SIAM Review
Vol. 3, No. 2, April, 1961

THE USE OF LINEAR GRAPHS IN GAUSS ELIMINATION[1]

S. Parter[2]

1. Introduction

The recent interest in various "block" iterative methods has been based on the ability to directly solve certain special systems of equations. As Varga [5] has remarked, the direct solution of tri-diagonal matrix equations has been extremely useful. Upon studying the methods in use one is struck by the fact that in almost all cases the matrices attacked have a special form which makes Gauss elimination, and hence the corresponding factorization, particularly simple and elegant.

In most discussions of these problems the concept of "sparseness" appears. Yet, two matrices which are equally "sparse" (and equally well-conditioned) may behave entirely differently when attacked by Gauss elimination; e.g., consider a five-diagonal matrix which has non-zero elements only on the main diagonal and the two diagonals directly above and below, and the five-diagonal matrix which appears in the Laplace difference equation, having some zero diagonals between the non-zero diagonals. As we shall see (Examples 2 and 6), the first of these is easily handled by Gauss elimination, while in the latter case the situation soon becomes very complicated, messy and unwieldly.

The major purpose of this note is to introduce a procedure to study the effect of "sparseness" in Gauss elimination and which enables one to choose an optimal elimination scheme and/or to determine if elimination is a practical approach. This is done by associating a linear graph $G(A)$, a "picture", with each matrix A.

In many problems the role of the associated graph is completely apparent. For example K. Friedrich [3] (see pp. 532 and 533) made use of these concepts in his work on the *normal equations* for Geodetic nets. In that case the "pictures" normally associated with the phenomena being studied are equivalent to the associated graph. In Section 2 we develop this association between graph and matrix and illustrate the viewpoint with a discussion of several examples of problems of current interest.

In Section 3 we develop some of the properties of special graphs, the "trees". These properties of trees are exploited to invert and factor a matrix associated with a tree. This discussion is of a practical nature, involving programming a digital computer to solve this problem.

In view of our interest in the mechanics of Gauss elimination, throughout most of this work we assume the matrices involved are sufficiently well-conditioned so that Gauss elimination is always possible. However, in Section 4 we

[1] Received by the editors October 28, 1958 and, in revised form, October 28, 1960.

[2] Cornell Computing Center and Department of Mathematics, Cornell University, Ithaca, New York. (A major portion of this work was done while the author was at the Brookhaven National Laboratory, Upton, L. I., New York.)

will make a few remarks concerning "trees" and the breakdown of the Gauss elimination procedure.

2. FORMULATION AND EXAMPLES

Consider a system of N equations in N unknowns

$$Ax = y, \tag{2.1}$$

where $A = (a_{ij})$ is an N by N matrix and x and y are column vectors of dimension N. With each index i we associate a "point" P_i. For $i \neq j$, if either a_{ij} or a_{ji} is not zero we connect P_i to P_j by an "arc". This configuration of "points" and "arcs" is called a linear graph G, or $G(A)$.

It should be noted that the graph $G(A)$ is invariant under simultaneous permutation of rows and columns of A. If the indices of the points of $G(A)$ are permuted the graph $G(A)$ then corresponds to the same system of equations, provided only we also perform the same permutation on the unknown (the columns of A) and the equations (the rows of A and y).

The graph $G(A)$ is said to be "connected" if it is a connected geometric object. From the above remarks it is easy to see that if $G(A)$ is not connected, such a permutation exhibits A as the direct sum of smaller matrices and one really has several smaller problems to consider. Hence, we shall assume $G(A)$ is connected.

For simplicity we shall frequently say the point "i" rather than P_i.

