COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 29 (1981) 97–107
NORTH-HOLLAND PUBLISHING COMPANY

PARALLEL SOLUTION OF LINEAR SYSTEMS BY QUADRANT INTERLOCKING FACTORISATION METHODS

D.J. EVANS

Department of Computer Studies, Loughborough University of Technology, Loughborough, Leicestershire, U.K.

A. HADJIDIMOS and D. NOUTSOS

Department of Mathematics, University of Ioannina, Ioannina, Greece

Received 19 March 1981

This paper presents a parallel algorithm for the solution of linear systems and determinant evaluation suitable for use on the proposed parallel computers of the future. The new method can be considered as extending the novel matrix factorisation strategies introduced by Evans and Hatzopoulos [1] and Evans and Hadjidimos [2] in which quadrant interlocking factors are considered instead of the more usual LU factors of triangular decomposition.

1. Introduction

Recently Evans and Hatzopoulos [1] and Evans and Hadjidimos (see [2] and [3]) have introduced two new factorisation techniques for a matrix A. These are known as Quadrant Interlocking Factorisation techniques (abbreviated to QIF1 and QIF2 respectively) and are such that the second is a modification of the first whilst both are suitable for the solution of the non-singular linear system

$$Ax = b \tag{1.1}$$

on a Parallel Computer of the type "Single Instruction Stream-Multiple Data Stream" (SIMD) (see Flynn [4] and Stone [5]). Both QIF techniques are in most cases better than the well known ones given by Sameh and Kuck [6], Lambiotte [7], Evans and Hatzopoulos [8] etc.

In this paper, we present a new version of the QIF2 technique, namely NQIF2, which constitutes an improvement over the old QIF2 procedure in as much as that both the total number of time steps and the maximum number of processors working in parallel required for the solution of the system (1.1) are decreased considerably. In addition, we present a parallel algorithm for the evaluation of the determinant of a matrix A by using the QIF1 technique. Such an algorithm based on QIF1 has so far been believed as not possible (see [2]). Finally, the various tables given in the text make comparisons amongst all these three techniques so that the best one to use in each case is readily produced.

2. The new (N)QIF2 factorisation

We consider system (1.1) where A is a non-singular matrix of order n with elements $a_{i,j}$, $i,j=1(1)n$ and x and b two n-dimensional vectors with x unknown and b known and corresponding components x_i and b_i, $i=1(1)n$. We now assume that there exists three matrices W, D and Z, each of order n, such that

$$A = WDZ. \tag{2.1}$$

In (2.1) the W-factor is the same factor as in the QIF2 technique (see [2]), namely of the form

$$W = \begin{bmatrix} 1 & & & & & & 0 \\ w_{21} & 1 & & \mathbf{0} & & 0 & w_{2n} \\ w_{31} & w_{32} & 1 & & 0 & w_{3,n-1} & w_{3n} \\ \vdots & \vdots & \vdots & \ddots \quad \cdot^{\cdot^{\cdot}} & \vdots & \vdots & \vdots \\ w_{n-2,1} & w_{n-2,2} & w_{n-2,3} & & 1 & w_{n-2,n-1} & w_{n-2,n} \\ w_{n-1,1} & w_{n-1,2} & & \mathbf{0} & & 1 & w_{n-1,n} \\ w_{n1} & & & & & & 1 \end{bmatrix}, \tag{2.2}$$

D is a diagonal matrix with elements d_i, $i=1(1)n$ and the Z-factor is of the same type as in the QIF2 technique with the only exception being that it has units on the main diagonal. Thus, the matrix Z in (2.1) is of the form

$$Z = \begin{bmatrix} 1 & z_{12} & z_{13} & \cdots & z_{1,n-2} & z_{1,n-1} & z_{1n} \\ & 1 & z_{23} & \cdots & z_{2,n-2} & z_{2,n-1} & \\ & & 1 & \cdots & z_{3,n-2} & & \\ & \mathbf{0} & & & & \mathbf{0} & \\ & & 0 & \cdots & 1 & & \\ & 0 & z_{n-1,3} & \cdots & z_{n-1,n-2} & 1 & \\ 0 & z_{n2} & z_{n3} & \cdots & z_{n,n-2} & z_{n,n-1} & 1 \end{bmatrix}, \tag{2.3}$$

where it is pointed out that the product DZ in (2.1) is in fact the factor Z in the QIF2 technique (see [2]).

