Aplikace matematiky

Jitka Segethová

Elimination on sparse symmetric systems of a special structure

Aplikace matematiky, Vol. 17 (1972), No. 6, 448–460

Persistent URL: http://dml.cz/dmlcz/103436

ELIMINATION ON SPARSE SYMMETRIC SYSTEMS OF A SPECIAL STRUCTURE*

JITKA SEGETHOVÁ

(Received September 30, 1971)

INTRODUCTION

The systems of linear algebraic equations which arise in solving differential equations by finite element or finite difference method usually have matrices which are sparse and have certain regular structure. Solving such a system by elimination, we try to use these properties of the corresponding matrix. It is desirable to find an ordering of rows and columns and an algorithm for solving the system to minimize storage requirements and the number of operations performed during the elimination. In this paper, the problem of finding a permutation of rows and columns and an algorithm for such an ordered system of equations is discussed.

There exist some approaches to this problem in which the sparsity is used to some extent. One of them is a very general approach when the optimal (or nearly optimal) ordering is sought and then the algorithm for solving the ordered system treats the matrix element by element to perform only necessary operations. This case and the case of bandmatrices are compared. In connection with this comparison another approach is introduced in Sec. 2. A type of a matrix more general than a bandmatrix is proposed as well as the means to order the rows and columns to get this form. The examples of matrices reordered by the given procedure and the results are discussed in Sec. 3.

For the sake of brevity, the paper is concerned only with symmetric matrices. Most results may be formulated also for non-symmetric matrices without difficulties.

1. A BRIEF SURVEY OF TECHNIQUES USED

In solving differential equations by the finite element or finite difference method we finally obtain systems of linear algebraic equations, with large sparse matrices

*) This research was sponsored in part by the National Aeronautics and Space Administration under Grant NGL-21-002-008 to the Computer Science Center of the University of Maryland.

of certain regular zero-nonzero structure, which are to be solved. When solving these systems by Gauss elimination we want to use the sparsity as well as the regularity of the zero-nonzero structure of the given matrix. Our aim is to reduce storage requirements and the number of operations performed during the elimination. Fewer operations take less time and result in less roundoff error.

There are two problems connected with this aim: First, to find an elimination procedure operating only (or almost only) on such elements of the given matrix (with a given ordering of rows and columns) which are involved in the course of elimination. Second, to find an ordering of rows and columns for the given matrix so that the employment of some of elimination procedures is as efficient as possible.

The work necessary for finding a proper elimination procedure and a proper ordering of rows and columns would be superfluous when we are to solve only a single system with reasonable storage requirements. The approaches surveyed in this section and that suggested in the following section show to be very efficient when we – as usual – solve many systems with matrices of a similar structure which differ from each other only by some parameter, e.g. the mesh size h. Then it is desirable to find a general rule for ordering such classes of matrices (see an example in Sec. 3).

In order to preserve a general point of view, we suppose in this paper that the elements of the matrix of a system are given. However, there are various procedures for solving large sparse systems that operate simultaneously with the evaluation of the elements of the matrix (cf. e.g. [6]).

Let us discuss the two problems mentioned above.

(I) Let us have the system

$$Ax = y$$

of linear algebraic equations. Let $A = \{a_{ij}\}_{i,j=1}^{n}$ be a real symmetric non-singular sparse matrix of order n. The problem is to find an algorithm allowing us to operate only (or almost only) on such elements of A which are involved and changed in the elimination process itself.

(Ia) One of the simplest ways to solve this problem is to consider the matrix A as a bandmatrix. The structure of the matrix as well as the number of operations are given by two parameters: the order n of the matrix and the width $2m + 1$ of the band where

$$m = \max_{(i,j)\in S} (j - i)\,, \quad S = \{(i,j) \mid 1 \leqq i < j \leqq n\,, a_{ij} \neq 0\}.$$

In connection with the elimination, the bandmatrices offer well-known important advantages, e.g. very simple data handling. In particular, a change of the parameter h results only in a change of the parameters n, m.