In Gauss elimination each variable x_i is attached to a definite equation, the ith. Choosing an ordering of the integers 1 through N, which may be taken as the natural ordering after an appropriate permutation, one systematically obtains smaller systems involving only "later" unknowns. That is, one solves the first equation for x_1 in terms of the remaining variables, obtaining a "back-substitution" equation. One uses this equation to eliminate x_1 from the remaining equations obtaining a system in the remaining $N - 1$ unknowns. Then the same procedure is applied to x_2, etc. The coefficient a_{ii}' (since the matrix A has been modified in the course of the elimination, in general $a_{ii}' \neq a_{ii}$) of x_i when one solves for x_i from the ith equation is called the pivot. In the remainder of this section it is always assumed that the matrix A has the property that Gauss elimination by pivoting on elements of the main diagonal is always possible *in any order*. For example, if A is positive definite, A has this property. In general, this condition requires a special analysis of the particular problem at hand.

THEOREM I. *Upon elimination of x_i from the subsequent equations, the new graph G^1 of the remaining system is contained in the graph $\tilde{G}$ obtained from G by:*

1. *eliminating the point "i",*
2. *pair-wise connecting (by arcs) all points which previously were connected to "i".*

PROOF. Assuming $a_{ii} \neq 0$, Gauss elimination proceeds as follows:

$$x_i = -1/a_{ii} \sum_{k \neq i} a_{ik}x_k + y_i/a_{ii}, \tag{2.2}$$

and substitute (2.2) into each of the other equations. If the point "j" is not directly connected to "i", $a_{ji} = 0$, and no substitution need be made. On the

other hand, if the points "k" and "j" are both connected to "i" the substitution (2.2) introduces x_j into the kth equation and x_k into the jth equation. Thus a_{jk}' and a_{kj}' (the new coefficients) are not zero and "j" is connected to "k".

Remark. Under fortuitous circumstances some pairs of points previously connected may become disconnected. However, in general this is *not* the case and almost always $G^1 = \tilde{G}$.

Remark. In certain special cases, such as the triangular matrices and related matrices, the graph $G^1 \neq \tilde{G}$, not by accident, but by the structure of the problem. However, we will not discuss this situation here and always treat the matter as though $G^1 = \tilde{G}$.

To see the use of Theorem I, consider the following example: The matrix A is of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 \\ a_{41} & 0 & 0 & a_{44} & 0 \\ a_{51} & 0 & 0 & 0 & a_{55} \end{bmatrix}, \tag{2.3}$$

and the associated graph $G(A)$ is given by

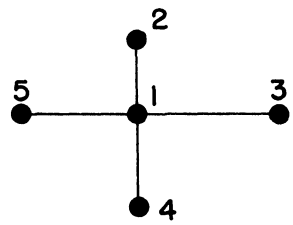


FIGURE 1.

If one applies Gauss elimination in the natural order and chooses to eliminate x_1 first, the back substitution equation is

$$x_1 = -\frac{1}{a_{11}} \{a_{12}\, x_2 + a_{13}\, x_3 + a_{14}\, x_4 + a_{15}\, x_5\} + \frac{y_1}{a_{11}}. \tag{2.3b}$$

The matrix of the reduced system of equations now (generally) has the form

$$A' = \begin{bmatrix} a_{22}' & a_{23}' & a_{24}' & a_{25}' \\ a_{32}' & a_{33}' & a_{34}' & a_{35}' \\ a_{42}' & a_{43}' & a_{44}' & a_{45}' \\ a_{52}' & a_{53}' & a_{54}' & a_{55}' \end{bmatrix}. \tag{2.3c}$$

The new associated graph has the form shown in Fig. 2

However, in order to obtain (2.3c) from the equations themselves, one must perform a large number of algebraic computations, as in the elimination of x_1.

On the other hand, if we apply Theorem I to (2.3a), we immediately obtain

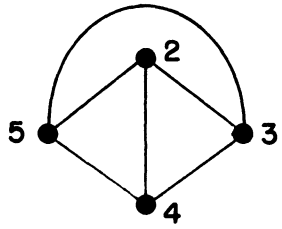


FIGURE 2.

the "picture" (2.3d), and we know that the reduced matrix *at worst* has the form (2.3c) *without the use of any algebraic computations.*

Let us now choose to eliminate x_2. Using Theorem I we find the graph of the associated matrix is of the form

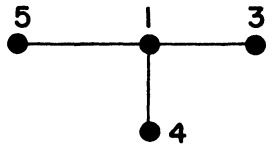


FIGURE 3.

