INFORMATION AND COMPUTATION 120, 237–250 (1995)

On Parallel Computations with Banded Matrices*

VICTOR Y. PAN†

Department of Mathematics and Computer Science, Lehman College, City University of New York, New York, New York 10468
E-mail:vpan@lcvax.bitnet

AND

ISDOR SOBZE† AND ANTOINE ATINKPAHOUN†

Department of Computer Science, Graduate School and University Center, City University of New York, 33 West 42 Street, New York, New York 10036

We devise parallel algorithms for solving a banded linear system of equations and for computing the determinant of a banded matrix, substantially improving the previous record computational complexity estimates of [E]. Our algorithms are in NC or RNC and support new record bounds on the parallel time complexity of these computations and simultaneously the bounds on their *potential work* (the product of time and processor bounds), which match (within constant or logarithmic factors) the record sequential time bounds for the same computations. Moreover, these algorithms are in NC^1 or RNC^1 if the bandwidth is a constant.

1. INTRODUCTION

Solving banded linear systems of equations is among the most frequent operations in practice of scientific and engineering computations and has long been a subject of intensive research (see [GL] for a survey and further bibliography). A $k \times h$ matrix is called *banded* if k and h largely exceed its *bandwidth*, which is defined in terms of its *lower bandwidth* and its *upper bandwidth* as follows: A matrix $A = (a_{ij})$ has lower (respectively, upper) bandwidth $m_- = m_-(A)$ (respectively, $m_+ = m_+(A)$) if m_- (respectively, m_+) is the minimum nonnegative integer such that $a_{ij} = 0$ for $i > j + m_-$ (respectively, $j > i + m_+$). The sum $m = m_+ + m_- = m(A)$ is called the bandwidth of A. If $a_{i,j} \neq 0$ as long as $i = j + m_-(A)$ (respectively, $j = i + m_+(A)$), then the matrix A has a lower (respectively, upper) *edge*.

Our main focus in this paper is the computational problem denoted $LIN \cdot SOLVE = LS(n, m)$: Given a nonsingular $n \times n$ matrix A, with bandwidth m, and a vector $\mathbf{b}$ of dimension n, compute the solution $\mathbf{x} = A^{-1}\mathbf{b}$ to the linear system $A\mathbf{x} = \mathbf{b}$.

The nonsingularity assumption of $LIN \cdot SOLVE$ can be verified by solving any of the two following problems (also of independent interest):

$DET = D(n, m)$: Given an $n \times n$ matrix A, with bandwidth m, compute its determinant, $\det A$;

$|DET|^2 = |D(n, m)|^2$: For an $n \times n$ matrix A, defined in a field of characteristic 0 and having the bandwidth m, compute $|\det A|^2$.

Frequently, one has to solve several linear systems $A\mathbf{x} = \mathbf{b}(i)$ for the same nonsingular banded matrix A and for several vectors $\mathbf{b}(i)$. We will solve this problem in two stages:

1. $PREPROCESS = P(n, m)$: given a nonsingular $n \times n$ matrix A with bandwidth m, compute a set Γ of parameters, implicitly defining the inverse matrix A^{-1} and independent of the vector $\mathbf{b}(i)$ (various choices of the set Γ will be specified in Sections 3, 7, and 8 and Appendix B).

2. $BACK \cdot SOLVE = B(n, m)$: given a nonsingular $n \times n$ matrix A with bandwidth m, a vector $\mathbf{b}(i)$, and the set Γ of parameters output by $PREPROCESS$, $P(n, m)$, compute the vector $A^{-1}\mathbf{b}(i)$.

The current record asymptotic estimates on the sequential arithmetic time complexity of both $LS(n, m)$ and $D(n, m)$ are $\Omega(nm)$ and $O(nM(m)/m)$, where $M(m) = o(m^{2.376})$. Here $\Omega(nm)$ is the information lower bound, which follows since the solution depends on $\Omega(nm)$ input values, and $O(nM(m)/m)$ is an upper bound supported by the block Gaussian elimination algorithm, where the blocks are of the size $O(m) \times O(m)$. (In fact, one needs to apply some techniques for avoiding the inversion of some singular blocks in the block elimination; such techniques can be taken from this paper.)

* The results of this paper were presented at the 5th Annual ACM–SIAM Symposium on Discrete Algorithms, Arlington, VA, January 1994.

† Supported by NSF Grant CCR 9020690 and PSC CUNY Award 662478.

0890-5401/95 $12.00

In the present paper we do not try to attack the outstanding problem of eliminating the gap between the cited lower and upper bounds but attempt to reach parallel acceleration of the known sequential solutions of $LS(n, m)$ and $D(n, m)$, without exploding the processor bound. After the next prerequisits (stating related definitions and previous results), we will formalize our goal and our results. (In particular, because of the above objectives, our definitions of the work efficiency and work optimality below deviate slightly from those of [KR]; that is, in our definitions we refer to the current upper bounds T on the sequential time, in contrast to the reference to the current lower bounds in [KR]. Note, however, that the lower and upper bounds match (within a constant factor) and, consequently, our definitions coincide with those of [KR] in the important case where $m = O(1)$.) We will state all the complexity estimates in the form $(t_A, p_A) = O(t, p)$, which shows that a computational problem $A = A(n, m)$ can be solved in time $t = t(n, m)$ using $p = p(n, m)$ processors, both t and p estimated within constant factors, assuming the arithmetic EREW PRAM model of parallel computing [KR; PP; PP, a]. We will also assume a variant of Brent's scheduling principle [KR; PP; PP, a], hereafter referred to as the *B-principle* and expressed by the implication $O(t, sp)$ implies $O(st, p)$ for any $s \geqslant 1$.

Stating our results, we will use the known estimates (from [BP, P, CW, P91, P92, KP91, KP92]) for the parallel complexity of the computational problems $M(n, q, r)$, $I(q)$, and $D(q)$, denoting $n \times q$ by $q \times r$ matrix multiplication, $q \times q$ matrix inversion, and the evaluation of the determinant of a $q \times q$ matrix, respectively:

$$(t_{M(n,q,r)}, p_{M(n,q,r)}) = O(\log q, nqrh^{\omega-3}), \tag{1.1}$$

$$(t_{I(q)}, p_{I(q)}) = O(\phi_q(\mathbf{F}), p_{M(q,q,q)}), \tag{1.2}$$

$$(t_{D(q)}, p_{D(q)}) = O(\phi_q(\mathbf{F}), p_{M(q,q,q)}), \tag{1.3}$$

provided that $h = \min\{n, q, r\}$, $2 \leqslant \omega < 2.376$, and that the computation is performed over the field of constants $\mathbf{F}$; $\phi_q(\mathbf{F}) \leqslant \log^2 q$ if $\mathbf{F}$ has characteristic 0, $\phi_q(\mathbf{F}) \leqslant \log^4 q$ for any $\mathbf{F}$. The bounds (1.2), (1.3) have been obtained in [P91, P92, KP91, KP92], using Las Vegas randomized algorithms.

We also recall the concepts of algorithms in RNC^k and NC^k [that is, of randomized and deterministic algorithms (respectively) having computational cost bounded by $O((\log L)^k, L^d)$ for two constants k and d, independent of the input length (size) L], of the *potential work* of a parallel algorithm (that is, of the product $w = tp$ of the time and processor bounds), and of the *work efficiency* and *work optimality* of a parallel algorithm expressed by the equations $w = O((\log L)^c T)$, for a constant c, and $w = O(T)$, respectively, where T is the record sequential time bound for the solution of the same computational problem (see [PP, PP, a]). We may formalize our goal as the design of work optimum or, at least, work efficient algorithms for *LIN · SOLVE* and *DET* that are in NC^k or RNC^k for some constant k, which should be made as small as possible. The recent work of [E] was a major step towards this goal. This work has established the current record bounds on the randomized parallel complexity of *LIN · SOLVE* and *DET*:

$$(t_{D(n,m)}, p_{D(n,m)}) = O(T^*(n), P^*(n, m)),$$

$$(t_{LS(n,m)}, p_{LS(n,m)}) = O(T^*(n), P^*(n, m)),$$

$$T^*(n) = (\log^3 n)\, \psi(n)\, t_{I(n)},$$

$$P^*(n, m) = \left(\frac{n}{m}\right) p_{I(m)} \log^{O(1)} n,$$

$$\psi(n) = \begin{cases} 1, & \text{if } |\mathbf{F}| \geqslant n; \\ (1 + \log\log_{|\mathbf{F}|} n) \log_{|\mathbf{F}|} n, & \text{otherwise.} \end{cases}$$

In the present paper, we substantially improve these estimates of [E]. We develop two main approaches. In the first (presented in Sections 3–6), we reach the Las Vegas randomized bounds

$$O\left(\left(\log \frac{n}{m}\right) t_{I(m)}, \left(\frac{n}{m}\right) p_{I(m)}\right) \tag{1.4}$$

for *LIN · SOLVE* and

$$O\left(\left(\log \frac{n}{m}\right)(t_{I(m)} + t_{D(m)}), \frac{(n/m)(p_{D(m)} + p_{I(m)})}{\log(n/m)}\right) \tag{1.5}$$

for *DET* (in both cases over any field of constants). Equations (1.4) and (1.5) imply that our parallel algorithms for *LIN · SOLVE* (respectively, *DET*) are in RNC (and even in RNC^1 if m is a constant) and are *work efficient* (respectively, *optimum*) according to the cited definitions of [KR]: their potential work, $w_{\mathrm{LS}}(n, m) = O((n/m)(\log(n/m))\, t_{I(m)} p_{I(m)})$ (for *LIN · SOLVE*) and $W_{\mathrm{DET}}(n, m) = O((n/m)(t_{I(m)} + t_{D(m)})(p_{I(m)} + p_{D(m)}))$ (for *DET*) matches the record sequential time bounds of [E] for *DET* and *LIN · SOLVE*. Furthermore, a simple inspection shows that our algorithms can be implemented in sequential time $O((n/m)(\log(n/m))\, T_{I(m)})$ (respectively, $O((n/m)(T_{I(m)} + T_{D(m)}))$), where $T_{I(m)}$ (respectively, $T_{D(m)}$) denotes the sequential time required for computing the inverse (respectively, the determinant) of an $m \times m$ matrix, which substantially improves the previous record bound $O((n/m)\, T_{I(m)} \log^3 n)$ (respectively, $O((n/m)\, T_{D(m)} \log^3 n)$) of [E].

