THE ROLE OF PARTITIONING IN THE NUMERICAL SOLUTION OF SPARSE SYSTEMS*

by

James R. Bunch
Cornell University
Ithaca, New York

and

Donald J. Rose
University of Denver
Denver, Colorado

TR 72 - 122

February, 1972

THE ROLE OF PARTITIONING IN THE NUMERICAL SOLUTION OF SPARSE SYSTEMS*

James R. Bunch, Cornell University

Donald J. Rose, University of Denver

1. INTRODUCTION

In the study of the numerical solution of sparse linear systems, several authors (Harary [1962A], Dulmage and Mendelsohn [1962A], Steward [1965A], Willoughby [1972A]) have considered the problem of ordering a sparse matrix so that it has a block structure which can be exploited efficiently for the solution of the system. We call this the "partitioning problem" and consider the related arithmetic and memory costs. In Section 3 we show that if the directed graph of the system is strongly connected then partitioning saves no arithmetic operations in an LU factorization but may save storage. On the other hand, if the directed graph of the system fails to be strongly connected, a reduction in both arithmetic and storage is possible. Some graph-theoretic preliminaries and useful operation counts are given in Section 2.

Closely related to partitioning is the problem of "modification" or "tearing". Modification is an attempt to arrive at a solution to a linear system synthetically by solving a slightly different ("torn") system and then modifying the solution. There have been many expositions of the "method of modification", especially in contexts where it is known as Kron's method. The reader is referred, in particular, to the work of Bennett [1965A], Branin [1959A], Steward [1965A], and Zielke [1968A]. There has been little comparison of the computational complexity of modification with that of ordinary Gaussian elimination (allowing for re-ordering); in Section 4 we consider these complexity questions. We show, for example, that single-element tearing of symmetric systems (such as those arising in circuit theory) is rarely advantageous when the torn system is

* Partial support under contracts NSF GU35 and NSF GP-28271.

solved by (symmetric) elimination. For general (non-symmetric) systems, we consider the relation between tearing and partitioning.

Throughout the paper our techniques are algebraic and combinatorial. We view partitioning and tearing in the context of Gaussian elimination, which leads us directly to complexity considerations.

2. PRELIMINARIES

In this section we associate a directed graph (digraph) with a sparse square matrix, and interpret the LU factorization (Gaussian eliimination) of the matrix graph-theoretically. The discussion is similar to that in Rose [1971A] for symmetric matrices. Our notions of digraph, (directed) paths, and (directed) cycles follow those of Harary [1969A].

Let M be an n x n (sparse) matrix such that PMP^t has an LU factorization for any permutation matrix P. We define a directed graph of M, $G(M) \doteq (X,\mathcal{A})$ with vertex set X and arc set $\mathcal{A}$ as follows:

A vertex $x_i \in X$ is associated with row i of M, and $(x_i,x_j) \in \mathcal{A}$ (an arc from x_i to x_j is in G) if $m_{ij} \neq 0$ and $i \neq j$. Note that the vertices X are regarded as ordered: i.e., $X = \{x_i\}_{i=1}^{n}$. When X of G(M) is not regarded as ordered, G(M) represents the equivalence class PMP^t.

Let the matrix M be written as $M = \begin{bmatrix} a & c^t \\ r & \bar{M} \end{bmatrix}$

where a is 1 x 1, r and c are (n-1) x 1 and $\bar{M}$ is (n-1) x (n-1). Then the first step of the LU factorization of M can be written as

$$(2.1) \qquad M = \begin{bmatrix} 1 & 0 \\ r/a & I \end{bmatrix} \cdot \begin{bmatrix} a & c^t \\ 0 & \bar{M}-rc^t/a \end{bmatrix}.$$