We leave it to the reader to construct the form of the associated matrix.

In this case elimination of x_1 creates a full matrix, while the elimination of x_2 introduces no new non-zero elements. Given any matrix A, an optimal elimination is one which introduces as few new arcs as possible. The following examples illustrate how the drawing of the associated graph facilitates finding an optimal elimination scheme.

Example 1. Tridiagonal Matrices. The matrix A has the form

$$A = \begin{bmatrix} x & x & & & \\ x & x & x & 0 & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & 0 & x & x & x \\ & & & x & x \end{bmatrix}, \tag{2.4}$$

where the x's denote non-zero elements.

The graph $G(A)$ is a "chain":

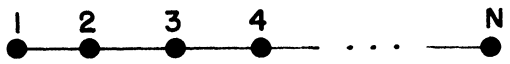


FIGURE 4.

One may successively eliminate "1", "2", $\cdots$, "N" (or "N", "$N-1$", $\cdots$, "1"). This is precisely the well-known method of solving such problems. Since

"i" is always an end point after x_{i-1} has been eliminated, the ith equation (at that time) always is of the form

$$a_{ii}X_i + a_{ii+1}X_{i+1} = Y_i ,$$

and the back substitution formulae are of the form

$$\begin{aligned} X_n &= F_n , \\ X_k &= E_k X_{k+1} + F_k . \end{aligned} \tag{2.4b}$$

Example 2. The Five-diagonal Matrices. We consider now the five-diagonal matrices which have non-zero elements only on the main diagonal and the two diagonals immediately above and below the main diagonal. The matrix A has the form

$$A = \begin{bmatrix} x & x & x & & & & & & \\ x & x & x & x & & & & O & \\ x & x & x & x & x & & & & \\ & \cdot & \cdot & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot & \\ & & & & & x & x & x & x & x \\ & O & & & & & x & x & x & x \\ & & & & & & & x & x & x \end{bmatrix} , \tag{2.5}$$

where once more, the x's denote non-zero elements. The graph has the form

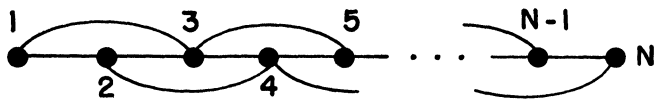


FIGURE 5.

which may be drawn also as shown in Fig. 6 (if N is even).

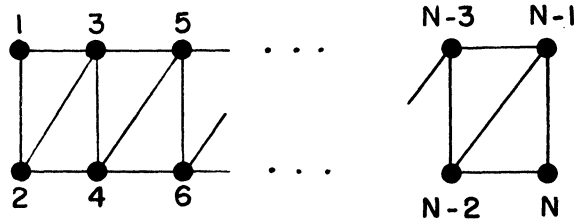


FIGURE 6.

One may successively eliminate "1", "2", $\cdots$, "N" without introducing new

arcs. The back substitution formulae take the form

$$X_n = F_n\,,$$

$$X_{n-1} = E_{n-1}X_n + F_{n-1}\,, \tag{2.5c}$$

$$X_k = E_kX_{k+1} + G_kX_{k+2} + F_k\,.$$

These matrices appear in the "line" methods for the biharmonic equations. Equations (2.5c) are precisely those in use, see [1].

In the remaining examples we suggest the reader construct the form of the matrix from the given graph.

The matrices considered in the next example arise in the S2LOR "two-line" method for Laplace's difference equation as discussed by Varga [2], [5].

Example 3. Two-line Laplace Matrix. In the Laplace difference equation consider the equations associated with the mesh-points on two horizontal lines. Each equation relates X_i to exactly three other unknowns and the graph is just the grid

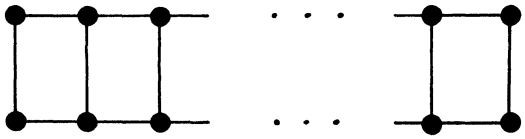


FIGURE 7.