The motivations for the creation of the new technique were the following three points (i) the observation that if the basic algorithmic step for the factorisation of the matrix A by using the QIF2 technique were split into two half steps then the number of time steps as well as the maximum number of processors working in parallel at the same time would become much smaller; (ii) the adoption of the form $A = WDW^t$ for the real symmetric positive definite case in QIF2 could have readily come from the adoption of the form WDZ for the general case, and (iii) the observation that if the previous general form were adopted the solution of a system with matrix coefficient DZ in two stages would require many less time steps than the corresponding solution of the system with coefficient matrix Z in the QIF2 technique.

3. The computation of the W-, D- and Z-matrices

From (2.1) we have that

$$A = \sum_{i=1}^{n} d_i W_i Z_i^{\mathrm{T}}, \tag{3.1}$$

where W_i and Z_i, $i = 1(1)n$ are the column vectors of the matrices W and Z^{t} respectively. Now we define the matrices A_k and $A_{k+1/2}$, $k = 1(1)[\frac{1}{2}(n-1)]$, with the symbol $[\alpha]$ denoting the largest integer not greater than the real number α, such that

$$\begin{aligned} &A_1 = A, \\ &A_k = A - \sum_{i=1}^{k-1} d_i W_i Z_i^{\mathrm{t}} - \sum_{i=n-k+2}^{n} d_i W_i Z_i^{\mathrm{t}}, \quad k = 2(1)[\tfrac{1}{2}(n-1)], \\ &A_{k+1/2} = A - \sum_{i=1}^{k} d_i W_i Z_i^{\mathrm{t}} - \sum_{i=n-k+2}^{n} d_i W_i Z_i^{\mathrm{t}}, \quad k = 1(1)[\tfrac{1}{2}(n-1)]. \end{aligned} \tag{3.2}$$

It can be easily seen, from (3.2), that the following relationships also hold:

$$\left.\begin{aligned} A_{k+1/2} &= A_k - d_k W_k Z_k^{\mathrm{t}}, \\ A_{k+1} &= A_{k+1/2} - d_{n-k+1} W_{n-k+1} Z_{n-k+1}^{\mathrm{t}}, \end{aligned}\right\} \quad k = 1(1)[\tfrac{1}{2}(n-1)]. \tag{3.3}$$

From relationships (3.2) it can be also deduced that the first and last $(k-1)$ rows and columns of any A_k matrix defined there are zero. So are the first k and the last $(k-1)$ rows and columns of any $A_{k+1/2}$ matrix.

If we consider all the relationships above the NQIF2 algorithm for finding the elements of three matrices W, D and Z becomes

$$\begin{aligned} &\text{(a)}\quad d_k = a_{k,k}^{(k)}, \\ &\text{(b)}\quad z_{k,j} = a_{k,j}^{(k)}/d_k, \qquad w_{j,k} = a_{j,k}^{(k)}/d_k, \quad j = k+1(1)n-k+1, \\ &\text{(c)}\quad A_{k+1/2} = A_k - A_k^{(k)} Z_k^{\mathrm{t}}, \\ &\text{(d)}\quad d_{n-k+1} = a_{n-k+1,n+k+1}^{(k+1/2)} \\ &\text{(e)}\quad z_{n-k+1,j} = a_{n-k+1,j}^{(k+1/2)}/d_{n-k+1}, \\ &\qquad\; w_{j,n-k+1} = a_{j,n-k+1}^{(k+1/2)}/d_{n-k+1}, \quad j = k+1(1)n-k, \\ &\text{(f)}\quad A_{k+1} = A_{k+1/2} - A_{n-k+1}^{(k+1/2)} Z_{n-k+1}^{\mathrm{t}}, \end{aligned} \tag{3.4}$$

for all $k = 1(1)[\frac{1}{2}(n-1)]$. In (3.4) $a_{p,q}^{(m)}$ denotes the element of the A_m matrix in the position (p, q), while $A_i^{(l)}$ denotes the ith column vector of the A_l matrix. The reader is reminded that in order to complete the evaluation of all the elements of the matrices concerned one more