If G(M) is the directed graph of M, the elimination graph G_y is obtained from G by deleting y together with its incident arcs and adding an arc (x,z) whenever there exists a (directed) x, z path of length 2 containing y. G_y is the graph of the matrix obtained by "eliminating" the variable corresponding to y in Gaussian elimination; e.g., for eqn. (2.1)

$$G_{x_1} \text{ is the graph of } \bar{M} - rc^t/a.$$

We will call M a perfect elimination matrix if there exists a permutation matrix P such that $A = PMP^t$ and

$$(2.2) \qquad a_{ij} \neq 0 \text{ and } a_{ki} \neq 0 \Longrightarrow a_{kj} \neq 0$$

$$\text{for } 1 \leq i < j \leq n \text{ and } 1 \leq i < k \leq n.$$

We then have immediately

Proposition 1. If M is a perfect elimination matrix and $A = LU$ then $a_{ij} = 0 \Longrightarrow \ell_{ij} = 0$ and $u_{ij} = 0$. (There is no "fill-in" in Gaussian elimination.)

When M is symmetric, then G(M) can be considered an undirected graph, and perfect elimination matrices can be characterized in terms of the cycle and separator structure of G(M); see Rose [1971A]. For unsymmetric M such a characterization remains elusive; the following necessary (but not sufficient) condition is straightforward.

Proposition 2. Let M be a perfect elimination matrix with diagraph G(M). Then every cycle of G(M) of length $\ell \geq 3$ has a chord.

A diagraph G is strongly connected if there exists a path from any vertex x_i to any vertex x_j. If G is not strongly connected, the strong components of G are maximal strongly connected subgraphs; see Harary [1969A], chapter 16. If the digraph G(M) of a matrix M fails to be strongly connected, then there exists a permutation matrix P such that PMP^t is block upper triangular; the diagonal blocks are represented by the strong components of G. The following observation follows from the definition of an elimination graph.

Proposition 3. If G(M) is strongly connected, then so is every elimination graph G_y, $y \in X$.

Thus, when G(M) is strongly connected, "elimination" is required at each column step; i.e., the pivot column is never null below the diagonal

Let M be an nxn matrix with digraph G(M) and let $G_1, \ldots, G_{n-1}$ be the sequence of elimination graphs defined recursively by $G_0 = G(M)$ and $G_i = (G_{i-1})_{x_i}$. Futhermore,

let $r_i = | \{y \in X_{i-1}: (x_i, y) \in \mathcal{A}_{i-1},\ G_{i-1} = (X_{i-1}, \mathcal{A}_{i-1})\} |$

and $c_i = | \{y \in X_{i-1}: (y, x_i) \in \mathcal{A}_{i-1},\ G_{i-1} = (X_{i-1}, \mathcal{A}_{i-1})\} |$

be the out-degree and in-degree, respectively, of vertex x_i in the elimination graph G_{i-1}. Counting operations is now straightforward and we have

Proposition 4. Let M and $\{r_i\}_{i=1}^{n-1}$ be as above. Counting multiplications and divisions as multiplications and operations $a + 0$ ($a \neq 0$, which occurs whenever there is fill-in) as additions, we obtain:

(1) The $M = LU$ factorization requires

$$\sum_{i=1}^{n-1} (r_i + 1)\, c_i \text{ multiplications and}$$

$$\sum_{i=1}^{n-1} r_i c_i \text{ additions;}$$

(2) Given an n-vector k, the backsolving operations

$$Ly = k$$
$$Ux = y$$

require

$$n + \sum_{i=1}^{n-1} (r_i + c_i) \text{ multiplications and}$$

$$\sum_{i=1}^{n-1} (r_i + c_i) \text{ additions.}$$

Proposition 4 is similar to operation counts given in Rose [1971A] for the LDL^t decomposition of a symmetric positive definite matrix. When M is symmetric (we consider G(M) as undirected), $r_i = c_i = d_i$, and, by taking advantage of symmetry, we can solve $Mx = k$, where $M = LDL^t$, with

$$\sum_{i=1}^{n-1} \frac{1}{2} d_i(d_i + 3) + (2\sum_{i=1}^{n-1} d_i + n) \text{ multiplications and}$$

$$\sum_{i=1}^{n-1} \frac{1}{2} d_i(d_i + 1) + 2\sum_{i=1}^{n-1} d_i \text{ additions, where}$$

the first term in each sum gives the factorization operations and the second term gives the backsolving operations.

