)} = r_{n-1}^{(i-1)},$$

where

$$\left[b_{n-1}^{(i-1)}, c_{n-1}^{(i-1)}, t_{2j}^{(i-1)}, t_{2j+1}^{(i-1)}; r_{n-1}^{(i-1)}\right] := \left[b_{n-1}^{(i-1)}, c_{n-1}^{(i-1)}, t_{2j}^{(i-1)}, 0; r_{n-1}^{(i-1)}\right]$$
$$-\delta_{2j-1}^{(i-1)}\left[v_{2j-1}^{(i)}, u_{2j-1}^{(i)}, c_{2j-1}^{(i)}, d_{2j-1}^{(i)}; r_{2j-1}^{(i)}\right].$$

For the second set of eliminations, consider:

unmodified

$$E_{2j+2}^{(i)} : v_{2j+2}^{(i)} x_{n-1}^{(i-1)} + u_{2j+2}^{(i)} x_n^{(i-1)} + e_{2j+2}^{(i)} x_{2j}^{(i)} + a_{2j+2}^{(i)} x_{2j+1}^{(i)} + b_{2j+2}^{(i)} x_{2j+2}^{(i)}$$
$$+c_{2j+2}^{(i)} x_{2j+3}^{(i)} + d_{2j+2}^{(i)} x_{2j+4}^{(i)} = r_{2j+2}^{(i)}.$$

Now, for the desired eliminations in (ii):

modified

$$E_{2j+1}^{(i)} := E_{2j+1}^{(i)} - \lambda_{2j+1}^{(i)} E_{2j}^{(i)},\ \lambda_{2j+1}^{(i)} = a_{2j+1}^{(i)}/b_{2j}^{(i)} \Rightarrow$$

$$v_{2j+1}^{(i)} x_{n-1}^{(i-1)} + u_{2j+1}^{(i)} x_n^{(i-1)} + b_{2j+1}^{(i)} x_{2j+1}^{(i)} + c_{2j+1}^{(i)} x_{2j+2}^{(i)} + d_{2j+1}^{(i)} x_{2j+3}^{(i)}$$
$$= r_{2j+1}^{(i)},$$

where

$$\left[v_{2j+1}^{(i)}, u_{2j+1}^{(i)}, b_{2j+1}^{(i)}, c_{2j+1}^{(i)}; r_{2j+1}^{(i)}\right] := \left[v_{2j+1}^{(i)}, u_{2j+1}^{(i)}, b_{2j+1}^{(i)}, c_{2j+1}^{(i)}; r_{2j+1}^{(i)}\right]$$
$$-\lambda_{2j+1}^{(i)}\left[v_{2j}^{(i)}, u_{2j}^{(i)}, c_{2j}^{(i)}, d_{2j}^{(i)}; r_{2j}^{(i)}\right].$$

$$E_{2j+2}^{(i)} := E_{2j+2}^{(i)} - \mu_{2j+2}^{(i)} E_{2j}^{(i)},\ \mu_{2j+2}^{(i)} = e_{2j+2}^{(i)}/b_{2j}^{(i)} \Rightarrow$$

$$v_{2j+2}^{(i)} x_{n-1}^{(i-1)} + u_{2j+2}^{(i)} x_n^{(i-1)} + a_{2j+2}^{(i)} x_{2j+1}^{(i)} + b_{2j+2}^{(i)} x_{2j+2}^{(i)} + c_{2j+2}^{(i)} x_{2j+3}^{(i)}$$
$$+d_{2j+2}^{(i)} x_{2j+4}^{(i)} = r_{2j+2}^{(i)},$$

where

$$\left[v^{(i)}_{2j+2}, u^{(i)}_{2j+2}, a^{(i)}_{2j+2}, b^{(i)}_{2j+2}; r^{(i)}_{2j+2}\right] := \left[v^{(i)}_{2j+2}, u^{(i)}_{2j+2}, a^{(i)}_{2j+2}, b^{(i)}_{2j+2}; r^{(i)}_{2j+2}\right]$$
$$-\mu^{(i)}_{2j+2}\left[v^{(i)}_{2j}, u^{(i)}_{2j}, c^{(i)}_{2j}, d^{(i)}_{2j}; r^{(i)}_{2j}\right].$$