Table 1
Factorisation of a matrix

Method	Num. of time steps	Max. num. of proc.
QIF1	$3(n-1)-1.5(1+(-1)^n)$	$2(n-2)^2$
QIF2	$4.5(n-1)-0.75(1+(-1)^n)$	$2(n-2)^2$
NQIF2	$3(n-1)$	$(n-1)^2$

stage, for $k=[\frac{1}{2}(n+1)]$, is necessary. To be more specific, if n is odd, step (3.4a) has to be carried out in order to find the value of the central element d_k, while if n is even, steps (3.4a)–(3.4d) have to be carried out in order to find the central elements d_k, $z_{k,k+1}$, $w_{k+1,k}$ and d_{k+1}. Here, it has to be made clear that in steps (3.4c) and (3.4f) use of the column vectors $A_k^{(k)}$ and $A_{n-k+1}^{(k+1/2)}$ was made instead of the products d_kW_k and $d_{n-k+1}W_{n-k+1}$ respectively as was indicated from relationships (3.3). This was permitted because of the second sets of relationships in steps (3.4b) and (3.4e) and thus a saving of two extra parallel calculations at each stage of the algorithm (3.4) was made.

If we adopt the basic assumption (as in [1, 2, 3] and [6]) that a parallel replacement statement requires negligible time, while any other parallel arithmetic operation needs the same time step we can very easily determine that the total number of time steps for the complete factorisation of the matrix A is equal to $6[\frac{1}{2}(n-1)]$ if n is odd and $6[\frac{1}{2}(n-1)]+3$ if n is even. Both these expressions are equal to $3(n-1)$ for any n. At the same time the maximum number of processors working in parallel needed to perform all the operations required is readily seen to be $(n-1)^2$.

In Table 1 below we give the number of time steps as well as the maximum number of processors required for the complete factorisation of the matrix A by using each one of the three methods QIF1, QIF2 and NQIF2.

From Table 1 it is readily seen that the QIF1 technique is always better than QIF2 for all $n \geq 3$. Comparing now the QIF1 and NQIF2 techniques we can see that for any n odd both techniques require the same number of time steps while for n even, QIF1 requires three time steps less than NQIF2 does. Comparing now the maximum number of processors used it is easily found out that only for $n=3$ and 4 is QIF1 better than NQIF2 while for $n \geq 5$ NQIF 2 is the best of the two.

Before we close this section we have to point out that the factorisation (2.1), where D is a diagonal matrix and the other two factors are of the forms (2.2) and (2.3) respectively, is unique. The proof is omitted here.

4. Solution of the linear system

For the solution of the linear system (1.1) we introduce the auxiliary vectors y and u, with components y_i, u_i, $i=1(1)n$ respectively, so that in view of (2.1), it is equivalent to solving the following three simpler systems

$$Wy=b\,, \tag{4.1}$$

$$Du=y \tag{4.2}$$

and

$$Zx = u\,. \tag{4.3}$$

To solve system (4.1) we let $Wy = b = b^{(1)}$ so that

$$\sum_{i=1}^{n} y_i W_i = b\,.$$

We then introduce the vectors $b^{(k+1/2)}$ and $b^{(k)}$, for $k = 1(1)[\frac{1}{2}(n+1)]$, such that

$$b^{(1)} = b, \qquad b^{(k+1/2)} = b^{(k)} - y_k W_k, \qquad b^{(k+1)} = b^{(k+1/2)} - y_{n-k+1} W_{n-k+1}\,.$$

Hence, the NQIF2 algorithm for the solution of system (4.1) is the following

$$\begin{aligned}
&\text{(a)} \quad y_k = b_k^{(k)}\,,\\
&\text{(b)} \quad b^{(k+1/2)} = b^{(k)} - y_k W_k\,,\\
&\text{(c)} \quad y_{n-k+1} = b_{n-k+1}^{(k+1/2)}\,,\\
&\text{(d)} \quad b^{(k+1)} = b^{(k+1/2)} - y_{n-k+1} W_{n-k+1}\,,
\end{aligned} \tag{4.4}$$

for $k = 1(1)[\frac{1}{2}(n-1)]$. If n is odd, step (4.4a) has to be executed for $k = [\frac{1}{2}(n+1)]$ to find the element in the middle y_k, while if n is even, steps (4.4a)–(4.4c) have to be executed for the same value of $k = [\frac{1}{2}(n+1)]$ to find the two centre elements y_k and y_{k+1}. Under the assumptions of Section 3, concerning the time needed for a parallel operation, we can very easily find that for the evaluation of the components of the vector y, a total number of $4[\frac{1}{2}(n-1)] + 1 + (-1)^n = 2(n-1)$ time steps and a maximum number of $(n-1)$ processors are required.

