Note

On the Use of Preconditioned Conjugate Gradient Methods for Red–Black Ordered Five-Point Difference Schemes

1. Introduction

The red–black (or checkerboard) ordering has certain advantages with regard to arithmetic and demand of storage, in particular for large, out-of-core systems of linear equations corresponding to the five-point difference approximation of the Poisson equation. This is so because on an $m \times n$ grid, $m \leqslant n$, one may reduce the problem to a, possibly in-core, system of half the number of unknowns, without increasing the bandwidth, $2m + 1$. Only this latter system has to be factorized. We refer to, for example, McDonald and Trimble [1] for this and a similar ordering, wherein other relevant references also may be found.

In particular, when m is large, it may be of interest to use an incomplete factorization of this reduced system and to solve it by iterative refinement. We show that this is possible and that the number of iterations grow as $O(h^{-1/2})$, as the average mesh size $h \to 0$, if a conjugate gradient method is used for the iterative refinements. (We assume that we subdivide all mesh sides in the same fraction given a coarse mesh to start with.)

For elongated grids, where m is small relative to n, the incomplete factorization is relatively more accurate than for the case where $m = n$. Hence fewer iterations are needed and, as we will see, the method is still competitive with a full factorization method. For a certain incomplete factorization method used below the demand of storage is about 7.5mn.

2. Incomplete Factorization of Red–Black Ordered Difference Equations

Given a grid Ω_h, consisting of rectangular elements, we number the points as on a checkerboard (see Fig. 1). For simplicity we describe the ordering on a rectangular $m \times n$ grid, where $m \leqslant n$. The matrix, corresponding to the difference operator of the Laplacian or of the more general operator

$$\frac{\partial}{\partial x}\left(a\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(b\frac{\partial}{\partial y}\right), \qquad a \geqslant a_0 > 0, \qquad b \geqslant b_0 > 0,$$

then takes the form,

$$A = \begin{bmatrix} D_1 & B^T \\ B & D_2 \end{bmatrix},$$

0021-4991/80/050284–06$02.00/0

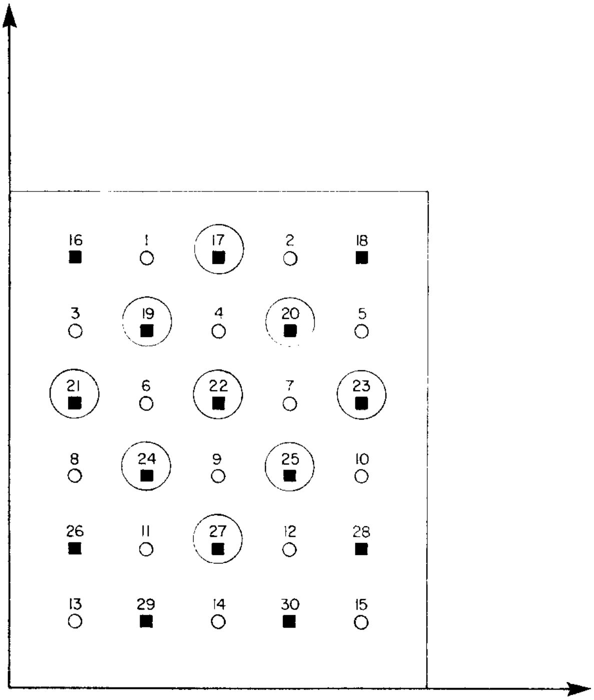

FIG. 1. The red–black ordering for $m = 5$, $n = 6$.

where D_1 and D_2 are positive diagonal matrices. Since we may multiply A by

$$\begin{bmatrix} D_1^{-1/2} & 0 \\ 0 & D_2^{-1/2} \end{bmatrix}$$

from the left and from the right, making corresponding transformations of the unknown and source vectors, we may, without limitation, assume that $D_1 = D_2 = I$, the identity matrix. Hence we have

$$A \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},$$

where

$$A = \begin{bmatrix} I & B^T \\ B & I \end{bmatrix}.$$

We now eliminate u_1 from the second set of equations, using $u_1 = -B^T u_2 + f_1$ from the first. Thus we have to solve the reduced system

$$(I - BB^T)\, u_2 = f_2 - Bf_1\,. \tag{2.1}$$

We observe that $\| BB^T \|_\infty \leqslant \| B \|_\infty \| B^T \|_\infty \leqslant 1$, and that $I - BB^T$ is an M matrix. (Actually, it is quite easy to see that this matrix corresponds to a nine-point difference operator, as indicated by the circles in Fig. 1, the variables being eliminated in the indicated order 1–15.) Hence there exist numerically stable incomplete factorizations of this matrix (see, for example, Axelsson [2], Meijerink and van der Vorst [3] and Gustafsson [4]). Let LL^T be such an incomplete factorization.

3. The Preconditioned Conjugate Gradient Method

We now solve (2.1) by iteration, using a preconditioned conjugate gradient method. At each step of this we solve the two systems of equations corresponding to the triangular matrices L and L^T.