$$E^{(i-1)}_{n} := E^{(i-1)}_{n} - \sigma^{(i-1)}_{2j} E^{(i)}_{2j},\ \sigma^{(i-1)}_{2j} = s^{(i-1)}_{2j}/b^{(i)}_{2j} \Rightarrow$$

$$e^{(i-1)}_{n} x^{(i-1)}_{n-2} + a^{(i-1)}_{n} x^{(i-1)}_{n-1} + b^{(i-1)}_{n} x^{(i-1)}_{n} + s^{(i-1)}_{2j+1} x^{(i)}_{2j+1} + s^{(i-1)}_{2j+2} x^{(i)}_{2j+2}$$
$$= r^{(i-1)}_{n},$$

where

$$\left[a^{(i-1)}_{n}, b^{(i-1)}_{n}, s^{(i-1)}_{2j+1}, s^{(i-1)}_{2j+2}; r^{(i-1)}_{n}\right] := \left[a^{(i-1)}_{n}, b^{(i-1)}_{n}, s^{(i-1)}_{2j+1}, 0; r^{(i-1)}_{n}\right]$$
$$-\sigma^{(i-1)}_{2j}\left[v^{(i)}_{2j}, u^{(i)}_{2j}, c^{(i)}_{2j}, d^{(i)}_{2j}; r^{(i)}_{2j}\right].$$

$$E^{(i-1)}_{n-1} := E^{(i-1)}_{n-1} - \rho^{(i-1)}_{2j} E^{(i)}_{2j},\ \rho^{(i-1)}_{2j} = t^{(i-1)}_{2j}/b^{(i)}_{2j} \Rightarrow$$

$$e^{(i-1)}_{n-1} x^{(i-1)}_{n-3} + a^{(i-1)}_{n-1} x^{(i-1)}_{n-2} + b^{(i-1)}_{n-1} x^{(i-1)}_{n-1} + c^{(i-1)}_{n-1} x^{(i-1)}_{n} + t^{(i-1)}_{2j+1} x^{(i)}_{2j+1}$$
$$+ t^{(i-1)}_{2j+2} x^{(i)}_{2j+2} = r^{(i-1)}_{n-1},$$

where

$$\left[b^{(i-1)}_{n-1}, c^{(i-1)}_{n-1}, t^{(i-1)}_{2j+1}, t^{(i-1)}_{2j+2}; r^{(i-1)}_{n-1}\right] := \left[b^{(i-1)}_{n-1}, c^{(i-1)}_{n-1}, t^{(i-1)}_{2j+1}, 0; r^{(i-1)}_{n-1}\right]$$
$$-\rho^{(i-1)}_{2j-1}\left[v^{(i)}_{2j}, u^{(i)}_{2j}, c^{(i)}_{2j}, d^{(i)}_{2j}; r^{(i)}_{2j}\right].$$

At the end of the above elimination stage, there results the following 2×2 block tridiagonal core system of size $2p \times 2p$:

$$\begin{bmatrix} b^{(1)}_{n-1} & c^{(1)}_{n-1} & t^{(1)}_{n-1} & t^{(1)}_{n} & & & \\ a^{(1)}_{n} & b^{(1)}_{n} & s^{(1)}_{n-1} & s^{(1)}_{n} & & & \\ v^{(2)}_{n-1} & u^{(2)}_{n-1} & b^{(2)}_{n-1} & c^{(2)}_{n-1} & t^{(2)}_{n-1} & t^{(2)}_{n} & \\ v^{(2)}_{n} & u^{(2)}_{n} & a^{(2)}_{n} & b^{(2)}_{n} & s^{(2)}_{n-1} & s^{(2)}_{n} & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \\ & & & v^{(p)}_{n-1} & u^{(p)}_{n-1} & b^{(p)}_{n-1} & c^{(p)}_{n-1} \\ & & & v^{(p)}_{n} & u^{(p)}_{n} & a^{(p)}_{n} & b^{(p)}_{n} \end{bmatrix} \begin{bmatrix} x^{(1)}_{n-1} \\ x^{(1)}_{n} \\ x^{(2)}_{n-1} \\ x^{(2)}_{n} \\ \cdot \\ x^{(p)}_{n-1} \\ x^{(p)}_{n} \end{bmatrix}$$

$$= \begin{bmatrix} r_{n-1}^{(1)} \\ r_n^{(1)} \\ r_{n-1}^{(2)} \\ r_n^{(2)} \\ \cdot \\ r_{n-1}^{(p)} \\ r_n^{(p)} \end{bmatrix} \tag{2.2}$$

