Optimized Cyclic Reduction for the Solution of Linear Tridiagonal Systems on Parallel Computers*

P. AMODIO†
Dipartimento di Matematica, Università di Bari
Via Orabona 4, 70125 Bari, Italy

(Received and accepted September 1992)

Abstract—A parallel version of the cyclic reduction algorithm for the solution of tridiagonal linear systems is presented. The original problem is divided into subproblems which may be solved almost independently. Synchronizations among the processors involved is only needed to solve a reduced tridiagonal system whose dimension depends on the number of processors.

Numerical tests have been performed on a linear array of processors. The obtained speedups show that this is the best possible parallel implementation of the cyclic reduction and one of the fastest algorithms for the solution of tridiagonal systems on a parallel computer with medium grain parallelism.

Keywords—Tridiagonal linear systems, Cyclic reduction, Parallel computers.

1. INTRODUCTION

The cyclic reduction algorithm was introduced [1–3] in several forms as an alternative to the $LU$ factorization for the solution of tridiagonal linear systems on vector and parallel computers.

It has already been proved [4,5] that the stability properties of the cyclic reduction are the same as the $LU$ factorization without pivoting (for banded systems, the pivoting compromises the band structure). Therefore, the cyclic reduction algorithm is stable if the coefficient matrix is diagonally dominant, or weakly diagonally dominant and irreducible, or symmetric and positive definite.

The vector implementation of the cyclic reduction is both simple and efficient, even if the dimension of the blocks becomes smaller as the reduction continues. On the contrary, its parallel implementation is not so immediate, especially on distributed memory computers, due to the communications among the processors to provide synchronization at every step of the reduction [6,7]. The large number of synchronizations makes this algorithm slow with respect to other parallel algorithms, for example the partition methods [6].

In [8,9], the cyclic reduction has been used as a scalar algorithm in a partitioned matrix.

In this paper, we propose a parallel version of the algorithm to minimize synchronizations and to provide obvious stability properties. The idea is to delay communications among the processors until the solution of a reduced tridiagonal system of dimension which depends on the number of

---

*Work supported by the Ministero della Ricerca Scientifica, 40% project, by the European Community, ESPRIT project (Parallel Computing Action), and by the "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo. Sottoprogetto: Calcolo Scientifico per Grandi Sistemi" of C.N.R.

†The parallel algorithm was implemented in collaboration with Tiziano Politi. The author thanks Paulene Butts for her help in the preparation of the manuscript.
processors. The communication delay is obtained by considering opportune partitioning of the coefficient matrix.

The paper is organized as follows: in Section 2, a block representation of the cyclic reduction algorithm is considered. The obtained factorization is used in Section 3 to develop the parallel algorithm. Section 4 contains numerical tests to compare the parallel cyclic reduction with the known scalar implementation of the algorithm (to make evident the efficiency of the parallel algorithm) and with the LU factorization.

The elements involved in the parallel algorithm need several sub and superscripts. Denote with \( x_{k,j} \) the \( j \)th element of a vector \( x \) obtained on the processor \( i \) after \( k \) steps of reduction.

### 2. THE CYCLIC REDUCTION ALGORITHM

Let us briefly illustrate [4] a block representation of the cyclic reduction algorithm for the solution of the linear system

\[ Mx = f, \]

where

\[
M = \begin{pmatrix} a_1 & c_1 \\ b_1 & a_2 & \ddots \\ & \ddots & \ddots & \ddots \\ & & b_{n-1} & a_n \end{pmatrix}
\]  

and, to simplify the notation, \( n = 2^{k+1} - 1 \).

Consider an \( n \times n \) odd-even permutation matrix \( P_1 \), that is a matrix which transforms the sequence \( 1, \ldots, n \) in the sequence \( 1, 3, \ldots, n, 2, 4, \ldots, n-1 \) \((n \text{ is odd})\). By means of \( P_1 \), we obtain the following \( 2 \times 2 \) block factorization of \( M \)

\[
M = P_1^T \begin{pmatrix} A_1 & T_1 \\ S_1 & B_1 \end{pmatrix} P_1 = P_1^T \begin{pmatrix} I & \widetilde{I} \\ S_1 & A_1^{-1} \end{pmatrix} \begin{pmatrix} I & M_1 \\ T_1 & I \end{pmatrix} P_1 = P_1^T L_1 D_1 U_1 P_1,
\]

where \( A_1 \) and \( B_1 \) are diagonal matrices containing, respectively, the odd and the even main diagonal entries of \( M \); \( S_1 \) and \( T_1 \) are bidiagonal, and \( M_1 = B_1 - S_1 A_1^{-1} T_1 \) is block tridiagonal of dimension \((2^k - 1) \times (2^k - 1)\).

