STATE UNIVERSITY OF NEW YORK AT STONY BROOK

COLLEGE OF ENGINEERING

REPORT No. 118

THE GAUSSIAN ELIMINATION AND SPARSE SYSTEMS

by

R. P. Tewarson

AUGUST 16, 1968

REPORT NO. 118

THE GAUSSIAN ELIMINATION AND SPARSE SYSTEMS

by

R. P. Tewarson

AUGUST 16, 1968

This research was supported by the National Aeronautics and Space Administration, Washington, D. C., Grant No. NGR-33-015-013.

THE GAUSSIAN ELIMINATION AND SPARSE SYSTEMS[+]

R. P. Tewarson[*]

Abstract. The problem of minimizing the number of new non-zero elements created during the forward course (no new elements are created in the back substitution) of the Gaussian Elimination for the solution of sparse equations is discussed. A pivot choice that leads to the minimum number of new non-zero elements under a convexity assumption is given. Some other methods for the determination of near optimum pivot choices are also described.

1. Introduction. Let us consider the solution of the system of simultaneous linear equations

$$Ax = b, \tag{1.1}$$

where A is a non-singular matrix of order n, x and b are n element column vectors. An excellent introduction to the various methods of solving (1.1) is given in [1]. For a more detailed treatment the reader is referred to [2]. The case of ill-conditioned systems of the type (1.1) is discussed in [3]. If the system (1.1) possesses any number of the following properties: A is rectangular, rank of A less than n, the right hand side b does not lie in the column space (the range $\mathfrak{R}(A)$) of A; then [4] can be used, e.g., if the generalized least square solution of (1.1) is desired.

In this paper we will assume that A is sparse and non-singular and some sort of row-column scaling has already been done in (1.1). A

[*] State University of New York at Stony Brook, Stony Brook, N.Y.

[+] This paper was presented at the Symposium on Sparse Matrices and their Applications, Sept. 9-10, 1968, sponsored by the International Business Machines, New York, N.Y. This research was supported in part by the National Aeronautics and Space Administration, Washington, D. C. Grant No. NGR-33-015-013.

simple method for scaling is given in [1, pp. 37-46]. More efficient, but complicated, methods are given in [5,6,7]. We will discuss the Gaussian Elimination (GE) for the solution of (1.1) because it is not only simple to implement on a computer but also gives fairly good results for the amount of computational work that is required [2, pp. 244-246].

In general, during the forward course of the Gaussian Elimination new non-zero elements are created. The back substitution part of the GE does not lead to any new non-zero elements. Our problem is to minimize the total number of such non-zero elements created during the entire forward course of the GE. Minimizing the number of non-zero elements created during the forward course of GE leads not only to less roundoff errors (since computations involving zero are exact in most computers) but also saves the computer storage. Because, usually the storage released by the column being eliminated at a particular stage of the GE, is not sufficient to store the additional non-zero elements created in the remaining columns. Minimizing the number of such new non-zero elements decreases the roundoff error not only in the forward course but also in the back substitution part of the GE, since whenever there is a zero element in the column under consideration no operations are performed on the corresponding element of the right hand side.

2. Main Results. Let $A^{(k)}$ be the square sub-matrix obtained from A after the GE has been performed on k - 1 columns, where $k = 1,2,\dots,n$. Clearly $A^{(k)}$ is an mxm matrix, where $m = n - k + 1$ and $A^{(1)} \equiv A$. Let the matrix B_k be obtained by replacing each non-zero element of $A^{(k)}$ by unity. B_k is called the incidence matrix associated with $A^{(k)}$. If U_k denotes a column vector of m ones, then we can define an m element column vector $r^{(k)}$