(Ib) Given a matrix of an arbitrary zero-nonzero structure, it is possible to employ an algorithm operating only on those elements which are actually involved in the elimination process (see e.g. Gustavson [5]). Therefore, the matrix must be treated element by element and the structure of such a matrix as well as the number of operations are given by the positions of every nonzero element. Apparently the use of this algorithm can entail a considerable amount of work. This approach is very general (in the sequel this algorithm for solving the system by elimination is called "general algorithm") and it may be particularly suitable for those matrices whose nonzero elements occur in no regular structure.

Another approach is introduced in the following section.

(II) In order to make use of the advantages of special elimination procedures, the matrix A cannot be usually processed in an arbitrary form. A permutation that transforms the original ordering of rows and columns with regard to the minimization of the number of operations and the number of nonzero elements created (i.e. storage requirements) is sought. An elimination procedure is then applied to this permuted matrix B.

All the practicable methods for finding such a permutation perform the minimization only to a certain extent. In particular, they usually use only local criteria that need not yield the global minimum. Such resulting orderings of rows and columns are called "nearly optimal".

Let us keep the notations of paragraph (I). Moreover, let us suppose that the elimination can be performed with an arbitrary ordering of rows and columns, i.e. with any matrix $B = PAP^T$ where P is a permutation matrix.*) This assumption is fulfilled e.g. by a positive definite matrix.

Let us recall several concepts of the graph theory. Let us denote by $G = (X, E)$ a graph of the matrix A where $X = \{x_i\}_{i=1}^n$ is the set of the nodes of the graph (x_i corresponds to the ith row of A) and E is the set of the edges of the graph (the edge $\{x_i, x_j\}$ belongs to $E = E(X)$ if and only if $a_{ij} \neq 0$, $i < j$). Let G be connected (i.e. A is irreducible). A graph is said to be ordered if a permutation

$$p = \begin{pmatrix} 1, 2, \dots, n \\ i_1, i_2, \dots, i_n \end{pmatrix}$$

of the set $\{1, 2, \dots, n\}$ is given. The permutation p defines an ordering $\{x_{i_k}\}_{k=1}^n$ of the elements $x_i \in X$.

Further let us write

$$N(x) = \{y \in X \mid \{x, y\} \in E\} \cup \{x\},$$

$$D(x) = \{\{y, z\} \mid y, z \in N(x), y \neq z, y \notin N(z)\}.$$

Obviously $N(x)$ is the set of neighbors of the node x.

*) This is equivalent to the following assumption: Let B_k ($k = 1, \dots, n - 1$) be the principal submatrices of B consisting of the first k rows and columns of the matrix B and let $\det B_k \neq 0$; $k = 1, \dots, n - 1$ for any $B = PAP^T$.

Let $y \in X$. We call $G_y = (X - \{y\}, E(X - \{y\}) \cup D(y))$ the graph obtained by eliminating the node y from G. It is well known that the graph $\tilde{G}_{x_i}$, corresponding to the matrix arising from A by eliminating x_i using the ith equation and omitting the ith row and ith column, is a partial graph of G_{x_i}.

It may be easily seen that new nonzero elements may be created during the elimination. Apparently the number of nonzero elements created in this way depends on the ordering of the rows and columns of A, i.e. on the permutation p. This fact and the suggestion that graph theory might be a suitable way to study the elimination is due to Parter [7].

Let us denote the number of the elements of the sets $N(x_i)$, $D(x_i)$ in the graph G by $b_i(G)$, $d_i(G)$ respectively. The number $b_i(G)$ is said to be the degree of the node x_i in the graph G, the number $d_i(G)$ is usually said to be the "fill in". Let G_k be the graph obtained by eliminating the node x_{i_k} from the graph G_{k-1}, $G_0 = G$. Then the number of operations in the forward course of Gauss elimination is equal to

$$\tfrac{1}{2}\sum_{k=1}^{n-1} b_{i_k}(G_{k-1})\,(b_{i_k}(G_{k-1}) + 5) + n$$

multiplications (or divisions) and

$$\tfrac{1}{2}\sum_{k=1}^{n-1} b_{i_k}(G_{k-1})\,(b_{i_k}(G_{k-1}) + 3)$$

additions, and in the backward course $\sum_{k=1}^{n-1} b_{i_k}(G_{k-1})$ multiplications and the same number of additions. (We suppose that a row is divided by its diagonal element in the forward course.)