Since the modified incomplete factorization MIC methods described in Gustafsson [4] are asymptotically fastest, we will use them in this report. These factorizations are realized by moving (and adding to) the diagonal of L^T entries that appear in undesired positions and thus cause fill-in, during the LL^T decomposition of a matrix A. For diagonally dominant matrices these factorization algorithms are stable, i.e., the entries of diag(L) are positive. The positions where fill-in is actually allowed can be chosen in advance. We use the notation MIC(d) to indicate that L contains d more nonzero diagonals than the lower part of A. It is shown in [4] that if LL^T is derived by a MIC factorization of a diagonally dominant matrix A corresponding to a second-order differential equation problem, the eigenvalues of $(LL^T)^{-1} A$ are bounded below by $O(1)$ and above by $O(h^{-1})$. In the present application $I - BB^T$ is a diagonally dominant M-matrix. Hence the eigenvalues λ_i of $(LL^T)^{-1} (I - BB^T)$ satisfy $O(1) < \lambda_i < O(h^{-1})$, $h \to 0$, $i = 1, 2, \ldots, mn/2$. Hence the spectral condition number κ is $O(h^{-1})$, $h \to 0$, and the conjugate gradient method, preconditioned by LL^T, will converge in $O(h^{-1/2})$ iterations. Different preconditioned conjugate gradient methods are described in, for example, Axelsson [5, 6] and Gustafsson [4]. The following one-step version was used (for the solution of a system $Au = f$):

Let u^0 be arbitrary, calculate $r^0 = Au^0 - f$, solve $LL^T\gamma^0 = r^0$, and calculate $\gamma_0 = (\gamma^0, \gamma^0)$, $d^0 = -\gamma^0$, $d1^0 = Ad^0$, $\sigma_0 = (r^0, \gamma^0)$. Then for $l = 0, 1, \ldots,$ calculate $\lambda_l = \sigma_l/(d^l, d1^l)$, $u^{l+1} = u^l + \lambda_l d^l$, $r^{l+1} = r^l + \lambda_l dl^l$, solve $LL^T\gamma^{l+1} = r^{l+1}$ and calculate $\gamma_{l+1} = (\gamma^{l+1}, \gamma^{l+1})$, $\sigma_{l-1} = (r^{l+1}, \gamma^{l+1})$, $\beta_{l+1} = \sigma_{l+1}/\sigma_l$, $d^{l+1} = -\gamma^{l+1} + \beta_{l+1}d^l$, $d1^{l+1} = Ad^{l+1}$.

Here (.,.) is the usual inner product $(x, y) = x^Ty$. As stopping criterion we have used $\gamma_l = \epsilon^2\gamma_0$ and as incomplete factorization LL^T of $A = I - BB^T$ the MIC(2) algorithm was used. As initial approximation u^0 we have chosen $u^0 = (LL^T)^{-1}f$. Apparently, since LL^T resembles A, this is always a good choice.

We observe that the matrix A has only to be formed in order to calculate the incomplete factorization matrix LL^T, but multiplications Ad^{l+1} can be performed from $A = I - BB^T$.

We also observe that the vector γ^l may be used for storage of Ad^l. Hence, apart from the matrices B and L, only four vectors of size $mn/2$ have to be stored, and the total requirement of central storage is 7.5 mn.

The number of multiplications (and additions) per iteration is 14 mn. We remark that the MIC(2) method, applied directly on the natural (rowwise or columnwise) ordered mesh, demands 21 mn operations. The factorization work is 13 mn and 6 mn operations, respectively.

As follows from, for example, Axelsson [5], an upper bound for the number of iterations needed to decrease the relative residual error to $\| r^l \|/\| r^0 \| = \epsilon$ is

$$l = \text{int}[\tfrac{1}{2}\kappa^{1/2} \ln 2/\epsilon + 1].$$

where r^l is the residual vector at the lth iterative step and κ is the spectral condition number of $L^{-T}AL^{-1}$. This number is about the same for the original matrix, corresponding to a natural ordering, and for $I - BB^T$, both preconditioned by some MIC factorization algorithm. The number of iterations l that were actually needed in a computer run for $\epsilon = 10^{-6}$ is given in Table I for different values of m and n. The corresponding result for the usual ordering is given in Table II. A model Poisson problem was used. More general problems show the same behavior; see [4].