Moreover, if we solve more than an order of m linear systems $A\mathbf{x}(i) = \mathbf{b}(i)$ with the same $n \times n$ matrix A having bandwidth m, we obtain a further improvement: we solve *PREPROCESS* at the same randomized Las Vegas cost

(1.4), and we give a solution of *BACK · SOLVE* at deterministic cost

$$O\left((\log n)(\log m), \frac{mn}{(\log n)(\log m)}\right). \tag{1.6}$$

To obtain these improvements, we applied several techniques distinct from the ones of [E]. In particular, we introduced special preprocessing based on a 2×2 block factorization of the banded input matrix A and on utilizing some auxiliary matrices, such as $I_F(n, m)$ and $I_I(n, m)$, defined in Section 2, which helped us to reveal and to exploit the sparseness structure of the input matrix A and its diagonal blocks. In addition, we achieved the required nonsingularity of the auxiliary block matrices at a substantially lower computational cost than that would have been required by the techniques of [E], since the symmetrization and the technique of randomization of [Sc, Z] enable us to avoid the auxiliary computation of matrix ranks. Our techniques can also be combined with the 3×3 factorization of [E] to obtain the same asymptotic estimates for *LIN · SOLVE* and *DET* (see Appendix B and/or [PSA, S]), although the latter approach leads to a little more complicated code, and over arbitrary fields it requires to involve larger random matrices. Furthermore, since for any banded matrix A having bandwidth m, a family of $O(m)$-separators for a graph associated with A is readily available, we arrive at the bound $O((\log n)\, t_{I(m)}, p_{I(m)} n/(m \log n))$ for *LIN · SOLVE* over the fields of characteristic 0, by applying the parallel nested dissection algorithm of [GLi, GT] (also compare [PR85, PR93]). Yet another alternative derivation of the same asymptotic complexity estimate for *LIN · SOLVE* over the field of characteristic zero relies on the known block cyclic reduction algorithm (see Appendix A). Comparison of the two latter algorithms, that is, the nested dissection algorithm and the block cyclic reduction algorithm, reveals that they are very close to each other in the case of banded linear systems and can be made identical by means of an appropriate choice of a separator family for the nested dissection and of block sizes for the block cyclic reduction.

Under some mild additional assumptions, we obtain further results:

(a) We *deterministically* obtain (1.4) for *LIN · SOLVE* and (1.5) for $|DET|^2$ over the fields of characteristic zero in which the Hermitian transpose $A^* = A^{\mathrm{H}}$ of a matrix A is readily available. (In the real number field and in its subfields, $A^{\mathrm{H}} = A^{\mathrm{T}}$.) Hereafter we will assume the latter property (of the availability of A^{H}) by default, wherever we deal with the fields of characteristic zero, and we will use the notation A^{H} for the Hermitian transpose.

(b) The problem *LIN · SOLVE* for a block bidiagonal linear system (which includes the banded triangular case) can be solved, by means of a variant of the block cyclic reduction algorithm, at deterministic computational cost bounded (over any field of constants) by

$$O\left(t_{I(m)} + \left(\log \frac{n}{m}\right)(\log m), \frac{(n/m)\, p_{I(m)} t_{I(m)}}{t_{I(m)} + (\log(n/m))(\log m)}\right), \tag{1.7}$$

at the preprocessing stage, and by

$$O\left(\left(\log \frac{n}{m}\right)(\log m), \frac{mn}{(\log(n/m))(\log m)}\right), \tag{1.8}$$

at the subsequent backsolving stage.

(c) We extend the latter deterministic solution of the block bidiagonal linear systems to the deterministic solution, over any field of constants, for the problems *LIN · SOLVE**, *PREPROCESS**, and *BACK · SOLVE**, which denote *LIN · SOLVE*, *PREPROCESS*, and *BACK · SOLVE*, respectively, in the special case where the input matrix A has a lower and/or an upper edge. The latter requirement holds for a large class of banded matrices; in particular, it typically holds for the matrices encountered in applications to PDEs and ODEs (see [A, LP]). Our extension leads to deterministic solution algorithms that perform at a computational cost bounded by

$$O\left(\left(\log \frac{n}{m}\right)(\log m) + t_{I(m)}, \left(\frac{n}{m}\right) \frac{p_{I(m)} t_{I(m)}}{(\log(n/m))(\log m) + t_{I(m)}}\right) \tag{1.9}$$

for *PREPROCESS** and *LIN · SOLVE**, and by (1.8) for *BACK · SOLVE**. (Note the improvement against (1.4).)

In Section 9, we show an extension of (1.8) and (1.9) to the complexity estimates for the approximate solution of banded linear systems over the fields of characteristic 0 and for the exact solution over the rationals.

Let us summarize: By using various new techniques, we substantially improved the algorithms of [E]. In particular, in contrast to [E], we reached processor optimality and, if m is a constant, the RNC^1 time bound $O(\log n)$. Moreover, our algorithms are deterministic over the fields of characteristic 0, and we improved the solution of several linear systems with the same banded matrix, by using preprocessing algorithms.

We organize our presentation in the following order: After some preliminary results in Section 2, we preprocess a strongly nonsingular input matrix A in Section 3. We use the results of this preprocessing for the solution of *BACK · SOLVE* in Section 4 and *DET* in Section 5. In Section 6,

we relax the strong nonsingularity condition imposed on the input matrix A in Sections 3–5. In Section 7, we treat the block bidiagonal case, and we solve *PREPROCESS** and *BACK·SOLVE** in Section 8. In Section 9, we show a reduction of *LIN·SOLVE* to *LIN·SOLVE**. Sections 8 and 9 follow [P93a]. In Appendixes A and B, we present two alternative algorithms for *LIN·SOLVE* and *DET*. The algorithm of Appendix A is over the fields of characteristic 0 and is based on block cyclic reduction. The algorithm of Appendix B is based on a 3×3 representation and factorization of the input matrix.

Sections 7–9 can be read independently of Sections 2–6.

2. DEFINITIONS AND AUXILIARY RESULTS

Hereafter, 0 denotes the null matrices, I_q the $q\times q$ identity matrix, $I_F(k,q)$ $(k\geqslant q)$ the $k\times k$ matrix $\left(\begin{smallmatrix} I_q & 0\\ 0 & 0\end{smallmatrix}\right)$, $I_L(k,q)$ $(k\geqslant q)$ the $k\times k$ matrix $\left(\begin{smallmatrix} 0 & 0\\ 0 & I_q\end{smallmatrix}\right)$, $\mathrm{diag}(U,V)$ the matrix $\left(\begin{smallmatrix} U & 0\\ 0 & V\end{smallmatrix}\right)$, and $W^{\mathrm T}$ and $W^{\mathrm H}$ the transpose and the Hermitian transpose of a matrix W, respectively.

PROPOSITION 2.1. *Let q, r, s, r_1, s_1, r_2, s_2 be seven positive integers such that $r=r_1+r_2$, $s=s_1+s_2$, $q\leqslant r$, $q\leqslant s$; let B be a $q\times q$ matrix; and let $W=\left(\begin{smallmatrix} W_{11} & W_{12}\\ W_{21} & W_{22}\end{smallmatrix}\right)$ be an $r\times s$ matrix, where W_{11} is an $r_1\times s_1$ matrix. Then*

$$I_F(r,r_1)\,W=\begin{pmatrix} W_{11} & W_{12}\\ 0 & 0\end{pmatrix},$$

$$I_L(r,r_2)\,W=\begin{pmatrix} 0 & 0\\ W_{21} & W_{22}\end{pmatrix},$$

$$WI_F(s,s_1)=\begin{pmatrix} W_{11} & 0\\ W_{21} & 0\end{pmatrix},$$

$$WI_L(s,s_2)=\begin{pmatrix} 0 & W_{12}\\ 0 & W_{22}\end{pmatrix},$$

$$I_F(r,r_1)\,WI_F(s,s_1)=\begin{pmatrix} W_{11} & 0\\ 0 & 0\end{pmatrix},$$

$$I_F(r,r_1)\,WI_L(s,s_2)=\begin{pmatrix} 0 & W_{12}\\ 0 & 0\end{pmatrix},$$

$$I_L(r,r_2)\,WI_F(s,s_1)=\begin{pmatrix} 0 & 0\\ W_{21} & 0\end{pmatrix},$$

$$I_L(r,r_2)\,WI_L(s,s_2)=\begin{pmatrix} 0 & 0\\ 0 & W_{22}\end{pmatrix}.$$

Moreover, if the matrices $(B\ \ 0)$ and $(0\ \ B)$ [respectively, $\left(\begin{smallmatrix} B\\ 0\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0\\ B\end{smallmatrix}\right)$] have size $q\times r$ (respectively, $s\times q$), then

$$(B\ \ 0)=(B\ \ 0)\,I_F(r,q),\quad (0\ \ B)=(0\ \ B)\,I_L(r,q),$$

$$\begin{pmatrix} B\\ 0\end{pmatrix}=I_F(s,q)\begin{pmatrix} B\\ 0\end{pmatrix},\qquad \begin{pmatrix} 0\\ B\end{pmatrix}=I_L(s,q)\begin{pmatrix} 0\\ B\end{pmatrix}.$$

DEFINITION 2.1. A $k\times k$ submatrix of a matrix W formed by rows and columns $i_1,\ldots,i_k$ of W for any k-tuple $(i_1,\ldots,i_k)$ is called *principal*. A matrix is *strongly nonsingular* if all its principal submatrices are nonsingular.