3. PARTITIONING

It has been established (see Householder [1964A], Section 5.2) that partitioning may be regarded as block Gaussian elimination. In the context of sparse matrices, partitioning may save storage and/or arithmetic depending on the connectivity of the graph G(M).

Let M be an n x n matrix partitioned as

$$M = \begin{bmatrix} A_{11} & C_1 \\ R_1 & A_{22} \end{bmatrix}$$

where A_{11} is m x m. Block elimination factors M as

$$(3.1) \qquad M = \begin{bmatrix} I & 0 \\ R_1A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & C_1 \\ 0 & \Delta \end{bmatrix}$$

where $\Delta = A_{22} - R_1A_{11}^{-1}C_1$.

One may then proceed recursively partitioning and factoring the lower order matrix Δ to obtain a block LU factorization of M. However, in sparse matrix applications it is rarely the case that A_{11}^{-1}

of (3.1) is available. Furthermore, it is not desirable to compute A_{11}^{-1} since this is usually more costly than computing the LU factorization of A_{11}, and A_{11}^{-1} is generally full while A_{11} is generally sparse. If, instead, A_{11} is factored, $A_{11} = L_1U_1$, we are lead immediately to the alternate factorization

$$(3.1)' \qquad M = \begin{bmatrix} L_1 & 0 \\ R_1U_1^{-1} & I \end{bmatrix} \begin{bmatrix} U_1 & L_1^{-1}C_1 \\ 0 & \Delta \end{bmatrix}$$

which we recognize as the partial LU factorization of M after the first m variables have been eliminated. It follows that partitioning saves no arithmetic operations over the usual LU factorization. This is the case when the digraph G(M) is strongly connected since we then have $R_1 \neq 0$ for any partitioning. Given that the computation would be done as indicated by eqn. (3.1)', what, then, is the advantage of "block" elimination? We show now that by reorganizing the calculation we can save storage but possibly at a cost of more arithmetic operations.

The factorization shown in eqn. (3.1)' denotes the first m steps of "point" Gaussian elimination. Suppose that the calculation of the matrices $R_1U_1^{-1}$ and $L_1^{-1}C_1$ has been executed step by step and they have been stored separately as auxiliary temporary storage. After these first m steps we store only L_1, U_1, and Δ. We then proceed to decompose Δ as before using the same auxiliary storage for all subsequent matrices $R_iU_i^{-1}$ and $L_i^{-1}C_i$. At the conclusion of the calculation, we have in storage only the matrices L_i, U_i, R_i and C_i. We have saved storage because R_i and C_i are sparser than $R_iU_i^{-1}$ and $L_i^{-1}C_i$. Now consider the calculation required to solve $Mx = k$; for convenience, consider the case of only two blocks. We backsolve

$$\begin{bmatrix} L_1 & 0 \\ R_1U_1^{-1} & L_2 \end{bmatrix} \begin{bmatrix} U_1 & L_1^{-1}C_1 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

by computing

(a) $L_1y_1 = k_1$

(b) $U_1\hat{y}_1 = y_1$

(c) $\hat{k}_2 = k_2 - R_1\hat{y}_1$

(d) $L_2y_2 = \hat{k}_2$

(e) $U_2x_2 = y_2$

(f) $L_1\hat{x}_2 = C_1x_2$

(g) $U_1x_1 = y_1 - \hat{x}_2$.

Note that the number of arithmetic operations necessary to compute steps (a) - (g) depends in part on the relative sparseness of L_i, L_i^{-1}, U_i, U_i^{-1}. That is, even though $R_iU_i^{-1}$ and $L_i^{-1}C_i$ have been previously computed but not stored (and hence lost), it is possible that steps (a) - (g) can require fewer arithmetic operations than ordinary backsolving. This will be the case if and only if the sum of the nonzero entries in the upper triangular part of U_1 and in the strictly lower triangular part of L_1 are less than the total fill-in caused by U_1^{-1} in $R_1U_1^{-1}$ and by L_1^{-1} in $L_1^{-1}C_1$.