(IIa) The algorithms used for finding an optimal ordering of rows and columns (i.e. an optimal permutation p) of a bandmatrix are oriented to the minimization of the bandwidth m. An efficient and simple algorithm yielding a nearly optimal ordering is discussed by Cuthill and McKee [4]. For further algorithms, see also Alway and Martin [1], Rosen [9], and others.

(IIb) As a rule, the minimization of $\sum_{k=1}^{n-1} d_{i_k}(G_{k-1})$ serves as a criterion for finding an optimal permutation p. Proceeding in this way, we minimize the storage requirements and the number of backsolving operations in elimination. In general, the number of operations in the forward course is not minimized. Every algorithm yielding the optimal ordering is, naturally, very complicated (see e.g. [8]).

Two algorithms are used in practice and recommended by many authors: the algorithm of the minimal degree (minimizing $b_{i_k}(G_{k-1})$'s) and the algorithm of the minimal fill-in (minimizing $d_{i_k}(G_{k-1})$'s). The essential advantages of these algorithms are their simplicity and small number of operations. The algorithm of the minimal degree needs less operations than the latter but need not yield the optimal ordering even in the case when there exists an ordering for which $\sum_{k=1}^{n-1} d_{i_k}(G_{k-1}) = 0$. In general, both these algorithms give only a nearly optimal permutation p. For details, see [8], [11], [12].

2. PIPEMATRICES

Let us try another approach. Let us consider a symmetric matrix A of this special zero-nonzero structure:

$$A = \left[\begin{array}{cccccccccc}
\times & & & \times & & \times & & & \times & \\
 & \times & & \times & & \times & \times & & \times & \\
 & & \times & \times & & \times & \times & & \times & \\
\times & \times & \times & \times & & \times & \times & & \times & \\
 & & & & \times & \times & \times & & \times & \\
\times & \times & \times & \times & \times & \times & \times & & \times & \times \\
 & \times & \times & \times & \times & \times & \times & & \times & \times \\
 & & & & & & & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\
 & & & & & \times & \times & \times & \times & \times
\end{array}\right].$$

Only the nonzero elements are indicated (by the ×'s). Let $A^{(k)} = \{a_{ij}^{(k)}\}_{i,j=1}^{n}$ be the matrix arising from an $n \times n$ symmetric matrix A after the first k steps of the Gauss elimination. Then $a_{ij}^{(k)} = a_{ji}^{(k)}$ for $i, j = k + 1, \ldots, n$; $k = 1, \ldots, n - 1$. Thus in the elimination process applied to any symmetric system it is sufficient to operate only on the elements on and above the diagonal.

In the elimination process applied to the symmetric system with the matrix of the structure considered, only the nonzero elements on and above the diagonal must be operated on. No other element in this part of the matrix is changed by the elimination process. With respect to the structure of the matrix considered, no nonzero element is created during the elimination, i.e. fill in equals zero.

For our further considerations it is useful to introduce the following concepts:

Definition 2.1. *Let A be an $n \times n$ symmetric matrix, let $a_{ii} \neq 0$; $i = 1, \ldots, n$. Let us put*

$$m_l = \min_{a_{il} \neq 0} i\,; \quad l = 1, \ldots, n\,.$$

Then the vectors $(a_{m_l,l}, a_{m_l+1,l}, \ldots, a_{l,l})$ are said to be pipes of the matrix A.

Definition 2.2. *An $n \times n$ symmetric matrix A is said to be a pipematrix with full pipes (or to be in a pipematrix form with full pipes) if the following condition is satisfied for all $1 \leqq i < j \leqq n$:*

$$(1) \qquad a_{ij} \neq 0 \Rightarrow a_{kj} \neq 0 \quad \text{for} \quad k = i + 1, \ldots, j\,.$$

Remark 2.1. From Definitions 2.1 and 2.2 it follows that the pipes of the pipematrix with full pipes consist of only nonzero elements. The example of a pipematrix with full pipes was considered in the beginning of this section.

Remark 2.2. Let A be a symmetric matrix the diagonal elements of which are nonzero. Let the numbers m_l; $l = 1, \ldots, n$ of Definition 2.1 be given. Applying the Gauss elimination to such a matrix A, it is sufficient to operate only on the pipes of A. For this purpose, zero-nonzero structure of A is sufficiently described by the numbers m_l. The matrix A considered in this way will be called a pipematrix.

