International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455451

The parallel solution of banded linear equations by the new quadrant interlocking factorisation (q.i.f.) method
D. J. Evans a; A. Hadjidimos b; D. Noutsos b
a Department of Computer Studies, Loughborough University of Technology, Loughborough, Leicestershire, U.K
b Department of Mathematics, University of Ioannina, Ioannina, Greece

Online Publication Date: 01 January 1981

To cite this Article Evans, D. J., Hadjidimos, A. and Noutsos, D.(1981)'The parallel solution of banded linear equations by the new quadrant interlocking factorisation (q.i.f.) method',International Journal of Computer Mathematics,9:2,151 — 161
To link to this Article DOI: 10.1080/00207168108803237
URL: http://dx.doi.org/10.1080/00207168108803237

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
The Parallel Solution of Banded Linear Equations by the New Quadrant Interlocking Factorisation (Q.I.F.) Method

D. J. EVANS
Department of Computer Studies, Loughborough University of Technology, Loughborough, Leicestershire, UK

and

A. HADJIĐIMOS and D. NOUTSOS
Department of Mathematics, University of Ioannina, Ioannina, Greece

Received October, 1991

In this paper, the authors extend the applicability of the new quadrant interlocking factorisation method [7] to the solution of the banded systems of linear equations which occur in the finite difference and finite element discretisation of engineering problems.

KEY WORDS: Banded Linear Systems, Quadrant Interlocking Factorisation (Q.I.F.)

C.R. CATEGORIES: 5.14, 6.22

1. INTRODUCTION

After the introduction of the Folding Algorithm by Evans and Hatzopoulos [1], the same authors conceived the idea of a novel technique (see [2]), called the Quadrant Interlocking Factorisation (QIF1) technique, suitable for the numerical solution of a linear system on a Parallel Computer of the type “Single Instruction Stream-Multiple Data Streams” (SIMD) (see Flynn [3] and Stone [4]). A modification to the QIF1 technique, called the QIF2 method, was given later in a paper by Evans and Hadjidimos [5]. This paper was followed by a further new process by the same authors [6] in which both the QIF1 and QIF2
techniques were applied to certain banded, symmetric and centro-symmetric systems.

Very recently the present authors have improved drastically the QIF2 technique [7] by proposing a new method called the NQIF2 technique. The new method is in many respects at least as good as the one introduced by Sameh and Kuck [8] which, in turn, is based on a parallel LU-type factorisation. In [7] various comparisons amongst the three Quadrant Interlocking Factorisation techniques were made which showed that in every case examined either the QIF1 or NQIF2 technique was the best one to use.

The purpose of the present paper is (i) to apply the NQIF2 technique for the numerical solution of certain banded linear systems, (ii) to apply both the QIF1 and NQIF2 techniques for the evaluation of the determinant of a banded matrix, and (iii) to compare the QIF1 and NQIF2 techniques from the point of view of the total number of time steps as well as the maximum number of processors required in each case. Thus, the results given in [7] are generalised and extended in directions which have not been considered so far.

2. NQIF2 TECHNIQUE FOR GENERAL BANDED SYSTEMS

We consider the system (2.1)

\[ A x = b, \]

(2.1)

where \( A \) is a non-singular banded matrix of order \( n \) and semibandwidth \( p,d(n) \) and \( x \) and \( b \) are two \( n \)-dimensional vectors with \( x \) unknown and \( b \) known and corresponding components \( x_i, b_i \) \( i=1(1)n \). We now assume that there exists three matrices \( W, D, Z \) such that

\[ A = WDZ. \]

(2.2)