(called the row count vector) as

$$r^{(k)} = B_k U_k \tag{2.1}$$

and also the m element row vector $c^{(k)}$ (called the column vector) as

$$c^{(k)} = U^T_k B_k, \tag{2.2}$$

where the transpose of U_k is denoted by U^T_k. Note that $r^{(k)}_i$ (the ith element of $r^{(k)}$) gives the total number of non-zero elements in the ith row of $A^{(k)}$; also $c^{(k)}_j$ (the jth element of $c^{(k)}$) gives the total number of non-zero elements in the jth column of $A^{(k)}$. In order to get $A^{(k+1)}$ from $A^{(k)}$, a suitable pivot, say $a^{(k)}_{pq}$, is chosen in $A^{(k)}$ and after transforming all $a^{(k)}_{iq}$, $i \neq p$ to zero by elementary row operations, the pth row and the qth column are dropped. The pivot position (p,q) is saved, which makes it unnecessary to move the pivot to the top left hand corner before performing the GE. The following theorem is useful in minimizing the number of additional non-zero elements created when $A^{(k)}$ is transformed to $A^{(k+1)}$.

Theorem 2.1. The total number of new non-zero elements that are created in the kth step of the GE if $a^{(k)}_{pq}$ is chosen as a pivot is equal to the corresponding element $g^{(k)}_{pq}$ of the matrix $G^{(k)}$, where

$$G^{(k)} = B_k \hat{B}^T_k B_k \quad \text{with} \quad \hat{B}^T_k = U_k U^T_k - B^T_k. \tag{2.3}$$

Proof: The total number of elements of $A^{(k)}$, both zero and non-zero, for which $a^{(k)}_{iq} \neq 0$ and $a^{(k)}_{pj} \neq 0$ is equal to $r^{(k)}_p c^{(k)}_q$. If $\alpha^{(k)}_{pq}$ is the total number of non-zero elements amongst the above, then

$$g^{(k)}_{pq} = r^{(k)}_p c^{(k)}_q - \alpha^{(k)}_{pq}$$

gives the number of new non-zero elements created when $a_{pq}^{(k)}$ is chosen as a pivot. If β_{ij} denotes the i^{th} row j^{th} column element of B_k, then

$$\alpha_{pq}^{(k)} = \sum_{i=1}^{m} \sum_{j=1}^{m} \beta_{pj}\beta_{ij}\beta_{iq},$$

and therefore

$$g_{pq}^{(k)} = r_p^{(k)} c_q^{(k)} - \sum_{i=1}^{m} \sum_{j=1}^{m} \beta_{pj}\beta_{ij}\beta_{iq}.$$

By using (2.1) and (2.2) in the above equation, we can write it in the matrix form

$$\begin{aligned} G^{(k)} &= B_k U_k U_k^T B_k - B_k B_k^T B_k \\ &= B_k (U_k U_k^T - B_k^T) B_k = B_k \hat{B}_k^T B_k. \end{aligned}$$

In view of the above theorem, it is evident that in order to minimize the number of new non-zero elements created at the k^{th} stage of the GE, we must choose for the pivot

$$g_{pq}^{(k)} = \min_{i,j} g_{ij}^{(k)} \text{ for all } i, \; j \text{ with } a_{ij}^{(k)} \neq 0. \qquad (2.4)$$

If we make the tacit assumption that the growth of new non-zero elements is convex, then for each k if we choose the pivot according to (2.4) viz., local minimum, then we will get the global minimum for the entire course of the GE. It is our experience that the pivot choice (2.4) leads to substantial decrease in the number of new non-zero elements.

In the beginning, when k is small, it usually happens that $g_{pq}^{(k)} = 0$. For example, if there are 'singletons' in columns or rows viz., $r_p^{(k)} = 0$ and/or $c_q^{(k)} = 0$. It also happens if for all s with

$a_{ps}^{(k)} \neq 0$, $a_{is}^{(k)} \neq 0$ whenever $a_{iq}^{(k)} \neq 0$. In this case column q is said to have a 'contained intersection' with column s. In any event, $g_{pq}^{(k)} = 0$ implies that the choice of $a_{pq}^{(k)}$ for the pivot at the kth stage of GE will lead to no new non-zero elements. If $g_{pq}^{(k)} = 0$ for more than one row-column pair (p,q), then we choose the pair which has the least number of other such pairs in the row and the column associated with it. This in effect 'saves' the other pivots of zero growth for subsequent stages of the GE.