The same operations are repeated on \( M_1 \) and, recursively, on each tridiagonal sub-matrix (obtained after \( j-1 \) steps of reduction)

\[
M_{j-1} = \begin{pmatrix} a_{j-1,1} & c_{j-1,1} \\ b_{j-1,1} & a_{j-1,2} & \ddots \\ & \ddots & \ddots & \ddots \\ & & b_{j-1,h-1} & c_{j-1,h-1} \\ & & & b_{j-1,h-1} & a_{j-1,h} \end{pmatrix}, \quad h = 2^{k-j+2} - 1
\]

by considering an odd-even permutation matrix \( Q_j \) in order to have

\[
Q_j M_{j-1} Q_j^T = \begin{pmatrix} A_j & T_j \\ S_j & B_j \end{pmatrix},
\]

To represent the factorization of the \( n \times n \) matrix

\[
D_{j-1} = \begin{pmatrix} I & M_{j-1} \\ M_{j-1} & I \end{pmatrix},
\]

we define the \( n \times n \) matrices

\[
P_j = \begin{pmatrix} I & Q_j \\ Q_j & I \end{pmatrix}, \quad L_j = \begin{pmatrix} I & I \\ S_j A_j^{-1} & I \end{pmatrix}, \quad D_j = \begin{pmatrix} I & I \\ M_j & I \end{pmatrix}, \quad U_j = \begin{pmatrix} I & A_j T_j \\ A_j T_j & I \end{pmatrix},
\]

(2.5)
where the main diagonal blocks of $L_j$, $D_j$, and $U_j$ are squares of dimension $2^{k+1} - 2^{k-j+2}$ on the first row, $2^{k-j+1}$ on the second, and $2^{k-j+1} - 1$ on the third. It results in $D_{j-1} = P_j^T L_j D_j U_j P_j$, and $D_j$ has a tridiagonal block $M_j$. After $k$ steps, $D_k$ is diagonal ($M_k$ is a $1 \times 1$ block) and the factorization ends. Summarizing, $M$ is factored as

$$M = P_1^T L_1 P_2^T L_2 \cdots P_k^T L_k D_k U_k P_k \cdots U_2 P_2 U_1 P_1.$$  \hspace{1cm} (2.6)

The following algorithm solves (2.1) by means of the factorization (2.6). In the next section, we shall refer to it as the scalar cyclic reduction algorithm.

**Algorithm 2.1. Scalar Cyclic Reduction Algorithm**

\[
M_0 = M, \quad x_0 = f
\]

% reduction step

\[
\text{for } j = 1, k \\
\text{define an odd-even permutation matrix } Q_j \text{ of dimension equal to the dimension of } M_{j-1} \\
determine the blocks } A_j, B_j, S_j \text{ and } T_j \text{ such that (2.4) is verified} \\
V_j = S_j A_j^{-1} \\
M_j = B_j - V_j T_j \\
\begin{pmatrix} y_j^{-1} \\ y_j \end{pmatrix} = Q_j y_{j-1} \\
y_j = y_j^{-1} - V_j y_{j-1}
\]

% solution of the reduced system

\[
\text{determine } x_k \text{ such that } M_k x_k = y_k
\]

% back substitution step

\[
\text{for } j = k, 1, \text{ step } -1 \\
x_{j-1} = Q_j^T \begin{pmatrix} A_j^{-1} (y_j^{-1} - T_j x_j) \\ x_j \end{pmatrix}
\]

end

\[
x = x_0
\]

Figure 2.1 represents the steps of cyclic reduction to obtain the solution.

From Algorithm 2.1, we observe that the cyclic reduction factorization exists and the solution of (2.1) is unique if the odd elements on the main diagonal of each $M_j$, $j = 0, \ldots, k$, are different from 0. Sufficient conditions are, for example, the diagonal dominance, or the weakly diagonal dominance and the irreducibility. These conditions are also sufficient for the stability [4,5].
3. THE PARALLEL VARIANT FOR \( p \) PROCESSORS

Consider an \( n \times n \) tridiagonal matrix \( M \), where \( n = p \cdot 2^{k+1} - 1 \). Suppose that the hypotheses stated in the previous section for the existence and the stability of the scalar cyclic reduction factorization are satisfied for \( M \).