These matrices are defined as follows: the matrix \( D \) is diagonal of order \( n \) with elements \( d_i, i=1(1)n \) while the forms of the \( W \) and \( Z \) matrices are given below in (2.3) and (2.4) (see also [7]). Thus, we have

\[
W = \begin{bmatrix}
1 & 0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \frac{w_{1,2}}{w_{1,1}} & \frac{w_{2,2}}{w_{1,1}} & \frac{w_{3,3}}{w_{1,1}} & \ldots & \frac{w_{n,n}}{w_{1,1}} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \frac{w_{2,1}}{w_{1,1}} & \frac{w_{3,1}}{w_{1,1}} & \frac{w_{4,1}}{w_{1,1}} & \ldots & \frac{w_{n,1}}{w_{1,1}} \\
0 & 0 & 0 & 0 & \ldots & 1 \\
0 & 0 & 0 & 0 & \ldots & 1 \\
\end{bmatrix}
\]
and

\[
Z = \begin{bmatrix}
1 & z_{12} & z_{13} & \cdots & z_{1,n-2} & z_{1,n-1} & z_{1,n} \\
1 & z_{23} & z_{2,2} & \cdots & z_{2,n-2} & z_{2,n-1} \\
1 & \cdots & z_{3,n-2} \\
0 & \cdots & 0 \\
0 & \cdots & 1 \\
0 & z_{n-1,3} & \cdots & z_{n-1,n-2} & z_{n,n-1} & 1 \\
0 & z_{n2} & z_{n3} & \cdots & z_{n,n-2} & z_{n,n-1} & 1
\end{bmatrix}.
\]

(2.4)

We can easily infer that because of the banded form of matrix \( A \) in (2.1) both matrices \( W \) and \( Z \) above are banded ones of the same semibandwidth \( p \).

As was made in [6], for the QIF1 and QIF2 techniques, we shall assume that during the factorisation process by the NQIF2 technique, no breakdown occurs so that we do not have to interchange columns for numerical stability reasons and therefore destroy the semibandwidth \( p \) form of the original matrix. Two sufficient conditions to avoid breakdown are (i) the matrix \( A \) to be diagonally dominant (see Theorem 1 and 2 of Section 5 of [5]) and (ii) the matrix \( A \) to be a real symmetric positive definite one (see Theorem 1 of Section 4 of [7]).

3. THE COMPUTATION OF \( W, D \) AND \( Z \) MATRICES

From (2.2) we have that

\[
A = \sum_{i=1}^{n} d_i W_i Z_i^T,
\]

(3.1)

where \( W_i \) and \( Z_i \mid i=1(1)n \) are the column vectors of the matrices \( W \) and \( Z^T \) respectively. Now, we define the matrices \( A_k, A_{k+1/2} \mid k=1(1)[n-1/2] \) (where \([a]\) denotes the largest integer not greater than the real number \( a \)) such that

\[
A_1 = A
\]

\[
A_k = A - \sum_{i=1}^{k-1} d_i W_i Z_i^T - \sum_{i=n-k+2}^{n} d_i W_i Z_i \quad \mid k = 2(1) \left[ \frac{n-1}{2} \right]
\]

(3.2)

\[
A_{k+1/2} = A - \sum_{i=1}^{k} d_i W_i Z_i^T - \sum_{i=n-k+2}^{n} d_i W_i Z_i^T \quad \mid k = 1(1) \left[ \frac{n-1}{2} \right].
\]
From (3.2) we obtain

\[ A_{1,1} = A_0 - d_1 \bar{w}_1 \bar{Z}_1 \]
\[ A_{1,2} = A_0 - d_2 \bar{w}_2 \bar{Z}_2 \]
\[ \vdots \]
\[ A_{k-1,1} = A_0 - d_{k-1} \bar{w}_{k-1} \bar{Z}_{k-1} \]

and from the same relationships it is easily seen that the first and last \((k-1)\) rows and columns of any \(A_k\) matrix as well as the first \(k\) and the last \((k-1)\) rows and columns of any \(A_{k-1}\) matrix are zero.