In general, the condition (1) need not be fulfilled for a pipematrix. If some zeros appear in the pipes nonzero elements may be created during the elimination process. The algorithm for solving linear algebraic system with pipematrices treats only the elements in the pipes and operates on all of them.

Remark 2.3. It is more general to consider an arbitrary matrix as a pipematrix than as a bandmatrix. Let us have an $n \times n$ symmetric matrix A whose structure is given by the numbers m_l; $l = 1, \ldots, n$. Then considering it as a pipematrix, we operate on $P = \sum_{l=1}^{n} (l - m_l) + n$ elements during the Gauss elimination while considering it as a bandmatrix, we operate in the best case on $Q = (m + 1)(n - m) + \frac{1}{2}m(m + 1)$ elements where $2m + 1$ is the width of the band (see (Ia) of Sec. 1). For the pipematrix A, $m = \max_l (l - m_l)$ and

$$P = \sum_{l=1}^{m} (l - m_l) + \sum_{l=m+1}^{n} (l - m_l) \leqq$$
$$\leqq \tfrac{1}{2}m(m - 1) + m(n - m) + n = Q$$

so that considering an arbitrary symmetric matrix as a pipematrix, we generally perform fewer or at most the same number of operations as if we consider it as a bandmatrix.

Remark 2.4. It is less general to treat a matrix as a pipematrix than to treat it by the general algorithm, element by element. The zero-nonzero structure of a pipematrix is described by the positions of the whole pipes (i.e. the vectors), which makes this algorithm as well as its use (handling data) simpler than the general one.

Now the problem arises to find a permutation p (or, equivalently, a permutation matrix P or an ordering of rows and columns) by which the matrix A given would be reordered into the form of a pipematrix whose pipes include as few zero elements as possible.

Let a matrix A satisfy all the assumptions of the previous section (i.e. it is a symmetric irreducible matrix which can be eliminated with an arbitrary order of rows and columns). Suppose A can be permuted into the pipematrix form with full pipes. The following procedure gives the permutation by which the matrix is reordered into the form with full pipes.

Procedure 1

Step 1

1. Set $S_1^1 = \{x_i \mid d_i(G_0) = 0\}$.

(According to the assumptions about the matrix A there exists at least one subscript i for which $d_i(G_0) = 0$.)

2. Select $x_{i_1} \in S_1^1$ arbitrarily.

3. $S^1 = \{x_m \mid x_m \in N(x_{i_1})\}$. Eliminate the node x_{i_1} from the graph G_0 (i.e., in the permutation p being constructed this i_1 corresponds to the integer 1).

Step $k = 2, \ldots, n - 1$

1. Set $R^k = \{x_j \mid x_j \in N(x_m)$ for all $x_m \in S^{k-1}\}$ ($R^k \neq \emptyset$ because $S^{k-1} \subset R^k$).

2. $S_1^k = \{x_j \mid x_j \in R^k, d_j(G_{k-1}) = 0\}$. If $S_1^k = \emptyset$ set $S_1^{k-1} = S_1^{k-1} - \{x_{i_{k-1}}\}$ and repeat the step $k - 1$.

3. If $S_1^k \neq \emptyset$ then select $x_{i_k} \in S_1^k$ arbitrarily.

4. Set $S^k = \{x_m \mid x_m \in N(x_{i_k})\}$. Eliminate the node x_{i_k} from the graph G_{k-1} (i.e., in the permutation p being constructed this i_k corresponds to the integer k).

Remark 2.5. In practice it may be advantageous to use the following additional criteria in part 2 of step 1 and part 3 of step $k = 2, \ldots, n - 1$.