PROPOSITION 2.2 [GL, p. 140]. *Over the field of complex numbers, as well as over any of its subfields, and, consequently, over any field of characteristic 0 in which the Hermitian transpose $A^{\mathrm H}$ of a matrix A is readily available, $W^{\mathrm H}W$ and $(W^{\mathrm H}W)^{-1}$ are strongly nonsingular matrices for any nonsingular matrix W.*

PROPOSITION 2.3. *For any $k\times k$ matrices W and V, we have $m(WV)\leqslant m(W)+m(V)$, $m(W^{\mathrm H})=m(W)$, so that $m(W^{\mathrm H}W)\leqslant 2m(W)$. Furthermore, $m(W_k)\leqslant m(W)$ for any principal submatrix W_k of W.*

PROPOSITION 2.4. *A $km\times km$ matrix A, with $m_-(A)<m$, $m_+(A)<m$, can be represented as a $k\times k$ block tridiagonal matrix with $m\times m$ blocks. For any $k\times k$ block tridiagonal matrix A, with $m\times m$ blocks, we have $m_-(A)\leqslant 2m-2$, $m_+(A)\leqslant 2m-2$.*

3. PREPROCESSING FOR THE SOLUTION OF A BANDED LINEAR SYSTEM (USING A 2×2 BLOCK DECOMPOSITION OF THE INPUT MATRIX)

Hereafter, A denotes an $n\times n$ banded matrix having bandwidth m. We will call the blocks in the 2×2 block representation

$$A=\begin{pmatrix} A_{11} & A_{12}\\ A_{21} & A_{22}\end{pmatrix} \tag{3.1}$$

balanced if the matrices A_{11} and A_{22} have sizes $n_1\times n_1$ and $n_2\times n_2$, respectively, with $n_1=\lceil n/2\rceil$, $n_2=n-n_1$. Up to Section 6, we assume that A is strongly nonsingular, so that A and A^{-1} have the factorizations

$$A=\begin{pmatrix} I_{n_1} & 0\\ A_{21}A_{11}^{-1} & I_{n_2}\end{pmatrix}\begin{pmatrix} A_{11} & 0\\ 0 & S\end{pmatrix}\begin{pmatrix} I_{n_1} & A_{11}^{-1}A_{12}\\ 0 & I_{n_2}\end{pmatrix}, \tag{3.2}$$

$$A^{-1}=\begin{pmatrix} I_{n_1} & -A_{11}^{-1}A_{12}\\ 0 & I_{n_2}\end{pmatrix}\begin{pmatrix} A_{11}^{-1} & 0\\ 0 & S^{-1}\end{pmatrix}\begin{pmatrix} I_{n_1} & 0\\ -A_{21}A_{11}^{-1} & I_{n_2}\end{pmatrix}, \tag{3.3}$$

$$S=A_{22}Y, \tag{3.4}$$

$$Y=I_{n_2}-A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}, \tag{3.5}$$

where S is called the Schur complement of A_{11} in A, and S^{-1} is a block of A^{-1}.

Note that nonsingularity of Y follows from nonsingularity of the matrices A, A_{11} and A_{22}.

PROPOSITION 3.1. *If the first m and the last m columns of the matrices A_{11}^{-1} and A_{22}^{-1} are known, then the matrix Y can be computed at a cost bounded by*

$$O(t_{M(n/2,m,m)}, p_{M(n/2,m,m)}), \tag{3.6}$$

and the matrix Y^{-1} can be computed at a cost bounded by

$$O\left(t_{I(m)} + t_{M(n/2,m,m)}, \frac{t_{I(m)}p_{I(m)} + p_{M(n/2,m,m)}t_{M(n/2,m,m)}}{t_{I(m)} + t_{M(n/2,m,m)}}\right). \tag{3.7}$$

Proof. Observe that the matrix A_{12} has the form $\left(\begin{smallmatrix} 0 & 0 \\ L & 0 \end{smallmatrix}\right)$ and A_{21} has the form $\left(\begin{smallmatrix} 0 & U \\ 0 & 0 \end{smallmatrix}\right)$, where L and U are $m \times m$ matrices, and apply Proposition 2.1 to deduce that

$$\begin{aligned} A_{12} &= I_L(n_1, m)\, A_{12} = A_{12} I_F(n_2, m) \\ &= I_L(n_1, m)\, A_{12} I_F(n_2, m), \end{aligned} \tag{3.8}$$

$$\begin{aligned} A_{21} &= I_F(n_2, m)\, A_{21} = A_{21} I_L(n_1, m) \\ &= I_F(n_2, m)\, A_{21} I_L(n_1, m). \end{aligned} \tag{3.9}$$

Then combine (3.8), (3.9), and (3.5) to deduce that

$$\begin{aligned} Y = I_{n_2} &- A_{22}^{-1} I_F(n_2, m)\, A_{21} L_F(n_1, m) \\ &\times A_{11}^{-1} I_L(n_1, m)\, A_{12} I_F(n_2, m). \end{aligned} \tag{3.10}$$

The latter equation implies the bound (3.6) on the cost of computing Y. It also implies that Y is a block triangular matrix of the following format where Y_{11} is an $m \times m$ matrix:

$$Y = \begin{pmatrix} Y_{11} & 0 \\ Y_{21} & I_{n_2 - m} \end{pmatrix}. \tag{3.11}$$

Since Y is nonsingular, it follows that the block Y_{11} is nonsingular, and from (3.11), we obtain that

$$Y^{-1} = \begin{pmatrix} Y_{11}^{-1} & 0 \\ -Y_{21}Y_{11}^{-1} & I_{n_2 - m} \end{pmatrix}. \tag{3.12}$$

This immediately implies the bound (3.7) on the cost of the computation of Y^{-1}. (Note that only the first m columns of Y and of Y^{-1} need be computed.) ∎

DEFINITION 3.1. Let A be a strongly nonsingular $n \times n$ matrix and recursively define

$$A_{i_1 i_1 \cdots i_k i_k} = \begin{cases} A, & k = 0, \\ (A_{i_1 i_1 \cdots i_{k-1} i_{k-1}})_{i_k i_k} & k > 0, \end{cases} \tag{3.13}$$

where $i_k = 1, 2$, and $C_{i_k i_k}$ denotes the two diagonal blocks [that is, the blocks (1.1) and (2.2)] in the balanced 2×2 representation of the matrix C. Then recursively define the set of matrices

$$\Gamma_2(A) = \begin{cases} A^{-1} & \text{if } n < 2m, \\ \gamma_2(A) \cup \Gamma_2(A_{11}) \cup \Gamma_2(A_{22}) & \text{otherwise;} \end{cases} \tag{3.14}$$

$$\gamma_2(A) = \{A^{-1} I_F(n, m),\ A^{-1} I_L(n, m),\ Y^{-1}\}.$$

Define *preprocessing*-2 of matrix A [based on the 2×2 block decompositions (3.2)–(3.5)] as the computation of the set $\Gamma_2(A)$. (In Section 4, we will prove that $\Gamma_2(A)$ is a solution to the *PREPROCESS* problem for the matrix A.)

THEOREM 3.1. *The complexity of preprocessing-2 is bounded by* (1.4).

Proof. Expand (3.3) and then apply (3.8), (3.9), and Proposition 2.1 to deduce that

$$\begin{aligned} &A^{-1} I_F(n, m) \\ &= \begin{pmatrix} A_{11}^{-1} I_F(n_1, m) + A_{11}^{-1} A_{12} Y^{-1} A_{22}^{-1} A_{21} A_{11}^{-1} I_F(n_1, m) & 0 \\ -Y^{-1} A_{22}^{-1} A_{21} A_{11}^{-1} I_F(n_1, m) & 0 \end{pmatrix} \\ &= \begin{pmatrix} A_{11}^{-1} I_F(n_1, m) + A_{11}^{-1} I_L(n_1, m)\, A_{12} Y^{-1} & \\ \quad \times A_{22}^{-1} I_F(n_2, m)\, A_{21} A_{11}^{-1} I_F(n_1, m) & 0 \\ -Y^{-1} A_{22}^{-1} I_F(n_2, m)\, A_{21} A_{11}^{-1} I_F(n_1, m) & 0 \end{pmatrix}; \end{aligned} \tag{3.15}$$

$$\begin{aligned} &A^{-1} I_L(n, m) \\ &= \begin{pmatrix} 0 & -A_{11}^{-1} A_{12} Y^{-1} A_{22}^{-1} I_L(n_2, m) \\ 0 & Y^{-1} A_{22}^{-1} I_L(n_2, m) \end{pmatrix} \\ &= \begin{pmatrix} 0 & -A_{11}^{-1} I_L(n_1, m)\, A_{12} Y^{-1} A_{22}^{-1} I_L(n_2, m) \\ 0 & Y^{-1} A_{22}^{-1} I_L(n_2, m) \end{pmatrix}. \end{aligned} \tag{3.16}$$

Let $t_1(n, m)$ denote the number of parallel steps and let $p_1(n, m)$ denote the number of processors needed for the computation of $\Gamma_2(A)$. Combining the recursive definition (3.14) and the recursive formulae (3.15) and (3.16) with Proposition 3.1 leads to the following recursive estimates, where c_1 and c_2 are two constants:

$$\begin{aligned} &t_1(n, m) \\ &\quad \leqslant \begin{cases} t_{I(m)}, & n \leqslant 2n \\ t_1(n/2, m) + c_1 t_{I(m)} + c_2 t_{M(n/2,m,m)}, & \text{otherwise,} \end{cases} \\ &p_1(n, m) \\ &\quad \leqslant \begin{cases} p_{I(m)}, & n \leqslant 2m \\ \max(2p_1(n/2, m), p_{I(m)}, p_{M(n/2,m,m)}), & \text{otherwise.} \end{cases} \end{aligned}$$

Recursive application of the latter relations and the B-principle yields the desired complexity bound (1.4). ∎

Remark 3.1. At each parallel step, the computation of each triple in the set $\Gamma_2(A)$ only involves the results

obtained at the previous recursive step. Therefore, we may overwrite these previous results by the current ones, to keep the storage space complexity for the entire preprocessing computation bounded by $O(mn)$.

4. SOLVING A PREPROCESSED BANDED LINEAR SYSTEM

THEOREM 4.1. *If the set $\Gamma_2(A)$ output by preprocessing-2 is available, then the complexity of the subsequent computation of $A^{-1}\mathbf{b}$, for a given vector $\mathbf{b}$, is bounded by* (1.6).

Proof. Let $\mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}$, where $\mathbf{b}_1$ and $\mathbf{b}_2$ are two vectors of dimensions n_1 and n_2, respectively. Expand (3.3), then multiply both sides by $\mathbf{b}$, and apply (3.8) and (3.9) to deduce that

$$A^{-1}\mathbf{b} = \begin{pmatrix} A_{11}^{-1}\mathbf{b}_1 + A_{11}^{-1}A_{12}Y^{-1}A_{22}^{-1}A_{21}A_{11}^{-1}\mathbf{b}_1 - A_{11}^{-1}A_{12}Y^{-1}A_{22}^{-1}\mathbf{b}_2 \\ -Y^{-1}A_{22}^{-1}A_{21}A_{11}^{-1}\mathbf{b}_1 + Y^{-1}A_{22}^{-1}\mathbf{b}_2 \end{pmatrix}$$

$$= \begin{pmatrix} A_{11}^{-1}\mathbf{b}_1 + A_{11}^{-1}I_L(n_1, m)\, A_{12}Y^{-1}A_{22}^{-1}I_F(n_2, m) \\ \times A_{21}A_{11}^{-1}\mathbf{b}_1 - A_{11}^{-1}I_L(n_1, m)\, A_{12}Y^{-1}A_{22}^{-1}\mathbf{b}_2 \\ -Y^{-1}A_{22}^{-1}I_F(n_2, m)\, A_{21}A_{11}^{-1}\mathbf{b}_1 + Y^{-1}A_{22}^{-1}\mathbf{b}_2 \end{pmatrix}.$$

Due to preprocessing, the matrices Y^{-1}, $A_{11}^{-1}I_L(n_1, m)$ and $A_{22}^{-1}I_F(n_2, m)$ are available. Therefore, given the vectors $A_{11}^{-1}\mathbf{b}_1$ and $A_{22}^{-1}\mathbf{b}_2$, the computation of $A^{-1}\mathbf{b}$ only requires a constant number of multiplications of matrices of sizes at most $m \times q$ by vectors of dimensions at most q, where $q \leqslant \lceil n/2 \rceil$.