As is known, the basic \(NQIF2\) factorization algorithm consists of the following 6 steps (see [7]):

1. For \(k = 1\) \((1)\left[\left(\frac{n-1}{2}\right)\right]\), where

\[ a_{ij} \] denotes the element of the matrix \(A_{k-1}\) in the \((q, r)\) position, while

\[ \bar{a}_{ij} \] denotes the \(i\)th column vector of the \(A_{k-1}\) matrix. Since the original matrix is banded we therefore have

\[ d_{ij} = 0 \]

for \(i, j = 1 \((1)\)\left[\left(\frac{n-1}{2}\right)\right]\) and \(p \) and it can easily be proved by induction that for \(k = 1\) \((1)\left[\left(\frac{n-p}{2}\right)\right]\) the basic algorithm, presented previously in (3.4), reduces to the following:

2. For \(k = 1\) \((1)\) \((\frac{n}{2})\), we then apply algorithm (3.4) for \(k = \left(\frac{n}{2}\right) + 1\) \((1)\) \((\frac{n-1}{2})\). To complete the computation of all the elements concerned one more stage for \(k = \left(\frac{n-1}{2}\right)\) is necessary. To be more specific, if \(n\) is odd, step (3.4a) is executed to find

\[ d_1 = a_{11} - d_1 \bar{w}_1 \bar{Z}_1 \]
\[ z_{11} = d_{11} \]

...
the value $d_i$ while if $n$ is even, steps (3.5a)-(3.5d) are executed to find the elements $d_i, d_{i+1}, d_{i+2}, \ldots$ and $u_{n-i+1}$.

As is usually made (see [2], [5], [6], [7] and [8]) it is assumed that a parallel replacement statement requires negligible time while any other parallel arithmetic operation or comparison between two numbers needs the same time step. Thus, to find the total number of time steps for the factorisation process we observe that when $k = \lfloor n/2 \rfloor$ (under the assumption that $p < \lfloor n/2 \rfloor$) the non-zero elements of the two products of the matrices appearing in (3.5c) do not overlap. This simply suggests that for the aforementioned values of $k$ instead of making the two subtractions in two time steps we can make them simultaneously in one. So, the total number of time steps for the complete factorisation of the matrix $A$ can be found equal to

$$3(n-1) - 2 \left[ \frac{n-p}{2} - 0.5 \left( \frac{n}{2} - p \right) - 0.5 \right]$$

(3.6)

At the same time a maximum number of $\max \{2p^2, 4p\}$ processors working in parallel are needed to perform all the operations involved.

Following a similar reasoning and bearing in mind the results in [6] and [7] we have that a total number of

$$3(n-1) - 1.5(1 + (-1)^{p}) - \frac{n-p}{2} - 0.5 \left( \frac{n}{2} - p \right) - 0.5 \left( \frac{n}{2} - p \right)$$

(3.7)

time steps are required for the complete factorisation of $A$ by using the QIF1 technique while the maximum number of processors is given by the expression max $\{2p^2, 2(p+1)\}$.

Comparing now the QIF1 and QIF2 techniques, we have that from the point of view of the total number of time steps, (3.6) and (3.7) give that for $n$ odd they are equivalent, while for $n$ even, QIF1 is slightly better. From the point of view of the maximum number of processors working in parallel they are equivalent for any value of $n$ and $p$.

4. SOLUTION OF THE LINEAR SYSTEM

To solve system (2.1) we introduce the auxiliary vectors $y$ and $u$, with components $y_i, u_i$ for $i = 1 \ldots n$, so that in view of (2.2) it is sufficient and necessary to solve the following three simpler systems

$$W y = h$$

(4.1)
To solve system (4.1) we let

\[ W_y = b - b^{(1)} \]

so that \( \sum_{i=1}^{n} y_i W_i = b \) and we then introduce, as in [7], the vectors \( b^{(k+1/2)} \) and \( b^{(k+1)} \) such that

\[ b^{(k+1)} - b^{(k+1/2)} = b^{(k)} - y_{k+1} W_{k+1}, \quad b^{(k+1/2)} = b^{(k)} + y_{k+1} W_{k+1}. \]