The core system (2.2) can be solved as follows. For $i = 1(1)p-1$, consider the eliminations:

$$E_n^{(i)} := E_n^{(i)} - \left(a_n^{(i)}/b_{n-1}^{(i)}\right) E_{n-1}^{(i)} \Rightarrow$$

$$\left[b_n^{(i)}, s_{n-1}^{(i)}, s_n^{(i)}; r_n^{(i)}\right] := \left[b_n^{(i)}, s_{n-1}^{(i)}, s_n^{(i)}; r_n^{(i)}\right]$$
$$- \left(a_n^{(i)}/b_{n-1}^{(i)}\right) \left[c_{n-1}^{(i)}, t_{n-1}^{(i)}, t_n^{(i)}; r_{n-1}^{(i)}\right],$$

$$E_{n-1}^{(i+1)} := E_{n-1}^{(i+1)} - \left(v_{n-1}^{(i+1)}/b_{n-1}^{(i)}\right) E_{n-1}^{(i)} \Rightarrow$$

$$\left[u_{n-1}^{(i+1)}, b_{n-1}^{(i+1)}, c_{n-1}^{(i+1)}; r_{n-1}^{(i+1)}\right] := \left[u_{n-1}^{(i+1)}, b_{n-1}^{(i+1)}, c_{n-1}^{(i+1)}; r_{n-1}^{(i+1)}\right]$$
$$- \left(v_{n-1}^{(i+1)}/b_{n-1}^{(i)}\right) \left[c_{n-1}^{(i)}, t_{n-1}^{(i)}, t_n^{(i)}; r_{n-1}^{(i)}\right],$$

$$E_n^{(i+1)} := E_n^{(i+1)} - \left(v_n^{(i+1)}/b_{n-1}^{(i)}\right) E_{n-1}^{(i)} \Rightarrow$$

$$\left[u_n^{(i+1)}, a_n^{(i+1)}, b_n^{(i+1)}; r_n^{(i+1)}\right] := \left[u_n^{(i+1)}, a_n^{(i+1)}, b_n^{(i+1)}; r_n^{(i+1)}\right]$$
$$- \left(v_n^{(i+1)}/b_{n-1}^{(i)}\right) \left[c_{n-1}^{(i)}, t_{n-1}^{(i)}, t_n^{(i)}; r_{n-1}^{(i)}\right].$$

Then, we do the following eliminations:

$$E_{n-1}^{(i+1)} := E_{n-1}^{(i+1)} - \left(u_{n-1}^{(i+1)}/b_n^{(i)}\right) E_n^{(i)} \Rightarrow$$

$$\left[b_{n-1}^{(i+1)}, c_{n-1}^{(i+1)}; r_{n-1}^{(i+1)}\right] := \left[b_{n-1}^{(i+1)}, c_{n-1}^{(i+1)}; r_{n-1}^{(i+1)}\right]$$
$$- \left(u_{n-1}^{(i+1)}/b_n^{(i)}\right) \left[s_{n-1}^{(i)}, s_n^{(i)}; r_n^{(i)}\right],$$

$$E_n^{(i+1)} := E_n^{(i+1)} - \left(u_n^{(i+1)}/b_n^{(i)}\right) E_n^{(i)} \Rightarrow$$

$$\left[a_n^{(i+1)}, b_n^{(i+1)}; r_n^{(i+1)}\right] := \left[a_n^{(i+1)}, b_n^{(i+1)}; r_n^{(i+1)}\right]$$
$$- \left(u_n^{(i+1)}/b_n^{(i)}\right) \left[s_{n-1}^{(i)}, s_n^{(i)}; r_n^{(i)}\right].$$