3. Practical considerations. The computation of $G^{(k)}$ given by (2.3) is not difficult. To save the working storage, each element of B_k can be represented by one bit, instead of a full word in the computer because all the elements of B_k are either zero or one. The computation itself can be simplified by programming techniques and noting that $\hat{B}_k^T$ is obtained from B_k^T by changing each of its unit elements to zero and vice-versa. Computing $G^{(k)}$ at each stage k of the GE and then selecting the pivot leading to the minimum growth, involves a lot of work but, as we pointed out in the previous section, gives excellent results. In any method for minimizing the number of non-zero elements created during the GE, we would like to get the maximum amount of information prior to pivoting with minimum effort and working storage. The choice of a method depends on several factors e.g., programming effort, compile time (especially for one shot use), storage available for the compiled program, solution time, available data and working storage etc. A discussion of these factors and several other methods for minimizing the number of new non-zero elements is given in [8].

If we do not want to compute $G^{(k)}$ at each stage k and are willing to settle for somewhat less of a decrease in the growth of non-zero elements, then we can proceed as follows. Compute $G^{(1)}$ and perform the steps of GE until all available pivots with $g_{pq}^{(1)} = 0$ have been utilized. In sparse matrices there are usually several such elements e.g., due to 'singletons' and 'contained columns'. Let $A^{(\ell)}$ be the matrix left after all available pivots with $g_{pq}^{(1)} = 0$ have been used, then as in [8], we compute the row vector

$$D^{(\ell)} = V^T_\ell B^T_\ell B_\ell, \tag{3.1}$$

and select the columns of $A^{(\ell)}$ in the order of the ascending values of the elements of the vector $D^{(\ell)}$. In each column, the pivot chosen is the one whose corresponding element in the row count vector $r^{(\ell)}$ is minimum. The probability updating given in [9] is used for updating the row count vector after the elimination on the chosen column. This avoids the evaluation of $r^{(k)}$ for each k from the corresponding B_k. We notice that in the process of transforming $A^{(1)}$ to $A^{(\ell)}$ no additional non-zero elements were created. Therefore, we do not need to construct B_ℓ afresh from $A^{(\ell)}$, but obtain it easily from B, by deleting the appropriate rows and columns.

An alternative method to treat the problem discussed in this paper is to first transform A to as much upper triangular form as possible by means of row-column permutations and then choose the pivots on or near the leading diagonal. This method is given in [10]. The tolerance on the pivot size is discussed in [8].

4. Concluding Remarks. When using the methods suggested in this and other papers given in the reference,it should be realized that the structure of A, in general, affects the efficiency of the chosen method. This is

especially true if the non-zero elements of A are not randomly distributed. In view of the above mentioned facts the choice of a method is not easy. Perhaps is is best left to the user provided he is made aware of as many methods as possible and of course their pitfalls.

REFERENCES

1. G. E. Forsythe and C. B. Moler, Computer Solution of Linear Algebraic Systems, Prentice Hall, Englewood Cliffs, N.J.,(1967).

2. J. H. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, London, (1965).

3. R. P. Tewarson and B. Ramnath, Some Comments on the Solution of Linear Equations, College of Engineering Report No. 117, SUNY, Stony Brook, N.Y.

4. R. P. Tewarson, On Computing Generalized Inverses, Computing (1968), to appear.

5. D. R. Fulkerson and P. Wolfe, An Algorithm for Scaling Matrices, SIAM Rev., 4(1962), pp. 142-146.

6. F. L. Baur, Optimally Scaled Matrices, Numer. Math., 5(1963), pp. 73-87.

7. E. E. Osborne, On Pre-Conditioning of Matrices, J. ACM, 7(1960),pp. 338-345.

8. R. P. Tewarson, Solution of A System of Simultaneous Linear Equations with a Sparse Coefficient Matrix by Elimination Methods, Nord. Tids. Inf. Bhld. (BIT), 7(1967), pp. 226-239.

9. R. P. Tewarson, On the Product Form of Inverses of Sparse Matrices, SIAM Rev., 8(1966), pp. 336-342.

10. R. P. Tewarson, Row-column Permutation of Sparse Matrices, Computer Journal, 10(1967), pp. 300-305.