Let $t_2 = t_2(n, m)$ denote the number of parallel steps required for the computation of $A^{-1}\mathbf{b}$, assuming that preprocessing-2 has been performed, and let $p_2 = p_2(n, m)$ denote the corresponding number of required processors. The above argument leads us to the following estimates for the pair (t_2, p_2), where c is a constant:

$$t_2(n, m) \leqslant \begin{cases} t_{M(m, m, 1)}, & n \leqslant 2m \\ t_2(n/2, m) + ct_{M(n/2, m, 1)}, & \text{otherwise,} \end{cases}$$

$$p_2(n, m) \leqslant \begin{cases} p_{M(m, m, 1)}, & n \leqslant 2m \\ \max(2p_2(n/2, m), p_{M(n/2, m, 1)}), & \text{otherwise.} \end{cases}$$

Recursive application of the latter relations and the B-principle yields the desired complexity bound (1.6). ∎

Remark 4.1. The problem $BACK \cdot SOLVE$, for s linear systems, can be defined by replacing the vectors $\mathbf{x}$ and $\mathbf{b}$ by $n \times s$ matrices. Due to Remark 3.1, the storage space complexity of $LIN \cdot SOLVE$ is thus bounded by $S = S_p + O(nm + ns)$, where S_p bounds the space required for the storage of all the values output at the preprocessing stage, and actually S_p is the dominating term of S.

5. SOLVING *DET*

THEOREM 5.1. *The complexity of DET is bounded by* (1.5).

Proof. Deduce from (3.2) and (3.4) that

$$\det A = \det A_{11} \det A_{22} \det Y. \tag{5.1}$$

Moreover, each of the matrices A_{11} and A_{22} has bandwidth at most m. Therefore, solving DET is reduced to two problems of half-size each, at the cost of computing the determinant of the $m \times m$ matrix Y; moreover, due to (3.11), $\det Y = \det Y_{11}$, and

$$I_F(n_2, m)\, YI_F(n_2, m) = \operatorname{diag}(Y_{11}, 0). \tag{5.2}$$

We may now define the set

$$\Gamma_D(A) = \begin{cases} \{\det A\} & \text{if } n < 2m, \\ \gamma_D(A) \cup \Gamma_D(A_{11}) \cup \Gamma_D(A_{22}) & \text{otherwise,} \end{cases}$$
$$\gamma_D(A) = \{\det A,\ Y^{-1},\ I_U(n, m)\, A^{-1} I_V(n, m))\}, \tag{5.3}$$

where the pair (U, V) takes on all the choices of the pairs (F, F), (F, L), (L, F), and (L, L).

Let $t_3(n, m)$ denote the number of parallel steps and $p_3(n, m)$ the number of processors needed for solving DET. By combining the equations (3.10), (3.15), (3.16), and (5.2), we arrive at the following complexity estimates for the pair $(t_3(n, m), p_3(n, m))$, where $n \geqslant 2m$:

$$t_3(n, m) \leqslant t_3(n/2, m) + t_{D(m)} + t_{I(m)} + t_{M(m, m, m)},$$
$$p_3(n, m) \leqslant \max(2p_3(n/2, m), p_{D(m)}, p_{I(m)}, p_{M(m, m, m)}).$$

Due to (1.1)–(1.3), we may ignore the terms $t_{M(m, m, m)}$ and $p_{M(m, m, m)}$. Recursive application of the latter estimates and the B-principle yields the complexity bound (1.5). ∎

6. HOW TO RELAX THE STRONG NONSINGULARITY ASSUMPTION

Shifting from A and DET to $A^{\mathrm{H}}A$ and $|DET|^2$ enables us to relax the assumption about strong nonsingularity of A, for computations over any field of characteristic zero in which the Hermitian transpose of a matrix is readily available. This follows from Propositions 2.2, 2.3 and the equations $A^{-1} = (A^{\mathrm{H}}A)^{-1}A^{\mathrm{H}}$, $\det(A^{\mathrm{H}}A) = |\det A|^2$. More

precisely, our algorithm of Section 5 (for *DET*) computes, as a by-product, the determinants of all the matrices $A_{i_1 i_1 \cdots i_k i_k}$, $Y_{i_1 i_1 \cdots i_k i_k}$ [see Definition 3.1 and Eq. (3.13)], unless at least one of these determinants is zero. In the latter case, we just output $\det A = 0$ [this equation follows, due to (5.1) and to its recursive extension]. Otherwise, if $\det A \neq 0$, all the matrices $A_{i_1 i_1 \cdots i_k i_k}$, $Y_{i_1 i_1 \cdots i_k i_k}$ are nonsingular, and our algorithms of Sections 3 and 4 solve *PREPROCESS* and *BACK · SOLVE*.

To relax the strong nonsingularity assumption in the case of *any* field of constants, we apply an alternative argument based on randomization. Again, we only need to ensure nonsingularity of the matrices A, Y, A_{11}, Y_{11}, A_{22}, Y_{22}, ... Furthermore, since the nonsingularity of $Y_{i_1 i_1 \cdots i_k i_k}$ follows from the nonsingularity of $A_{i_1 i_1 \cdots i_k i_k}$, $A_{i_1 i_1 \cdots i_k i_k 11}$ and $A_{i_1 i_1 \cdots i_k i_k 22}$, it is sufficient to ensure nonsingularity of $A_{i_1 i_1 \cdots i_k i_k 11}$ and $A_{i_1 i_1 \cdots i_k i_k 22}$, assuming that $A_{i_1 i_1 \cdots i_k i_k}$ is nonsingular.

We will next apply randomization to yield nonsingularity of the two diagonal blocks A_{11} and A_{22}, in the balanced 2×2 representation of A (this argument can be extended to all other diagonal blocks as well). More precisely, we will reach our goal (of insuring nonsingularity of the two diagonal blocks) by shifting from the matrix A to PA, where $P = \operatorname{diag}(I_{n_1 - m_+(A)}, R, I_{n_2 - m_-(A)})$, and R is a random $m \times m$ matrix. This is a valid transition because $\det P$ is readily available, $\det A = \det(PA)/\det P$, and $A^{-1} = (PA)^{-1} P$. Moreover, we immediately verify by inspection that the equations (3.2)–(3.5), (3.8), (3.9) (3.15), (3.16) and, consequently, the proofs of Proposition 3.1 and Theorem 3.1 are extended in the transition from the matrix A to the matrix PA, as long as the matrix PA is nonsingular, together with $(PA)_{11}$, $(PA)_{22}$, the two principal submatrices of PA, which are the two diagonal blocks of PA in its balanced 2×2 block representation.

We will now recursively apply this process of regularization via randomization: first to the two diagonal blocks $(PA)_{11}$ and $(PA)_{22}$, thus obtaining the matrices $B^{(i,j)} = P^{(i,j)}(PA)_{jj}$, $i, j = 1, 2$, then to the two diagonal blocks of $B_{11}^{(i,j)}$ and $B_{22}^{(i,j)}$ of each matrix $B^{(i,j)}$, and so on, until we arrive at the blocks of size less than $3m \times 3m$. Theorem 6.1 will show that in every recursive step it suffices to shift from the $k \times k$ input matrix B^* to the matrix $P^* B^*$ where P^* has the format

$$\operatorname{diag}(I_i, R^*, I_{k-i-m}), \tag{6.1}$$

provided that $k \geqslant 3m$, $i = \lceil k/2 \rceil$, and R^* is an $m \times m$ random matrix. Due to the middle position of the blocks R^* and to the bound $k \geqslant 3m$, the entire regularization process amounts to the transition from the input matrix A to the matrix $\tilde{P}A$, $\tilde{P} = \operatorname{diag}(D_1, D_2, ..., D_h)$, where each D_i is either the identity matrix or an $m \times m$ random matrix, and all the diagonal blocks in the balanced recursive factorization of $\tilde{P}A$ are nonsingular. It follows that $m(\tilde{P}A) \leqslant 3m$, and we may apply our solutions and our complexity estimates to *LIN · SOLVE* and *DET* with the input matrix $\tilde{P}A$. Clearly, the solutions for $\tilde{P}A$ are immediately extended to the ones for A. It remains to substantiate the claim that using the matrices of the format (6.1) as multipliers suffices to ensure the desired regularization of the diagonal blocks. This will follow from the next result.

THEOREM 6.1. *Let $A = W + \operatorname{diag}(U, 0, V)$ where U and V are $m \times m$ matrices and W is a nonsingular banded matrix of size $n \times n$, with bandwidth $m = m(W)$. Let $n_1 = \lfloor n/2 \rfloor$ and $n_2 = n - n_1$. Then there exists a permutation matrix R of size $m \times m$, such that the matrix $B = PA = \operatorname{diag}(I_{n_1 - m_+(W)}, R, I_{n_2 - m_-(W)})\, A$ has the balanced 2×2 block representation of the format* (3.1), *where the matrix B_{11} is nonsingular.*

Proof. Our proof follows [S] and was inspired by an argument of [E] applied in the existence proof for a permutation matrix with similar properties, but for a different, 3×3, block representation of the input matrix. Let A_L denote the $n \times n_1$ matrix $\binom{A_{11}}{A_{21}}$. Without loss of generality, we will assume that $m_+(A) = m_-(A)$ and set $m_+(A) = m_-(A) = k$. Since A is nonsingular, A_L has full (column) rank n_1. The top $n_1 - k$ rows of A_L are linearly independent, since the top $n_1 - k$ rows of A are linearly independent, and since these rows have no nonzero entries outside A_L. The top $n_1 + k$ rows of A_L have full rank, n_1, as well, because A_L has this rank and has no nonzero entries below these rows. It follows that there exists a set of n_1 rows of A_L that includes the first $n_1 - k$ top rows, as well as k of the next $2k = m$ rows, that form a nonsingular $n_1 \times n_1$ matrix. Consequently, there exists a permutation matrix R of size $2k \times 2k$ such that the leading principal $n_1 \times n_1$ submatrix of the matrix $B = \operatorname{diag}(I_{n_1 - k}, R, I_{n_2 - k})\, A$ is nonsingular. ∎

Similarly, nonsingularity of B_{22} (for a certain assignment of the entries of R) follows. Therefore, the matrices B_{11} and B_{22} are nonsingular for a generic choice of the matrix R, that is, for the matrix R filled with indeterminates. By the standard argument of [Sc, Z], the nonsingularity of B_{11} and of B_{22} then follows (with a high probability) for a random assignment of the values (from a fixed large set) to the entries of R. Similar results apply to subsequent stages of the recursive regularization of the diagonal blocks. The overall number of random parameters used in order to achieve the entire recursive factorization is $O(m^2(1 + 2 + 2^2 + \cdots + 2^{\log(n/m)})) = O(nm)$.