Now, because of the fact that the \( W \) matrix is a banded one with semibandwidth \( p \), the NQIF2 algorithm, for \( k = l \lfloor (n-p)/2 \rfloor \), has the form

\[ \begin{align*}
  a) & \quad y_k = b^{(k)} - y_{k+1} W_{k+1}, \\
  b) & \quad b^{(k+1)} = b^{(k)} - y_{k+1} W_{k+1}, \\
  c) & \quad y_{k+1} = b^{(k+1/2)} + y_{k+2} W_{k+2}.
\end{align*} \]

For \( k = l \lfloor (n-p)/2 \rfloor + 1 \lfloor (n-1)/2 \rfloor \), as in [7], the algorithm becomes

\[ \begin{align*}
  a) & \quad y_k = b^{(k)} - y_{k+1} W_{k+1}, \\
  b) & \quad b^{(k+1/2)} = b^{(k)} - y_{k+1} W_{k+1}, \\
  c) & \quad y_{k+1} = b^{(k+1/2)} + y_{k+2} W_{k+2}.
\end{align*} \]

In addition to the execution of the algorithms (4.4) and (4.5) for the appropriate values of \( k \), step (4.5a) has to be executed for \( k = l \lfloor (n+1)/2 \rfloor \), if \( n \) is odd, to find the element \( y_k \); while if \( n \) is even, steps (4.5a)-(4.5c) have to be executed for the same value of \( k \) to find the values \( y_k \) and \( y_{k+1} \). It can easily be seen then that the total number of time steps for the solution of (4.1) is equal to

\[ 2(n-1) - \left\lfloor \frac{n-p}{2} \right\rfloor - 0.5 \left\lfloor \frac{n}{2} \right\rfloor - 0.5 \left( \left\lfloor \frac{n}{2} \right\rfloor - p \right), \]

while the maximum number of processors working in parallel at the same time is \( 2p \).

System (4.2) is solved in one time step by using the simple algorithm

\[ u_i = y_i / d_i, \quad (i = 1 \ldots n) \]
and a number of \( n \) processors.

To solve system (4.3) we proceed as in [7]. Thus, 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 \( u^{(1)} \) and \( u^{(i+1/2)} | i = 1(1)[(n + 1)/2] \) such that

\[
\begin{align*}
  u^{(1)} &= u \\
  u^{(l-k+3/2)} &= u^{(l-k+1)} - x_{n-k+1}Z_{n-k+1}^* \\
  u^{(l-k+2)} &= u^{(l-k+3/2)} - x_kZ_k^*,
\end{align*}
\]

where \( l = [(n + 1)/2] \) and \( k = [(n + 1)/2](-1)1 \), except for \( n \) odd and \( k = l \) when we put \( u^{(3/2)} = u^{(1)} = u \) and \( u^{(2)} = u^{(1)} - x_iZ_i^* \). If \( n \) is odd, we start with

\[
\begin{align*}
  a) & \quad x_i = u_i^{(1)} \\
  b) & \quad u^{(2)} = u^{(1)} - x_iZ_i^*, \quad (4.6)
\end{align*}
\]

while if \( n \) is even, we start with

\[
\begin{align*}
  a) & \quad x_{i+1} = u_{i+1}^{(1)} \\
  b) & \quad u^{(3/2)} = u^{(1)} - x_{l+1}Z_{l+1}^* \\
  c) & \quad x_i = u_i^{(3/2)} \\
  d) & \quad u^{(2)} = u^{(3/2)} - x_iZ_i^*. \quad (4.7)
\end{align*}
\]

Then, the basic algorithm for \( k = [(n - 1)/2](-1)[(n - p)/2] + 1 \) is the following

\[
\begin{align*}
  a) & \quad x_{n-k+1} = u_{n-k+1}^{(l-k+1)} \\
  b) & \quad u^{(l-k+3/2)} = u^{(l-k+1)} - x_{n-k+1}Z_{n-k+1}^* \\
  c) & \quad x_k = u_k^{(l-k+3/2)} \\
  d) & \quad u^{(l-k+2)} = u^{(l-k+3/2)} - x_kZ_k^*. \quad (4.8)
\end{align*}
\]

