JOURNAL OF COMPUTATIONAL PHYSICS 44, 134–155 (1981)

Guidelines for the Usage of Incomplete Decompositions in Solving Sets of Linear Equations as They Occur in Practical Problems

J. A. MEIJERINK

Koninklijke/Shell, Exploratie en Produktie Laboratorium, Rijswijk, The Netherlands

AND

H. A. VAN DER VORST*

Academisch Computercentrum Rijksuniversiteit Utrecht, The Netherlands

Received November 12, 1979; revised April 28, 1981

This paper presents incomplete decompositions for various types of matrices as they occur in the implicit discretisation of practical problems. A review is given of methods for the usual five-point discretisation of a self-adjoint elliptic second-order partial differential equation in two dimensions on a square. The matrices which occur in this type of problem are symmetric M-matrices of very regular structure. The convergence behaviour of the different decompositions for this case is demonstrated by numerical experiments. The paper also gives decompositions for the following type of matrices: (i) Symmetric M-matrices of a different structure. (ii) Symmetric positive definite matrices. (iii) Non-symmetric matrices.

1. INTRODUCTION

In Ref. [8] the idea of constructing an approximation K for arbitrary sparse M-matrices[1] A by incomplete LU factorisation was introduced. The matrix K has the property that any system $Kv = w$ can be solved much easier than $Ax = b$. This led to the iterative method $x^{n+1} = x^n + K^{-1}(b - Ax^n)$ for the solution of $Ax = b$. Also it has been proven that the splitting $A = K - R$ is regular[2], which implies that the iterative method always converges. For symmetric matrices A, these decompositions were used as preconditionings for the conjugate gradient algorithm. In the examples given in Ref. [8] matrices were considered arising from five-point discretisation of a self-adjoint elliptic partial differential equation on a rectangular region. Only two different incomplete decompositions were demonstrated.

In this paper we present a more systematic review of the possible incomplete decompositions for that problem (Section 2.1). In addition, we shall discuss incom-

* Supported in part by the European Research Office, London, through Grant DA ERO-75-G-084.

[1] $A = (a_{ij})$ is an M-matrix if $a_{ij} \leqslant 0$ for $i \neq j$, A is non singular and $A^{-1} \geqslant 0$.

[2] The splitting $A = M - N$ is a regular splitting if M is non singular, $M^{-1} \geqslant 0$ and $N \geqslant 0$.

0021-9991/81/110134–22$02.00/0

plete decompositions for other types of matrices, e.g., M-matrices arising from problems with periodic boundary conditions (Section 2.2) and M-matrices with a more arbitrary structure (Sections 2.4 and 2.5). For this purpose we need an extension of the definition of an incomplete decomposition given in the proof of Theorem 2.3 in Ref. [8]. In the process defined there some off-diagonal elements were omitted after each elimination step. If, instead of omitting some off-diagonal elements, we replace them by negative elements which are smaller in absolute value, or if diagonal elements are replaced by larger ones, the construction process does not break down and the resulting incomplete decomposition 'LU' defines a regular splitting of A. This new process describes the extended concept of incomplete decompositions. A specific example of this process is the following: In the kth step of the Gaussian elimination process elements are eliminated with the kth row. This may cause three effects:

(i) zero off-diagonal elements turn to negative non-zero values;

(ii) non-zero off-diagonal elements become smaller (although larger in absolute value);

(iii) diagonal elements become smaller (but remain positive);

thus omitting to carry out the elimination corrections for some of the elements of the matrix results in an incomplete decomposition. Examples of this type of decomposition are given in Sections 2.1.2, 2.1.3, 2.4 and 2.5.

Other approximate factorisations are discussed by Stone [11], Dupont *et al.* [3], Gustafsson [5] and Wong [15]. They all introduce one or more parameters into the decomposition process to accelerate the convergence. In particular the condition numbers of their preconditioned matrices are in the limit proportional to the mesh parameter $1/h$ whereas the methods described in this paper lead to condition numbers proportional to $1/h^2$. The property of regular splitting is lost in most of their examples.

Kershaw [6] and Manteuffel [7] provide extensions for positive definite matrices. We shall describe these briefly and present another one in Section 3.

Incomplete decompositions for p.d.e.'s in three dimensions are treated in Section 2.3. In Section 4 algorithms for non-symmetric matrices are described.

In Section 5 convergence results as well as 'eigenvalue' information on the decompositions described in Section 2.1 are given for some specific examples.

2. Incomplete Decompositions for Symmetric M-Matrices

2.1. *Five-Point Discretisation of Elliptic Partial Differential Equations in Two Dimensions*

The linear equations in this section arise from five-point discrete approximation to the second-order self adjoint elliptic partial differential equation:

$$-\frac{\partial}{\partial x}\left(A(x,y)\frac{\partial}{\partial x}u(x,y)\right)-\frac{\partial}{\partial y}\left(B(x,y)\frac{\partial}{\partial y}u(x,y)\right)+C(x,y)\,u(x,y)=D(x,y) \tag{2.1.1}$$

with $A(x,y)$, $B(x,y)>0$, $C(x,y)\geqslant 0$ and $(x,y)\in R$, where R is a rectangular region. Along the boundary δR of R the boundary condition

$$\alpha(x,y)\,u(x,y)+\beta(x,y)\frac{\partial}{\partial n}u(x,y)=\gamma(x,y)$$

holds, with $\alpha(x,y)$, $\beta(x,y)\geqslant 0$ and $\alpha(x,y)+\beta(x,y)>0$ and where $\partial/\partial n$ is the outward derivative perpendicular to δR. The structure of the resulting symmetric M-matrix A of order N is shown in Fig. 1. The elements of the diagonal of A are denoted by a_i, those of the first upper diagonal by b_i and those of the mth upper diagonal by c_i, where i is the index of the row of A in which the respective elements occur, and m is the half bandwidth of the matrix. For the derivation of such linear systems see Ref. [14].

2.1.1. *Diagonal scaling.* The simplest permissible choice for an incomplete LU-decomposition K is the diagonal of A. The resulting conjugate gradient method is the same as the c.g. method applied on the matrix scaled by its diagonal. This scaling is in some respects optimal, since it approximately minimises the condition number of $K^{-1}A$ among all diagonal scalings [10]. If A has property (A) it is the optimal diagonal scaling [2, 4]. If the equation is scaled in advance, the number of multiplications is $10N$ per iteration. If not, this number will be $11N$. The total amount of storage is seven arrays of length N.

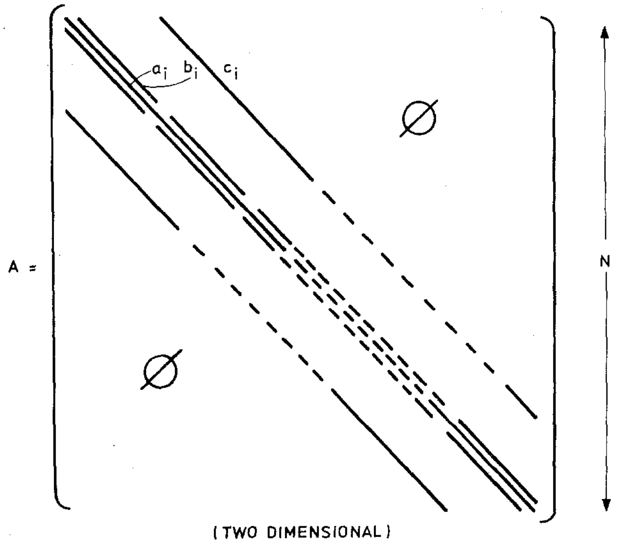


FIG. 1. Matrix of Section 2.1.

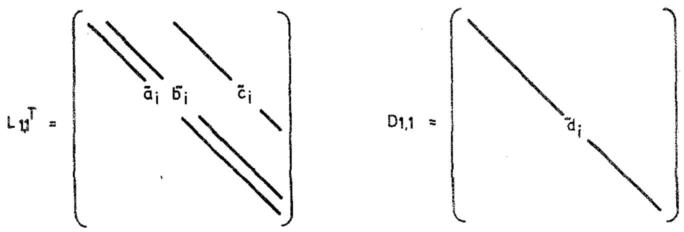


FIG. 2. Matrices of $L_{1,1}^T$ and $D_{1,1}$ (Section 2.1.2).