When the digraph G(M) of the nxn matrix M is not strongly connected, the situation is quite different. In this case PMP^t can always be partitioned as

$$\overline{M} = PMP^t = \begin{bmatrix} A_{11} & C_1 \\ 0 & A_{22} \end{bmatrix},$$

where A_{11} is mxm and P is a permutation matrix. Note that the LU factorization of $\overline{M}$ is

$$\overline{M} = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix} \begin{bmatrix} U_1 & L_1^{-1}C_1 \\ 0 & U_2 \end{bmatrix}$$

where $A_{11} = L_1U_1$ and $A_{22} = L_2U_2$. It is unnecessary, however, to compute $L_1^{-1}C_1$, since in order to solve $\overline{M}x = k$ we need only factor A_{11} and A_{22}. The solution to

$$\begin{bmatrix} L_1U_1 & C_1 \\ 0 & L_2U_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

is then found by backsolving

(a) $L_2y_2 = k_2$

(b) $U_2x_2 = y_2$

(c) $L_1y_1 = k_1 - C_1x_2$

(d) $U_1x_1 = y_1$

Hence, $L_1^{-1}C_1$ is never needed and since $L_1^{-1}C_1$ is less sparse than C_1, factoring just the irreducible blocks A_{ii} saves both storage and arithmetic.

Note that the success of any partitioning depends on identifying a convenient block structure. The task of identifying such a structure has its price, and this must be considered. For example, when

M is reducible we save operations and storage only if we can identify the strong components of G(M) and order them appropriately. (This is equivalent to finding the permutation matrix P such that PMP^t is block upper triangular.) In a sense, then, it is useful to regard partitioning in the context of optimal ordering; that is, finding an ordering of the digraph G(M) or set of subgraphs such that some desired measure of complexity is minimized.

4. MODIFICATION

Let $\hat{B}$ be an nxn matrix and consider the system

(4.1) $$\hat{B}x = k$$

We attempt to solve (4.1) by considering the "torn" matrix

(4.2) $$B = \hat{B} - \sigma uv^t$$

where $\sigma \neq 0$ is a scalar and u, v are nx1, and by using the method of modification which is based on the identity

(4.3) $$(B + \sigma uv^t)^{-1} = B^{-1} - (\sigma^{-1} + v^t B^{-1} u)^{-1} B^{-1} uv^t B^{-1} .$$

To solve (4.1) we factor

(a) $B = LU$ and backsolve

(b) $LUx_1 = k$

(c) $LUx_2 = u$

then

(d) $x = x_1 - \alpha x_2,\ \alpha = (\sigma^{-1} + v^t x_2)^{-1} v^t x_1$.

For symmetric $\hat{B}$ we consider symmetric B with $B = LDL^t$ and take v=u; (a) - (d) are altered appropriately.

Later we restrict our attention to "single-element tearing"; that is, we let $\sigma = \hat{b}_{ij}$, $u = e_i$, $v = e_j$, (where e_i is the i^{th} column of the identity matrix).

Thus $\hat{B} - B = \hat{b}_{ij} e_i e_j^t$.

In the symmetric case we let $-\sigma = \hat{b}_{ij} = \hat{b}_{ji}$, $u = e_i - e_j$, $v = u$; this is appropriate in network problems where a change in an off-diagonal element (corresponding to a resistor, say) necessitates a change in two diagonal entries b_{ii} and b_{jj}. Note that in single-element tearing the calculation of α in (d) above is trivial.

The method of modification may be related to partitioning (or bordering) as follows.

