COMPUTATIONAL MATHEMATICS
BANACH CENTER PUBLICATIONS, VOLUME 13
PWN–POLISH SCIENTIFIC PUBLISHERS
WARSAW 1984

A PARALLEL ALGORITHM FOR SOLVING BAND SYSTEMS AND MATRIX INVERSION

LADISLAV HALADA

Institute of Technical Cybernetics, Slovak Academy of Sciences, 809 31 Bratislava, Czechoslovakia

1. Introduction

Recently, several articles appeared regarding solving systems of linear equations $Ax = b$ on parallel computers where A is a real n by n dense, triangular or tridiagonal matrix ([1]–[6]). Essentially less effort has so far been given to handling this problem on parallel computers if the matrix had another structure, e.g. a band structure with non-scalar constant coefficients, although there exist studies of that type for serial computers.

Thus, while there exist several parallel algorithms requiring only $O(\log n)$ (1) time steps to solve a tridiagonal system, there are no such fast algorithms, for the time being, for band systems with bandwidth $2m+1$, where $m > 1$.

In this paper a new parallel direct algorithm for solving such systems is discussed. This algorithm on SIMD type parallel machine requires $(2+\log 2m)\log n + O(m\log m)$ time steps using no more than $(3m^2+m)n$ processors. For a tridiagonal systems this means $3\log n + O(1)$ steps using $4n$ processors, which is the least time we know of for solving tridiagonal systems. The algorithm can be formally interpreted as parallel shooting method. It is based on a factorization of A such as in [7]; in this way the problem of solving a system with a band matrix gets transformed to that of solving systems with a banded lower triangular matrix. The algorithm is a generalization of the algorithm to be published in [1] for a tridiagonal system.

We also develop a parallel direct algorithm for solving a system involving the inverse of such a band matrix. Its application is advan-

(1) Throughout this paper, $\log p = \lceil \log_2 p \rceil$, and time is measured in steps.

tageous if the computation of A^{-1} is a part of the solving of $Ax = b$ and it is necessary to know only certain selected rows or columns of A^{-1}. If A is a symmetric matrix, the solution of both problems together requires $(4+\log 4m^2)\log n + O(m\log m)$ time steps using $mn^2/4 + O(m^2 n)$ processors.

Throughout the paper we assume that any number of processors can be used at any time, but we give bounds on this number. All processors are assumed to perform the same operations; determining the maximum magnitude of two numbers can be performed in one time step.

2. A parallel band systems solver

Let us consider linear systems of equations $Ax = b$, where A is a non-singular band matrix of order n with bandwidth $2m+1$, i.e. $a_{ij} = 0$ for $|i-j| > m$ and $a_{i,i+m} \neq 0$ for $i = 1, 2, \ldots, n-m$. We assume a situation frequent in practice, $m \ll n$, or, in a worse case, $m < n/3$. Such a system can be written in the form

$$\begin{bmatrix} C & T \\ O_m & S \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \end{bmatrix} \tag{1}$$

where T and O_m are square matrices of order $n-m$ and m, respectively. Specifically, T can be a lower band triangular matrix with bandwidth $2m+1$ and O_m the zero matrix. The submatrices C and S are generally rectangular of size $(n-m)\times m$ and $m\times(n-m)$, respectively. The vectors $x = (x_1, x_2, \ldots, x_n)$ and $b = (b_1, b_2, \ldots, b_n)$ are partitioned into $x^{(i)}$ and $b^{(i)}$, $i = 1, 2$, conformably with A.

Because T is a non singular matrix, we get from the first equation of (1)

$$x^{(2)} = T^{-1}(b^{(1)} - Cx^{(1)}). \tag{2}$$

Let us assume the existence of a fast parallel method for the solution of triangular linear systems. Let $y^{(i)}$, $i = 0, 1, 2, \ldots, m$, be the solutions of such systems with the right-side vectors

$$T(y^{(0)}\, y^{(1)} \ldots y^{(m)}) = (b^{(1)} C) \tag{3}$$

Thus, (2) can be expressed as

$$x^{(2)} = y^{(0)} - \sum_{k=1}^{m} x_k y^{(k)}, \tag{4}$$

i.e. the last $n-m$ elements of x are linear combinations of the first m ones. When applying (4) to the second equation of (1) we get

$$-S(y^{(1)}\, y^{(2)} \ldots y^{(m)})x^{(1)} = b^{(2)} - Sy^{(0)}. \tag{5}$$

By LU decomposition of A the following equality can be shown:

$$\det(A) = \det(T)\cdot\det\big(S(y^{(1)}\, y^{(2)} \dots y^{(m)})\big)(-1)^{mn-m}.$$