2.1.2. *ICCG*(1, 1) *and* *SSOR*($\omega = 1$). Here the matrix K is chosen so that its decomposition factor L^T has the same sparsity pattern as the upper triangular part of A. This decomposition has been considered by many authors [3, 5, 8, 11, 13].

It is convenient to write this decomposition as $K_{1,1} = L_{1,1}D_{1,1}L_{1,1}^T$, where $L_{1,1}^T$ is an upper triangular matrix and $D_{1,1}$ a diagonal matrix equal to the inverse of the main diagonal of $L_{1,1}$. In common with the elements of A, those of $L_{1,1}$ are denoted by $\tilde{a}_i$, $\tilde{b}_i$ and $\tilde{c}_i$ and those of $D_{1,1}$ by $\tilde{d}_i$ (see Fig. 2). The following relations are easily verified:

$$\left.\begin{aligned} &\tilde{a}_i = \tilde{d}_i^{-1} = a_i - \tilde{b}_{i-1}^2 \tilde{d}_{i-1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m} \\ &\tilde{b}_i = b_i \quad \text{and} \quad \tilde{c}_i = c_i \end{aligned}\right\} \quad \text{for} \quad i = 1, 2, \ldots, N.$$

In these relations the non-defined elements are zero. Only extra storage for the elements $\tilde{d}_i$ is required. The resulting hybrid conjugate gradient method is called *ICCG*(1, 1). The indices are used to indicate that there is one non-zero diagonal next to the main diagonal and one non-zero diagonal at the outer side of the band. This is

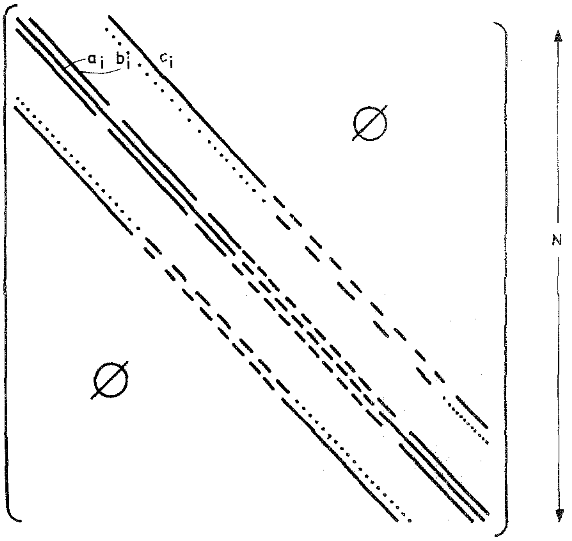


FIG. 3. Matrix $K_{1,1}$ (Section 2.1.3).

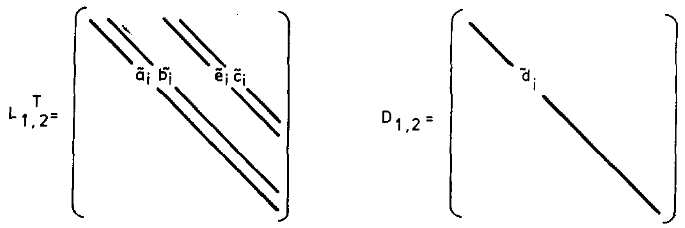


FIG. 4. Matrices $L^T_{1,2}$ and $D_{1,2}$ Section 2.1.3).

ICCG(0) of Ref. [8]. The number of multiplications for this method is $16N$ multiplications per iteration, and the memory requirements are $8N$ words.

The $SSOR(\omega = 1)$ decomposition arises if all Gaussian elimination corrections are neglected. Thus $SSOR(\omega = 1)$ is an example of the extended class of incomplete decompositions mentioned in Section 1. The number of multiplications remains the same as for *ICCG*(1, 1), but one array of storage has been saved. For the use of SSOR as a preconditioning technique see Ref. [1].

2.1.3. *ICCG*(1, 2). The matrix $K_{1,1} = L_{1,1}D_{1,1}L^T_{1,1}$ of the previous section is a matrix equal to A, except for two diagonals adjacent to the outermost two diagonals, as indicated by the dotted lines in Fig. 3. By including non-zero entries on those lines in L and L^T, we expect to improve the approximate decomposition. This approximation will be written as

$$K_{1,2} = L_{1,2}D_{1,2}L^T_{1,2},$$

where $D_{1,2}$ is the diagonal matrix equal to the inverse of the main diagonal of $L^T_{1,2}$. The elements of $L^T_{1,2}$ are denoted by $\tilde{a}_i, \tilde{b}_i, \tilde{e}_i$ and $\tilde{c}_i$ and those of $D_{1,2}$ by $\tilde{d}_i$ (see Fig. 4). The elements $\tilde{a}_i$, $\tilde{b}_i, \tilde{c}_i, \tilde{d}_i$ and $\tilde{e}_i$ can be computed recursively from

$$\left.\begin{aligned}
\tilde{a}_i = \tilde{d}_i^{-1} &= a_i - \tilde{b}^2_{i-1}\tilde{d}_{i-1} - \tilde{e}^2_{i-m+1}\tilde{d}_{i-m+1} - \tilde{c}^2_{i-m}\tilde{d}_{i-m}\\
\tilde{b}_i &= b_i - \tilde{c}_{i-m+1}\tilde{d}_{i-m+1}\tilde{e}_{i-m+1}\\
\tilde{e}_i &= -\tilde{c}_{i-1}\tilde{d}_{i-1}\tilde{b}_{i-1}\\
\tilde{c}_i &= c_i
\end{aligned}\right\}\quad \text{for } i = 1, 2,\ldots, N.$$

The non-defined elements are all zero.

Storage is required for three arrays of length N. The number of multiplications necessary for each iteration step of the resulting hybrid conjugate gradient method *ICCG*(1, 2) is equal to $18N$, and the memory requirements are $10N$ words.

In order to save computer storage (one array) the relatively small correction on $\tilde{b}_i$ can be omitted. Thus $\tilde{b}_i = b_i$. This is another example of the extended class of incomplete decompositions and will be denoted by *ICCG*(1*, 2).

2.1.4. *ICCG*(1, 3). The matrix $K_{1,2}$ is equal to A, except for the two dotted diagonals as indicated in Fig. 5. These non-zero elements can be eliminated by

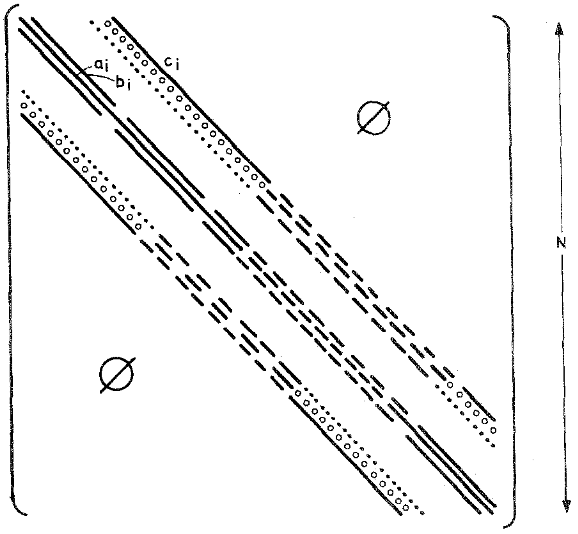


FIG. 5. Matrix $K_{1,2}$ (Section 2.1.4).

introducing an extra diagonal in L^T (see Fig. 6). The elements on these diagonals will be denoted by $\tilde{f}_i$. This incomplete decomposition will be denoted by

$$K_{1,3} = L_{1,3} D_{1,3} L^T_{1,3}.$$

The elements of $D_{1,3}$ and $L^T_{1,3}$ can be computed from:

$$\left.\begin{aligned}
\tilde{a}_i = \tilde{d}_i^{-1} &= a_i - \tilde{b}^2_{i-1}\tilde{d}_{i-1} - \tilde{f}^2_{i-m+2}\tilde{d}_{i-m+2} - \tilde{e}^2_{i-m+1}\tilde{d}_{i-m+1} \\
&\quad - \tilde{c}^2_{i-m}\tilde{d}_{i-m}, \\
\tilde{b}_i &= b_i - \tilde{c}_{i-m+1}\tilde{d}_{i-m+1}\tilde{e}_{i-m+1} - \tilde{e}_{i-m+2}\tilde{d}_{i-m+2}\tilde{f}_{i-m+2}, \\
\tilde{f}_i &= -\tilde{e}_{i-1}\tilde{d}_{i-1}\tilde{b}_{i-1}, \\
\tilde{e}_i &= -\tilde{c}_{i-1}\tilde{d}_{i-1}\tilde{b}_{i-1}, \\
\tilde{c}_i &= c_i,
\end{aligned}\right\} \quad i = 1, 2, \ldots, N.$$

The non-defined elements are zero.