For \( k = [(n - p)/2](-1)1 \) the algorithm (4.8) reduces to

\[
\begin{align*}
  a) & \quad x_k = u_k^{l-k+1}, \quad x_{n-k+1} = u_{n-k+1}^{(l-k+1)} \\
  b) & \quad u^{(l-k+2)} = u^{(l-k+1)} - x_kZ_k^* - x_{n-k+1}Z_{n-k+1}^*, \quad (4.9)
\end{align*}
\]

where step (4.9b) is not executed when \( k = 1 \). In (4.9) we can observe that the two half steps of the basic algorithm (4.8) have not been used and also that because of the form of the banded matrix \( Z \), the two subtractions can be performed in one time step. Thus, from (4.6)–(4.9) we find out that the
solution of system (4.3) can be obtained in a total number of \(2(n-1)-2[n-p-2]\) time steps by using a maximum number of \(2p\) processors.

By virtue of the analysis made so far we come to the conclusion that the complete solution of system (2.1), after the factorisation has taken place, by using NQIF2 technique requires a total number of

\[
4(n-1)-\frac{n-p}{2}-0.5\left(\frac{n}{2}\right)-0.5\left(\frac{n}{2}+p\right)+1
\]

(4.10)
time steps and a maximum number of \(\max\{2p, n\}\) processors working in parallel at the same time. The corresponding numbers when using QIF1 technique are

\[
4.5(n-1)-6.75\{1+(-1)^\rho\}-3\left(\frac{n-p}{2}\right)-0.5\left(\frac{n}{2}\right)-0.5\left(\frac{n}{2}+p\right)+1
\]

(4.11)
and \(2p\) respectively (see [6] and [7]).

Consequently for the solution of system (2.1) we have from (4.10) and (4.11) that NQIF2 is always better than QIF1 except for \(n=4\) when they are equivalent. However, from the point of view of the maximum number of processors both methods are equivalent unless \(p < n/2\) (which is usually the case) where QIF1 is the best of the two.

Taking into account the results so far we have from (3.6), (3.7), (4.10) and (4.11) that the total number of time steps for both the factorisation and the solution of the system are

\[
7(n-1)-5\left(\frac{n-p}{2}\right)-\frac{n}{2}-0.5\left(\frac{n}{2}+p\right)+1
\]

(4.12)
for the NQIF2 technique and

\[
7.5(n-1)-22.5\{1+(-1)^\rho\}-5\left(\frac{n-p}{2}\right)-\frac{n}{2}-0.5\left(\frac{n}{2}+p\right)+1
\]

(4.13)
for the QIF1 one. A simple comparison of the two expressions in (4.12) and (4.13) shows that NQIF2 is better than QIF1 for all \(n\) except for \(n = 4, 6\) and \(8\) where QIF1 is the best of the two and for \(n = 10\) where both techniques are equivalent. Taking now into account the maximum number of processors working in parallel at the same time we have that these numbers are \(\max\{2p^2, 4p, n\}\) for NQIF2 and \(\max\{2p^2, 2(p+1)\}\) for QIF1. Thus, from this point of view QIF1 is better than NQIF2 unless \(n \leq 2p^2\) when the two techniques are equivalent.
5. EVALUATION OF THE DETERMINANT OF A BANDED MATRIX

To evaluate $\det(A)$ by using the NQIF2 technique first we make the factorisation of $A$ as in Section 5 and then we compute it from the product (see [7])

$$\det(A) = \prod_{i=1}^{n} d_i$$

The product in (5.1) can be computed in $\lceil \log n \rceil$ parallel time steps by using $\lceil n/2 \rceil$ processors working in parallel at the same time. The symbol $\lceil x \rceil$ just used denotes the smallest integer not less than the real number $x$, while the logarithm is taken to the base 2.