Theorem 1. Let x be the solution to the system (4.1). Then x can be augmented to satisfy the system

(4.4) $$\begin{bmatrix} B & u \\ v^t & -\sigma^{-1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k \\ 0 \end{bmatrix}$$

Proof. We may rewrite (4.4) as

$$\begin{bmatrix} I & -\sigma u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} B+\sigma uv^t & 0 \\ v^t & -\sigma^{-1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} I & -\sigma u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} k \\ 0 \end{bmatrix}$$

which is equivalent to

$$\begin{bmatrix} B+\sigma uv^t & 0 \\ v^t & -\sigma^{-1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k \\ 0 \end{bmatrix} \quad \text{or}$$

$$(B + \sigma uv^t)\, x = k, \; v^t x - \sigma^{-1}\, y = 0. \qquad \text{qed.}$$

Note also that the LU factorization of (4.4) leads to

$$\begin{bmatrix} B & u \\ v^t & -\sigma^{-1} \end{bmatrix} = \begin{bmatrix} L & 0 \\ v^t U^{-1} & 1 \end{bmatrix} \begin{bmatrix} U & L^{-1}u \\ 0 & -\sigma^{-1}-v^t B^{-1} u \end{bmatrix},$$

and that the operations in the block back solution of

$$(4.5) \quad \begin{bmatrix} L & 0 \\ v^t U^{-1} & 1 \end{bmatrix} \begin{bmatrix} U & L^{-1}u \\ 0 & -\sigma^{-1}-v^t B^{-1} u \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} k \\ 0 \end{bmatrix}$$

can be identified exactly with the operations in solving (a) - (d) above. Hence modification may be regarded as solving the system (4.4) by block Gaussian elimination rather than "backwards" Gaussian elimination as indicated in the proof of Theorem 1. In the context of optimal ordering, "tearing" is of no advantage if there exists an optimal ordering which orders the variable y first in the system (4.4). We can be more precise in the case that B and $\hat{B}$ are symmetric.

We begin by considering the following:

Example 1. Let T_n be an n x n symmetric tridiagonal matrix and $\hat{T}_n = T_n + \sigma(e_1 - e_n)(e_1 - e_n)^t$. Using the operation counts for symmetric elimination given in Section 2, we find that $\sim$ 10n multiplications are required to solve $\hat{T}_n x = k$ directly. However, to solve $\hat{T}_n x = k$ by using (a) - (d) above requires only $\sim$9n multiplications (n multiplications are required for step (d)). Note that n multiplications are saved because $u = v = e_1 - e_n$ is a sparse vector and some of the operations in the block backsolving in (4.5) need not be repeated in the calculation of $-\sigma^{-1} - v^t B^{-1} u$. We show now that the matrix $\hat{T}_n$ is essentially the only case when tearing is advantageous.

In what follows the graphs $G(\hat{B})$ and $G(B)$ will be the undirected graphs of the symmetric matrices $\hat{B}$ and B of eqns. (4.1) and (4.2).

In comparing operations for solving (4.1) by elimination and by modification [(a) - (d)], there is no loss in assuming that B is a perfect elimination matrix and that the ordering implied by B is a monotone transitive ordering*. We will need the following lemma which extends a theorem of A. Hoffman (see Rose [1971A]).

Lemma. Let G(B) be a monotone transitive ordered graph with degree sequence d_i, $1 \leq i \leq n-1$. Then there exists an ordering for G(B) and a corresponding degree sequence (for $G(\hat{B})$) $\hat{d}_i$, $1 \leq i \leq n-1$, such that $d_i \leq \hat{d}_i \leq d_i + 1$.

Proof. $G(\hat{B}) = (X,\hat{E})$ has single edge $e = xy$ which is not an edge of $G(B) = (X,E)$. $G(\hat{B})$ may be triangulated by adding edges yw for all $w \in X$; let $T(\hat{B}) = (X, \hat{E} \cup \hat{F})$ be this triangulated graph. From the results in Rose [1971A], there exists a monotone transitive ordering for G(B) such that $x_n = y$ (i.e., y is ordered last) with degree sequence equivalent to that of d_i, $1 \leq i \leq n-1$ above. It is easy to see that the same ordering is monotone transitive for T. By construction the degree sequence, d_i^T, $1 \leq i \leq n-1$, for T satisfies $d_i \leq d_i^T \leq d_i + 1$, hence also the degree sequence $\hat{d}_i$. qed.