7. BLOCK BIDIAGONAL LINEAR SYSTEMS

In this section, A denotes a nonsingular $k \times k$ block matrix of the following format, where $k \geqslant 2$, A_i $(0 \leqslant i \leqslant k-1)$ and B_j $(1 \leqslant j \leqslant k-1)$ are $m \times m$ blocks:

$$A=\begin{pmatrix} A_0 & B_1 & 0 & \cdots & & 0 \\ 0 & A_1 & B_2 & \cdots & & 0 \\ 0 & 0 & A_2 & \cdots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \\ 0 & 0 & \cdots & & A_{k-2} & B_{k-1} \\ 0 & 0 & \cdots & & 0 & A_{k-1} \end{pmatrix}. \tag{7.1}$$

Assume that the matrix A of (7.1) is nonsingular and observe that $\det A=\prod_{i=0}^{k-1}\det A_i$, so that all the diagonal blocks A_i of A are nonsingular too.

Let us hereafter assume (with no loss of generality) that $k=2^r$, for an integer r, and let us write

$$\begin{aligned} D_{k,m}&=\operatorname{diag}(A_0, A_1, \ldots, A_{k-1}),\\ V_i&=A_{i-1}^{-1}B_i,\ i=1,\ldots,k-1, \end{aligned} \tag{7.2}$$

$$V=D_{k,m}^{-1}A=\begin{pmatrix} I & V_1 & 0 & \cdots & & 0 \\ 0 & I & V_2 & \cdots & & 0 \\ 0 & 0 & I & V_3 & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \\ 0 & 0 & \cdots & & I & V_{k-1} \\ 0 & 0 & \cdots & & 0 & I \end{pmatrix} \tag{7.3}$$

and define the *preprocessing-bd* of a matrix A of (7.1) as the computation of the matrices $V_i^{(0)}=V_i$ of (7.2), for $i=1,\ldots,k-1$, complemented by the computation of the matrices

$$V_{i+1}^{(s+1)}=V_{2i+1}^{(s)}V_{2i+2}^{(s)},\quad \text{for } i=0,1,\ldots,2^{r-s-1}-1,$$

with

$$V_{2^{r-s}}^{(s)}=0,\quad \text{for } s=0,1,\ldots,r-h.$$

PROPOSITION 7.1. *Preprocessing-bd can be performed at deterministic computational cost bounded by*

$$O\left(t_{I(m)}+(\log k)(\log m),\ \frac{kp_{I(m)}t_{I(m)}}{t_{I(m)}+(\log k)(\log m)}\right).$$

Next, we will estimate the cost of solving a linear system $A\mathbf{x}=\mathbf{b}$ with the matrix A of (7.1), provided that the preprocessing-bd of A has been performed.

PROPOSITION 7.2. *Let A be a nonsingular matrix of the format* (7.1). *Suppose that the preprocessing-bd of A has been performed. Then the deterministic computational complexity of solving the linear system $A\mathbf{x}=\mathbf{b}$ is bounded by*

$$O\left((\log k)(\log m),\ \frac{km^2}{(\log k)(\log m)}\right). \tag{7.4}$$

Proof. Our proof is based on a variant of the block cyclic reduction algorithm. Since we have preprocessing-bd done, we now shift to the solution of the equivalent linear system

$$V\mathbf{x}=\mathbf{c}, \tag{7.5}$$

where $\mathbf{c}=D_{k,m}^{-1}\mathbf{b}$, and $D_{k,m}$ and V are the matrices of (7.2), (7.3). [Clearly the computational cost of multiplication of the matrix $D_{k,m}^{-1}$ by the vector $\mathbf{b}$ is within the bound (7.4).] We will keep assuming (with no loss of generality) that $k=2^r$. Let S_r denote the linear system (7.5), represented by the following 2^r equations where each of the vectors $\mathbf{x}^{(h)}$ and $\mathbf{c}^{(h)}$ has dimension m:

$$\mathbf{x}^{(h)}+V_{h+1}\mathbf{x}^{(h+1)}=\mathbf{c}^{(h)},\quad 0\leqslant h\leqslant 2^r-1. \tag{7.6}$$

If we multiply the equation (7.6), for $h=2i+1$, by the block $(-V_{2i+1})$ $(0\leqslant i\leqslant 2^{r-1})$ and add the resulting equation to the equation (7.6), for $h=2i$, then we will arrive at the following linear system, to be denoted S_{r-1} and consisting of 2^{r-1} equations:

$$\begin{aligned} &\mathbf{x}^{(2i)}-V_{2i+1}V_{2i+2}\mathbf{x}^{(2i+2)}\\ &\quad=\mathbf{c}^{(2i)}-V_{2i+1}\mathbf{c}^{(2i+1)},\quad 0\leqslant i\leqslant\frac{k}{2}-1,\ V_k=0. \end{aligned} \tag{7.7}$$

As soon as we solve this linear system, we may immediately obtain the values of the vectors $\mathbf{x}^{(1)}$, $\mathbf{x}^{(3)}$, $\mathbf{x}^{(5)}$, ..., $\mathbf{x}^{(k-1)}$ satisfying the vector equation (7.6). Indeed, we just need to substitute the known values of the vectors $\mathbf{x}^{(2i)}$ into (7.6) and concurrently perform at first $k/2$ multiplications of $m\times m$ matrices V_{2i+1} by vectors $\mathbf{x}^{(2i+1)}$ and then $k/2$ subtractions of the resulting vectors from $\mathbf{b}^{(2i+1)}$, for $i=0,\ldots,(k/2)-1$. These steps are performed at the cost $O(\log m, km^2/\log m)$. Due to this reduction of the input size, we have the following recurrence relations for the deterministic parallel complexity (t_k, p_k) of solving the system (7.6):

$$\begin{aligned} t_k&\leqslant t_{k/2}+O(\log m),\\ p_k&\leqslant p_{k/2}+O(km^2/\log m). \end{aligned}$$

Recursive application of the latter bounds using the B-principle yields the bound (7.4). ∎

Now repeat the proof of Proposition 7.2, with the vector $\mathbf{b}$ replaced by an $n\times m$ matrix; then apply Proposition 7.1 and obtain

PROPOSITION 7.3. *For a nonsingular block bidiagonal matrix A of the format* (7.1), *the first m rows of its inverse A^{-1} can be computed at deterministic cost bounded by*

$$O(t^*, kp_{I(m)}/t^*),\quad t^*=t_{I(m)}+(\log k)(\log m). \tag{7.8}$$

PROPOSITION 7.4. *The estimates of Propositions* 7.1–7.3, *for* $k = n/m$, *apply to the solution of a nonsingular linear system* $A\mathbf{x} = \mathbf{b}$ *with an* $n \times n$ *triangular matrix* A *having its bandwidth equal to* $m(A) = m$.

8. SOLUTION OF *LIN · SOLVE** VIA THE REDUCTION TO THE BLOCK BIDIAGONAL FORM

Assume that A, the $n \times n$ input matrix to the problem *LIN · SOLVE**, has a lower edge. Let $q = m\_(A) \leqslant m = m(A)$, define the $(n+q) \times (n+q)$ matrix $B = \begin{pmatrix} V & A \\ 0 & W \end{pmatrix}$, where $V = \begin{pmatrix} I_q \\ 0 \end{pmatrix}$, $W = (0 \;\; I_q)$, and consider the auxiliary linear system

$$B\begin{pmatrix} \mathbf{0} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{z} \end{pmatrix}, \tag{8.1}$$

where the vector $\mathbf{z}$ will be chosen later, to make the system (8.1) equivalent to the original linear system $A\mathbf{x} = \mathbf{b}$. We observe that the matrix B is nonsingular (since the matrix A has a lower edge) and has bandwidth $m(B) \leqslant m = m(A)$. Let $B^{-1} = \begin{pmatrix} G & H \\ K & L \end{pmatrix}$, where H is a $q \times q$ block. From the nonsingularity of A and the factorization

$$\begin{aligned} B &= \begin{pmatrix} V & A \\ 0 & W \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ WA^{-1} & I \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & -WA^{-1}V \end{pmatrix} \begin{pmatrix} A^{-1}V & I \\ I & 0 \end{pmatrix}, \end{aligned} \tag{8.2}$$

we deduce that $H = -(WA^{-1}V)^{-1}$, which implies nonsingularity of H.

LEMMA 8.1. *Set*

$$\mathbf{z} = -H^{-1}G\mathbf{b}. \tag{8.3}$$

Then the two linear systems $A\mathbf{x} = \mathbf{b}$ *and* (8.1) *are equivalent to each other.*

Proof. The nonsingular system $A\mathbf{x} = \mathbf{b}$ is the subsystem of (8.1) formed by the first n equations of (8.1). Substitution of $\mathbf{x} = A^{-1}\mathbf{b}$ into (8.1) defines the unique vector $\mathbf{z}$ satisfying (8.1). On the other hand, the vector $\mathbf{z}$ must satisfy the first q linear equations of the system $B^{-1}\begin{pmatrix} \mathbf{b} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} G & H \\ K & L \end{pmatrix}\begin{pmatrix} \mathbf{b} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \end{pmatrix}$; that is, this vector must satisfy the q equations $G\mathbf{b} + H\mathbf{z} = \mathbf{0}$. This defines a unique vector $\mathbf{z}$ of (8.3). ∎

We have reduced the original linear system $A\mathbf{x} = \mathbf{b}$ to the banded triangular linear system (8.1), by performing the following stages:

1. (*PREPROCESS · REDUCE*): First compute the first q rows of the matrix B^{-1}, that is, compute the matrix $I_F(n+q, q)\, B^{-1} = (G \;\; H)$ (compare Proposition 2.1); then compute the matrix H^{-1}.