Finally, for $i = p$, we need the following elimination:

$$E_n^{(p)} := E_n^{(p)} - \left(a_n^{(p)}/b_{n-1}^{(p)}\right) E_{n-1}^{(p)} \Rightarrow$$

$$\left[b_n^{(p)}; r_n^{(p)}\right] := \left[b_n^{(p)}; r_n^{(p)}\right] - \left(a_n^{(p)}/b_{n-1}^{(p)}\right) \left[b_n^{(p)}; r_n^{(p)}\right].$$

At the end of the above eliminations, the core system (2.2) is reduced to the following:

$$\begin{bmatrix} b_{n-1}^{(1)} & c_{n-1}^{(1)} & t_{n-1}^{(1)} & t_n^{(1)} & & & \\ 0 & b_n^{(1)} & s_{n-1}^{(1)} & s_n^{(1)} & & & \\ 0 & 0 & b_{n-1}^{(2)} & c_{n-1}^{(2)} & t_{n-1}^{(2)} & t_n^{(2)} & \\ 0 & 0 & 0 & b_n^{(2)} & s_{n-1}^{(2)} & s_n^{(2)} & \\ & . & . & . & . & . & \\ & & & 0 & 0 & b_{n-1}^{(p)} & c_{n-1}^{(p)} \\ & & & 0 & 0 & 0 & b_n^{(p)} \end{bmatrix} \begin{bmatrix} x_{n-1}^{(1)} \\ x_n^{(1)} \\ x_{n-1}^{(2)} \\ x_n^{(2)} \\ . \\ x_{n-1}^{(p)} \\ x_n^{(p)} \end{bmatrix} = \begin{bmatrix} r_{n-1}^{(1)} \\ r_n^{(1)} \\ r_{n-1}^{(2)} \\ r_n^{(2)} \\ . \\ r_{n-1}^{(p)} \\ r_n^{(p)} \end{bmatrix}. \tag{2.3}$$

The solution of the reduced core system (2.3) can now be obtained by back substitution.

Once the core system has been solved, the blocks of equations, for $i = 1, ..., p$, uncouple into

$$
\begin{bmatrix}
 & v_1^{(i)} & u_1^{(i)} \\
 & v_2^{(i)} & u_2^{(i)} \\
 & \cdot & \\
 & \cdot & \\
 & v_{n-2}^{(i)} & u_{n-2}^{(i)}
\end{bmatrix} \mathbf{x}^{(i-1)}
$$

$$
+ \begin{bmatrix}
b_1^{(i)} & c_1^{(i)} & d_1^{(i)} & & \\
 & b_2^{(i)} & c_2^{(i)} & d_2^{(i)} & \\
 & \cdot & \cdot & \cdot & \\
 & & \cdot & \cdot & \cdot \\
 & & b_{n-2}^{(i)} & c_{n-2}^{(i)} & d_{n-2}^{(i)}
\end{bmatrix} \mathbf{x}^{(i)} = \mathbf{r}^{(i)}. \tag{2.4}
$$

Thus, the solution of each block can be obtained in parallel.

2.1. The Algorithm

We now describe the algorithm resulting from the above method. Set

$$
u_1^{(i)} = a_1^{(i)},\ u_2^{(i)} = e_2^{(i)},\ v_1^{(i)} = e_1^{(i)},\ v_2^{(i)} = 0;\ s_1^{(i)} = c_n^{(i)},
$$
$$
s_2^{(i)} = d_n^{(i)},\ t_1^{(i)} = d_{n-1}^{(i)},\ t_2^{(i)} = 0.
$$

Omit the computation of the fill-ins u and v in the first block for $i = 1$ and omit the computation of the fill-ins s and t in the last block for $i = p$.

Step 1. Elimination stage.