Appealing to the lemma we see that we may consider the matrices $\hat{B}$ and B (re-ordered) to satisfy

$$\text{(4.6)} \qquad \hat{B} - B = \hat{b}_{in} (e_i - e_n)(e_i - e_n)^t ;$$

that is, the torn elements are in the last row and column. In fact, without loss of generality we can assume more; namely,

$$\text{(4.7)} \qquad \hat{B} - B = \hat{b}_{1n}(e_1 - e_n)(e_1 - e_n)^t ,$$

since if $1 \leq i \leq n$ in (4.6) then we can partition $\hat{B}$ (and B) to have the factored form

$$\text{(4.8)} \qquad \hat{B} \begin{bmatrix} L & 0 \\ RL^{-t}D^{-1} & I \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & \hat{\Delta} \end{bmatrix} \begin{bmatrix} L^t & D^{-1}L^{-1}R^t \\ 0 & I \end{bmatrix}$$

where

$$\hat{B} = \begin{bmatrix} B_{11} & R^t \\ R & \hat{B}_{22} \end{bmatrix} , \quad B_{11} = LDL^t \text{ is } (i-1) \times (i-1)$$

---

* See Rose [1971A] for details of these graph-theoretic notions. Such an assumption is possible because once any ordering (e.g., optimal ordering) for B is chosen the operation counts are the same as those for some B that is a perfect elimination matrix (with the chosen ordering).

and $\hat{\Delta} = \hat{B}_{22} - RB_{11}^{-1} R^t$. Then we need only solve $\hat{\Delta}\, w = b$ for appropriate w and b to complete the solution to $\hat{B}x = k$. To solve $\hat{B}x = k$ by tearing and modification directly would require more operations than by partitioning as in (4.8) and solving only $\hat{\Delta}\, w = b$ by tearing and modification; i.e., this presents the method of modification in its most advantageous form.

We then have:

Theorem 2. Let $\hat{B}$ and B be as in (4.7) and $G(\hat{B})$, G(B) and d_i for $1 \leq i \leq n-1$ be as above. Let $m(LDL^t)$ be the number of multiplications necessary to solve the system $\hat{B}x = k$ by symmetric elimination (see Section 2) and m(tear) be the number of multiplications necessary for solution by tearing and modification (a) - (d). Then

$$m(\text{tear}) < m(LDL^t) \text{ only if } \sum_{i=1}^{n-1} d_i < 2(n-2).$$

Corollary 1. If $d_i \geq 2$ for $1 \leq i \leq n-2$ and $d_{n-1} = 1$, then

$$m(\text{tear}) > m(LDL^t).$$

Corollary 2. If $m(\text{tear}) < m(LDL^t)$, then

$$m(LDL^t) - m(\text{tear}) \leq 2(n-2) - \sum_{i=1}^{n-1} d_i \leq n-5 < n.$$

Hence, if the graph G(B) is 2-connected, modification and tearing never reduce operations; furthermore, the best possible reduction is fewer than n multiplications. The situation for additions is similar.

Proof of Theorem 2. Recall we have re-ordered $\hat{B}$ and B so so that $d_i \leq \hat{d}_i \leq d_i + 1$. Hence $m(LDL^t) \leq \sum_{i=1}^{n-1} \frac{1}{2}(d_i+1)(d_i+4)$

$$+ 2\sum_{i=1}^{n-1} (d_i+1) + n = \sum_{i=1}^{n-1} \frac{1}{2}(d_i^2 + 9d_i) + 5n - 4.$$

From (a) - (d) we count (disregarding the multiplications in computing α)

$$m(\text{tear}) = \sum_{i=1}^{n-1} \frac{1}{2} d_i(d_i+3) + 4\sum_{i=1}^{n-1} d_i + 3n = \sum_{i=1}^{n-1} \frac{1}{2}(d_i^2 + 11d_i) + 3n.$$

Thus

$$m(\text{tear}) < m(LDT^t) \text{ only if } \sum_{i=1}^{n-1} d_i < 2(n-2). \qquad \text{qed.}$$