Notice that if the points are renumbered so that the points on the upper line are the consecutive odd-numbered points and those on the lower line are the consecutive even points, the graph is "almost" the graph of (2.5b). This graph is more sparse. An optimal elimination scheme is just the consecutive elimination in the renumbered order, once more giving back substitution formulae of the form (2.5c).

Here we see an interesting fact—the extra "sparseness" of (2.6) over (2.5b) is of little help; i.e. the "missing" arcs must be introduced during the calculation. Another example which illustrates that certain kinds of "sparseness" may be an illusion is the "loop".

Example 4. The "Loop". The graph is a loop, i. e.

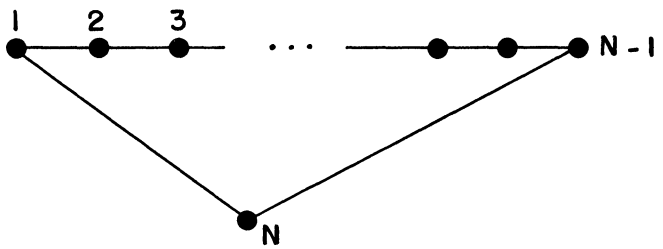


FIGURE 8.

One optimal elimination scheme is to successively eliminate in the natural order obtaining formulae of the form

$$X_N = F_N, \qquad X_{N-1} = E_{N-1}X_N + F_N,$$

(2.7b)

$$X_k = E_kX_{k+1} + G_kX_N + F_k.$$

Compare this to the "triangulated loop" which follows.

Example 5. Triangulated Loop. The graph has the form

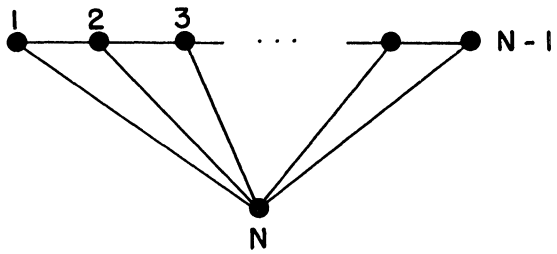


FIGURE 9.

i.e. every point of (2.7a) is also connected to "N". If we undertake elimination in the natural order we obtain formulae of the form (2.7b). We also notice that the "extra" sparseness of Example 4 does not make that problem very much simpler than this one as in that problem the extra "arcs" must be introduced during the calculation.

It is well known that despite its sparse character the five-diagonal matrix for the Laplace difference equation is not easily inverted by elimination. This is easily seen if we consider the associated graph.

Example 6. Laplace's Difference Equation. In this case $G(A)$ is just the grid. The difficulties become apparent from consideration of the following "small" problem:

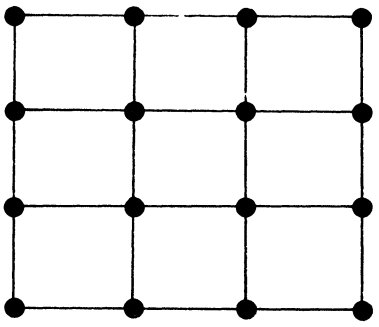


FIGURE 10.

Example 7. The Two-line Biharmonic Matrix. (See [1] or [5] for the exact equations.) If in the biharmonic difference equation we consider the equations asso-

ciated with the mesh-points on two horizontal lines, we obtain quite a complicated graph, the construction of which we leave to the reader. Following this construction, the reader will observe that just as in Example 3, if the points on the upper line are chosen as consecutive odd points and those on the lower line as consecutive even points, the natural ordering gives an optimal elimination scheme. Moreover, as in Example 3, a new arc must be constructed upon the elimination of every odd point.

3. TREES

A connected linear graph G which contains no loops is called a tree.