For $i = 1, ..., p$ do in parallel:

For $j = 1(1)\dfrac{n}{2} - 1$:

(i) $\left[\alpha_{2j}^{(i)}, \beta_{2j+1}^{(i)}, \gamma_{2j-1}^{(i-1)}, \delta_{2j-1}^{(i-1)}\right] = \left(1/b_{2j-1}^{(i)}\right)\left[a_{2j}^{(i)}, e_{2j+1}^{(i)}, s_{2j-1}^{(i-1)}, t_{2j-1}^{(i-1)}\right]$,

$$
\begin{bmatrix}
v_{2j}^{(i)} & u_{2j}^{(i)} & b_{2j}^{(i)} & c_{2j}^{(i)} & r_{2j}^{(i)} \\
v_{2j+1}^{(i)} & u_{2j+1}^{(i)} & a_{2j+1}^{(i)} & b_{2j+1}^{(i)} & r_{2j+1}^{(i)} \\
a_n^{(i-1)} & b_n^{(i-1)} & s_{2j}^{(i-1)} & s_{2j+1}^{(i-1)} & r_n^{(i-1)} \\
b_{n-1}^{(i-1)} & c_{n-1}^{(i-1)} & t_{2j}^{(i-1)} & t_{2j+1}^{(i-1)} & r_{n-1}^{(i-1)}
\end{bmatrix}
$$

$$:= \begin{bmatrix} v_{2j}^{(i)} & u_{2j}^{(i)} & b_{2j}^{(i)} & c_{2j}^{(i)} & r_{2j}^{(i)} \\ v_{2j+1}^{(i)} & u_{2j+1}^{(i)} & a_{2j+1}^{(i)} & b_{2j+1}^{(i)} & r_{2j+1}^{(i)} \\ a_n^{(i-1)} & b_n^{(i-1)} & s_{2j}^{(i-1)} & s_{2j+1}^{(i-1)} & r_n^{(i-1)} \\ b_{n-1}^{(i-1)} & c_{n-1}^{(i-1)} & t_{2j}^{(i-1)} & t_{2j+1}^{(i-1)} & r_{n-1}^{(i-1)} \end{bmatrix}$$

$$- \begin{bmatrix} \alpha_{2j}^{(i)} \\ \beta_{2j+1}^{(i)} \\ \gamma_{2j-1}^{(i-1)} \\ \delta_{2j-1}^{(i-1)} \end{bmatrix} \left[v_{2j-1}^{(i)}, u_{2j-1}^{(i)}, c_{2j-1}^{(i)}, d_{2j-1}^{(i)}, r_{2j-1}^{(i)}\right].$$

(ii) $\left[\lambda_{2j+1}^{(i)}, \mu_{2j+2}^{(i)}, \sigma_{2j}^{(i-1)}, \rho_{2j}^{(i-1)}\right] = \left(1/b_{2j}^{(i)}\right)\left[a_{2j+1}^{(i)}, e_{2j+2}^{(i)}, s_{2j}^{(i-1)}, t_{2j}^{(i-1)}\right],$

$$\begin{bmatrix} v_{2j+1}^{(i)} & u_{2j+1}^{(i)} & b_{2j+1}^{(i)} & c_{2j+1}^{(i)} & r_{2j+1}^{(i)} \\ v_{2j+2}^{(i)} & u_{2j+2}^{(i)} & a_{2j+2}^{(i)} & b_{2j+2}^{(i)} & r_{2j+2}^{(i)} \\ a_n^{(i-1)} & b_n^{(i-1)} & s_{2j+1}^{(i-1)} & s_{2j+2}^{(i-1)} & r_n^{(i-1)} \\ b_{n-1}^{(i-1)} & c_{n-1}^{(i-1)} & t_{2j+1}^{(i-1)} & t_{2j+2}^{(i-1)} & r_{n-1}^{(i-1)} \end{bmatrix}$$

$$:= \begin{bmatrix} v_{2j+1}^{(i)} & u_{2j+1}^{(i)} & b_{2j+1}^{(i)} & c_{2j+1}^{(i)} & r_{2j+1}^{(i)} \\ v_{2j+2}^{(i)} & u_{2j+2}^{(i)} & a_{2j+2}^{(i)} & b_{2j+2}^{(i)} & r_{2j+2}^{(i)} \\ a_n^{(i-1)} & b_n^{(i-1)} & s_{2j+1}^{(i-1)} & s_{2j+2}^{(i-1)} & r_n^{(i-1)} \\ b_{n-1}^{(i-1)} & c_{n-1}^{(i-1)} & t_{2j+1}^{(i-1)} & t_{2j+2}^{(i-1)} & r_{n-1}^{(i-1)} \end{bmatrix}$$