2. (*BACK · REDUCE*): Compute the vector $-H^{-1}G\mathbf{b}$.

We call the computation at stage 1 preprocessing because it does not involve the vector $\mathbf{b}$. The computational cost of performing stages 1 and 2 is bounded by $O(t^*, (n/q)\, P_{l(q)}/t^*)$, $t^* = t_{l(q)} + (\log(n/q))(\log q)$, for stage 1 (due to Proposition 7.2), and by $O(\log n, nq/\log n)$, for stage 2. The next theorem summarizes the computational costs of both the reduction of the original system to the triangular linear system and the subsequent solution of the latter system.

THEOREM 8.1. *A nonsingular* $n \times n$ *matrix* A, *having bandwidth* $m = m(A)$ *and having at least one edge, can be preprocessed at a computational cost bounded by* (1.9) (*by applying Proposition* 7.2 *and the reduction algorithm of this section*). *After that, for any fixed vector* $\mathbf{b}$ *of dimension* n, *the linear system* $A\mathbf{x} = \mathbf{b}$ *can be solved at a computational cost bounded by* (1.8).

Remark 8.1. The estimates of Theorem 8.1 can be extended immediately to the case where the entries of A are block (matrices); in particular, to the case of a nonsingular block tridiagonal input matrix A all of whose subdiagonal and/or superdiagonal blocks are nonsingular (compare Proposition 2.4).

9. EXTENSIONS FROM *LIN · SOLVE** TO *LIN · SOLVE*

In this section rounding to the nearest integer will be assumed as a unit cost operation. For the extension from *LIN · SOLVE** to *LIN · SOLVE* we may apply various modifications of the techniques of the variable artificial diagonal of [P85, P87]. Consider a $k \times k$ block tridiagonal matrix with $m \times m$ blocks,

$$A = \begin{pmatrix} A_0 & B_0 & 0 & \cdots & & 0 \\ C_1 & A_1 & B_1 & \cdots & & 0 \\ 0 & C_2 & A_2 & B_2 & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \\ 0 & 0 & \cdots & & A_{k-2} & B_{k-2} \\ 0 & 0 & \cdots & & C_{k-1} & A_{k-1} \end{pmatrix}. \tag{9.1}$$

The class of such matrices includes, in particular, all the $(km) \times (km)$ matrices A with $m\_(A) \leqslant m$, $m\_{+}(A) \leqslant m$. With no loss of generality, assume that

$$\det A_i \neq 0, \quad \text{for all } i, \tag{9.2}$$

and transform the matrix A into a block matrix polynomial $A(\lambda) = A + \lambda Z^{2m}$ in the scalar parameter λ, where Z denotes the $(km) \times (km)$ matrix filled with zeros, except for its first subdiagonal, filled with ones, so that $A(\lambda)$ is obtained from A by inserting the matrix λI_m below each block C_i.

Moreover, $A(\lambda)$ is a matrix polynomial of degree 1 with a lower edge, and $(A(\lambda))^{-1} \bmod \lambda = A^{-1}$. If the computation is over a field of characteristic 0, we may set $\lambda = \varepsilon$, for a sufficiently small positive ε, and compute an approximate solution, $A^{-1}(\varepsilon)\,\mathbf{b}$, to a linear system $A\mathbf{x} = \mathbf{b}$. If the computation is over the rationals, then we may shift to the integer input by scaling, choose a sufficiently small ε, and then apply the techniques of [UP] to recover the exact solution $A^{-1}\mathbf{b}$ from its approximation $A^{-1}(\varepsilon)\,\mathbf{b}$. Furthermore, $\operatorname{adj} A = A^{-1} \det A$ is an integer matrix, so we may easily recover $\det A$ and $\operatorname{adj} A$ via rounding off (their approximations by) $\det A(\varepsilon)$ and $\operatorname{adj} A(\varepsilon)$ to the nearest integers. An alternative way is to fix a prime p and to compute the vectors $\mathbf{x}(p) = A^{-1}(p)\,\mathbf{b}$ and $A^{-1}(p)\,\mathbf{b} \bmod p = \mathbf{x}(p) \bmod p$. Then again, we may compute $\det A(p) \bmod p$ and $\operatorname{adj} A(p)\,\mathbf{b} \bmod p$ and recover $\det A$ and $(\operatorname{adj} A)\,\mathbf{b}$ if p is sufficiently large. Such a computation may fail if $\det A \bmod p = 0$, but this may occur only with a small probability if p is sufficiently large and $\det A \neq 0$ (see [P, BP]). Thus, over the rationals, we extend our arithmetic complexity estimates, (1.9) for *PREPROCESS* and (1.8) for *BACK·SOLVE*, to any $n \times n$ matrix with bandwidth m.

Remark 9.1. We may decrease the precision of the latter computation, $\lceil \log p \rceil$, if we first replace p with several primes $p_1, \ldots, p_k$ such that $p_1 \cdots p_k > p$, then perform the above computation modulo p_i for $i = 1, \ldots, k$, and recover $\det A \bmod p$ and $\operatorname{adj} A \bmod p$ by means of the Chinese remainder theorem. Alternatively, we may apply p-adic lifting of [MC].

Remark 9.2. Suppose that only s entries equal to 0 prevent a matrix A from having lower or upper edges. Then we may replace these entries by λ, solve the resulting system $A(\lambda)\,\mathbf{x} = \mathbf{b}$, and output $\mathbf{x} = \mathbf{x}(\lambda) \bmod \lambda = A^{-1}\mathbf{b}$. If s is small, then the computation of $\mathbf{x}(\lambda)$ only involves polynomials in λ of small degrees (as well as the ratios of such polynomials). In particular, the overall complexity of the solution is still bounded according to (1.8) and (1.9) if $s = O(1)$, that is, if s is bounded by a fixed constant independent of m and n.

APPENDIX A: THE BLOCK CYCLIC REDUCTION ALGORITHM

Let A denote a $k \times k$ block tridiagonal matrix with $m \times m$ blocks, defined by (9.1), and recall the classical block cyclic reduction algorithm [H, BGN, GL] for a linear system $A\mathbf{x} = \mathbf{b}$, defined over a field of characteristic 0. Assume nonsingularity of the diagonal blocks of the matrix A and of the smaller coefficient matrices (such as $A^{(1)}$ below) of the auxiliary linear systems that arise in the recursive applications of the block cyclic reduction. This restriction can be relaxed by means of the symmetrization techniques (see Section 6 and Remark A.1).

First rewrite the linear system $A\mathbf{x} = \mathbf{b}$ in the block form

$$C_i\mathbf{x}_{i-1} + A_i\mathbf{x}_i + B_i\mathbf{x}_{i+1} = \mathbf{b}_i, \quad i = 0, 1, \ldots, k-1, \tag{A.1}$$

where

$$C_0 = B_{k-1} = 0, \quad \mathbf{x}_{-1} = \mathbf{x}_k = \mathbf{0},$$
$$\mathbf{x} = (\mathbf{x}_0^{\mathrm{T}}, \ldots, \mathbf{x}_{k-1}^{\mathrm{T}}), \quad \mathbf{b} = (\mathbf{b}_0^{\mathrm{T}}, \ldots, \mathbf{b}_{k-1}^{\mathrm{T}})^{\mathrm{T}}, \tag{A.2}$$

and $\mathbf{x}_i, \mathbf{b}_i$ are m-dimensional vectors. The apply the following algorithm:

ALGORITHM A.1. (k, m)-Block Cyclic Reduction. For a given pair of a matrix A and a vector $\mathbf{b}$, defining a linear system (A.1), compute the inverses of the matrices A_{2j} for all j, the $k_1 \times k_1$ block matrix

$$A^{(1)} = \begin{pmatrix} A_0^{(1)} & B_0^{(1)} & 0 & \cdots & & 0 \\ C_1^{(1)} & A_1^{(1)} & B_1^{(1)} & \cdots & & 0 \\ 0 & C_2^{(1)} & A_2^{(1)} & B_2^{(1)} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & & A_{k_1-2}^{(1)} & B_{k_1-2}^{(1)} \\ 0 & 0 & \cdots & & C_{k_1-1}^{(1)} & A_{k_1-1}^{(1)} \end{pmatrix}, \tag{A.3}$$

and the k_1-dimensional vector

$$\mathbf{b}^{(1)} = (\mathbf{b}_j^{(1)}, j = 0, \ldots, k_1 - 1), \tag{A.4}$$

where

$$A_j^{(1)} = A_{2j+1} - C_{2j+1}A_{2j}^{-1}B_{2j} - B_{2j+1}A_{2j+2}^{-1}C_{2j+2}, \tag{A.5}$$
$$B_j^{(1)} = -B_{2j+1}A_{2j+2}^{-1}B_{2j+2}, \tag{A.6}$$
$$C_j^{(1)} = -C_{2j+1}A_{2j}^{-1}C_{2j}, \tag{A.7}$$
$$\mathbf{b}_j^{(1)} = \mathbf{b}_{2j+1} - C_{2j+1}A_{2j}^{-1}\mathbf{b}_{2j} - B_{2j+1}A_{2j+2}^{-1}\mathbf{b}_{2j+2},$$
$$j = 0, 1, \ldots, k_1 - 1; \quad k_1 = \lfloor k/2 \rfloor. \tag{A.8}$$

The computation of the matrices $A^{(1)}$ and A_{2j}^{-1}, for all j, will be called *preprocessing-bcr*; it can be performed at a cost bounded by $O(t_{I(m)}, kp_{I(m)})$. The vector $\mathbf{b}^{(1)}$ can be computed (after such preprocessing), at a cost bounded by

$$O(\log m, m^2 k/\log m). \tag{A.9}$$

When we solve the linear system

$$A^{(1)}\mathbf{x}^{(1)} = \mathbf{b}^{(1)}, \tag{A.10}$$

we obtain the subvector $\mathbf{x}^{(1)} = (\mathbf{x}_{2j+1}^{(1)}, j = 0, 1, \ldots)$ of the solution vector $\mathbf{x}$ for the original system $A\mathbf{x} = \mathbf{b}$. After the substitution of the vector $\mathbf{x}^{(1)}$ into the linear system (A.1), we may easily compute the remaining components of the

vector $\mathbf{x}$, at a cost bounded by (A.9). On the other hand, the systems (A.1) and (A.10) have the same format, and we may recursively apply Algorithm A.1 to the system (A.10) until we arrive at a system with the matrix of a size at most $m \times m$. Then we solve this system as a general linear system of m equations and extend the solution to the evaluation of the vector $\mathbf{x}$. Summarizing and using the B-principle and Proposition 2.4, we obtain

THEOREM A.1. *The block tridiagonal linear system* $A\mathbf{x} = \mathbf{b}$ *of* (A.1) *can be solved by means of* (*the block cyclic reduction*) *Algorithm A.1, provided that all the auxiliary diagonal blocks that appear in the recursive process of the application of this algorithm are nonsingular. The computational cost of this solution is bounded by*:

(a) $O((\log k)\, t_{I(m)}, kp_{I(m)}/\log k)$, *at all stages independent of the vector* $\mathbf{b}$ (*and called preprocessing*), *and*

(b) $O((\log k)(\log m), km^2/(\log k)(\log m))$, *at all other stages* (*called backsolving*).