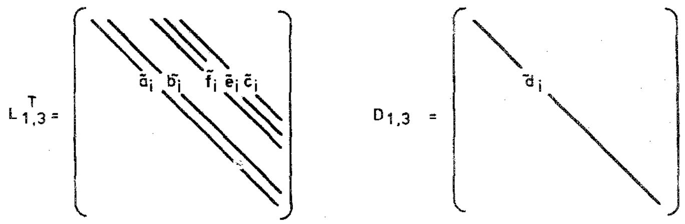


FIG. 6. Matrices $L^T_{1,3}$ and $D_{1,3}$ (Section 2.1.4).

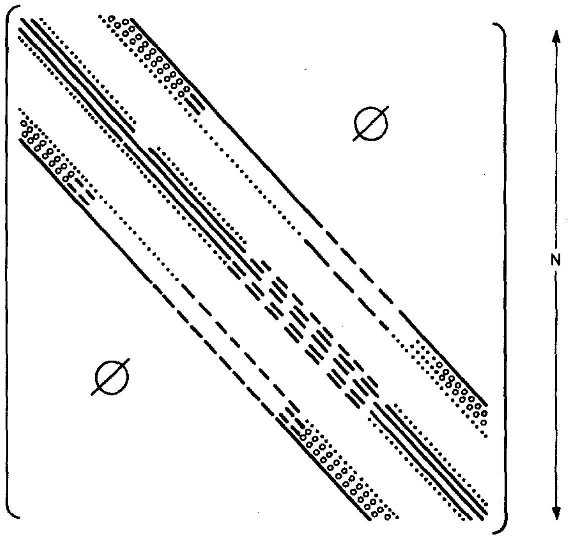


FIG. 7. Matrix $K_{1,3}$ (Section 2.1.5).

The resulting $ICCG(1, 3)$ process necessitates about $20N$ multiplications per iteration and requires 11 arrays of length N.

2.1.5. $ICCG(2, 4)$. Unlike $K_{1,1}$ and $K_{1,2}$, $K_{1,3}$ differs from A by four non-zero diagonals (Fig. 7). To eliminate these, two more non-zero diagonals in L^T are necessary. The elements on these diagonals are denoted by $\tilde{h}_i$ and $\tilde{g}_i$ (Fig. 8). This incomplete decomposition is written as

$$K_{2,4} = L_{2,4} D_{2,4} L_{2,4}^T$$

and the elements of $L_{2,4}^T$ and $D_{2,4}$ follow from

$$\left.\begin{aligned}
\tilde{a}_i &= \tilde{d}_i^{-1} = a_i - \tilde{b}_{i-1}^2 \tilde{d}_{i-1} - \tilde{h}_{i-2}^2 \tilde{d}_{i-2} - \tilde{g}_{i-m+3}^2 \tilde{d}_{i-m+3} - \tilde{f}_{i-m+2}^2 \tilde{d}_{i-m+2} \\
&\qquad - \tilde{e}_{i-m+1}^2 \tilde{d}_{i-m+1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m}, \\
\tilde{b}_i &= b_i - \tilde{h}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1} - \tilde{f}_{i-m+3} \tilde{d}_{i-m+3} \tilde{g}_{i-m+3} - \tilde{e}_{i-m+2} \tilde{d}_{i-m+2} \tilde{f}_{i-m+2} \\
&\qquad - \tilde{c}_{i-m+1} \tilde{d}_{i-m+1} \tilde{e}_{i-m+1}, \\
\tilde{h}_i &= -\tilde{e}_{i-m+3} \tilde{d}_{i-m+3} \tilde{g}_{i-m+3} - \tilde{c}_{i-m+2} \tilde{d}_{i-m+2} \tilde{f}_{i-m+2}, \\
\tilde{g}_i &= -\tilde{e}_{i-2} \tilde{d}_{i-2} \tilde{h}_{i-2} - \tilde{f}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1}, \\
\tilde{f}_i &= -\tilde{c}_{i-2} \tilde{d}_{i-2} \tilde{h}_{i-2} - \tilde{e}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1}, \\
\tilde{e}_i &= -\tilde{c}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1}, \\
\tilde{c}_i &= c_i,
\end{aligned}\right\} \quad i = 1, 2, \dots, N.$$

The non-defined elements are zero.

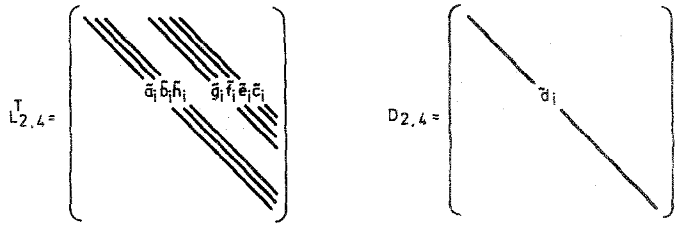

$L^T_{2,4}=$
$\tilde{a}_i\tilde{b}_i\tilde{h}_i$
$\tilde{g}_i\tilde{f}_i\tilde{e}_i\tilde{c}_i$
$D_{2,4}=$
$\tilde{d}_i$

FIG. 8. Matrices $L^T_{2,4}$ and $D_{2,4}$ (Section 2.1.5).

The resulting $ICCG(2, 4)$ process necessitates about $24N$ multiplications per iteration and requires 13 arrays of length N.

If instead of the two extra diagonals $\tilde{h}$ and $\tilde{g}$ in L^T we take only the $\tilde{h}$ diagonal, the $ICCG(3)$ method of Ref. [1] results. This method is denoted in this report by $ICCG(2, 3)$.

2.1.6. *Some other decompositions.* Proceeding in this manner we obtain a sequence of incomplete decompositions $K_{3,6}$, $K_{5,9}$, $K_{8,14}$, $K_{13,22}$, etc, resulting in an increasingly rapid convergence. From the indices we see that the number of non-zero diagonals grows rapidly and thus the amount of work. However, only adding the two diagonals next to those of the previous decomposition will cover most of the convergence improvement. In this way, $K_{3,5}$, $K_{4,6}$, etc., together with the corresponding $ICCG$ methods $ICCG(3,5)$, $ICCG(4,6)$ etc., are developed.

Up to now we have always added complete diagonals in the decomposition process. The diagonals, however, contain their non-zero entries only in a block structure (see Fig. 9). In particular, this implies that in our terminology a complete Choleski factorisation is equivalent to "incomplete decomposition" with $2m - 2$ extra "diagonals."

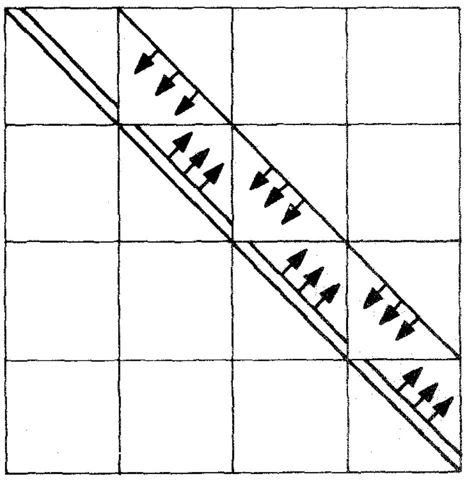


FIG. 9. Structure of Choleski factorisation.