$$- \begin{bmatrix} \lambda_{2j+1}^{(i)} \\ \mu_{2j+2}^{(i)} \\ \sigma_{2j}^{(i-1)} \\ \rho_{2j}^{(i-1)} \end{bmatrix} \left[v_{2j}^{(i)}, u_{2j}^{(i)}, c_{2j}^{(i)}, d_{2j}^{(i)}, r_{2j}^{(i)}\right].$$

Step 2. Solution of the core system

For $i = 1(1)p - 1$:

(i) $\begin{bmatrix} b_n^{(i)} & s_{n-1}^{(i)} & s_n^{(i)} & r_n^{(i)} \\ u_{n-1}^{(i+1)} & b_{n-1}^{(i+1)} & c_{n-1}^{(i+1)} & r_{n-1}^{(i+1)} \\ u_n^{(i+1)} & a_n^{(i+1)} & b_n^{(i+1)} & r_n^{(i+1)} \end{bmatrix} :=$

$$\begin{bmatrix} b_n^{(i)} & s_{n-1}^{(i)} & s_n^{(i)} & r_n^{(i)} \\ u_{n-1}^{(i+1)} & b_{n-1}^{(i+1)} & c_{n-1}^{(i+1)} & r_{n-1}^{(i+1)} \\ u_n^{(i+1)} & a_n^{(i+1)} & b_n^{(i+1)} & r_n^{(i+1)} \end{bmatrix}$$

$$-\left(1/b_{n-1}^{(i)}\right)\begin{bmatrix} a_n^{(i)} \\ v_{n-1}^{(i+1)} \\ v_n^{(i+1)} \end{bmatrix}\left[c_{n-1}^{(i)}, t_{n-1}^{(i)}, t_n^{(i)}, r_{n-1}^{(i)}\right].$$

(ii) $$\begin{bmatrix} b_{n-1}^{(i+1)} & c_{n-1}^{(i+1)} & r_{n-1}^{(i+1)} \\ a_n^{(i+1)} & b_n^{(i+1)} & r_n^{(i+1)} \end{bmatrix} := \begin{bmatrix} b_{n-1}^{(i+1)} & c_{n-1}^{(i+1)} & r_{n-1}^{(i+1)} \\ a_n^{(i+1)} & b_n^{(i+1)} & r_n^{(i+1)} \end{bmatrix}$$

$$-\left(1/b_n^{(i)}\right)\begin{bmatrix} u_{n-1}^{(i+1)} \\ u_n^{(i+1)} \end{bmatrix}\left[s_{n-1}^{(i)}, s_n^{(i)}, r_n^{(i)}\right].$$

(iii) for $i = p$

$$\left[b_n^{(p)}, r_n^{(p)}\right] := \left[b_n^{(p)}, r_n^{(p)}\right] - \left(a_n^{(p)}/b_{n-1}^{(p)}\right)\left[c_{n-1}^{(p)}, r_n^{(p)}\right].$$

(iv) Solve the reduced core system:
$x_n^{(p)} = r_n^{(p)}/b_n^{(p)}$,
$x_{n-1}^{(p)} = \left(r_{n-1}^{(p)} - c_{n-1}^{(p)}x_n^{(p)}\right)/b_{n-1}^{(p)}$,
and for $i = p-1$ down to 1
$x_n^{(i)} = \left(r_n^{(i)} - s_{n-1}^{(i)}x_{n-1}^{(i+1)} - s_n^{(i)}x_n^{(i+1)}\right)/b_n^{(i)}$,

$$x_{n-1}^{(i)} = \left(r_{n-1}^{(i)} - c_{n-1}^{(i)}x_n^{(i)} - t_{n-1}^{(i)}x_{n-1}^{(i+1)} - t_n^{(i)}x_n^{(i+1)}\right)/b_{n-1}^{(i)}.$$

Step 3. Solution of uncoupled subsystems:

