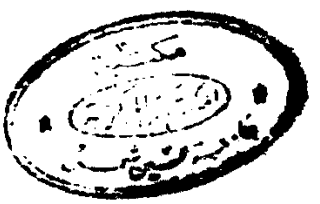
ON THE SOLUTION OF LARGE SPARSE SETS OF LINEAR EQUATIONS

THESIS SUBMITTED IN PARTIAL
FULFILMENT OF THE REQUIREMENTS FOR THE
AWARD OF THE M.Sc. DEGREE

By
ADEL ABD EL-HAKEM ALY EL-KADY

SUBMITTED AT
AIN SHAMS UNIVERSITY
FACULTY OF SCIENCE



JULY 1979

M.Sc. COURSES

STUDIED BY THE AUTHOR (FEB. 1977 - FEB.1978)

(AT AIN SHAMS UNIVERSITY)

(i) Functional analysis

3 hours weekly for two semisters.

(ii) Algebraic and differential topology

4 hours weekly for one semister.

(iii) Ordinary differential equations

3 hours weekly for one semister.

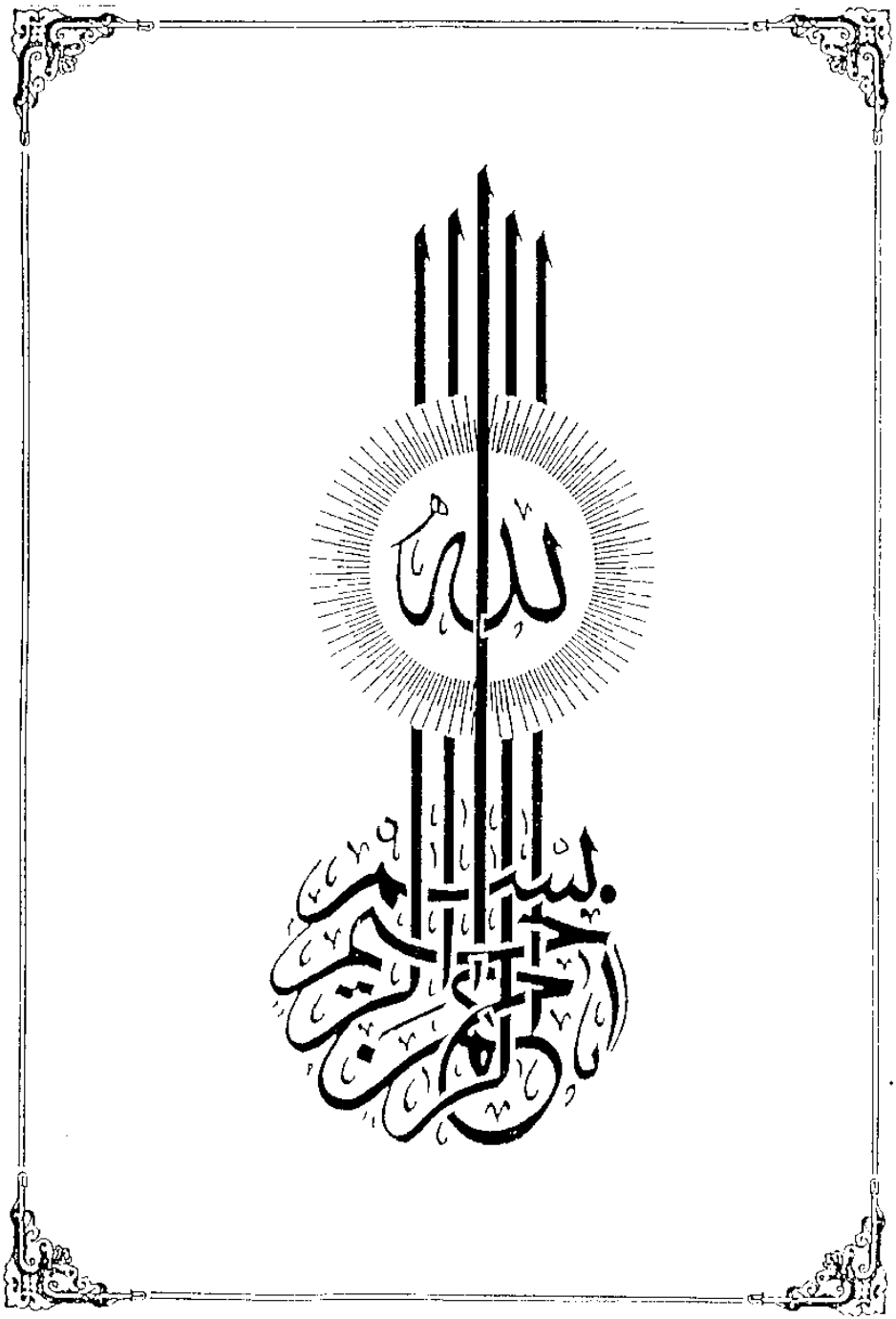
(iv) Theory of functions of matrices

3 hours weekly for one semister.

(v) The algebraic eigenvalue problem

4 hours weekly for one semister.