We substitute

2. Set $S_2^1 = \{x_i \mid x_i \in S_1^1, b_i(G_0) = \min_{x_q \in S_1{}^1} b_q(G_0)\}$ and select $x_{i_1} \in S_2^1$ arbitrarily.

for part 2 of step 1 and

3. Set $S_2^k = \{x_j \mid x_j \in S_1^k, x_j \in S^{k-\pi(x_j)}, \pi(x_j) = \max_{x_q \in S_1{}^k} \pi(x_q)\}$ where $\pi(x_q)$ is an integer uniquely determined by the condition $x_q \in S^{k-\pi(x_q)}$, $x_q \notin S^{k-\pi(x_q)-1}$. Set $S_3^k = \{x_j \mid x_j \in S_2^k, b_j(G_{k-1}) = \min_{x_q \in S_2{}^k} b_q(G_{k-1})\}$. Select $x_{i_k} \in S_3^k$ arbitrarily.

for part 3 of step $k = 2, \ldots, n - 1$. Using Procedure 1 modified in the above way, we may require less time to complete the process.

If A cannot be permuted into the pipematrix form with full pipes we may employ Procedure 2 obtained from Procedure 1 by substitution of

1. Set $S_1^1 = \{x_i \mid d_i(G_0) = \min_{q=1,\ldots,n} d_q(G_0)\}$.

for part 1 of step 1 and substitution of

2. Set $S_1^k = \{x_j \mid x_j \in R^k, d_j(G_{k-1}) = \min_{x_q \in R^k} d_q(G_{k-1})\}$.

for part 2 of step $k = 2, \ldots, n - 1$. The set S_1^k in part 2 of step $k = 2, \ldots, n - 1$ is never empty.

Remark 2.6. Using Procedure 2 with the additional criteria given in Remark 2.5, we may obtain a better ordering.

The procedure shown above have the following important properties.

Theorem 2.1. *Let A be a symmetric matrix. If there exists a permutation matrix P such that PAP^T is a pipematrix with full pipes then Procedure* 1 *gives the corresponding permutation p.*

Proof. Procedure 1 gives the possibility to try in each step k the finite number of all the orderings $\{x_{j_1}, \ldots, x_{j_k}\}$ of nodes such that $x_{j_l} \in S_1^l$ for $l = 1, \ldots, k$. According to the assumptions, there exists at least one ordering $\{x_{i_1}, \ldots, x_{i_n}\}$ where $x_{i_l} \in S_1^l$ for $l = 1, \ldots, n$. Obviously the ordering $\{x_{i_1}, \ldots, x_{i_k}\}$ is one of the orderings $\{x_{j_1}, \ldots \ldots, x_{j_k}\}$.

Theorem 2.2. *Let A be a symmetric irreducible matrix. Let it be possible to perform the elimination on any matrix QAQ^T where Q is an arbitrary permutation matrix. Let P be a permutation matrix given by either Procedure* 1 *or* 2. *Let us solve the system of linear algebraic equations with the matrix PAP^T by the algorithm for pipematrices. Then the same number of operations is required as when using the general algorithm (e.g.* [5]*) where the matrix is treated element by element.*

Proof. The pipes of the matrix ordered by the permutation p given by Procedure 1 or 2 do not involve zeros which are not changed by the elimination process. This follows from part 1 of step $k = 2, \ldots, n - 1$ where the set R^k is constructed.

Remark 2.7. The only time-consuming part of Procedure 1 may be part 2 of step $k = 2, \ldots, n - 1$ where it can theoretically happen that we go back to the very beginning of the procedure several times. However, in the computations performed, where a matrix was successfully permuted into the form with full pipes, only Procedure 2 was used and, therefore, this problem did not arise.

Remark 2.8. If Procedure 2 is applied to the matrices which cannot be permuted into the form with full pipes or if we do not require the matrix with full pipes as a result, it may give an acceptable ordering in some cases as will be shown on examples in Sec. 3.

Remark 2.9. Procedure 2 is not very time-consuming because only a certain set of nodes is tested in each step.

3. EXAMPLES

Procedure 2 was applied to several types of matrices. The results have been very interesting. Let us show two typical examples where the procedure yields a satisfactory ordering. In Procedure 2, the additional criteria given in Remark 2.5 were used in part 2 of step 1 and part 3 of step $k = 2, \ldots, n - 1$. When "Select $x_{i_k} \in S_2^1$ or S_3^k arbitrarily" is recommended in these parts, the node with the lowest subscript in the original ordering was selected.