THEOREM II. *If Gauss elimination is always possible, there exists a one-term successive elimination scheme, i.e. one of the form*

$$X_N = F_N,$$

$$X_k = E_k X_{f(k)} + F_k, \tag{3.1}$$

where $f(k)$ is some single-valued integer-valued function of k, if $G(A)$ is a tree.

PROOF. If $G(A)$ is a tree, consider the effect of eliminating any endpoint. If i is an end-point, the ith equation reads

$$a_{ii}X_i + a_{ij}X_j = Y_i$$

for some unique value of j. Upon elimination of X_i from this equation we find an equation of the form (3.1) with $f(i) = j$. Moreover, the new graph is again a tree. Thus, repeated elimination of end-points gives our result.

We will show how to calculate effectively the coefficients E_k and F_k occurring in the one-term elimination scheme (3.1). This requires some additional facts about the structure of trees. If G is a tree it is possible to define a "distance" $d(i,j)$, between points: the minimal number of arcs in any path between i and j. For example, if $a_{ij} \neq 0$, then $d(i, j) = 1$.

Given a tree G and a chosen "last-point" N', one can define a single-valued, integer-valued function, the successor $S(i)$, for $i \neq N' \cdot S(i)$ satisfies the relations

$$d(i, S(i)) = 1,$$

$$d(i, N') = d(S(i), N') + 1,$$

i.e. the successor of i is that unique neighbor of i which is nearer to N'. *Note:* N' has no successor, and end-points (other than N', if N' is an end-point) are not successors.

The function $S(i)$ completely describes the tree. Given $S(i)$ one can construct the tree and conversely, given the tree G and a choice of N' one can construct $S(i)$.

Definition I. An ordering of the points is called a monotone ordering if

$$i < j \text{ implies } d(i, N') \geqq d(j, N').$$

LEMMA 1. *In a monotone ordering,*

$$N' = N, \qquad i < S(i).$$

It is always possible to achieve a monotone ordering since renumbering the points is permitted (see Section 2).

Let the ordering of the points be a monotone ordering. Let the function $S(i)$ be given. We propose to define recursively coefficients E_k, F_k so that (3.1) holds with $f(i) = S(i)$. We assume, for the moment, that the Gauss elimination procedure does not break down.

Let

$$\begin{aligned} D_k &= a_{kk} + \sum_{S(i)=k} a_{ki} E_i , \\ E_k &= -\frac{a_{k\,S(k)}}{D_k}, \\ M_k &= Y_k - \sum_{S(i)=k} a_{ki} F_i , \\ F_k &= \frac{M_k}{D_k}, \end{aligned} \tag{3.2}$$

where Y_k are the components of y in (2.1).

Notice that $D_1 = a_{11}$ and that at the time we are dealing with point k (since $i < S(i) = k$) the quantities E_i (where $S(i) = k$) have already been computed. But even more important we see that while the formulas (3.2) seem to require the knowledge of the multi-valued inverse of $S(i)$, this can be avoided as follows: If we initially set $D_k = M_k = 0$ for all k and then, immediately after computing E_i and F_i, form $a_{S(i)i}E_i$ and $a_{S(i)i}F_i$ and add these terms to the partial sums $D_{S(i)}$ and $M_{S(i)}$ respectively, we obtain D_k and M_k at the appropriate time without any references to the multi-valued $S^{-1}(k)$. (See the flow diagram given in the appendix.) It is then only a computation to verify that E_k and F_k so computed do indeed give the solution of (2.1) in the form

$$\begin{aligned} x_N &= F_N , \\ x_k &= E_k x_{S(k)} + F_k . \end{aligned}$$

Note: The assumption that the Gauss elimination procedure does not break down is equivalent to $D_k \neq 0$, for all k.

Every systematic Gauss elimination corresponds to a "factoring" of the matrix A into triangular factors. In this case we find

$$A = LR,$$

where

$$\begin{aligned} L_{ij} &= \frac{a_{ij}}{D_j}\delta_{S(j)\,i} + \delta_{ij} , \\ R_{jk} &= D_j\delta_{jk} + a_{jk}\delta_{k\,S(j)} , \end{aligned}$$

and

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Using Lemma 1 one may readily verify that L is a lower triangular matrix and R is an upper triangular matrix.