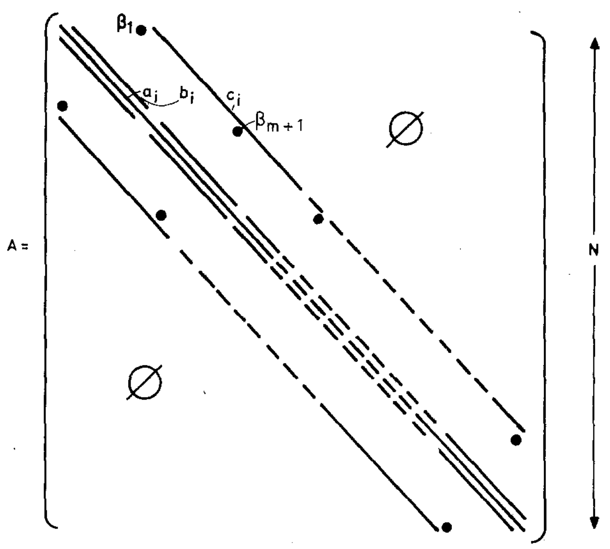


FIG. 10. Matrix A for periodic boundary conditions (Section 2.2).

2.2. *Five-Point Discretisation for Problems with Periodic Boundary Conditions*

The linear equations in this section arise in the same way as the linear equations in Section 2.1, except that a periodic boundary condition holds in at least one direction. In the examples we have restricted ourselves to a periodic boundary condition in the x-direction. This boundary condition gives rise to additional elements in the matrix A, as indicated in Fig. 10. These extra elements are denoted by β_i. Since i is the row index, only $\beta_1, \beta_{m+1}, \beta_{2m+1},\ldots$ are non-zero. Again A is an M-matrix.

2.2.1. *ICCG*(1, 1). In common with the non-periodic case in 2.1.1 an incomplete decomposition can be constructed in cases where the upper triangular factor has the same sparsity structure as the upper triangular part of A. This decomposition is written as

$$K_{1,1} = L_{1,1} D_{1,1} L_{1,1}^T.$$

The elements of $L_{1,1}^T$ and $D_{1,1}$ can be calculated from

$$\begin{aligned}&\tilde{a}_i = \tilde{d}_i^{-1} = a_i - \tilde{b}_{i-1}^2 \tilde{d}_{i-1} - \tilde{\beta}_{i-m+1}^2 \tilde{d}_{i-m+1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m},\\ &\tilde{b}_i = b_i, \qquad \tilde{c}_i = c_i, \qquad \tilde{\beta}_i = \beta_i,\end{aligned} \qquad i = 1, 2,\ldots, N.$$

Non-defined elements are zero. Note that the term $\tilde{\beta}_{i-m+1}^2 \tilde{d}_{i-m+1} \neq 0$ only, if $i = m$, $2m$, $3m$,.... The resulting *ICCG*(1, 1) process again takes approximately $16N$ multiplications per iteration, and needs $8N$ words.

2.2.2. *ICCG*(1, 2). The matrix $K_{1,1}$ of 2.2.1 has elements as indicated in Fig. 11. To annihilate these elements in L^T, non-zero elements are required in these places. These elements are denoted as shown in Fig. 12 by $\tilde{a}_i$, $\tilde{b}_i$, $\tilde{c}_i$, $\tilde{\beta}_i$, $\tilde{e}_i$, $\tilde{\gamma}_i$, $\tilde{\delta}_i$ and the elements of $D_{1,2}$ by $\tilde{d}_i$. They can be calculated from

$$\left.\begin{aligned}
\tilde{a}_i = \tilde{d}_i^{-1} &= a_i - \tilde{b}_{i-1}^2 \tilde{d}_{i-1} - \tilde{e}_{i-m+1}^2 \tilde{d}_{i-m+1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m}\\
&\quad - \tilde{\delta}_{i-1}^2 \tilde{d}_{i-1} \text{ (only for } i = m+1, 2m+1, 3m+1, \text{--)}\\
&\quad - \tilde{\gamma}_{i-m+2}^2 \tilde{d}_{i-m+2} - \tilde{\beta}_{i-m+1}^2 \tilde{d}_{i-m+1} \text{ (only for } i = m, 2m \text{ ---)},\\
\tilde{b}_i &= b_i - \tilde{c}_{i-m+1} \tilde{d}_{i-m+1} \tilde{e}_{i-m+1} \text{ (only for } i \neq m, 2m \text{ ---)},\\
\tilde{c}_i &= c_i,\\
\tilde{e}_i &= -\tilde{c}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1} \text{ (only for } i \neq 1, m+1 \text{ ---)},
\end{aligned}\right\} \quad i = 1, 2, 3, \ldots N.$$

$$\begin{aligned}
\tilde{\delta}_i &= -\tilde{e}_{i-m+2} \tilde{d}_{i-m+2} \tilde{\gamma}_{i-m+2} - \tilde{c}_{i-m+1} \tilde{d}_{i-m+1} \tilde{\beta}_{i-m+1}, && i = m, 2m, \text{---},\\
\tilde{\gamma}_i &= -\tilde{\beta}_{i-1} \tilde{d}_{i-1} \tilde{b}_{i-1}, && i = 2, m+2, 2m+2, \text{---},\\
\tilde{\beta}_i &= \beta_i - \tilde{c}_{i-1} \tilde{d}_{i-1} \tilde{\delta}_{i-1}, && i = 1, m+1, 2m+1, \text{---}.
\end{aligned}$$

Note that $\tilde{b}_m, \tilde{b}_{2m}, \text{---}$ and $\tilde{e}_1, \tilde{e}_{m+1}, \text{---}$ are nonexistent. The resulting *ICCG* process takes $18N$ multiplications per iteration and requires roughly three arrays of length N for the incomplete decomposition.

2.2.3. *ICCG*(1, 1) *periodic.* The incomplete decomposition of 2.2.1 does not have a periodic structure. To obtain a K with a periodic structure we write

$$K_p = (L_p + D_p^{-1})\, D_p (L_p^T + D_p^{-1}).$$

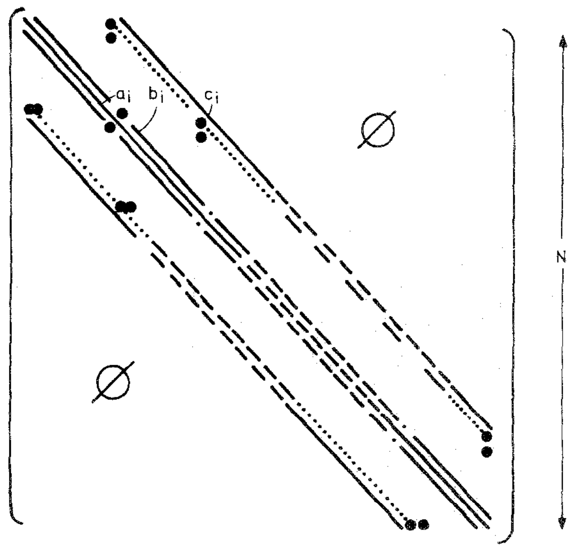


FIG. 11. Matrix $K_{1,1}$ (Section 2.2.2).