Introduce an \((n + p - 1) \times n\) matrix \( R_p \)

\[
R_p = \begin{pmatrix}
I_{h-1} & 1 & \\
1 & I_{h-1} & \\
& 1 & \\
& & \ddots & 1 \\
& & & \ddots & I_{h-1}
\end{pmatrix},
\]

where \( h = 2^{k+1} \) and \( I_{h-1} \) represents the identity matrix of order \( h - 1 \). By means of \( R_p \), \( M \) is factored as

\[
M = R_p^T N R_p,
\]

where \( N \) is \((n + p - 1) \times (n + p - 1)\) block diagonal

\[
N = \begin{pmatrix}
M^{(1)} & & \\
& \ddots & \\
& & M^{(p)}
\end{pmatrix}
\]

and the blocks \( M^{(i)} \) are tridiagonal

\[
M^{(1)} = \begin{pmatrix}
a_1^{(1)} & c_1^{(1)} & \\
b_1^{(1)} & a_2^{(1)} & c_2^{(1)} & \\
& \ddots & \ddots & \ddots \\
& & b_{h-1}^{(1)} & a_h^{(1)}
\end{pmatrix}, \quad M^{(p)} = \begin{pmatrix}
a_0^{(p)} & c_0^{(p)} & \\
b_0^{(p)} & a_1^{(p)} & c_1^{(p)} & \\
& \ddots & \ddots & \ddots \\
& & b_{h-2}^{(p)} & a_{h-1}^{(p)}
\end{pmatrix},
\]

\[
M^{(i)} = \begin{pmatrix}
a_0^{(i)} & c_0^{(i)} & \\
b_0^{(i)} & a_1^{(i)} & c_1^{(i)} & \\
& \ddots & \ddots & \ddots \\
& & b_{h-1}^{(i)} & a_h^{(i)}
\end{pmatrix}, \quad i = 2, \ldots, p - 1.
\]

The elements of (3.2) and (2.2) are connected by the equalities

\[
a_j^{(i)} = a_{(i-1)h+j}, \quad j = 1, \ldots, h - 1,
\]
\[
b_j^{(i)} = b_{(i-1)h+j}, \quad j = 0, \ldots, h - 1, \quad i = 1, \ldots, p,
\]
\[
c_j^{(i)} = c_{(i-1)h+j}, \quad j = 0, \ldots, h - 1,
\]

and

\[
a_h^{(i)} + a_0^{(i+1)} = a_{ih} \quad i = 1, \ldots, p - 1.
\]

We apply the factorization (2.6) to each block \( M^{(i)} \), thus obtaining

\[
M^{(i)} = F_1^{(i)T} L_1^{(i)} F_2^{(i)T} L_2^{(i)} \ldots F_k^{(i)T} L_k^{(i)} D_k^{(i)} U_k^{(i)} P_{k+1}^{(i)} \ldots U_2^{(i)} P_2^{(i)} U_1^{(i)} P_1^{(i)}.
\]
The structure of the blocks $P_j^{(i)}$, $L_j^{(i)}$, $D_j^{(i)}$ and $U_j^{(i)}$ is the same as in (2.5). Since the indexes of the coefficients of $M^{(i)}$, for $i = 2, \ldots, p - 1$, go from 0 to $h$, the odd-even permutation matrix $P_1^{(i)}$ transforms the sequence 0, 1, \ldots, $h$ into the sequence 1, 3, \ldots, $h - 1, 0, 2, \ldots, h$ and the indexes of the matrix $M_1^{(i)}$ go from 0 to $h/2 = 2^k$. Because of the partitioning of the coefficient matrix, the odd elements on the main diagonal of each $M_j^{(i)}$ are exactly the odd elements of the matrix $M_{j-1}$ in (2.3), that is, 

\[ a_{j-1,2l-1}^{(i)} = a_{j-1,(1-1)h+2l-1}, \quad i = 1, \ldots, p, \quad j = 1, \ldots, k \quad l = 1, \ldots, 2^{k-i+1}. \]