4. THE VANISHING OF SOME D_k

We now turn to the problem which occurs when $D_k = 0$ for some k. Consider first the simplest tree, the "chain" of examples 1 (2.4a).

Since at the time "k" is eliminated it is an end-point there is no loss in generality in assuming $k = 1$. Then $D_k = a_{11} = 0$, and the first equation reads

$$a_{12}x_2 = y_1 .$$

Moreover, if A is non-singular neither a_{12} or a_{21} vanishes. Thus, one can either

a) solve for x_2, eliminate x_2 from the 3rd equation, solve the reduced "chain" in Fig. 11 and determine x_1 from the second equation (after x_3 is known)

$$a_{21}x_1 = -a_{22}x_2 - a_{23}x_3 + Y_2 ,$$

or

b) solve for x_2, eliminate x_2 from both the first and third equations, setting

$$x_1 = -1/a_{21}a_{22}x_2 + y_{2/a_{21}} - a_{23/a_{21}}x_3$$

and obtaining a formula

$$x_1 = E_1'x_3 + F_1',$$

and continue on to solve the reduced chain. Thus one always has formulae of the form (3.1).

FIGURE 11.

In the case of a general tree, after pivoting on x_{k-1}, "k" is an end-point, the vanishing of D_k implies that the reduced kth equation reads

$$a_{kS(k)}x_{S(k)} = y_k .$$

Hence, one can obtain $x_{S(k)}$ directly. Furthermore, if $D_j \neq 0$ for all j such that $S(j) = S(k)$ one may proceed somewhat as in the case of a chain: obtain E_j and F_j for all j such that $S(j) = S(k)$ and since $x_{S(k)}$ is known, obtain x_k from the formula

$$x_k = E_k'x_{SS(k)} + F_k',$$

which is obtained from the $S(k)$th equation. If also the matrix associated with the tree containing $S[S(k)]$ obtained by removing the connecting arc between $S(k)$ and $S[S(k)]$ is non-singular, one then proceeds to solve that tree.

It is not too difficult to show that if A is non-singular all of the above conditions are met and one can *indeed* define an inductive algorithm for solving (2.1) which can easily be programmed for a digital computer. However, this algorithm would depend on the ability of the machine to determine if a *computed* quantity (i.e. D_k) is precisely zero. Since this is a subtle question, we turn instead to conditions which are sufficient to avoid such pitfalls.

THEOREM III. *If $G(A)$ is a tree and*

$$| a_{ii} | = \epsilon_i + \sum_{i \neq j} | a_{ij} |,$$

where

$$\epsilon_i \geqq \epsilon_0 > 0,$$

then Gauss elimination is always possible, and

$$| E_i | \leqq 1,$$

$$| D_i | \geqq \epsilon_0 + | a_{iS(i)} |.$$

PROOF. Direct inductive calculation using (3.2).

THEOREM IV. *If $G(A)$ is a tree and*

$$| a_{ii} | = \epsilon_i + \sum_{i \neq j} | a_{ji} |,$$

where

$$\epsilon_i \geqq \epsilon_0 > 0,$$

then Gauss elimination if always possible, and

$$| D_i | \geqq \epsilon_0 + | a_{S(i)i} |.$$

PROOF. Direct inductive calculation using (3.2). Theorems III and IV are extensions of results of Richtmyer for "chains", see [4].