The total solution work per unknown with $\epsilon = 10^{-3}$ grows as in Fig. 2, where also the corresponding numbers for the natural ordering and for a complete factorization

TABLE I

The Number of Iterations for the MICCG(2) Method on the Reduced System, $\epsilon = 10^{-6}$

n	8	16		32			40		
m	8	8	16	8	16	32	10	20	40
l	4	5	6	5	7	9	6	8	10

TABLE II

The Number of Iterations for the MICCG(2) Method on the Usual Ordered System, $\epsilon = 10^{-6}$

n	8	16		32			40		
m	8	8	16	8	16	32	10	20	40
l	5	6	8	7	9	11	8	10	12

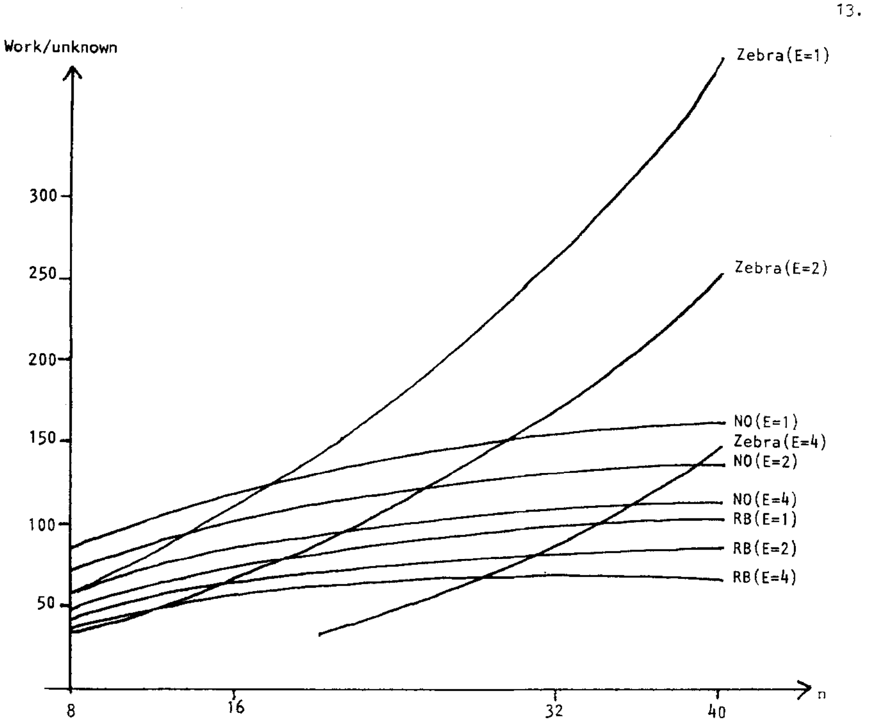

FIG. 2. The total work per unknown for the MICCG(2) method on the reduced red–black ordered system (RB), the MICCG(2) method on the natural ordered system (NO), and the Zebra algorithm described in [1] for different values of n and $E = n/m$, $\epsilon = 10^{-3}$ (in the iterative methods).

of a reduced system (a one-way dissection method, called Zebra method), as described in McDonald and Trimble [1], are given. In practice a relative accuracy of three decimal digits often suffices. Hence, although the latter method in general gives full precision the comparison may be adequate. In any case, the number of iterations grows only linearly with the number of correct digits.

4. CONCLUSION

We conclude that the red–black ordering may be advantageous also in connection with incomplete factorizations. Furthermore, since for the complete factorization the factorization work is $\mathcal{O}(nm^3)$ (actually $\sim\frac{1}{2}nm^3 - \frac{1}{4}m^3$), and the work in our method is $\mathcal{O}(nm^{1.5})$, $m \to \infty$, if the ratio m/n is constant, it is also superior to that method. Only for small systems and for very small bandwidths is the complete factorization slightly faster. The accuracy depends on the relative accuracy, ϵ in the iterative method. For instance, for $n = 40$, $m = 10$, and $\epsilon = 10^{-3}$, the number of operations per unknown in our method is only 63 multiplications and additions.

We finally remark that our method, incomplete factorization and a conjugate gradient method, as well as the Zebra method and similar methods are very favorable also on diagonally dominant nonsymmetric matrices, like those arising from (modified) upwind approximations of diffusion–convective equations, for arbitrary large Peclet numbers (see Axelsson and Gustafsson [7]).

References

1. A. E. McDonald and R. H. Trimble, *Soc. Petrol. Engrg. J.* (August 1977), 300–310.
2. O. Axelsson, *Comput. Math. Appl. Mech. Engrg.* **9** (1976), 123–137.
3. J. A. Meijerink and H. A. van der Vorst, *Math. Comput.* **31** (1977), 148–162.
4. I. Gustafsson, *BIT* **18** (1978), 142–156.
5. O. Axelsson, "On Preconditioning and Convergence Acceleration in Sparse Matrix Problems," CERN 74-10, Geneva, 1974.
6. O. Axelsson, *in* "Sparse Matrix Techniques, Copenhagen, 1976" (V. A. Barker, Ed.), Lecture Notes in Mathematics, Springer-Verlag, New York/Berlin, 1977.
7. O. Axelsson and I. Gustafsson, *J. Inst. Math. Appl.* **23** (1979), 321–337.

Received: November 9, 1978; revised: March 8, 1979

O. Axelsson and I. Gustafsson
Chalmers University of Technology,
Göteborg, Sweden