In parallel do

For $i = 1$

for $j = n-2$ down to 1

$$x_j^{(1)} = \left(r_j^{(1)} - c_j^{(1)}x_{j+1}^{(1)} - d_j^{(1)}x_{j+2}^{(1)}\right)/b_j^{(1)},$$

and, for $i \in \{2, ..., p\}$

for $j = n-2$ down to 1

$$x_j^{(i)} = \left(r_j^{(i)} - c_j^{(i)}x_{j+1}^{(i)} - d_j^{(i)}x_{j+2}^{(i)} - u_j^{(i)}x_n^{(i-1)} - v_j^{(i)}x_{n-1}^{(i-1)}\right)/b_j^{(i)}.$$

2.2. Arithmetical Operations Counts

We next report on the arithmetical operations counts involved in the serial and parallel version of the above described algorithm. We first consider the counts for serial implementation.

The major part of the computational effort is in Step 1 of the elimination stage. Column eliminations in each block, for $i = 1(1)p$, $j = 1(1)\frac{n}{2} - 1$, involve 6N-12p additions, 6N-12p multiplications and 2N-4p divisions. The computation of the fill-ins s and t involves 8N-8n-17p+17 additions, 10N-10n-20p+20 multiplications and 2N-2n-4p+4 divisions. The computation of the fill-ins u and v involves 2N-2n-5p+5 additions and 4N-4n-8p+8 multiplications. Thus, Step 1 involves a total of 16N-10n-34p+22 additions, 20N-14n-40p+28 multiplications and 4N-2n-8p+4 divisions.

In Step 2, eliminations involve 22p-20 additions, 22p-20 multiplications and 5p-4 divisions. The solution of the reduced core system involves 5p-4 additions, 5p-4 multiplications and 2p divisions. Thus, Step 2 involves a total of 27p-24 additions, 27p-24 multiplications and 7p-4 divisions.

Step 3 involves a total of 4N-2n-8p+4 additions, 4N-2n-8p+4 multiplications and N-2p divisions.

The above counts are summarized in Table 1.

	+	×	÷
Step 1	16N-10n-34p+22	20N-14n-40p+28	4N-2n-8p+4
Step 2	27p-24	27p-24	7p-4
Step 3	4N-2n-8p+4	4N-2n-8p+4	N-2p
Totals	20N-12n-15p+2	24N-16n-21p+8	5N-2n-3p

Table 1: Serial arithmetical operations counts.

The grand total of all the operations is 49N-30n-39p+10. It is interesting to note that for $p = 1$, this grand total reduces to 19N-29 which is the count for the serial algorithm as noted in Section 1.1.

A count for the parallel algorithm can be done as above. In Step 1, column elimination in each block, for $j = 1(1)\frac{n}{2} - 1$, involve 6n-12 additions, 6n-12 multiplications and 2n-4 divisions. Computation of the fill-ins s and t involves 8n-17 additions, 10n-20 multiplications and 2n-4 divisions. The computation of the fill-ins u and v involves 2n-5 additions and 4n-8 multiplications. Thus, the total of operations in Step 1 is 16n-34 additions, 20n-40 multiplications and 4n-8 divisions. For Step 2 the operations counts remain the same as those given above. In Step 3, the computation of the solution in each block involves 4n-8 additions, 4n-8 multiplications and n-2 divisions. These counts for a parallel implementation of the algorithm are summarized in Table 2.

	+	×	÷
Step 1	16n-34	20n-40	4n-8
Step 2	27p-24	27p-24	7p-4
Step 3	4n-8	4n-8	n-2
Totals	20n+27p-66	24n+27p-72	5n+7p-14

Table 2: Parallel operations counts.

Thus, the grand total of all operations for a parallel implementation of the present algorithm is $49\left(\frac{N}{p}\right) + 61p - 152$.

In terms of efficiency of processor utilization as defined by Kuck [13], for a general number of processors, the present parallel algorithm can achieve an efficiency slightly in excess of 38%. This result should be placed in the context of the efficiency for parallel tridiagonal solvers; as noted in Section 1, no parallel algorithm for tridiagonal systems is known with efficiency exceeding 47%.