R. H. Makar



C O N T E N T S

- - -

PREFACE

The thesis deals with direct methods of solving large sparse sets of linear equations. It consists of three chapters and an appendix.

Chapter I involves a general review of the basic Gaussian elimination and LU factorization methods considering both band and general sparse matrices and considering the error analysis of the results. It also involves Stewart's algorithm for modifying the pivot elements as well as the Zollenkopf bi-factorization method.

Chapter II involves three separate topics. §1, is concerned with the partial pivoting strategies for the decomposition of symmetric indefinite matrices. §2, deals with the interesting problem of the capacitance matrix technique. §3, deals with the very interesting algebraic structure due to Carré for the minimal-cost path problem and its solution by a triangularization method.

Chapter III deals with the (important) least squares problem for solving over-determined sparse sets of linear equations. Several methods are explained among which are the Peters-Wilkinson method, the Householder method; the Givens method without square roots and the modified Gram-Schmidt method. A comparison between a number of the methods is also included.

The appendix includes the description of a number of schemes commonly used for storing large sparse matrices.

The thesis has been prepared under the kind supervision of Prof. Dr. Ragy H. Makar, to whom I wish to express my sincerest gratitude and thankfulness.

July 1979.

CHAPTER I

THE GAUSSIAN ELIMINATION AND LU FACTORIZATION METHODS APPLIED TO SPARSE MATRICES AND THE BI-FACTORIZATION METHOD.

§§ 1-4, are a general review of the basic Gaussian elimination and LU factorization methods applied to sparse matrices. They consider, respectively, band and circulant band matrices, general sparse matrices, the use of triangular factors and error analysis of the results. § 5, involves an algorithm due to Stewart, modifying the pivot elements. §6, explains the bi-factorization method due to Zollenkopf.

1. Band and circulant band matrices

In considering direct methods of solving a large sparse set of linear equations $Ax = b$, it is useful to distinguish two main types of matrix, the band matrix and the general sparse matrix. We assume that the matrix A has elements a_{ij} and that it is square and of order n. A band matrix may be defined by the conditions

(1) $a_{ij} = 0$ for $i - j \geqslant s$ or $j - i \geqslant t$.

The total band-width is $k = s + t - 1$, and the band is symmetrically placed about the main diagonal if $s = t$. An example with $s = 2$, $t = 3$, $n = 7$ is the matrix

$$(2) \qquad A = \begin{bmatrix} x & x & x & & & & \\ x & x & x & x & & & \\ & x & x & x & x & & \\ & & x & x & x & x & \\ & & & x & x & x & x \\ & & & & x & x & x \\ & & & & & x & x \end{bmatrix}$$

where the symbol x denotes elements which may be non-zero. In many cases there are zero elements within the band, but it is generally convenient to ignore them in the solution process. An extension of the band form which is not difficult to handle is the "circulant band" matrix, which has additional non-zero elements in some or all of the positions (i, j) for which

$$(3) \qquad n + i - j < s \quad \text{or} \quad n + j - i < t.$$

An example with the same parameters as (2) is

$$(4) \qquad A = \begin{bmatrix} x & x & x & & & & x \\ x & x & x & x & & & \\ & x & x & x & x & & \\ & & x & x & x & x & \\ & & & x & x & x & x \\ x & & & & x & x & x \\ x & x & & & & x & x \end{bmatrix}.$$

The band form is useful only when the band-width k is considerably less than the order n. If the non-zero elements are not restricted to a fairly narrow band, the matrix is of general sparse type, and it requires some what different storage arrangements and solution methods. Both types of matrix may be either symmetric or unsymmetric; in the symmetric case special methods are available if the matrix is also positive definite.

The principal direct methods of solution which will be considered in this chapter are

(a) Gaussian elimination with interchanges.

(b) Triangular factorization without interchanges, which is stable for symmetric positive-definite matrices and for diagonally dominant matrices.

Direct methods of solution are easily adapted to deal with band matrices, if any zeros within the band are ignored. For many purposes it is convenient to split the solution process into two stages.

(i) Triangularization of the matrix, by elimination, or factorization.

(ii) Operations on the right-hand side to obtain the solution.

Alternatively, the elimination or reduction operations can be carried out on b at the same time as on A, leaving

only the back-substitution to the second stage. This has the advantage that we need not preserve the multipliers in Gaussian elimination and so less storage is required. However, if these quantities are retained, they can be used repeatedly to obtain solutions corresponding to any number of right-hand sides, which is useful in certain iterative processes. It should be noted that the triangular factors occupy much less space than the inverse matrix, which is generally full, and so we avoid forming the inverse explicitly unless it is absolutely essential.