Hence, the non-singularity of the matrix $S(y^{(1)} \dots y^{(m)})$ follows immediately from the assumptions. The elements of this matrix and the vector on the right-hand side of (5) can be solved by extension of the triangular system of (3) to the system

$$\begin{bmatrix} T & O_{n-m} \\ S & I_m \end{bmatrix} (z^{(0)}\, z^{(1)} \dots z^{(m)}) = \begin{bmatrix} b^{(1)} & C \\ b^{(2)} & O_m \end{bmatrix}. \tag{6}$$

Thus we get the following assertion.

LEMMA 1. *Let A be a non singular band matrix of order n with bandwidth $2m+1$ and suppose that the elements of the uppermost line above the diagonal are non-zero. Then the unknowns of the system $Ax = b$ fulfil the equation*

$$\begin{bmatrix} z_{n-m+1}^{(1)} & z_{n-m+1}^{(2)} & \cdots & z_{n-m+1}^{(m)} \\ z_{n-m+2}^{(1)} & z_{n-m+2}^{(2)} & \cdots & z_{n-m+2}^{(m)} \\ \cdots & \cdots & \cdots & \cdots \\ z_n^{(1)} & z_n^{(2)} & \dots & z_n^{(m)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_m \end{bmatrix} = \begin{bmatrix} z_{n-m+1}^{(0)} \\ z_{n-m+2}^{(0)} \\ \cdot \\ \cdot \\ \cdot \\ z_n^{(0)} \end{bmatrix}, \tag{7}$$

$$x_{m+i} = z_i^{(0)} - \sum_{k=1}^{m} x_k z_i^{(k)}, \quad i = 1, 2, \dots, n-m, \tag{8}$$

where $z_i^{(k)}$ is the i-th component of the vector $z^{(k)}$ which is the solution of (6).

Hence, the direct parallel algorithm for solving $Ax = b$ of order n, can be divided into 3 *stages. We present it, evaluating the time consumed.*

Stage E1. *The solution of the lower band triangular system* (6) *of order n: There are fast parallel algorithms for solving such systems. Algorithm* II ([2]) *requires the smallest number of steps and processors. If we apply this algorithm for solving* (6) *in such a way that it solves simultaneously $m+1$ band triangular systems differing from each other only by the right-hand side, then this stage requires $(2+\log 2m)\log n-(1/2)(\log^2 2m+ +\log 2m)+3$ steps using no more than $(3m^2+m)n-8m^3$ processors.*

Stage E2. *The solution of the dense system* (7) *of order m: Solving this system by Gaussian elimination with pivoting requires $3m(\log m-1)+ +O(\log^2 m)$ steps using $(m-1)^2$ processors.*

Stage E3. *The computation of* (8): *The computation of* (8) *consists of $n-m$ independent scalar products of two vectors of order $m+1$. They can be performed in $1+\log(m+1)$ steps using $(m-1)(n-m)/2$ processors.*

Thus we have proved the following theorem.

THEOREM 1. *Let A be a matrix as in Lemma 1. Then the solving of $Ax = b$ requires $(2+\log 2m)\log n + O(m\log m)$ steps using no more than $(3m^2+m)n$ processors.*

We see that this means $3\log n + O(1)$ steps and $4n$ processors for a tridiagonal matrix. A comparison with Theorem 3.1 of [2] shows that the complexity of computation of a band system with bandwidth $2m+1$ and the lower band triangular systems with bandwidth $2m+1$ is the same, for $m \ll n$. A difference occurs only in the number of processors used. A band system requires 6 times more processors.

Although quite satisfactory from the viewpoint of algebraical complexity, like many shooting methods, this algorithm suffers from the exponential growth in roundoff error and from the possibility of over- or underflow. These drawbacks are due to the parallel algorithm for solving lower band triangular systems (6).