3. An Illustration of the Algorithm

To illustrate the working the parallel algorithm presented above, we consider an example for $N = 8$ with $A = pentadiag\{-1, -1, 4, -1, -1\}$, $\mathbf{r} = (2, 1, 0, 0, 0, 0, 1, 2)^\top$. We have selected a symmetric matrix purely for ease of illustration, and consider $p = 2$. The steps of the algorithm are shown below.

Step 1. Elimination stage $j = 1$:

$$\left[\begin{array}{cccc|cccccc}
4 & -1 & -1 & & & & & & : & 2 \\
0 & \frac{15}{4} & -\frac{5}{4} & -1 & & & & & & \frac{3}{2} \\
0 & 0 & \frac{46}{15} & -\frac{5}{3} & 0 & 0 & -\frac{1}{3} & -\frac{1}{15} & : & 1 \\
 & 0 & -\frac{5}{3} & \frac{46}{15} & 0 & 0 & -\frac{2}{3} & -\frac{1}{3} & : & \frac{2}{5} \\
\hline
 & & -1 & -1 & 4 & -1 & -1 & & : & 0 \\
 & & -\frac{1}{4} & -\frac{5}{4} & 0 & \frac{15}{4} & -\frac{5}{4} & -1 & : & 0 \\
 & & -\frac{1}{3} & -\frac{2}{3} & 0 & 0 & \frac{10}{3} & -\frac{4}{3} & : & 1 \\
 & & -\frac{1}{15} & -\frac{1}{3} & & 0 & -\frac{4}{3} & \frac{56}{15} & : & 2
\end{array}\right].$$

Step 2. Reduction of the core system:

$$\left[\begin{array}{cc|cccc} \frac{46}{15} & -\frac{5}{3} & -\frac{1}{3} & -\frac{1}{15} & : & 1 \\ -\frac{5}{3} & \frac{46}{15} & -\frac{2}{3} & -\frac{1}{3} & : & \frac{2}{5} \\ \hline -\frac{1}{3} & -\frac{2}{3} & \frac{10}{3} & -\frac{4}{3} & : & 1 \\ -\frac{1}{15} & -\frac{1}{3} & -\frac{4}{3} & \frac{56}{15} & : & 2 \end{array}\right] \sim$$

$$\left[\begin{array}{cc|cccc} \frac{46}{15} & -\frac{5}{3} & -\frac{1}{3} & -\frac{1}{15} & : & 1 \\ 0 & \frac{497}{230} & -\frac{39}{46} & -\frac{17}{46} & : & \frac{217}{230} \\ \hline 0 & 0 & \frac{203320}{68586} & -\frac{101890}{68586} & : & \frac{33810}{22862} \\ 0 & 0 & 0 & \frac{1367465}{10166} & : & \frac{1367465}{10166} \end{array}\right],$$

giving

$$x_4^{(2)} = 1,\ x_3^{(2)} = 1;$$

then,

$$x_4^{(1)} = 1,\ x_3^{(1)} = 1.$$

Step 3. Solution of the two uncoupled subsystems:

$$\left[\begin{array}{cccc|cccccc} 4 & -1 & -1 & & & & & & : & 2 \\ 0 & \frac{15}{4} & -\frac{5}{4} & -1 & & & & & & \frac{3}{2} \\ \hline & & -1 & -1 & 4 & -1 & -1 & & : & 0 \\ & & -\frac{1}{4} & -\frac{5}{4} & 0 & \frac{15}{4} & -\frac{5}{4} & -1 & : & 0 \end{array}\right]$$

gives in parallel

$$x_2^{(1)} = 1, \quad x_2^{(2)} = 1,$$
$$x_1^{(1)} = 1, \quad x_1^{(2)} = 1.$$

References

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

[2] 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.

[3] M.M. Chawla and M.A. Al-Zanaidi, An extended trapezoidal formula for the diffusion equation, *Computers Math. Applic.*, **38** (1999), 51-59.

[4] M.M. Chawla and M.A. Al-Zanaidi, An extended trapezoidal formula for the diffusion equation in two space dimensions, *Computers Math. Applic.*, **42** (2001), 157-168.

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

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

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

[10] 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.

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

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

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

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

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

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

[17] 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).