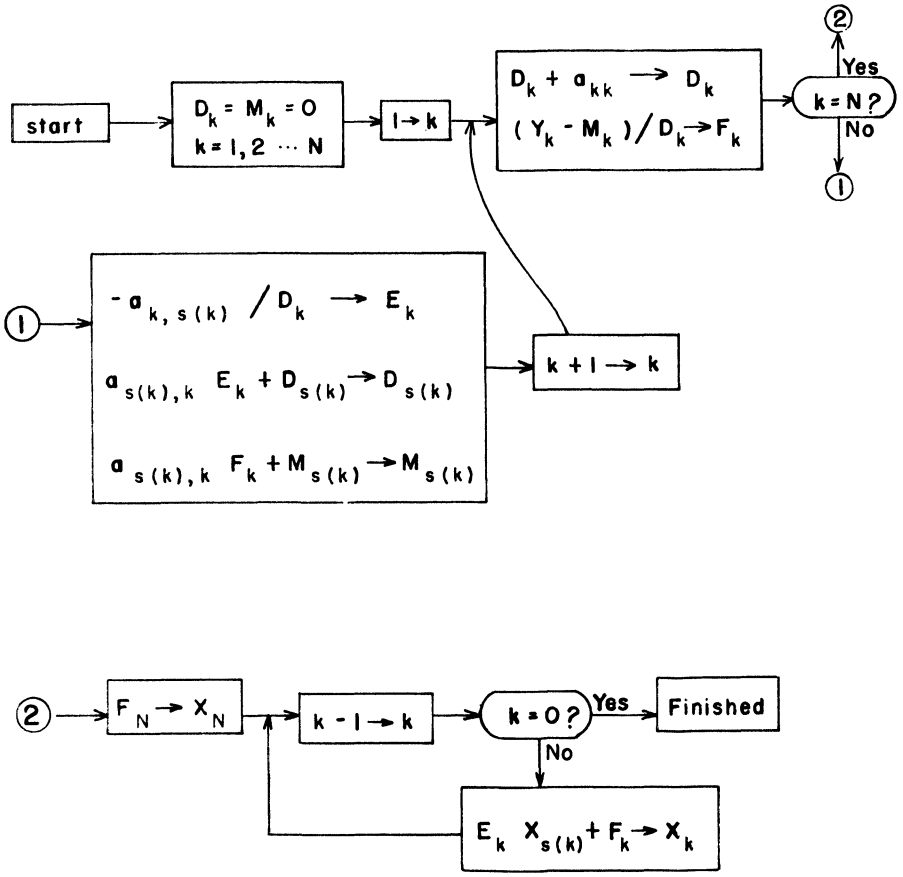
It should be noted that the hypothesis of Theorems III and IV are separately sufficient to guarantee that all the principal determinants of A are non-zero. And, as is well known, if none of the principal determinants of A vanish, Gauss elimination will proceed smoothly, i.e. for all k, $a_{kk}' = D_k \neq 0$.

However, when one is using a fast digital computer, it is not enough to know that a particular procedure gives the correct answer *if one computes with infinite accuracy*. It is desirable to have specific estimates on various quantities. This is particularly true for quantities which will be used as denominators in a division.

Thus, theorems III and IV tell us more than the fact that $D_k \neq 0$—they provide useful specific estimates.

APPENDIX

Flow chart for solving $Ax = y$ where $G(A)$ is a tree and one is assured $D_k \neq 0$ for all k.



REFERENCES

1. Conte, S. D. and Dames, R. T., *An alternating direction method for solving the biharmonic equation*, Math. Tables Aids Comput. Vol. XII, 1958, pp. 198–204.
2. Cuthill, E. H. and Varga, R. S., *A method of normalized block iteration*, J. Assoc. Comput. Mach., Vol. 6, 1959, pp. 236–244.
3. Friedrich, K. *Beiträge zur direkten und indirekten Auflosung der Normalgleichungen unter besonderer Berücksichtigung der geodätischen Netzausgleichung*, Zeitschrift für Vermessungswesen, Vol. 59, 1930, pp. 461–469, pp. 525–539, pp. 671–697.
4. Richtmyer, R. D., *Difference methods for initial-value problems*, Interscience Tracts in Pure and Applied Mathematics, Interscience Publishers, N. Y., 1957.
5. Varga, R. S., *Factorization and normalized iterative methods*, West. Elec. Corp. Report WAPD T-950, April, 1959.