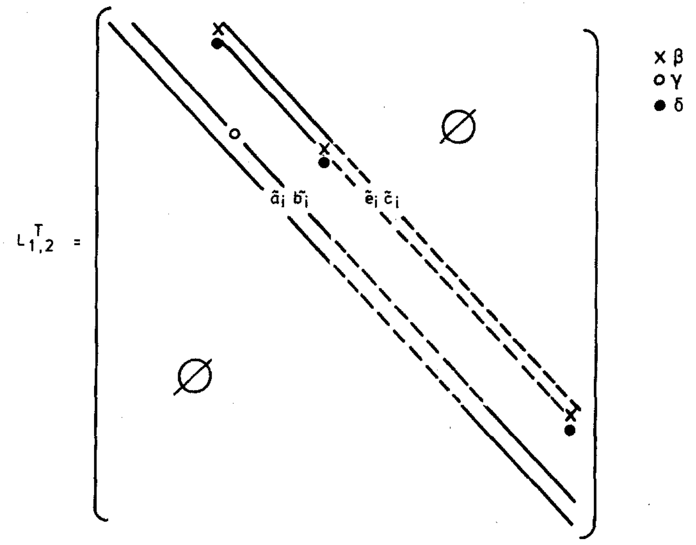


FIG. 12. Matrix $L_{1,2}^T$ (Section 2.2.2).

The periodic structure of the matrix L_p is given in Fig. 13. D_p is a diagonal matrix. The elements of L_p and D_p have to satisfy:

$$\tilde{b}_i = b_i, \qquad \tilde{c}_i = c_i, \qquad \tilde{\beta}_i = \beta_i \qquad \text{for } i = 1, 2, ---, N, \tag{2.2.3.1}$$

$$\tilde{a}_i = \tilde{d}_i^{-1} = a_i - \tilde{b}_{i-1}^2 \tilde{d}_{i-1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m} \qquad \text{for } i = 2, 3, --, m, m+2, m+3, --, 2m, 2m+2, ---, N, \tag{2.2.3.2}$$

$$\tilde{a}_i = \tilde{d}_i^{-1} = a_i - \tilde{\beta}_i^2 \tilde{d}_{i+m-1} - \tilde{c}_{i-m}^2 \tilde{d}_{i-m} \qquad \text{for } i = 1, m+1, 2m+1, --, N-m+1. \tag{2.2.3.3}$$

The $\tilde{d}_i$ cannot be calculated straightforwardly, since in the second formula $\tilde{d}_{i+m-1}$ is present. We can calculate them by substituting (2.2.3.3) into (2.2.3.2) for the next i, and continuing in this way, we find quadratic equations for the $\tilde{d}_{km}$. For $\tilde{d}_{km}$ we choose the largest root, since this choice results in smaller elements $\tilde{b}_{i-1}\tilde{d}_{i-1}^{-1}\tilde{c}_{i-1}$ in the error matrix $K_p - A$. We now give the derivation for the formula for $\tilde{d}_m$. The $\tilde{d}_{km}$ [illegible]

$$\tilde{d}_1^{-1} = w_1 - v_1 \tilde{d}_m \tag{2.2.3.4}$$

and (2.2.3.2) as

$$\tilde{d}_i^{-1} = w_i - v_i \tilde{d}_{i-1}, \qquad i = 2, --- m, \tag{2.2.3.5}$$

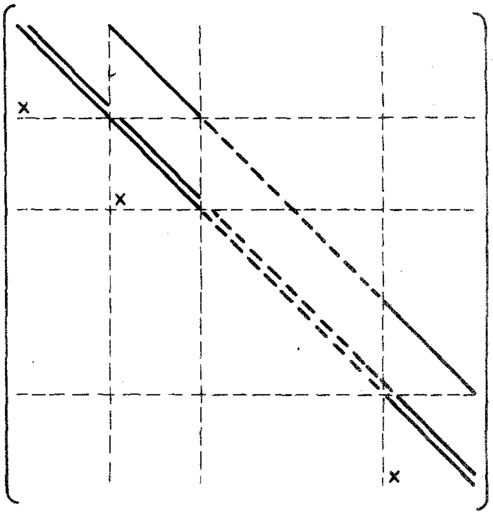


FIG. 13. Matrix L_p^T (Section 2.2.3).

From induction it follows that:

$$\tilde{d}_i = \frac{p_i + q_i \tilde{d}_m}{r_i + s_i \tilde{d}_m} \qquad \text{for } i = 1, 2, ---, m.$$

The coefficients p_i, q_i and s_i satisfy

$$p_1 = 1, \qquad q_1 = 0, \qquad r_1 = w_1, \qquad s_1 = -v_1, \qquad p_{i+1} = r_i, \qquad q_{i+1} = s_i,$$
$$r_{i+1} = w_{i+1} r_i - v_{i+1} p_i, \qquad s_{i+1} = w_{i+1} s_i - v_{i+1} q_i.$$

This leads to the quadratic equation in d_m with known coefficients p_m, q_m, r_m and s_m:

$$s_m \tilde{d}_m^2 + (r_m - q_m)\, \tilde{d}_m - p_m = 0,$$

from which the largest root can be calculated.

2.3. *Seven-Point Discretisations of Elliptic p.d.e.'s in Three Dimensions*

The seven-point discretisation for Eq. (2.1.1) in three dimensions leads in a similar way to a matrix with seven non-zero diagonals. The structure of this matrix A is shown in Fig. 14. The elements of the upper triangular part of A are denoted by a_i, b_i, c_i and e_i, where i is counted by the row. If n, m, k are the number of gridpoints in the x, y, z directions, respectively, the order of the matrix and the sizes of the blocks are nmk, nm and n.

2.3.1. *ICCG*(1, 1, 1). In common with the 2-*D* case the *ICCG*(1, 1, 1) factorisation is the one where the upper triangular factor has the same non-zero

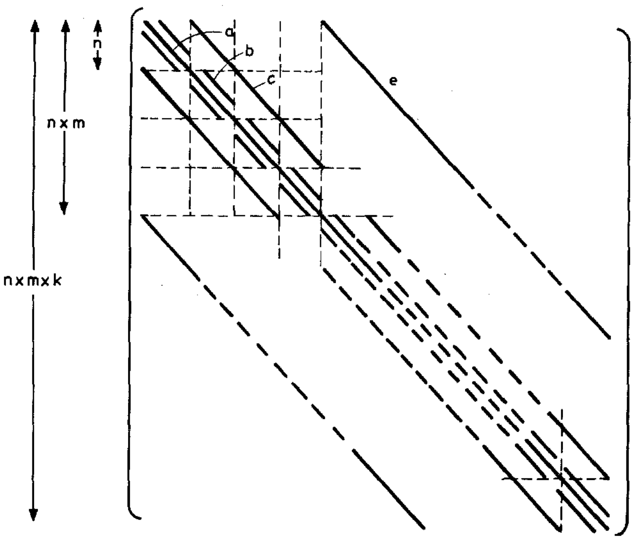


FIG. 14. Matrix A for three-dimensional problems.

structure as the upper triangular part of A. Again this decomposition is written as $K_{1,1,1} = L_{1,1,1} D_{1,1,1} L^T_{1,1,1}$, where $L^T_{1,1,1}$ is an upper triangular matrix and $D_{1,1,1}$ a diagonal matrix equal to the inverse of the main diagonal of $L^T_{1,1,1}$. The elements of $L^T_{1,1,1}$ are denoted by $\tilde{a}_i$, $\tilde{b}_i$, $\tilde{c}_i$ and $\tilde{e}_i$ and the elements of $D_{1,1,1}$ by $\tilde{d}_i$. These elements are given by the recurrency relations:

$$\begin{aligned} &\tilde{a}_i = \tilde{d}_i^{-1} = a_i - \tilde{b}^2_{i-1}\tilde{d}_{i-1} - \tilde{c}^2_{i-n}\tilde{d}_{i-n} - \tilde{e}^2_{i-mn}\tilde{d}_{i-mn} \\ &\tilde{b}_i = b_i, \qquad \tilde{c}_i = c_i \qquad \text{and} \qquad \tilde{e}_i = e_i \end{aligned} \qquad \text{for } i = 1, 2, ---, nmk.$$

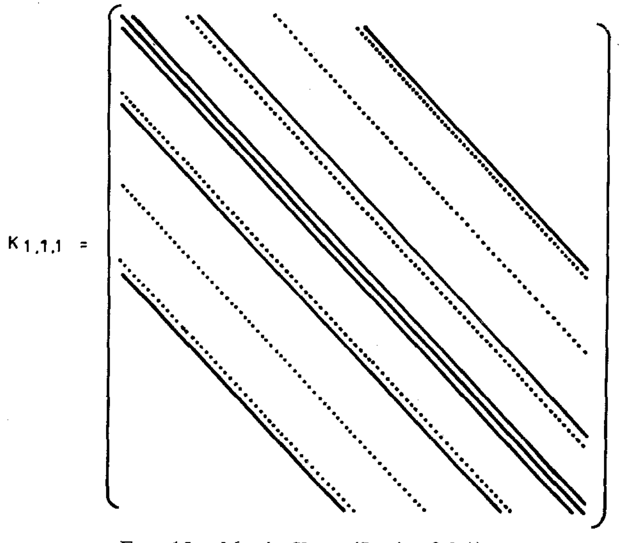


FIG. 15. Matrix $K_{1,1,1}$ (Section 2.3.1).