This means that the factorization (3.5) exists. Because of the dimension of the blocks $M^{(i)}$ and because of that stated previously, after $k$ steps of reduction 

\[
M_k^{(1)} = \begin{pmatrix} a_{k,1}^{(i)} \\ b_{k,0}^{(i)} \\ c_{k,0}^{(i)} \\ d_{k,1}^{(i)} \end{pmatrix}, \quad M_k^{(p)} = \begin{pmatrix} a_{k,0}^{(p)} \\ b_{k,0}^{(p)} \\ c_{k,0}^{(p)} \\ d_{k,1}^{(p)} \end{pmatrix},
\]

\[
M_k^{(i)} = \begin{pmatrix} a_{k,0}^{(i)} & b_{k,0}^{(i)} \\ c_{k,0}^{(i)} & d_{k,1}^{(i)} \end{pmatrix}, \quad i = 2, \ldots, p - 1.
\]

A relation exists between these elements and those of $M_k$ in (2.6) (consider that the dimension of $M$ is now $p \cdot 2^{k+1} - 1$ and therefore after $k$ steps of reduction $M_k$ has dimension $p - 1$), that is, 

\[
a_{k,1}^{(i)} + a_{k,0}^{(i+1)} = a_{k,i}, \quad i = 1, \ldots, p - 1,
\]

\[
b_{k,0}^{(i)} = b_{k,i}, \quad c_{k,0}^{(i)} = c_{k,i}, \quad i = 1, \ldots, p - 2.
\]

Now consider the solution of the linear system (2.1). If the blocks $M_k^{(i)}$ are nonsingular, then the solution of (2.1) by means of the factorization (3.1) and (3.5) exists. In fact, the linear system $R_p^T y = f$ has $\infty^{p-1}$ solutions. The consistency of the system $R_p x = N^{-1} y$ implies that $N^{-1} y \in \text{range}(R_p)$; this imposes $p - 1$ conditions and the solution $x$ is unique.

After referring to the notation of Algorithm 2.1 and the blocks $M^{(i)}$ in (3.2), let

\[
y^{(1)}_0 = \begin{pmatrix} y_0^{(1)} \\ \vdots \\ y_0^{(h)} \end{pmatrix}, \quad y^{(i)}_0 = \begin{pmatrix} y_0^{(i)} \\ \vdots \\ y_0^{(h)} \end{pmatrix}, \quad i = 2, \ldots, p - 1, \quad y^{(p)}_0 = \begin{pmatrix} y_0^{(p)} \\ \vdots \\ y_0^{(h)} \end{pmatrix}
\]

be the block components of the $(n + p - 1)$ solution of the system $R_p^T y_0 = f$, where 

\[
y_0^{(i)} = f_{(i-1)h+j}, \quad j = 1, \ldots, h - 1, \quad i = 1, \ldots, p,
\]

and

\[
y_0^{(i)} + y_0^{(i+1)} = f_{ik}, \quad i = 1, \ldots, p - 1.
\]

The vectors $y^{(i)}_0$, for $i = 2, \ldots, p - 1$, have two indeterminate components ($y_0^{(i)}$ and $y_0^{(i)}$ are not known). From that stated previously concerning the matrices $M^{(i)}$, a similar indetermination will also appear in the successive $y_j^{(i)}$ (the last element of $y_j^{(i)}$ and the first of $y_j^{(i+1)}$ are not known but their sum is known) obtained by solving the linear systems with the blocks $L_j^{(i)}$ as coefficient matrix in parallel.

At the $k^{th}$ step of reduction, it results that 

\[
y_k^{(1)} = \begin{pmatrix} y_k^{(1)} \\ y_k^{(i)} \end{pmatrix}, \quad y_k^{(p)} = \begin{pmatrix} y_k^{(p)} \\ y_k^{(i)} \end{pmatrix},
\]

\[
y_k^{(i)} = \begin{pmatrix} y_k^{(i)} \\ y_k^{(i)} \end{pmatrix}, \quad i = 2, \ldots, p - 1.
\]
In these vectors, only the sums \( y_{i,1}^{(i)} + y_{k,0}^{(i+1)} \) are a known quantity. Nevertheless, the indeterminacy may be continued until this point, since the indexes of the first and the last elements of each \( y_{i}^{(i)} \) are always even.