The same estimates, for $k = n/m$, *hold for the solution of the linear system* $A\mathbf{x} = \mathbf{b}$ *with a nonsingular* $n \times n$ *matrix* A *having bandwidth* $m(A) \leqslant m$. *The assumption about nonsingularity of the diagonal blocks can be relaxed, over any field of characteristic* 0, *by means of the transition from the linear system* $A\mathbf{x} = \mathbf{b}$ *to* $A^{\mathrm{H}}A\mathbf{x} = A^{\mathrm{H}}\mathbf{b}$, *and from* $|\det A|$ *to* $\det(A^{\mathrm{H}}A) = |\det A|^2$.

As already mentioned in the introduction, the block cyclic reduction amounts to block Gaussian elimination, for some appropriate block elimination ordering. Therefore, the block cyclic reduction defines a block factorization of A into a product of block triangular matrices. Consequently, $\det A$ equals the product of the determinants of the diagonal blocks of these triangular matrices and of the determinants of the permutation matrices involved; each of the latter determinants equals 1 or -1 and is readily available. From these observations we obtain that the bound of part (a) of Theorem A.1 also applies to the evaluation of $\det A$.

Remark A.1. The matrix $A^{(1)}$ is Hermitian if A is Hermitian (see [GL, p. 140]).

Remark A.2. A simple inspection reveals that the block cyclic reduction algorithm is identical with the nested dissection [GLi, GT] applied to *LIN · SOLVE* for banded matrices. To observe the equivalences of the two methods, recall that both nested dissection and block cyclic reduction amount to Gaussian elimination with specific elimination ordering. Therefore, we need only compare this ordering for the two methods. In the case of a tridiagonal linear system, the associated vertex graph for the nested dissection is given by the line with n consecutive vertices. The nested dissection immediately leads to the eliminations of all the even numbered vertices-variables, which is exactly the ordering also defined by the cyclic reduction. In the block tridiagonal case, the elimination process is the same, except that a block of vertices-variables replaces each single vertex.

Remark A.3. The time bound of part (a) of Theorem A.1 matches one of (1.4) and the processor bound of part (a) of Theorem A.1 improves one of (1.4) by a logarithmic factor. Note, however, that currently we only know how to ensure performing the block cyclic reduction over fields of characteristic 0, whereas (1.4) is proved over any field. Moreover, the preprocessing stage of the block cyclic reduction outputs information that is as large as the input information, thus doubling the entire storage space. The storage space can be actually decreased back to the original size (but at the expense of increasing the processor bound by a logarithmic factor) if one overwrites the original matrix A by the two matrices $A^{(1)}$ and $A^{(2)}$ obtained by excluding the odd-indexed block equations of (A.1) and then recursively applies the same proces to both matrices $A^{(1)}$ and $A^{(2)}$.

APPENDIX B: AN ALTERNATIVE ALGORITHM

In this appendix, we will modify the Algorithms of Sections 3–6 by using, in the spirit of [E], a 3×3 (rather than a 2×2) block representation of the $n \times n$ input matrix A; specifically, we let

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{pmatrix}, \tag{B.1}$$

where the matrices A_{11}, A_{22}, and A_{33} have sizes $n_1 \times n_1$, $m(A) \times m(A)$, and $n_3 \times n_3$, respectively, with $n_1 = \lceil (n - m(A))/2 \rceil$, $n_3 = n - m(A) - n_1$.

We will follow the line of Sections 3–6 with the changes due to the shift from (3.1) to (B.1). Slightly abusing the notation, we will reuse such symbols as A_{11}, A_{12}, A_{21}, A_{22}, n_1, and Y, now giving them a slightly different meaning.

We will apply the following consequence of Proposition 2.1:

COROLLARY B.1. *Let* A_{ij} *denote the blocks of* A *defined by* (B.1). *Then* $A_{12} = I_L(n_1, m)\, A_{12}$, $A_{21} = A_{21} I_L(n_1, m)$, $A_{23} = A_{23} I_F(n_3, m)$, *and* $A_{32} = I_F(n_3, m)\, A_{32}$.

We will assume that the input matrix A is strongly nonsingular and will consider the following factorizations (similar to those of Section 3 and of [E]):

$$A = \begin{pmatrix} I_{n_1} & 0 & 0 \\ A_{21}A_{11}^{-1} & I_m & A_{23}A_{33}^{-1} \\ 0 & 0 & I_{n_3} \end{pmatrix} \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & Y & 0 \\ 0 & 0 & A_{33} \end{pmatrix} \times \begin{pmatrix} I_{n_1} & A_{11}^{-1}A_{12} & 0 \\ 0 & I_m & 0 \\ 0 & A_{33}^{-1}A_{32} & I_{n_3} \end{pmatrix}, \tag{B.2}$$

$$A^{-1}=\begin{pmatrix} I_{n_1} & -A_{11}^{-1} & 0 \\ 0 & I_m & 0 \\ 0 & -A_{33}^{-1}A_{32} & I_{n_3} \end{pmatrix}\begin{pmatrix} A_{11}^{-1} & 0 & 0 \\ 0 & Y^{-1} & 0 \\ 0 & 0 & A_{33}^{-1} \end{pmatrix} \times\begin{pmatrix} I_{n_1} & 0 & 0 \\ -A_{21}A_{11}^{-1} & I_m & -A_{23}A_{33}^{-1} \\ 0 & 0 & I_{n_3} \end{pmatrix}, \tag{B.3}$$

$$Y=A_{22}-A_{21}A_{11}^{-1}A_{12}-A_{23}A_{33}^{-1}A_{32}. \tag{B.4}$$

Note that (B.2) and the nonsingularity of A imply the nonsingularity of Y.

DEFINITION B.1. Let A be an $n\times n$ (strongly nonsingular) matrix with bandwidth m, A_{ij} denote the blocks of A defined by (B.1), and Y be defined by (B.4). Define the set

$$\Gamma_3(A)=\begin{cases} A^{-1}, & n<4m, \\ [\gamma_3(A)]\cup\Gamma_3(A_{11})\cup\Gamma_3(A_{33}), & \text{otherwise;} \end{cases} \tag{B.5}$$

$$\gamma_3(A)=(A^{-1}I_F(n,m),\ A^{-1}I_L(n,m),\ Y^{-1}).$$

Define PREPROCESS-3 as the problem of computing the set $\Gamma_3(A)$.

THEOREM B.1. *The complexity of PREPROCESS-3 is bounded by* (1.4).

Proof. Expand (B.3) and then apply Corollary B.1 to deduce that

$$A^{-1}I_F(n,m)=\begin{pmatrix} A_{11}^{-1}I_F(n_1,m)+A_{11}^{-1}I_L(n_1,m)\,A_{12}Y^{-1}A_{21}A_{11}^{-1}I_F(n_1,m) & 0 \\ -Y^{-1}A_{21}A_{11}^{-1}I_F(n_1,m) & 0 \\ A_{33}^{-1}I_F(n_3,m)\,A_{32}Y^{-1}A_{21}A_{11}^{-1}I_F(n_1,m) & 0 \end{pmatrix}; \tag{B.6}$$

$$A^{-1}I_L(n,m)=\begin{pmatrix} 0 & A_{11}^{-1}I_L(n_1,m)\,A_{12}Y^{-1}A_{23}A_{33}^{-1}I_L(n_3,m) \\ 0 & -Y^{-1}A_{23}A_{33}^{-1}I_L(n_3,m) \\ 0 & A_{33}^{-1}I_L(n_3,m)+A_{33}^{-1}I_F(n_3,m)\,A_{32}Y^{-1}A_{23}A_{33}^{-1}I_L(n_3,m) \end{pmatrix}. \tag{B.7}$$

Combine (B.4) and Corollary B.1 to deduce that

$$Y=A_{22}-A_{21}I_L(n_1,m)\,A_{11}^{-1}I_L(n_1,m)\,A_{12} -A_{23}I_F(n_3,m)\,A_{33}^{-1}I_F(n_3,m)\,A_{32}. \tag{B.8}$$

Due to the recursive definition (B.5), the computation of the set $\Gamma_3(A)$ amounts to computing $\Gamma_3(A_{11})$, $\Gamma_3(A_{33})$ and $\gamma_3(A)$. Once the sets $\Gamma_3(A_{11})$ and $\Gamma_3(A_{33})$ are computed, the matrices $A_{11}^{-1}I_F(n_1,m)$, $A_{11}^{-1}I_L(n_1,m)$, $A_{33}^{-1}I_F(n_3,m)$, and $A_{33}^{-1}I_L(n_3,m)$ are known; then we may use (B.8) to compute the matrix Y at the cost of a constant number of $m\times m$ matrix multiplications. After that, due to the equations (B.6) and (B.7), the computation of the $\gamma_3(A)$ [and therefore also the computation of $\Gamma_3(A)$] will only require a single $m\times m$ matrix inversion and a constant number of $q\times m$ by $m\times m$ matrix multiplications, where $q\leqslant\lceil n/2\rceil$.

Let $t_1(n,m)$ bound the number of parallel steps and let $p_1(n,m)$ bound the number of processors needed for the computation of $\Gamma_3(A)$. The above argument leads to the following estimates, where, for simplicity, we assume that n is a power of 2 and where c_1, c_2 are two constants:

$$t_1(n,m)\leqslant\begin{cases} t_{I(m)}, & n<3m \\ t_1(n/2,m)+c_1t_{I(m)}+c_2t_{M(n/2,m,m)}, & \text{otherwise;} \end{cases}$$

$$p_1(n,m)\leqslant\begin{cases} p_{I(m)}, & n<3m \\ \max(2p_1(n/2,m),\,p_{I(m)},\,p_{M(n/2,m,m)}), & \text{otherwise.} \end{cases}$$

Recursive application of the latter estimates and the B-principle yields the complexity bound (1.4). ∎

Having solved PREPROCESS-3, we now easily solve the original linear system.