When we use the QIF2 technique to evaluate $\det(A)$ first we make the factorisation as in Section 2 of [6] and then we use the relationship (6.1) of [7] which is the following.

$$\det(A) = \prod_{k=1}^{[n+1]/2} z_{-k,k-1} z_{-k+1,k-1}$$

where the determinant corresponding to the value of $k=[(n+1)/2]$, for $n$ odd, reduces to the single element $z_{-k,k-1}$. Since, however, the matrix $Z$ is a banded matrix of semibandwidth $p$ we shall have

$$z_{-k,k-1} = z_{k,-k+1} = 0 \quad |k = \{1\} \left\lceil \frac{n-p}{2} \right\rceil$$

Because of (5.3), (5.2) becomes

$$\det(A) = \prod_{k=1}^{[n+1]/2} z_{-k,k-1} z_{-k+1,k-1} \prod_{k=[p/2]+1}^{[(n+1)/2]} z_{-k,k-1} z_{k,-k+1}$$

Since the determinants in (5.4) have been evaluated during the corresponding factorisation process, except the last determinant if $n$ is even, the RHS of (5.4) represents in fact the product of $\lceil (n+1)/2 \rceil \times \lceil (n-p)/2 \rceil$ numbers which can be computed in

$$\log \left( \left\lceil \frac{n+1}{2} \right\rceil \left\lceil \frac{n-p}{2} \right\rceil \right) + 1 + (-1)^p$$
time steps by using a maximum number of
\[ \left\lfloor \frac{\frac{n+1}{2}}{\frac{n+2}{2}} \right\rfloor \]

processors.

Taking into account the analysis of this section as well as the results of Section 3 we have that the total number of time steps required for the evaluation of \( \det(A) \) by using the NQIF2 technique is given by the expression
\[ 3(n - 1) - 2 \left[ \frac{n - p}{2} - 0.5 \left[ \frac{n}{2} - 0.5 \left( \frac{n}{2} - p \right) \right] + \log n \right], \quad (5.5) \]
while the corresponding number by using the QIF1 technique is
\[ 3(n - 1) - 0.5(1 + (-1)^{p-1}) - 2 \left[ \frac{n - p}{2} - 0.5 \left[ \frac{n}{2} - 0.5 \left( \frac{n}{2} - p \right) \right] + \log \left( \left\lfloor \frac{n+1}{2} \right\rfloor - \left\lfloor \frac{n-p}{2} \right\rfloor \right) \right], \quad (5.6) \]

Since as is easily found out
\[ \left\lfloor \log \left( \frac{n+1}{2} \right) \right\rfloor \leq \left\lfloor \log \left( \frac{n}{2} + \frac{n-p}{2} \right) \right\rfloor \]
\[ \leq \left\lfloor \log \left( \frac{n+1}{2} - \left\lfloor \frac{n-1}{2} \right\rfloor \right) \right\rfloor \leq \log n \quad (5.7) \]
and also that
\[ \left\lfloor \log \left( \frac{n+1}{2} \right) \right\rfloor \leq [\log n] - 1 \quad (5.8) \]

we have from (5.5)-(5.8) that from the point of view of the total number of time steps both techniques are equivalent or QIF1 is better than NQIF2 by one or at most two time steps. As far as the maximum number of processors working in parallel at the same time is concerned we have that for the NQIF2 technique this number is \( \max \left\lfloor \frac{2^n}{4p} \right\rfloor \) while for
PARALLEL BANDED LINEAR SYSTEM SOLVER

the QIF1 technique the corresponding number is

\[ \max \left\{ 2p^2, 2(p+1) \right\} \]

Comparing these two numbers we have that if (i) \( p=1 \) and \( \frac{n}{2} \leq 4 \) or (ii) if \( p > 1 \) and \( \frac{11}{2} \leq 2^p \) the two techniques are equivalent, while in any other case we obtain

\[ \left( \frac{n+1}{2} + \frac{n-p}{2} \right) = \left( \frac{n}{2} + \frac{3-\beta}{4} \right) \text{ if } n \text{ is odd,} \]

\[ \left( \frac{n}{2} + \frac{\beta}{4} \right) \text{ if } n \text{ is even.} \]

Thus, we have that if \( n \) is odd and \( p \leq 3 \) the two techniques are equivalent otherwise QIF1 technique is the best of the two techniques.

References