Now we must solve the linear sub-systems \( M_{k}^{(i)} \mathbf{x}_{k}^{(i)} = y_{k}^{(i)} \), for \( i = 1, \ldots, p \), that is, the linear system
\[
\begin{pmatrix}
M_{k}^{(1)} \\
\vdots \\
M_{k}^{(p)}
\end{pmatrix}
\begin{pmatrix}
\mathbf{x}_{k}^{(1)} \\
\vdots \\
\mathbf{x}_{k}^{(p)}
\end{pmatrix}
= 
\begin{pmatrix}
y_{k}^{(1)} \\
\vdots \\
y_{k}^{(p)}
\end{pmatrix}. 
\tag{3.9}
\]

To avoid the above stated indeterminacy, we consider the \( (p - 1) \times (2p - 2) \) matrix
\[
Z_{p} = 
\begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & 1 & 1 & \cdots & 1
\end{pmatrix}. 
\tag{3.10}
\]

Through \( Z_{p} \), we obtain both the non-singular \( p \times p \) matrix
\[
M_{k} = Z_{p} \begin{pmatrix}
M_{k}^{(1)} \\
\vdots \\
M_{k}^{(p)}
\end{pmatrix} Z_{p}^{T},
\]
which is the same obtained by applying \( k \) steps of scalar cyclic reduction to \( M \), and the vector
\[
y_{k} = Z_{p} \begin{pmatrix}
y_{k}^{(1)} \\
\vdots \\
y_{k}^{(p)}
\end{pmatrix} = 
\begin{pmatrix}
y_{k,1}^{(1)} + y_{k,0}^{(2)} \\
\vdots \\
y_{k,1}^{(p-1)} + y_{k,0}^{(p)}
\end{pmatrix}, 
\tag{3.11}
\]
which is uniquely defined. The linear system
\[
M_{k} \mathbf{x}_{k} = y_{k},
\tag{3.12}
\]
where
\[
M_{k} = 
\begin{pmatrix}
\alpha_{k,1}^{(1)} + \alpha_{k,0}^{(2)} & c_{k,0}^{(2)} \\
\beta_{k,0}^{(2)} & \alpha_{k,1}^{(2)} + \alpha_{k,0}^{(3)} \\
\vdots & \ddots & \ddots & \ddots \\
\beta_{k,0}^{(p-1)} & \alpha_{k,1}^{(p-1)} + \alpha_{k,0}^{(p)}
\end{pmatrix},
\]
has a unique solution \( \mathbf{x}_{k} \), from which we obtain the \( 2p - 2 \) vector
\[
\begin{pmatrix}
\mathbf{x}_{k}^{(1)} \\
\vdots \\
\mathbf{x}_{k}^{(p)}
\end{pmatrix}
= Z_{p}^{T} \mathbf{x}_{k}
= Z_{p}^{T} \begin{pmatrix}
\mathbf{x}_{k,1} \\
\vdots \\
\mathbf{x}_{k,p-1}
\end{pmatrix},
\tag{3.12}
\]
where
\[
\mathbf{x}_{k}^{(1)} = (x_{k,1}), \quad \mathbf{x}_{k}^{(p)} = (x_{k,p-1}), \\
\mathbf{x}_{k}^{(i)} = 
\begin{pmatrix}
x_{k,i-1} \\
x_{k,i}
\end{pmatrix}, \quad i = 2, \ldots, p - 1.
\tag{3.12}
The solution of (3.9) by means of the matrix $Z_\rho$ makes every choice of $a_h^{(i)}$ and $a_0^{(i+1)}$ which satisfies (3.4) correct. It is no necessary that the blocks $M^{(i)}$ in (3.2) are non singular. This is also true for $y_0^{(i)}$ and $y_0^{(i+1)}$ in (3.8). Moreover, the stability properties are maintained because at each step of the reduction the elements obtained by this algorithm are the same obtained by the scalar cyclic reduction algorithm.

The solution of (3.11) is the only scalar part of the algorithm. However, the dimension of $M_k$ depends only on the number of processors, and not on the dimension of the original problem. We do not insist on the solution of this reduced system because it depends on the parallel computer used.

For $j = k, k - 1, \ldots, 1$, the solution of the linear systems with $U_j^{(i)}$ as coefficient matrices proceeds in parallel on different processors, and the vector $x$ obtained from

\begin{equation}
\begin{aligned}
\varphi_{(i-1)h+j} &= x_{0,j}^{(i)} , \quad j = 1, \ldots, h - 1, \quad i = 1, \ldots, p \\
x_{ih} &= x_{0,h}^{(i)} = x_{0,0}^{(i+1)} = x_{k,i}^{(i)} , \quad i = 1, \ldots, p - 1
\end{aligned}
\end{equation}

satisfies $R_p x = x_0$, and hence it is the solution of (2.1).