A general unsymmetric band matrix may be stored initially in a rectangular array of dimensions n x k. The Gaussian elimination method is usually carried out using column pivoting, i.e. the variables are eliminated in order, and for each elimination the element of maximum modulus in the column is chosen as pivot. Complete pivoting, where the maximum element over the whole unreduced submatrix is used as pivot, is theoretically more satisfactory, because it gives smaller bounds for the perturbations due to rounding errors. But it is more complicated to program, and the theoretical advantage is not usually significant in practice. With column pivoting, the elimination of any variable involves only s rows of the matrix of which one is the pivotal row. It is readily seen from the form (2) that if we

interchange rows to bring the pivotal element onto the diagonal, the band-width above the diagonal may be increased by up to s-1 elements. So in the complete elimination process, the number of additional elements which may be introduced is approximately $(n - t)(s - 1)$. Taking the basic arithmetic operation as one multiplication + one addition, the total number of basic operations required for elimination is approximately

$$\begin{aligned} & n\,st \text{ without interchanges} \\ \text{and up to } & n\,s\,(s + t) \text{ with interchanges.} \end{aligned}$$

The number of operations performed on the vector b to obtain the solution is approximately

$$\begin{aligned} & n(s + t) \text{ without interchanges} \\ \text{and up to } & n(2s + t) \text{ with interchanges.} \end{aligned}$$

For general problems, it is easy to improve on the simple band elimination method by allowing for variable band-width. Each row is associated with markers giving the positions of its first and last non-zero elements, and all elements between them are treated as non-zero. The most economical arrangement is to have the band-width of A increasing with row number, because this minimizes the number of additional elements introduced during the elimination.

For the circulant form (4), the elimination process introduces additional elements in the last t - 1 rows and s - 1 columns of the matrix. If the elements outside the band are used as pivots, the last s + t - 2 columns may be filled up. The number of arithmetic operations is correspondingly increased, but the difference is not large.

Gaussian elimination is essentially equivalent to the factorization of A(after row permutations) into the product LU of a lower triangular matrix with unit diagonal, and an upper triangular matrix. The multipliers of the Gauss method are equal to the subdiagonal elements of L with the sign changed. In the case of symmetric matrices, it is possible to split A symmetrically into triangular factors, $A = LL^T$, where L is lower triangular but without a unit diagonal. This is usually called Cholesky factorization. However, unless A is positive definite, the method may be numerically unstable, because the diagonal elements of L may be small. It can also give imaginary elements in certain columns of L. So for matrices which are symmetric but not positive definite, it is better to use Gaussian elimination with interchanges, although this destroys the advantage of symmetry.

For the symmetric positive-definite case, we have s = t, and only the diagonal and superdiagonal elements of A need

be stored (a total of about n s elements). The triangular factor, L occupies exactly the same space as the original elements, if each row is treated as full. The number of arithmetic operations involved in solving the equations is approximately

$\frac{1}{2}$ n s^2 (+ n square roots) for factorization

and 2 n s for each right-hand side.

Again we can allow for variable band-width, and this is most advantageous if the band-width increases as we go down the matrix.

2. General sparse matrices

We now consider the case where the matrix has no definite band structure. Most techniques for dealing with sparse matrices require each element to be stored with its column number and possibly its row number as well. So the amount of data is two or three times more than the actual elements, and it has to be unpacked for every operation. Because of these complications it is often uneconomical to use sparse matrix techniques unless the density of non-zero elements is less than about one in five.

The direct method of elimination or factorization, will almost invariably introduce some additional elements, and the aim of many algorithms is to re-order the problem so as

to keep the number as small as possible. However, an algorithm is not satisfactory unless it has numerical stability as well, and it is more important to maintain accuracy than to minimize storage and arithmetic operations. If the answers are inaccurate it is no consolation that the method was fast.

The occasions for using the two main methods of § 1 are the same for sparse matrices as for band matrices. For Gaussian elimination, it is useful to carry along with the data a direotory giving the first and last non-zero elements in every row, and to amend this after each elimination. The operation of adding multipl es of the pivotal row to some other row is conveniently done by treating the pivotal row as sparse, and the other row as full between its end-points. It is difficult to estimate initially how much storage and how many arithmetic operations will be needed, particularly when interchanges are allowed. An upper bound can be given, but it is generally too large than what actually is the case.

Large-scale linear programming problems often have sparse matrices, and in solving by the Simplex Method it is usual to obtain the inverse of the submatrix corresponding to the basic variables in an implicit form. This form is obtained by a process equivalent to Jordan elimination. In Jordan's method (ignoring row interchanges) the i th stage involves the

elimination of the i th variable from the preceding as well as the following equations. Thus the i th step is equivalent to premultiplication of the current reduced matrix by a matrix of the form:

$$(5) \qquad J_i = \begin{bmatrix} 1 & & & x & & & \\ & 1 & & x & & & \\ & & 1 & x & & & \\ & & & 1 \; x & & & \\ & & & x & & & \\ & & & x & 1 & & \\ & & & x & & 1 & \\ & & & x & & & 1 \end{bmatrix} .$$

i th column

The subdiagonal elements of the matrices J_i are exactly the Gauss multipliers, and so they have the same amount of sparseness as in the Gauss method. But instead of the U matrix obtained by Gaussian elimination, we have the superdiagonal elements of the J_i, which are essentially the columns of U^{-1}. So the Jordan multipliers are likely to require more storage than the usual triangular factors.