Non–defined elements should be replaced by zeros. It can easily be seen that for major problems where the diagonals cannot be stored all together in core, the $\tilde{d}_i$ can be calculated by taking successively only parts of the order of $n \times m$ into core.

The resulting hybrid conjugate gradient method requires $20N$ multiplications per iteration, and $9N$ words for storage.

2.3.2. *Other decompositions for* 3-*D*. The matrix $K_{1,1,1} = L_{1,1,1}D_{1,1,1}L^T_{1,1,1}$, of the previous section is a matrix equal to A, except for six diagonals, as shown in Fig. 15 as dotted lines. We obtain another decomposition by including non-zero entries on these lines in L and L^T. The elements of L^T_3 are denoted by $\tilde{a}_i$, $\tilde{b}_i$, $\tilde{c}_i$, $\tilde{e}_i$, $\tilde{f}_i$, $\tilde{g}_i$ and $\tilde{h}_i$, as shown in Fig. 16, and can be calculated from:

$$\left.\begin{aligned}
\tilde{a}_i &= \tilde{d}_i^{-1} = a_i - \tilde{b}^2_{i-1}\tilde{d}_{i-1} - \tilde{h}^2_{i-n+1}\tilde{d}_{i-n+1} - \tilde{c}^2_{i-n}\tilde{d}_{i-n} \\
&\qquad - \tilde{g}^2_{i-nm+n}\tilde{d}_{i-nm+n} - \tilde{f}^2_{i-nm+1}\tilde{d}_{i-nm+1} - \tilde{e}^2_{i-nm}\tilde{d}_{i-nm}, \\
\tilde{b}_i &= b_i - \tilde{c}_{i-n+1}\tilde{d}_{i-n+1}\tilde{h}_{i-n+1} - \tilde{e}_{i-mn+1}\tilde{d}_{i-mn+1}\tilde{f}_{i-mn+1}, \\
\tilde{h}_i &= \tilde{c}_{i-1}\tilde{d}_{i-1}\tilde{b}_{i-1} - \tilde{f}_{i-mn+n}\tilde{d}_{i-mn+n}\tilde{g}_{i-mn+n}, \\
\tilde{c}_i &= c_i - \tilde{e}_{i-mn+n}\tilde{d}_{i-mn+n}\tilde{g}_{i-mn+n}, \\
\tilde{g}_i &= -\tilde{f}_{i-n+1}\tilde{d}_{i-n+1}\tilde{h}_{i-n+1} - \tilde{e}_{i-n}\tilde{d}_{i-n}\tilde{c}_{i-n}, \\
f_i &= -\tilde{e}_{i-1}\tilde{d}_{i-1}\tilde{b}_{i-1}, \\
\tilde{e}_i &= e_i,
\end{aligned}\right\} \quad i = 1, 2, --- mnk.$$

Six arrays of length N are required to store the non-zero diagonals of L^T. The resulting *ICCG* method requires $26N$ multiplications per iteration.

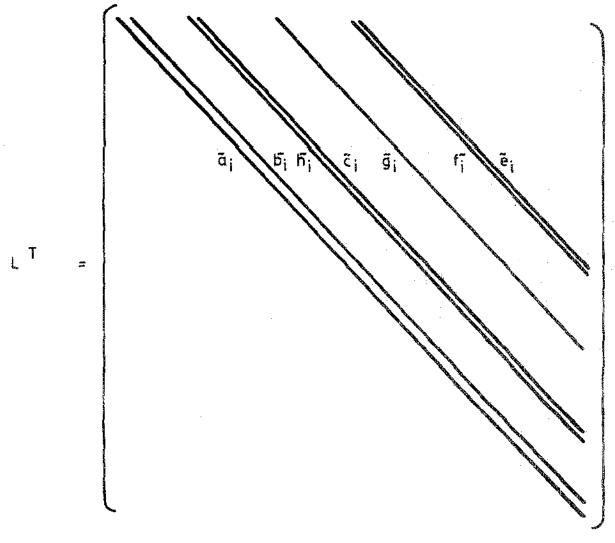


FIG. 16. Matrix L^T (Section 2.3.2).

Unfortunately, if we proceed in this manner the number of diagonals in the subsequent decompositions will increase rapidly. For instance, the next decomposition has 12 non-zero diagonals in its upper triangular part. The resulting *ICCG* method takes $36N$ multiplications per iteration.

2.4. *M-Matrices Arising from Five-Point Discretisations on Irregular Regions*

So far we have only considered discretisations on square regions. We are now going to comment on regions with internal boundaries (no-flow boundaries) or differently shaped regions. For convenience it will be assumed that the region consists of small squares.

An internal boundary is reflected by some extra zeros in the matrix, but the matrix remains a symmetric M-matrix, thus incomplete decompositions can be constructed as before. An internal boundary implies that there is no direct connection (no flow) between points on different sides of the boundary. This property is preserved in each of the above-mentioned decompositions. This in contrast with Stone's *SIP* method [11], where the use of the iteration parameter may cause a connection through a no-flow boundary.

Irregularly shaped regions can be extended in an obvious way to square regions with an internal boundary at the point of the original real boundary. The linear system arising from this extended region can be treated as before, bearing in mind that the extended parts do not require computational work. If the true boundary does not coincide with meshpoints, the discretisation may lead to an irregular non-zero structure. This will be considered in the following section.

2.5. *M-Matrices with an Irregular Non-zero Structure*

M-matrices with an irregular non-zero structure arise, for instance, from some finite-element methods on irregular meshes [12] and pipeline networks [9].

We write the matrix A as $A = L + D + U$, where L, U are strictly lower and upper triangular, respectively, and D the diagonal of A. If we omit all Gaussian elimination corrections on off-diagonal elements (see Section 1), then the incomplete decomposition is given by $K_0 = (L + D_0)\, D_0^{-1} (D_0 + U)$.

D_0 is determined by the relation that the diagonal of $K_0 - A$, which is equal to the diagonal of $D_0 + \text{diag}(LD_0^{-1} U) - D$, is zero.

If the matrix is symmetric, this decomposition can be combined with the conjugate gradient method. For non-symmetric matrices see Section 4.

3. Algorithms for Symmetric Positive Definite Matrices

If the matrix is not an M-matrix, the construction of an incomplete decomposition may fail because of the occurrence of non-positive diagonal elements [6]. Small positive diagonal elements are also undesirable because of stability problems.

Three different strategies which seem to overcome this problem have currently been proposed:

(i) If a diagonal element of less than a prescribed positive value is encountered during the construction of the incomplete LL^T decomposition then some already computed off-diagonal elements in the corresponding column of L^T are set to zero.

(ii) The diagonal element is enlarged if necessary [6], e.g., by neglecting some of the Gaussian elimination corrections (see Section 1).

(iii) We can also add αI to the matrix [7]. If α is large enough, the problems signalled will not occur.

Strategy (ii) has the advantage over (i) that the Gaussian elimination process is cut short 1 step later. Further if, e.g., K_0 as defined in Section 2.5 is applied strategy (i) needs extra memory to denote which elements have been set to zero.

Strategy (iii) has the disadvantage that the whole diagonal is affected whereas often only local corrections are required. We prefer strategy (ii) to the others.