On the other hand, the algorithm does not fail if any of the leading principal submatrices is singular. It can be also used for solving band systems with a different number of non-zero super and subdiagonal lines or with matrices of semi-band form, but then the elements of the uppermost line above the diagonal have to be non-zero.

3. A parallel matrix inversion algorithm

Let us assume that it is necessary, along with the solution of $Ax = b$, to know also some rows or columns of A^{-1}. Such a problem can occur for example in the case when we want to obtain the solution of an equation $Bv = w$ on the basis of the solution of $Ax = b$, where B differs only slightly from A, so that B is naturally thought of as a modification of A [8]. Therefore it is useful to know algorithms which make it possible, on the base of the solution of $Ax = b$, to compute some selected rows or columns of A^{-1}. In the sequel such a parallel algorithm will be designed. The procedure is based on the following assertion.

LEMMA 2. *Let A be a matrix as in Lemma 1. Then for the elements a_{ij} of the matrix A^{-1}, $i \leqslant j$, $i = 1, \ldots, n$, the following relations hold:*

I1.

$$\begin{bmatrix} a_{1,n-m+1} & a_{1,n-m+2} & \cdots & a_{1n} \\ a_{2,n-m+1} & a_{2,n-m+2} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m,n-m+1} & a_{m,n-m+2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} z_{n-m+1}^{(1)} & z_{n-m+1}^{(2)} & \cdots & z_{n-m+1}^{(m)} \\ z_{n-m+2}^{(1)} & z_{n-m+2}^{(2)} & \cdots & z_{n-m+2}^{(m)} \\ \cdots & \cdots & \cdots & \cdots \\ z_{n}^{(1)} & z_{n}^{(2)} & \cdots & z_{n}^{(m)} \end{bmatrix}^{-1}; \tag{9}$$

I2.

$$(10)\qquad [a_{j1}, a_{j2}, \ldots, a_{j,n-m}]T = -[a_{j,n-m+1}, a_{j,n-m+2}, \ldots, a_{jn}]S, \qquad j = 1, 2, \ldots, m;$$

I3.

$$(11)\quad a_{m+i,j} = -\sum_{k=1}^{m} a_{kj} z_i^{(k)}, \quad j = m+1, m+2, \ldots, n, \ i = 1, 2, \ldots, j-m.$$

Proof. Let us consider the matrix A^{-1} in the form

$$A^{-1} = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}$$

where M_2 and M_3 are square matrices of order m and $n-m$, respectively. Matrices M_1 and M_4 are partitioned conformably with M_2 and M_3.

I1. From the equation $AA^{-1} = I_n$ we have

$$(12)\qquad \begin{aligned} CM_2 + TM_4 &= O_{(n-m)\times m}, \\ SM_4 &= I_m. \end{aligned}$$

By the elimination of M_4 we obtain

$$-S(y^{(1)}\, y^{(2)} \ldots y^{(m)})M_2 = I_m,$$

i.e.

$$(13)\qquad M_2 = \left(-S(y^{(1)}\, y^{(2)} \ldots y^{(m)})\right)^{-1} = \begin{bmatrix} z_{n-m+1}^{(1)} & z_{n-m+1}^{(2)} & \ldots & z_{n-m+1}^{(m)} \\ \cdot & \cdot & \cdot & \cdot \\ z_n^{(1)} & z_n^{(2)} & \ldots & z_n^{(m)} \end{bmatrix}^{-1}.$$

I2. The validity of (10) follows from $A^{-1}A = I_n$ because we get from it $M_1 T = -M_2 S$.

I3. Let $a_{\cdot j}$ represent the jth column of A^{-1}, where $m < j \leqslant n$. Then $Aa_{\cdot j} = e_j$, where e_j is the jth column of I_n. According to Lemma 1 the last $n-m$ components of $a_{\cdot j}$ fulfil

$$(14)\qquad a_{m+i,j} = z_i^{(0j)} - \sum_{k=1}^{m} a_{kj} z_i^{(k)}, \quad i = 1, 2, \ldots, n-m,$$

where $y^{(0j)}$ is the solution of the band triangular system

$$\begin{bmatrix} T & O_{n-m} \\ S & I_m \end{bmatrix} y^{(0j)} = e_j.$$

It can be easily seen that $y_i^{(0j)} = 0$, for $i = 1, 2, \ldots, j-1$, and therefore (14) reduces to (11).