THEOREM B.2. *If A is an $n\times n$ strongly nonsingular input matrix with bandwidth m and if the output of PREPROCESS-3 is available, then the complexity of the subsequent computation of $A^{-1}\mathbf{b}$, for a given vector $\mathbf{b}$, is bounded by* (1.6).

Proof. Let n_1 and n_3 be defined as in (B.1), let $\mathbf{b}=(\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3)^{\mathrm{T}}$, and let $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ be vectors of dimension n_1, m, n_3, respectively. Expand (B.3), multiply by $\mathbf{b}$, and apply Corollary B.1 to deduce that

$$A^{-1}\mathbf{b}=\begin{pmatrix} A_{11}^{-1}\mathbf{b}_1-A_{11}^{-1}I_L(n_1,m)\,A_{12}Y^{-1}(-A_{21}A_{11}^{-1}\mathbf{b}_1+\mathbf{b}_2-A_{23}A_{33}^{-1}\mathbf{b}_3) \\ Y^{-1}(-A_{21}A_{11}^{-1}\mathbf{b}_1+\mathbf{b}_2-A_{23}A_{33}^{-1}\mathbf{b}_3) \\ A_{33}^{-1}\mathbf{b}_3-A_{33}^{-1}I_F(n_3,m)\,A_{32}Y^{-1}(-A_{21}A_{11}^{-1}\mathbf{b}_1+\mathbf{b}_2-A_{23}A_{33}^{-1}\mathbf{b}_3) \end{pmatrix}.$$

Due to preprocessing, the matrices $A_{11}^{-1}I_L(n_1,m)$, $A_{33}^{-1}I_F(n_3,m)$, and Y^{-1} are available. Therefore, given $A_{11}^{-1}\mathbf{b}_1$ and $A_{33}^{-1}\mathbf{b}_3$, the computation of $A^{-1}\mathbf{b}$ requires only a constant number of matrix multiplications, of size at most $m\times q$, by vectors of dimensions at most q, where $q\leqslant\lceil n/2\rceil$.

Let $t_2(n,m)$ bound the number of parallel steps and $p_2(n,m)$ bound the number of processors needed for the computation of $A^{-1}\mathbf{b}$, given the precomputed matrices. The above argument leads to the following estimates, where, for simplicity, we assume that n is a power of 2, and where c_3 is a constant:

$$t_2(n, m) \leqslant \begin{cases} t_{M(m, m, 1)}, & n \leqslant 4m, \\ t_2(n/2, m) + c_3 t_{M(n/2, m, 1)}, & n > 4m; \end{cases}$$

$$p_2(n, m) \leqslant \begin{cases} p_{M(m, m, 1)}, & n \leqslant 4m, \\ \max(2p_2(n/2, m), p_{M(n/2, m, 1)}), & n > 4m. \end{cases}$$

Recursive application of the latter estimates and the B-principle yields the complexity bound (1.6). ▮

Remark B.1. Our solution requires $O(nm)$ memory space for a single banded linear system $A\mathbf{x} = \mathbf{b}$ and $O(nm + ns)$ for s such systems $A\mathbf{x}(i) = \mathbf{b}(i)$, since in each recursive step of the computation, its output may overwrite its input.

Remark B.2. Our solution to PREPROCESS-3 can be extended to the complexity of computing $\det A$: Due to (B.2), $\det A = \det Y \det A_{11} \det A_{33}$; moreover, the matrix Y is available from PREPROCESS-3, so that the original problem is reduced to ones of the smaller sizes. By recursively applying this reduction and the B-principle, we obtain the bound (1.5) on the complexity of computing $\det A$.

Remark B.3. Actually, Theorems B.1 and B.2 require only that the blocks A_{12}, A_{21}, A_{23}, and A_{32} satisfy Corollary B.1 and that the two larger (northwestern and southeastern) diagonal blocks A_{11} and A_{33} of the matrix A be nonsingular, as well as the similar diagonal blocks of the matrices A_{11}, A_{33}, and so on. This remark can be exploited to relax the strong nonsingularity assumption, by using symmetrization over the filed of characteristic 0 and randomization over any field of constants. The argument essentially repeats that of Section 6, and we will only indicate the main point of the (small) difference: In the 3×3 block case, we probabilistically ensure nonsingularity of the (1, 1) and (3, 3) blocks by means of the transition from the input matrix A to the matrix PA, where $P = \operatorname{diag}(I_{n_1 - m}, R_1, R_3, I_{n_3 - m})$, and R_1 and R_3 are random $m \times m$ matrices.

ACKNOWLEDGMENT

The referee made several thoughtful and helpful comments.

Received June 10, 1993; final manuscript received March 20, 1995

REFERENCES

[A] W. F. Ames, "Numerical Methods for Partial Differential Equations," Academic Press, New York, 1977.

[BGN] B. L. Buzbee, G. H. Golub, and C. W. Nielson, On direct methods for solving Poisson's equations, *SIAM J. Numer. Anal.* 7, 627–656 (1970).

[BP] D. Biniand and V. Y. Pan, "Polynomial and Matrix Computations," Birkhäuser, Boston, 1994.

[CW] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progression, *J. Symbolic Comput.* **9** (3), 251–280 (1990).

[E] W. Eberly, On efficient band matrix arithmetic, *in* "Proceedings, 33rd Annual IEEE Symposium on Foundation of Computer Science (FOCS)," pp. 457–463, IEEE Comput. Sci. Press, 1992.

[GL] G. H. Golub and C. F. Van Loan, "Matrix Computations," Johns Hopkins Univ. Press, 1989.

[GLi] A. George and J. W.-H. Liu, "Computer Solution of Large Positive Definite Systems," Prentice–Hall, Englewood Cliffs, NJ, 1981.

[GT] J. R. Gilbert and R. E. Tarjan, The analysis of nested dissection algorithms, *Numer. Math.* **50**, 377–404 (1987).

[H] R. W. Hockney, A fast direct solution of Poisson equation using Fourier analysis, *J. Assoc. Comput. Mach.* **12**, 95–113 (1965).

[KP91] E. Kaltofen and V. Y. Pan, Processor efficient parallel solution of linear systems over an abstract field, *in* "Proceedings 3rd Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)," pp. 180–191, ACM Press, New York, 1991.

[KP92] E. Kaltofen and V. Y. Pan, Processor efficient parallel solution of linear systems. II. The positive characteristic and singular cases, *in* "Proceedings 33rd Annual IEEE Symposium on Foundation of Computer Science (FOCS)," in pp. 714–723, IEEE Comput. Soc. Press, 1992.

[KR] R. Karp and V. Ramachandran, Parallel algorithms for shared-memory machines, *in* "Handbook of Theoretical Computer Science," pp. 869–941, MIT Press, Cambridge, MA/North-Holland, Amsterdam, 1990.

[LP] L. Lapidus and G. F. Pinder, "Numerical Solutions of Partial Differential Equations in Science and Engineering," Wiley, New York, 1982.

[MC] R. T. Moenck and J. H. Carter, Approximate algorithms to derive exact solutions to systems of linear equation, *in* "Proceedings EUROSAM," pp. 63 73, Lecture Notes in Computer Science, Vol. 72, Springer-Verlag, Berlin, 1979.

[P] V. Y. Pan, Complexity of parallel matrix computations, *Theoret. Comput. Sci.* **54**, 65–85 (1987).

[P85] V. Y. Pan, Fast and efficient parallel algorithms for the exact inversion of integer matrices, *in* "Proceedings, 5th Conference on Foundations of Software Technology and Theoretical Computer Science," pp. 504–521, Lecture Notes in Computer Science, Vol. 206, Springer-Verlag, Berlin, 1985.

[P91] V. Y. Pan, Complexity of algorithms for linear systems of equations, *in* "Computer Algorithms for Solving Linear Algebraic Equations (The State of the Art)" (E. Spedicato, Ed.), pp. 27–56, NATO ASI Series, Series F: Computer and System Sciences, Vol. 77, Springer-Verlag, Berlin, 1991.

[P92] V. Y. Pan, Parametrization of Newton's iteration for computations with structured matrices and applications, *Comput. Math. (Appl.)*, **24** (3), 61–75 (1992).

[P93] V. Y. Pan, Parallel solution of sparse linear and path systems, *in* "Synthesis of Parallel Algorithms" (J. H. Reif, editor), Chap. 14, pp. 621–678, Morgan Kaufmann, San Mateo, CA (1993).

[P93a] V. Y. Pan, On solving block tridiagonal systems via reduction to block bidiadonal case, manuscript, 1993.

[PP] V. Y. Pan and F. P. Preparata, Supereffective slow-down of parallel computations, *in* "Proceedings, Proc. 4th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)," pp. 402–409, ACM Press, New York, 1992.

[PP, a] V. Y. Pan and F. P. Preparata, Work-preserving speed-up of parallel matrix computations, *SIAM J. Comput.* **24** (4), (1995).

[PR85] V. Y. Pan and J. H. Reif, Efficient parallel solutions of linear systems, *in* "Proceedings, 17th Annual ACM Symposium on Theory of Computing (STOC)," pp. 143–152, ACM Press, New York, 1985.

[PR93] V. Y. Pan and J. H. Reif, Fast and efficient parallel solution of sparse linear systems, *SIAM J. Comput.* **22** (6), 1227–1250 (1993).

[PSA] V. Y. Pan, I. Sobze, and A. Atinkpahoun, "Improved Band Matrix Computations," Technical Report TR-93-60, International Computer Science Institute, Berkeley, CA, 1993.

[S] I. Sobze, "Parallel Algorithms for Banded Linear Systems of Equations," Ph.D. Thesis, CUNY Graduate Center, New York, 1994.

[Sc] J. T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, *J. Assoc. Comput. Mach.* **27**, 701–717 (1980).

[UP] S. Ursic and C. Patarra, Exact solution of systems of linear equations with iterative methods, *SIAM J. Algebraic Discrete Methods* **4**, 111–115 (1983).

[Z] R. E. Zippel, Probabilistic algorithms for sparse polynomials, *in* "Proceedings EUROSAM'79," pp. 216–226, Lecture Notes in Computer Science, Vol. 72, Springer-Verlag, Berlin/New York, 1979.