We conclude by remarking that Theorem 2 is based upon the assumption that both B and $\hat{B}$ are factored in an LDL^t factorization; that is, the equations $\hat{B}x=k$ and the equations (a) - (d) are solved by "point" (symmetric) Gaussian elimination. If the torn systems $Bx_1 = k$ and $Bx_2 = u$ can be solved more efficiently, modification might be more attractive. For example, if the torn tridiagonal matrix, $T_{\dot{n}}$, of example 1 has the special form considered in Rose [1969A], the system $\hat{T}_n x = k$ can be solved with at most 6n multiplications. Similar cases may arise in circuit theory where one can solve a system by either the "circuit" method or "node" method.

We conjecture that a result similar to Theorem 2 holds for unsymmetric B and $\hat{B}$. In the unsymmetric case, however, it may be possible to tear in such a manner that G(B) has more strongly connected components than G($\hat{B}$). We may then save operations as shown in Section 3 since B could be made block upper triangular with more blocks than $\hat{B}$. The work of Steward [1962A, 1965A] may be interpreted in this light. Of course, any such tearing requires finding the strong components of G(B) in advance (see Purdom [1970A], Warshall [1962A], and Willoughby [1972A] for discussion of algorithms for finding strong components).

Some of these remarks are illustrated in the following:

Example 2. Let $\hat{M}$ be the 2n x 2n matrix

$$\hat{M} = \begin{bmatrix} T_1 & F \\ Z & T_2 \end{bmatrix}$$

where T_1 and T_2 are n x n tridiagonal, F is n x n and full, and Z is n x n with all elements zero except one: $Z_{n1} = x \neq 0$. Let M be the torn matrix $M \doteq \hat{M} - xe_n e_1^t$; then M is block upper triangular. To solve $\hat{M}x = k$ by ordinary Gaussian elimination requires $\sim 3n^2 + 16n$ multiplications, while solving $\hat{M}x = k$ by tearing and modification taking advantage of the block upper triangular structure of M requires $\sim 2n^2 + 16n$ multiplications.

BIBLIOGRAPHY

1972 Willoughby, R. A., "A matrix reducibility algorithm," Math. Comp., to appear.

1971 Read, R. (Editor) Graph Theory and Computing, Academic Press, New York.

Rose, D. J., "A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations," to appear in Graph Theory and Computing [Read (1971A)].

1970 Purdom, P., Jr., "A transitive closure algorithm," BIT 10, pp. 76-94.

1969 Harary, F., Graph Theory, Addison-Wesley, Reading, Massachusetts.

Rose, D. J., "An algorithm for solving a special class of tridiagonal systems of linear equations," Comm. ACM 12, pp. 234-236.

1968 Zielke, G., "Inversion of modified symmetric matrices," J. Assoc. Comput. Mach. 15, pp. 402-408.

1965 Bennett, J. M., "Triangular factors of modified matrices," Numer. Math. 7, pp. 217-221.

Steward, D. V., "Partitioning and tearing systems of equations," J. Soc. Indust. Appl. Math. Ser. B Numer. Anal. 2, pp. 345-365.

1964 Householder, A. S., The Theory of Matrices in Numerical Analysis, Blaisdell, New York.

1962 Dulmage, A. L., and Mendelsohn, N. S., "On the inversion of sparse matrices," Math. Comp. 16, pp. 494-496.

Harary, F., "A graph theoretic approach to matrix inversion by partitioning," Numer. Math. 4, pp. 128-135.

Steward, D. V., "On an approach to techniques for the analysis of the structure of large systems of equations," SIAM Rev. 4, pp. 321-342.

Warshall, S., "A theorem on Boolean matrices," J. Assoc. Comput. Mach. 9, pp. 11-12.

1959 Branin, F. H., Jr., "The relation between Kron's method and the classical methods of network analysis," IRE WESCON Convention Record, Part 2, pp. 1-29.

FINDING A MAXIMUM CLIQUE

Robert Tarjan

TR 72 - 123

March 1972

Department of Computer Science
Cornell University
Ithaca, New York 14850