Example 3.1. (The example of a matrix that can be reordered into the pipematrix with full pipes.) The matrix which arises from mesh refinement in one dimension when the solution of a certain boundary value problem is approximated by a kind of hill functions (see: Babuška [7], [8]) has a graph of the type in Fig. 3.1.

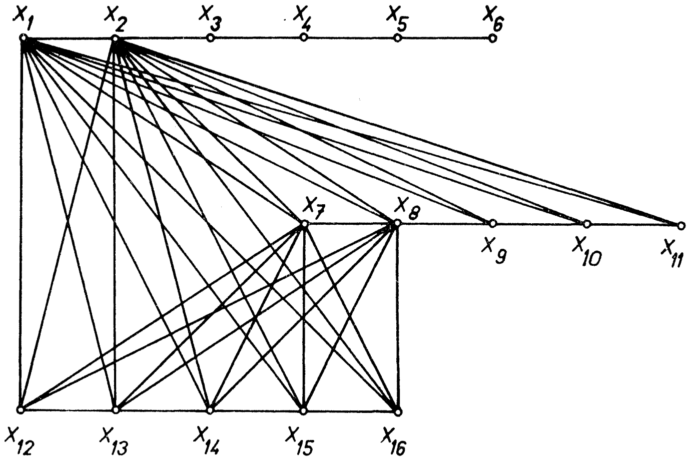

Fig. 3.1.

The corresponding matrix is given in Fig. 3.2. Procedure 2 gives the permutation p defining the ordering $\{x_{11}, x_{10}, x_9, x_{12}, x_{13}, x_{14}, x_1, x_8, x_7, x_{15}, x_{16}, x_2, x_3, x_4, x_5, x_6\}$ of the elements $x_i \in X$. The permuted matrix $B = PAP^T$ is shown in Fig. 3.3. Eliminating the system with the matrix B (by the elimination procedure for pipe-matrices) we get zero fill in.

$$A = \begin{bmatrix}
\times & \times & & & & & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & & & & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\
& \times & \times & \times & & & & & & & & & & & & \\
& & \times & \times & \times & & & & & & & & & & & \\
& & & \times & \times & \times & & & & & & & & & & \\
& & & & \times & \times & & & & & & & & & & \\
\times & \times & & & & & \times & \times & & & & \times & \times & \times & \times & \times \\
\times & \times & & & & & \times & \times & \times & & & \times & \times & \times & \times & \times \\
\times & \times & & & & & & \times & \times & \times & & & & & & \\
\times & \times & & & & & & & \times & \times & \times & & & & & \\
\times & \times & & & & & & & & \times & \times & & & & & \\
\times & \times & & & & & \times & \times & & & & \times & \times & & & \\
\times & \times & & & & & \times & \times & & & & \times & \times & \times & & \\
\times & \times & & & & & \times & \times & & & & & \times & \times & \times & \\
\times & \times & & & & & \times & \times & & & & & & \times & \times & \times \\
\times & \times & & & & & \times & \times & & & & & & & \times & \times
\end{bmatrix}$$

Fig. 3.2.

$$B = \begin{bmatrix}
\times & \times & & & & & \times & & & & & \times & & & & \\
\times & \times & \times & & & & \times & & & & & \times & & & & \\
& \times & \times & & & & \times & \times & & & & \times & & & & \\
& & & \times & \times & & \times & \times & \times & & & \times & & & & \\
& & & \times & \times & \times & \times & \times & \times & & & \times & & & & \\
& & & & \times & \times & \times & \times & \times & \times & & \times & & & & \\
\times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & & \\
& & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & & \\
& & & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & & \\
& & & & & \times & \times & \times & \times & \times & \times & \times & & & & \\
& & & & & & \times & \times & \times & \times & \times & \times & & & & \\
\times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & \\
& & & & & & & & & & & \times & \times & \times & & \\
& & & & & & & & & & & & \times & \times & \times & \\
& & & & & & & & & & & & & \times & \times & \times \\
& & & & & & & & & & & & & & \times & \times
\end{bmatrix}$$

Fig. 3.3.

We can obtain several orderings by which the matrix is reordered into the pipe-matrix form with full pipes by starting with various original orderings. Moreover, these particular examples show certain general features allowing us to construct permutations p for various problems of this kind not using the procedure of the previous section any more.