As can be readily seen system (4.2) is solved in one time step using n processors. More specifically we have

$$u_i = y_i/d_i\,, \quad i = 1(1)n\,.$$

Finally to solve system (4.3) we proceed as follows. We let $Zx = u = u^{(1)}$ so that

$$\sum_{i=1}^{n} x_i Z_i^* = u^{(1)}\,,$$

where Z_i^*, $i = 1(1)n$ are the column vectors of the matrix Z. We then introduce the vectors $u(i)$ and $u^{(i+1/2)}$, for $i = 1(1)[\frac{1}{2}(n+1)]$, such that

$$\begin{aligned}
&u^{(1)} = u\,,\\
&u^{(l-k+2/3)} = u^{(l-k+1)} - x_{n-k+1} Z_{n-k+1}^*\,,\\
&u^{(l-k+2)} = u^{(l-k+3/2)} - x_k Z_k^*\,,
\end{aligned}$$

where $l = [\frac{1}{2}(n+1)]$ and $k = [\frac{1}{2}(n+1)](-1)1$, except for n odd and $k = 1$ when we put

Table 2
Solution of the auxiliary systems

Method	Num. of time steps	Max. num. of proc.
QIF1	$4.5(n-1)-0.75(1+(-1)^n)$	$2(n-2)$
QIF2	$6(n-1)$	$2(n-2)$
NQIF2	$4n-3$	n

$u^{(3/2)}=u^{(1)}=u$ and $u^{(2)}=u^{(1)}-x_l Z_l^*$. If n is odd we start with

$$\begin{aligned} &\text{(a)} \quad x_l = u_l^{(1)}, \\ &\text{(b)} \quad u^{(2)} = u^{(1)} - x_l Z_l^*, \end{aligned} \tag{4.5}$$

while if n is even, we start with

$$\begin{aligned} &\text{(a)} \quad x_{l+1} = u_{l+1}^{(1)}, \\ &\text{(b)} \quad u^{(3/2)} = u^{(1)} - x_{l+1} Z_{l+1}^*, \\ &\text{(c)} \quad x_l = u_l^{(3/2)}, \\ &\text{(d)} \quad u^{(2)} = u^{(3/2)} - x_l Z_l^*. \end{aligned} \tag{4.6}$$

We then proceed by using the algorithm

$$\begin{aligned} &\text{(a)} \quad x_{n-k+1} = u_{n-k+1}^{(l-k+1)}, \\ &\text{(b)} \quad u^{(l-k+3/2)} = u^{(l-k+1)} - x_{n-k+1} Z_{n-k+1}^*, \\ &\text{(c)} \quad x_k = u_k^{(l-k+3/2)}, \\ &\text{(d)} \quad u^{(l-k+2)} = u^{(l-k+3/2)} - x_k Z_k^*, \end{aligned} \tag{4.7}$$

for $k=[\frac{1}{2}(n-1)](-1)1$, where step (4.7d) is not executed for $k=1$. Starting with (4.5) or (4.6), depending on whether n is odd or even respectively, and then applying algorithm (4.7) we can obtain the solution of system (4.3) in a total number of $4[\frac{1}{2}(n-1)]+1+(-1)^n=2(n-1)$ time steps by using a maximum number of $(n-1)$ processors working in parallel.

Thus for the complete solution of systems (4.1)–(4.3) the time steps and the maximum number of processors required for the three basic techniques are given in Table 2.

Table 3
Complete solution of the original system

Method	Num. of time steps	Max. num. of proc.
QIF1	$7.5(n-1)-2.25(1+(-1)^n)$	$2(n-2)^2$
QIF2	$10.5(n-1)-0.75(1+(-1)^n)$	$2(n-2)^2$
NQIF2	$7n-6$	$(n-1)^2$

When Tables 1 and 2 are combined we can form Table 3 in which the results for the complete solution of the original system (1.1) are given for the three Quadrant Interlocking Factorisation techniques.

As is readily seen from Table 3 the QIF1 technique is always better than QIF2. Comparing QIF1 with NQIF2 as far as the total number of time steps is concerned we can find out that for $n = 4$, 6, 8 and 10 QIF1 is the best of the two, for $n = 3$ and $n = 12$ they are equivalent, while for any other value of n NQIF2 is the best. Taking into consideration the maximum number of processors required we arrive at exactly the same conclusion as previously for the factorisation stage. That is, for $n = 3$ and 4 QIF1 is the best while for any other value of n, NQIF2 is the one which utilises a smaller maximum number of processors.