The process to obtain the solution on four processors is represented in Figure 3.1.

![Figure 3.1. Parallel cyclic reduction algorithm.](image)

The following summarizes the parallel algorithm.

**ALGORITHM 3.1. PARALLEL CYCLIC REDUCTION ALGORITHM**

% splitting of $M$ and $f$ among the processors

\begin{verbatim}
let the elements of $M^{(i)}$ and $y_0^{(i)}$ as in (3.3) and (3.7) and for i = 1, p - 1
\end{verbatim}

\begin{verbatim}
y_{0,h}^{(i)} = f_{ih} \quad y_{0,0}^{(i+1)} = 0
\end{verbatim}

\begin{verbatim}
a_{0,h}^{(i)} = a_{ih} \quad a_{0,0}^{(i+1)} = 0
\end{verbatim}

end

% reduction step (in parallel)

\begin{verbatim}
for i = 1, p

apply the reduction step of ALGORITHM 2.1 to $M^{(i)}$ and $y_0^{(i)}$
\end{verbatim}

end

% solution of the reduced system

\begin{verbatim}
determine $M_k$ and $y_k$
\end{verbatim}

\begin{verbatim}
solve the system (3.11)
\end{verbatim}

\begin{verbatim}
determine $x_k^{(i)}$ by means of (3.12)
\end{verbatim}

% backsubstitution step (in parallel)

\begin{verbatim}
for i = 1, p

apply the back substitution step of ALGORITHM 2.1 to $x_k^{(i)}$
\end{verbatim}

end

\begin{verbatim}
determine the solution $x$ by means of (3.13)
\end{verbatim}
4. Numerical Tests

The parallel algorithm was coded on a Multiputer (Microway) with 32 transputers INMOS T800-20, each one with 1 Mb of memory. The topology of interconnection used was a linear array with bidirectional communications [6]. The scalar tests were performed on a monoputer INMOS T800-20 with 16 Mb of memory.

To begin, the coefficient matrix has been partitioned among the processors to make the solution of large problems possible.

![Figure 4.1. Percentages of time to solve (3.3) with respect to the entire time spent to solve a system of dimension \( p \cdot 2^k \) (the number \( p \) of processors is \( 2^j, j = 1, \ldots, 5 \), \( k \) it ranges from 7 to 13).](image)

![Figure 4.2. Speedup of the problem.](image)

<table>
<thead>
<tr>
<th># of processors</th>
<th>log(_2) of the dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>0.87</td>
</tr>
<tr>
<td>4</td>
<td>0.51</td>
</tr>
<tr>
<td>8</td>
<td>0.48</td>
</tr>
<tr>
<td>16</td>
<td>0.46</td>
</tr>
<tr>
<td>32</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 4.1. Efficiency of the parallel algorithm.
The solution of (3.11) was obtained by factorizing $M_k$ in the form $UL$, where $U$ is upper and $L$ is a lower bidiagonal matrix. In this way, each processor sends two data to the previous processor when the upper bidiagonal system is solved, and one data to the next processor when the upper bidiagonal system is solved. The solution of (3.11) might be obtained by considering more efficient algorithms if one uses topologies of communications among the processors (see, for example, [10]).

For large problems, the execution time to solve (3.11) is a small part of the entire execution time (Figure 4.1).

For small problems, the slowness of the communication operations in the considered parallel computer compromises the efficiency of the algorithm. By using a different computer with high velocity communications, it is certainly possible to reduce the execution time of (3.11).

Table 4.1 shows the efficiency of the algorithm, obtained by the formula

\[ E_p = \frac{t_s}{t_p \cdot p}. \]

For large problems this algorithm is optimal.

The speedup of the problem (Figure 4.2) is given by the ratio between the time of scalar execution of the $LU$ factorization (that is the best scalar algorithm) and the execution time of the parallel version of the cyclic reduction algorithm here proposed. A comparison with the implementation on the same computer of other parallel methods (see [8,9]) proves the effectiveness of this parallel implementation of the cyclic reduction.

REFERENCES