Example 3.2. Suppose we have a matrix with graph G in Fig. 3.4 (which arises from using the five point difference formula in approximating certain boundary value problems on an L-shaped domain $D = D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5$). If we number the nodes in the so-called natural ordering (see Fig. 3.4) we obtain a band-matrix with $m = m_4$.

Let us number the nodes by Procedure 2. With this ordering, the number of operations performed during the elimination in $D_1 \cup D_2 \cup D_3$ is less than or equal to that for the bandmatrix with $m = \min(m_1, m_2 + m_3)$ and in $D_4 \cup D_5$ with $m = \min(m_3, m_4)$. This can result in a substantial time and operations reduction. Similar results are valid for various types of L-shaped domains which differ only in the ratios of m_1, m_2, m_3, m_4. Moreover, having matrices with the graph in Fig. 3.4 where only h (the mesh size) is different, it is not necessary to seek an ordering for every matrix because the procedure gives a general rule for orderings in the corresponding domains D_i with various magnitudes of h.

For example, let us consider the L-shaped domain D above. Then for every sub-domain there is a formula describing the relation between the coordinates of the node (x, y) in the net (it is also the node in the graph of the matrix) and the number

$Q(x, y)$ that is assigned to this node by the permutation p found. For example in D_1

$$Q(x, y) = \tfrac{1}{2}(x^2h^{-2} + y^2h^{-2} + xh^{-1} + yh^{-1}) + xyh^{-2} + xh^{-1} + \\ + (y - x)\, h^{-1}\, \delta_{0x} + \delta_{0y}$$

(where $\delta_{0a} = 1$ if $a = 0$ and $\delta_{0a} = 0$ if $a \neq 0$) is valid. Similar formulae are valid for the other parts of the domain.

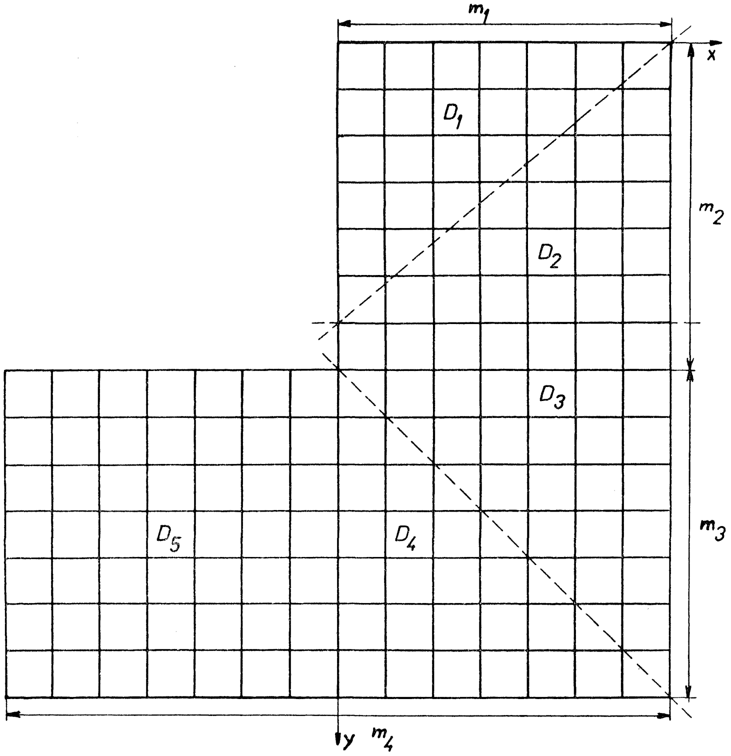

Fig. 3.4.

Let us have a matrix with graph G in Fig. 3.4. Let us permute the iows and columns using the permutation p found by Procedure 2 and let us solve the system by the algorithm for pipematrices. Also let us permute the rows and columns in the ordering given by the minimal degree algorithm and solve the system by the general algorithm. Then we have the following results ($m_1 = 5$, $m_2 = 5$, $m_3 = 5$, $m_4 = 12$, $n = 80$):

1. Ordered in the so-called "natural ordering" we get the bandmatrix with $m = 12$. Solved by the algorithm for bandmatrices $80 \times 13 = 1040$ elements are treated.