5. Basic remarks concerning the NQIF2 technique

When we consider the analysis made so far in the previous sections concerning the NQIF2 technique as well as some of the basic results in [2] we can make the following remarks:

(i) A possible breakdown during the factorisation process (see [2]) can be avoided provided that

$$a^{(k)}_{k,k},\ a^{(k+1/2)}_{n-k+1,n-k+1} \neq 0\,, \qquad k = 1(1)[\tfrac{1}{2}(n-1)] \quad \text{and} \quad a^{(l)}_{l,l} \neq 0\,,$$

with $l = [\frac{1}{2}(n+1)]$ if n is even. If the matrix A is a non-singular matrix then an interchanging column strategy can be adopted in the same way as suggested in [2].

(ii) The interchanging column strategy is not necessary if A is a diagonally dominant matrix. This was proved directly in [2] for the matrices A_k and indirectly for the matrices $A_{k+1/2}$ for all $k = 1(1)[\frac{1}{2}(n-1)]$, since these matrices, as was proved there, possess the same property of diagonal dominance.

(iii) On the other hand when A is a real symmetric positive definite matrix no breakdown occurs during the factorisation process. This was proved in [2] under the assumption that there existed matrices W and D (diagonal) such that $A = WDW^{t}$. Since the existence of these two matrices is not an obvious thing, especially in the present case of the NQIF2 technique, we show it by stating and proving the following theorem.

THEOREM. If A is a real symmetric positive definite matrix, then so are all matrices A'_k, $k = 1(1)[\frac{1}{2}(n+1)]$ or $A'_{k+1/2}$, $k = 1(1)[\frac{1}{2}(n-1)]$ and $k = [\frac{1}{2}(n+1)]$ if n is even. The matrices A'_k and $A'_{k+1/2}$ are obtained from the matrices A_k and $A_{k+1/2}$, defined in (3.2), *by deleting the first and the last $(k-1)$ zero rows and columns or the first k and the last $(k-1)$ zero rows and columns respectively.*

PROOF. First we define the unit matrix of order $(n-1)I_{n-1}$, the $(n-1)$-dimensional zero vector 0_{n-1}, and then we define the $(n-1)$-dimensional vectors W'_1 and $A'^{(1)}_1$ obtained from the vectors W_1 and $A^{(1)}_1$ respectively by deleting their first components. It is evident that $d_1 = a^{(1)}_{11} = a_{11} > 0$ and that $A_{3/2} = A_1 - A^{(1)}_1 W^{t}_1 = A_1 - d_1 W_1 W^{t}_1$. Therefore

$$A'_{3/2} = A'_1 - A'^{(1)}_1 W'^{t}_1 = A'_1 - d_1 W'_1 W'^{t}_1\,.$$

Now we define the non-singular matrix

$$B = \begin{bmatrix} 1 & -W''^{\mathrm{t}} \\ 0_{n-1} & I_{n-1} \end{bmatrix}$$

and we form the product

$$C = B^{\mathrm{t}}AB = \begin{bmatrix} a_{11} & A_1'^{(1)\mathrm{t}} - a_{11}W_1'^{\mathrm{t}} \\ A_1'^{(1)} - a_{11}W_1' & A_1' - A_1'^{(1)}W_1'^{\mathrm{t}} - W_1'(A_1'^{(1)\mathrm{t}} - a_{11}W_1'^{\mathrm{t}}) \end{bmatrix},$$

or equivalently

$$C = \begin{bmatrix} a_{11} & 0_{n-1}^{\mathrm{t}} \\ 0_{n-1} & A_{3/2}' \end{bmatrix}.$$

Since A is a real symmetric positive definite matrix so is the matrix $C = B^{\mathrm{t}}AB$. On the other hand it is known that if C is a real symmetric positive definite matrix then so is any submatrix obtained from it by deleting any rows of C and the corresponding columns (see e.g. [9, p. 21, Proof of Theorem 2.5]). Thus by deleting the first row and column of C we obtain $A_{3/2}'$ which is a real symmetric positive definite matrix. Hence $d_n = a_{n,n}^{(3/2)} > 0$. We now proceed in an obvious way from the right bottom corner of $A_{3/2}'$ and prove that A_2' is a real symmetric positive definite matrix in an analogous way. The complete proof of the Theorem can be given by induction.