4. Algorithms for Non-symmetric Matrices

If the matrix is non-symmetric, then an incomplete LU decomposition K can be constructed in a similar way as described previously for the symmetric matrices. Since symmetry and positive-definiteness are both required for the conjugate gradient algorithm, the CG algorithm can be applied to:

$$A^T K^{T-1} K^{-1} A x = A^T K^{T-1} K^{-1} b.$$

This algorithm requires twice as much work per iteration as the corresponding symmetric case and the upper bound for the number of iterations increases. It has been considered in more detail by Kershaw [6].

5. Numerical Experiments

To obtain an impression of the convergence behaviour of different incomplete decompositions, we have, for the $ICCG$ methods introduced in Section 2.1,

(i) compared the convergence results,

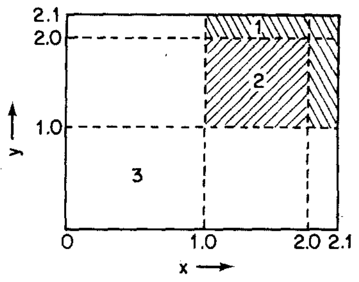
(ii) calculated the eigenvalue distribution of the preconditioned matrices $K^{-1}A$.

The two test problems were:

(i) *Problem* 1. The five-point discretisation of the Poisson equation $\Delta u = 0$ over $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$ with boundary conditions $\partial u/\partial x = 0$ for $x = 0$ and $x = 1$, $\partial u/\partial y = 0$ for $y = 1$ and $u = 1$ for $y = 0$. A uniform rectangular mesh was chosen, with $\Delta x = 1/31$ and $\Delta y = 1/31$, which resulted in a linear system of 992 equations.

The solution of this equation is known to be $u(x,y)=1$ and as initial starting vector for the iterative schemes a vector was chosen with all entries random between 0 and 1. This was done to prevent coincidental fast convergence.

(ii) *Problem* 2. This problem has been taken from Varga [14]. Equation (2.1.1) holds for R, where R is the square region $0 \leqslant x, y \leqslant 2.1$ as shown below:



On the boundary of R, the boundary conditions are $\partial u/\partial n = 0$. Further, $D(x,y)=0$ over R and the functions A, B and C are given by

Region	$A(x,y)$	$B(x,y)$	$C(x,y)$
1	1.0	1.0	0.02
2	2.0	2.0	0.03
3	3.0	3.0	0.05

A uniform rectangular mesh was chosen with 0.05 mesh spacing, so that a system of 1849 linear equations resulted. The solution of the system is known to be $u=0$. A vector similar to the one in problem 1 was chosen as starting vector.

In Tables I and II the convergence results are listed. In both tables we see that in general the amount of work decreases when extra diagonals are included according to the patterns described in Section 2. Including other diagonals does not lead to further improvements, e.g., *ICCG*(1, 4), *ICCG*(2, 5) and *ICCG*(3, 4). Taking into account the amount of storage required and the complexity of programming we conclude that *ICCG*(1,3) is a good choice.

Since the convergence behaviour depends on the eigenvalue distribution of the preconditioned matrix, where the condition number and clustering play an important role, a number of the largest and smallest eigenvalues have been calculated for the matrix of problem 1 preconditioned with several incomplete decompositions. The eigenvalues are all divided by the smallest eigenvalue $\lambda_{\min}$, because an upper bound for the convergence factor of the conjugate gradient method is given by $(\sqrt{c}-1)/(\sqrt{c}+1)$, where the condition number $c=\lambda_{\max}/\lambda_{\min}$. The distribution of these scaled eigenvalues has been plotted in Figs. 17–21.

TABLE I

Convergence Behaviour for Problem 1, $N = 992$[†]

Work per iteration	Method	Number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^{-3}$	Computational work to reduce $\|Ax_i - b\|_2 \leqslant 10^{-3}$	Number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$	Computational work to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$	Number of iterations to gain* 1 decimal	Computational work to gain 1 decimal
$16N$	$SSOR(\omega = 1)$	31	$496N$	52	$832N$	6.8	$109N$
	$ICCG(1,1)$	32	$512N$	44	$704N$	5.6	$90N$
$18N$	$ICCG(1^*, 2)$	20	$360N$	33	$594N$	4.3	$77N$
	$ICCG(1,2)$	19	$342N$	27	$486N$	3.6	$65N$
$20N$	$ICCG(1,3)$	15	$300N$	22	$440N$	2.9	$58N$
$22N$	$ICCG(2,3)$	12	$264N$	20	$440N$	2.4	$53N$
	$ICCG(1,4)$	14	$308N$	21	$462N$	2.6	$58N$
$24N$	$ICCG(2,4)$	10	$240N$	16	$384N$	2.1	$50N$
$26N$	$ICCG(2,5)$	9	$234N$	16	$416N$	1.9	$50N$
	$ICCG(3,4)$	9	$234N$	15	$390N$	1.9	$49N$
$28N$	$ICCG(3,5)$	8	$224N$	13	$364N$	1.7	$48N$
$30N$	$ICCG(3,6)$	8	$240N$	13	$390N$	1.6	$48N$
$32N$	$ICCG(4,6)$	7	$224N$	12	$384N$	1.4	$45N$
$36N$	$ICCG(5,7)$	6	$216N$	10	$360N$	1.24	$45N$
$40N$	$ICCG(5,9)$	6	$240N$	9	$360N$	1.13	$45N$
	$ICCG(6,8)$	5	$200N$	9	$360N$	1.05	$42N$
$44N$	$ICCG(7,9)$	5	$220N$	8	$352N$	.96	$42N$

[†] Note that the convergence rate is $0(N^{1.5})$, $N \to \infty$.

* Number of iterations to gain 1 decimal $= n/{}^{10}\log(r_0/r_n)$, where $n =$ number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$, $r_0 = \|Ax_0 - b\|_2$ and $r_n = \|Ax_n - b\|_2$.

TABLE II

Convergence Behaviour for Problem 2, $N = 1849$[†]

Work per iteration	Method	Number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^3$	Computational work to reduce $\|Ax_i - b\|_2 \leqslant 10^{-3}$	Number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$	Computational work to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$	Number of iterations to gain* 1 decimal	Computational work to gain 1 decimal
$16N$	$SSOR(\omega = 1)$	69	$1104N$	89	$1424N$	10.6	$169N$
	$ICCG(1,1)$	58	$928N$	72	$1152N$	8.7	$139N$
$18N$	$ICCG(1^*,2)$	45	$810N$	57	$1026N$	6.9	$124N$
	$ICCG(1,2)$	38	$684N$	47	$846N$	5.7	$103N$
$20N$	$ICCG(1,3)$	30	$600N$	38	$760N$	4.6	$92N$
$22N$	$ICCG(2,3)$	26	$572N$	33	$726N$	3.9	$86N$
	$ICCG(1,4)$	28	$616N$	35	$770N$	4.2	$93N$
$24N$	$ICCG(2,4)$	22	$528N$	29	$696N$	3.4	$82N$
$26N$	$ICCG(2,5)$	21	$546N$	27	$702N$	3.2	$83N$
	$ICCG(3,4)$	19	$494N$	25	$650N$	3.1	$79N$
$28N$	$ICCG(3,5)$	18	$504N$	22	$616N$	2.7	$76N$
$30N$	$ICCG(3,6)$	15	$450N$	21	$630N$	2.6	$77N$
$32N$	$ICCG(4,6)$	13	$416N$	19	$608N$	2.2	$70N$
$36N$	$ICCG(5,7)$	12	$432N$	16	$576N$	1.92	$69N$
$40N$	$ICCG(5,9)$	11	$440N$	16	$640N$	1.79	$72N$
	$ICCG(6,8)$	11	$440N$	15	$600N$	1.66	$66N$
$44N$	$ICCG(7,9)$	9	$396N$	13	$572N$	1.58	$70N$

[†] Note that the convergence rate is $0(N^{1.5})$, $N \to \infty$.