2. Ordered by Procedure 2 and solved by the algorithm for pipematrices, fill is 230.

3. Ordered in the ordering given by the minimal degree algorithm and solved by the general algorithm, fill is 188.

In comparison with 1, the fills in 2 and 3 are approximately the same, but in case 3 we require the general algorithm for solving the system, which means a disadvantage mentioned above.

The technique when the matrix is permuted by Procedure 1 or 2 into the pipematrix form and the system is solved in the corresponding way is advantageous if applied to certain matrices (as in Examples 3.1, 3.2) in comparison with both of the other approaches mentioned in Sec. 1.

The examples show that the algorithm for solving the system with a pipematrix and, in particular, its operation (handling input data) aie simpler than the general algorithm and its application. On the other hand, a pipematrix form is more general than a bandmatrix form, which, in turn, may be an advantage; particularly if the ordering has been found for a set of matrices with the same zero-nonzero structure as in the Example 3.2.

References

[1] *G. G. Alway, D. W. Martin:* An algorithm for reducing the bandwidth of the matrix of symmetric configuration. Computer J. *8* (1965—66), 264—272.

[2] *I. Babuška:* Finite element method for domains with corners. Computing *5* (1970), 264—273.

[3] *I. Babuška:* The rate of convergence for the finite element method. SIAM J. Numer. Anal. *8* (1971), 304—315.

[4] *E. Cuthill, J. McKee:* Reducing the bandwidth of sparse symmetric matrices. 1969 Summer National ACM Meeting Proceedings.

[5] *F. G. Gustavson, W. M. Liniger, R. A. Willoughby*: Symbolic generation of an optimal Crout algorithm for sparse systems of linear equations. Proc. of the Symposium on Sparse Matr. and Their Appl., IBM Watson Res. Center, 1968.

[6] *B. M. Irons*: A frontal solution program for finite element analysis. Internat. J. for Num. Meth. in Engineering *2* (1970), 5—32.

[7] *S. Parter:* The use of linear graphs in Gauss elimination. SIAM Rev. *3* (1961), 119—130.

[8] *D. J. Rose:* Symmetric elimination on sparse positive definite systems and the potential flow network problem. PhD thesis, Harvard University, Cambridge, Mass., 1970.

[9] *R. Rosen:* Matrix bandwidth minimization. ACM National Conference Proc., Las Vegas, Nevada, 1968.

[10] *J. Segethová:* Elimination for sparse symmetric systems of a special structure. Tech. Rep. 70—121, Comp. Sci. Center, University of Maryland, 1970.

[11] *W. R. Spillers, N. Hickerson:* Optimal elimination for sparse symmetric systems as a graph problem. Quart. Appl. Math. *26* (1968), 425—432.

[12] *R. P. Tewarson:* The Gaussian elimination and sparse systems. Proc. of the Symposium on Sparse Matr. and Their Appl., IBM Watson Res. Center, 1968.

Souhrn

ELIMINACE ŘÍDKÝCH SYMETRICKÝCH SOUSTAV SPECIÁLNÍ STRUKTURY

JITKA SEGETHOVÁ

Matice soustav lineárních algebraických rovnic, které vznikají při řešení diferenciálních rovnic metodou konečných prvků nebo konečných diferencí, jsou zpravidla řídké a mají jistou pravidelnou strukturu rozložení nenulových prvků. Při řešení takové soustavy zvolenou modifikací eliminace se snažíme nalézt vhodné pořadí řádků a sloupců matice, abychom využili vlastností dané matice k minimalizaci požadavků na paměť a počtu operací.

V článku je podán stručný přehled technik, užívaných pro eliminaci soustav s řídkými maticemi, a navržen další postup, který je obecnější než eliminace matice v pásovém tvaru, nezachází však s maticí prvek po prvku jako nejobecnější možný algoritmus. Je uveden postup pro nalezení „píšťalového tvaru“ matice, tj. pořadí řádků a sloupců, vhodného pro tuto modifikaci eliminace. Článek je doplněn numerickými příklady.

Author's address: Dr. *Jitka Segethová*, Matematicko-fysikální fakulta KU, Malostranské nám. 25, Praha 1.