(iv) It has to be pointed out that in the case where A is a real symmetric positive definite matrix then in (2.1) $Z = W^{\mathrm{t}}$ and the algorithm given in the previous sections applies in a straightforward manner. As is obvious the total number of time steps remains the same as in the general case while the maximum number of processors used becomes $(n-1)n/2$. As is also known, there does not exist a free root Choleski type analogue for the QIF1 technique.

(v) It is remarkable to point out that the NQIF2 technique is equivalent from either point of view (that is the number of time steps and the maximum number of processors required) to the LDU method of parallel solution given by Sameh and Kuck [6].

6. Evaluation of the determinant of a matrix

The evaluation of the det(A) by using the NQIF2 technique follows the same steps as in the case of the QIF2 technique (see [2]). The only difference is that A is given by (2.1) and therefore

$$\det(A) = \det(W)\det(D)\det(Z). \tag{6.1}$$

Since the matrix W in the NQIF2 technique is exactly the same as the W factor in QIF2 that for the factor Z in QIF2 we had that $\det(Z) = \Pi_{i=1}^{n} z_{i,i}$. Because of the fact that the matrix Z in the NQIF2 technique is of the same type as the corresponding factor in QIF2 then we immediately obtain for the present case that $\det(Z)=1$. Therefore (6.1) becomes

$$\det(A)=\det(D)=\prod_{i=1}^{n} d_i . \tag{6.2}$$

The product in (6.2) can be obtained in $\lceil \log n \rceil$ parallel time steps, where the symbol $\lceil \alpha \rceil$ denotes the smallest integer not less than the real number α. (The logarithm is taken to the base 2.)

The general belief so far has been that the determinant of a matrix A cannot be evaluated by using the QIF1 technique. This is not true as we shall show in the following sequel. Let W and Z be the two factors in the QIF1 factorisation (see [1]). We have that

$$\det(A)=\det(W)\det(Z). \tag{6.3}$$

Since the form of the W matrix in the QIF1 factorisation is the same as the W matrix in either the QIF2 or the NQIF2 factorisation apart from the fact that the elements in the positions $(n-k+1, k)$, $k=1(1)[n/2]$ are zeros, a proof similar to the one given in [2] leads to the conclusion that $\det(W)=1$. As is known the form of the Z matrix (see [1]) is given in (6.4)

$$Z=\begin{bmatrix} z_{11} & z_{12} & z_{13} & \cdots & z_{1,n-2} & z_{1,n-1} & z_{1n} \\ & z_{22} & z_{23} & \cdots & z_{2,n-2} & z_{2,n-1} & \\ & & z_{33} & \cdots & z_{3,n-3} & & \\ & \mathbf{0} & & & & \mathbf{0} & \\ & & z_{n-2,3} & \cdots & z_{n-2,n-2} & & \\ & z_{n-1,2} & z_{n-1,3} & \cdots & z_{n-1,n-2} & z_{n-1,n-1} & \\ z_{n1} & z_{n2} & z_{n3} & \cdots & z_{n,n-2} & z_{n,n-1} & z_{n,n} \end{bmatrix}. \tag{6.4}$$

To find $\det(Z)$ we multiply the first row of Z by a multiplier such that when this first row is added to the last row the new element in the position $(n, 1)$ becomes zero. As is known the determinant of the new matrix is equal to $\det(Z)$. If now we expand the new determinant according to the elements of its first column we obtain the result in (6.5):

$$\det(Z)=z_{11}\det\left(\begin{bmatrix} z_{22} & z_{23} & \cdots & z_{2,n-1} & 0 \\ & z_{33} & \cdots & 0 & \\ \mathbf{0} & & \cdots & & \mathbf{0} \\ z_{n-1,2} & z_{n-1,3} & \cdots & z_{n-1,n-1} & \\ z_{n2}-\dfrac{z_{n1}}{z_{11}}z_{12} & z_{n3}-\dfrac{z_{n1}}{z_{11}}z_{13} & \cdots & z_{n,n-1}-\dfrac{z_{n1}}{z_{11}}z_{1,n-1} & z_{n,n}-\dfrac{z_{n1}}{z_{11}}z_{1n} \end{bmatrix}\right). \tag{6.5}$$