The algebraical complexity of the stages I1–I3 is the following.

Stage I1. Any column of the inverse matrix of order m by Gaussian elimination with pivoting requires $3m(\log m-1)+O(\log^2 m)$ steps using $(m-1)^2$ processors. In the same time all columns can be computed using no more than $m(m-1)^2$ processors.

Stage I2. The system (10) can be readily adapted so that Algorithm II [2] may be used for its solution. Then this stage requires, with the computation of the right-side vector, $(2+\log 2m)\log(n-m)-(1/2)\times$ $\times(\log^2 2m+\log 2m-2\log m)+4$ steps and $3m^2n+mn-8m^3$ processors.

Stage I3. The computation of (11) consists of $(n-m)(n-m+1)/2$ independent scalar products of two vectors of length m, which can be performed using $(1/4)\big(mn^2-(2m-1)mn\big)+O(m^3)$ processors in $\log m+1$ steps.

Thus we can state a theorem.

THEOREM 2. *Let A be a matrix as in Lemma 1. Knowing the elements $z_i^{(j)}$, $j=1,2,\ldots,m$; $i=n-m+1,\ldots,n$, the upper triangular portion of A^{-1} can be computed in $(2+\log 2m)\log(n-m)+O(m\log m)$ steps using $(1/4)\big(mn^2-(2m-1)mn\big)+O(m^3)$ processors.*

From the above theorems it follows that for a symmetric matrix A and for an n which is a power of 2, the solution of $Ax=b$ and A^{-1} can be obtained in $(4+\log 4m^2)\log n+O(m\log m)$ steps using $mn^2/4+O(m^2n)$ processors. For a tridiagonal symmetric matrix this means $6\log n+O(1)$ steps using $n(n-1)/4+O(1)$ processors.

If A is not symmetric, but all the elements of the lowermost line below the diagonal are non-zero, it is also possible to compute the lower triangular portion of A^{-1} in such a way that the stages E1, I1, I2, I3, are applied to the matrix A^T (A^T denotes the transpose of A). The number of time steps remains the same but the number of processors is doubled in such a case.

The algorithm for computation of A^{-1} has the same drawbacks as the algorithm for solving $Ax=b$. Therefore it seems to be reasonable, when implementing these algorithms, to use a double word length. This will have a practical effect on stabilizing the algorithms and reducing the error bounds.

References

[1] A. H. Sameh and D. J. Kuck, *On stable parallel linear system solvers*, J. Assoc. Comput. Mach. 25.1 (1980), 81–91.

[2] A. H. Sameh and R. P. Brent, *Solving triangular system on a parallel computer*, SIAM J. Numer. Anal. 6 (1977), 1101–1113.

[3] S. C. Chen, D. J. Kuck and A. H. Sameh, *Practical parallel band triangular system solvers*, ACM Tran. on Math. Software 3 (1978), 270–277.

[4] S. Ch. Chen and D. J. Kuck, *Time and parallel processors bounds for linear recurrence systems*, IEEE Trans. Computers 24.7 (1975), 701–717.

[5] H. S. Stone, *An efficient parallel algorithm for the solution a tridiagonal linear system of equations*, J. Assoc. Comput. Mach. 20.1 (1973), 27–38.

[6] D. J. Evans and M. A. Hatzopoulos, *A parallel linear system solver*, Internat. J. Comput. Math. 7 (1979), 227–238.

[7] R. E. Bank and D. J. Rose, *An $O(n^2)$ method for solving constant coefficient boundary value problems in two dimensions*, SIAM J. Numer. Anal. 12.4 (1975), 529–540.

[8] B. L. Buzbee, *A capacitance matrix technique*; in: *Sparse matrix computations*, J. R. Bunch and D. J. Rose, Ed., Academic Press, New York 1976.

Presented to the Semester
Computational Mathematics
February 20 – May 30, 1980