* Number of iterations to gain 1 decimal $= n/{}^{10}\log(r_0/r_n)$, where $n =$ number of iterations to reduce $\|Ax_i - b\|_2 \leqslant 10^{-6}$, $r_0 = \|Ax_0 - b\|_2$ and $r_n = \|Ax_n - b\|_2$.

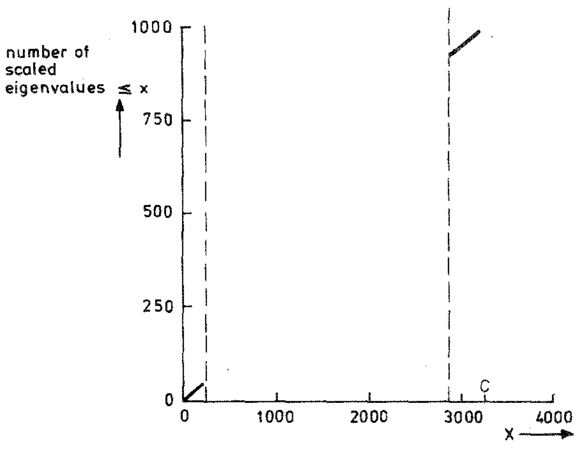


FIG. 17. Distribution of scaled eigenvalues of A.

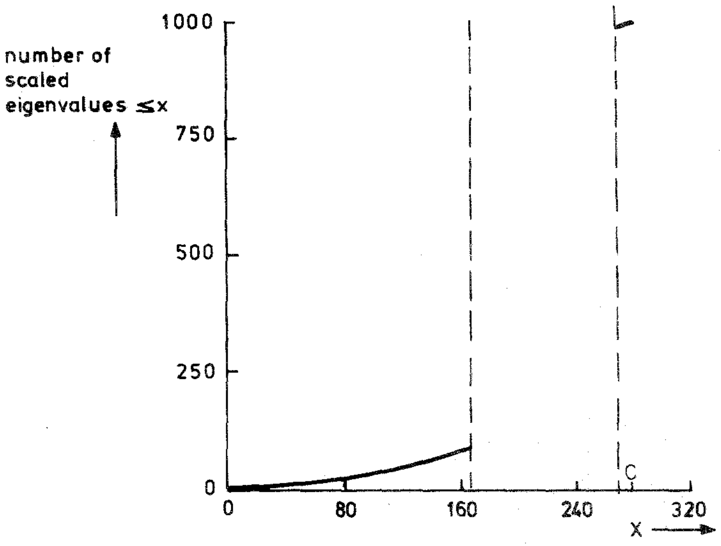


FIG. 18. Distribution of scaled eigenvalues of $K_{1,1}^{-1}A$.

From these figures we see that the condition c of the preconditioned matrix decreases rapidly with an increasing number of diagonals. Also, the majority of eigenvalues are concentrated in intervals which become smaller and smaller in relationship to c. Comparisons with other methods for both these examples have been described in Ref. [8].

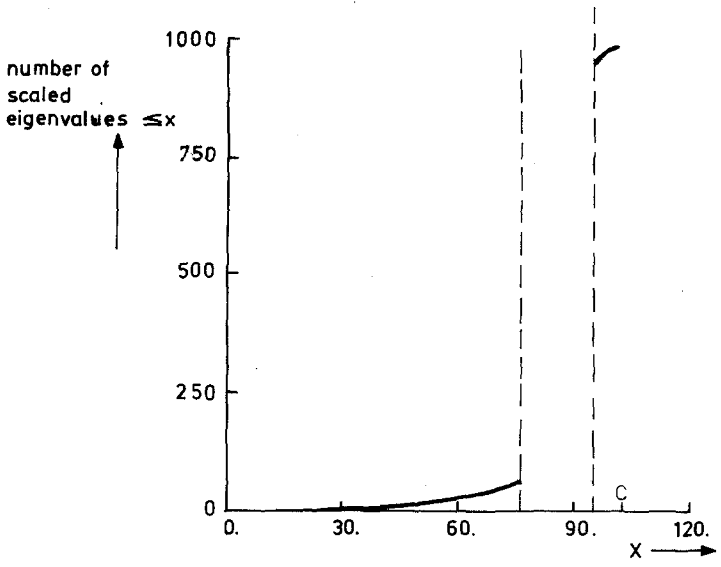


FIG. 19. Distribution of scaled eigenvalues of $K_{1,2}^{-1}A$.

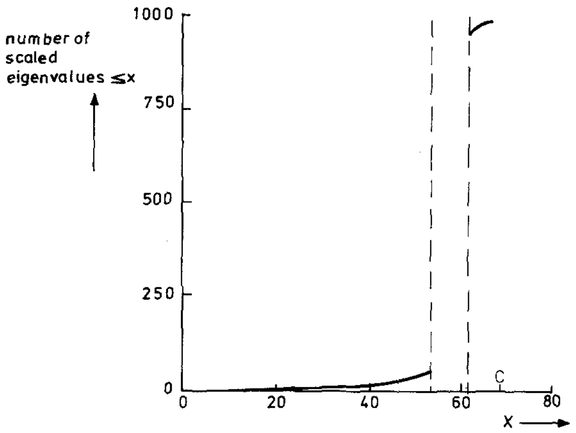


FIG. 20. Distribution of scaled eigenvalues of $K_{1,3}^{-1}A$.

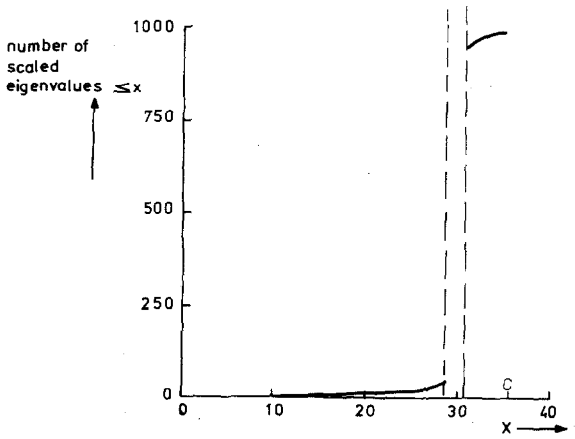


FIG. 21. Distribution of scaled eigenvalues of $K_{2,4}^{-1}A$.

References

1. O. Axelsson, *BIT* **13** (1972), 443–467.
2. P. Concus, G. H. Golub, and D. P. O'Leary, *in* 'Sparse Matrix Computations," Academic Press, New York/London, 1976.
3. T. Dupont, M. R. P. Kendall, and H. H. Rachford, *SIAM J. Numer. Anal.* **5** (1968), 559–573.
4. G. Forsythe and E. G. Strauss, *Proc. Amer. Math. Soc.* **6** (1955), 340–345.
5. I. Gustafsson, "A Class of 1st Order Factorisation Methods," Research Report, Dept. of Comp. Sciences, University of Göteborg.
6. D. S. Kershaw, *J. Comp. Phys.* **26**(1) (1978), 43–65.
7. T. A. Manteuffel, "The Shifted Incomplete Cholesky Factorisation," Sandia Laboratories Pub. SAND 78-8226, Livermore, May 1978.
8. J. A. Meijerink and H. A. van der Vorst, *Math. Comp.* **31**, No. 137 (1977), 148–162.
9. T. A. Porsching, *in* "Sparse Matrix Computations," Academic Press, New York, 1976.
10. A. van der Sluis, *Numer. Math.* **14** (1969), 14–23.
11. H. L. Stone, *SIAM J. Numer. Anal.* **5** (1968), 530–558.
12. G. Strang and G. J. Fix, "An Analysis of the Finite Element Method," Prentice–Hall, Englewood [illegible]
13. [illegible] Univ. of Wisconsin Press, Madison, Wisconsin, 1960.
14. R. S. Varga, "Matrix Iterative Analysis," Prentice–Hall, Englewood Cliffs N.J., 1962.
15. Y. S. Wong, *in* "Numerical methods in thermal problems," Proc. of the First Conf. 1979, Pineridge Press. Swansea, UK.