Now by expanding the determinant of the RHS of (6.5) according to the elements of the last column we obtain (6.6)

Table 4
Evaluation of the determinant of a matrix

Method	Num. of time steps	Max. num. of proc.
QIF1	$3(n-1)+\lceil\log[\frac{1}{2}(n+1)]\rceil-0.5(1+(-1)^n)$	$2(n-2)^2$
QIF2	$4.5(n-1)+\lceil\log n\rceil-0.75(1+(-1)^n)$	$2(n-2)^2$
NQIF2	$3(n-1)+\lceil\log n\rceil$	$(n-1)^2$

$$\det(Z)=\begin{vmatrix} z_{11} & z_{1n} \\ z_{n1} & z_{n,n} \end{vmatrix} \det\left(\begin{bmatrix} z_{22} & z_{23} & \cdots & z_{2,n-2} & z_{2,n-1} \\ & z_{33} & \cdots & z_{3,n-2} & \\ \mathbf{0} & & \cdots & & \mathbf{0} \\ & z_{n-2,3} & \cdots & z_{n-2,n-2} & \\ z_{n-1,2} & z_{n-1,3} & \cdots & z_{n-1,n-2} & z_{n-1,n-1} \end{bmatrix}\right). \tag{6.6}$$

By repeating the process a further $[\frac{1}{2}(n-1)]$ times we can finally obtain by induction that

$$\det(A)=\det(Z)=\prod_{k=1}^{[1/2(n+1)]}\begin{vmatrix} z_{k,k} & z_{k,n-k+1} \\ z_{n-k+1,k} & z_{n-k+1,n-k+1} \end{vmatrix}, \tag{6.7}$$

where in (6.7) the determinant corresponding to the value of $k=[\frac{1}{2}(n+1)]$ for n odd reduces to the single element $z_{k,k}$. Since each determinant in the RHS of (6.7) had been evaluated during the factorisation process, except the one corresponding to the value of $k=[\frac{1}{2}(n+1)]$ if n is even, they do not have to be evaluated again provided that we had kept the corresponding values. For the evaluation of the last determinant, for n even, two extra time steps are needed. Thus, the total number of time steps needed for the evaluation of the RHS of (6.7) is $\lceil\log[\frac{1}{2}(n+1)]\rceil+1+(-1)^n$.

Taking into consideration Table 1 and the result which we have just obtained we form Table 4 in which the total number of time steps and the maximum number of processors required in all three techniques are given.

As is readily found out from Table 4, QIF1 is always better than QIF2 technique. In addition to that, QIF1 is also always better than NQIF2 as far as the total number of times steps required are concerned for any value of $n \geq 3$. From the point of view of the maximum number of processors used, however, NQIF2 is the best of the two for any $n \geq 5$.

References

[1] D.J. Evans and M. Hatzopoulos, A parallel linear system solver, Internat. J. Comput. Math. 7 (1979) 227–238.
[2] D.J. Evans and A. Hadjidimos, A modification of the quadrant interlocking factorisation parallel method, Internat. J. Comput. Math. 8 (1980) 149–166.

[3] D.J. Evans and A. Hadjidimos, Parallel solution of certain banded, symmetric and centrosymmetric systems by using the quadrant interlocking factorisation method, T.R. No 71, Department of Computer Studies, Loughborough University of Technology, Loughborough, Leicestershire, U.K., 1978; Math. Comput. Similation (1981), in press.
[4] M.J. Flynn, Some computer organizations and their effectivness, IEEE Trans. on Computers, C-21 (1972) 948–960.
[5] H.S. Stone, Problems of parallel computation, in: J.F. Traub, ed., Complexity of Sequential and Parallel Numerical Algorithms (Academic Press, New York, 1973) pp. 1–16.
[6] A.H. Sameh and D.J. Kuck, Linear system solvers for parallel computers, T.R. No. UIUCDCS-R-75-701, Department of Computer Science, University of Illinois, U.S.A., 1975.
[7] J.J. Lambiotte, The solution of linear systems of equations on a vector computer, Ph.D. Thesis, University of Virginia, U.S.A., 1975.
[8] D.J. Evans and M. Hatzopoulos, The solution of certain banded systems of linear equations using the folding algorithm, Comput. J. 19 (1976) 184–187.
[9] D.M. Young, Iterative Solution of Large Linear Systems (Academic Press, New York, 1971).