JOURNAL OF COMPUTATIONAL PHYSICS **32**, 263–269 (1979)

Note

An ILUCG Algorithm Which Minimizes in the Euclidean Norm*

This paper presents an algorithm which solves sparse systems of linear equations of the form $A\mathbf{x} = \mathbf{y}$, where A is non-symmetric, by the Incomplete LU Decomposition-Conjugate Gradient (ILUCG) method. The algorithm minimizes the error in the Euclidean norm $\| \mathbf{x}_i - \mathbf{x} \|_2$, where $\mathbf{x}_i$ is the solution vector after the ith iteration and $\mathbf{x}$ the exact solution vector. The results of a test on one real problem indicate that the algorithm is likely to be competitive with the best existing algorithms of its type.

As is well known, the Incomplete Cholesky–Conjugate Gradient method (ICCG) has been found to be very effective in the solution of sparse systems of linear equatons of the form $A\mathbf{x} = \mathbf{y}$ [1] with A symmetric. This method is a much improved version of the conjugate gradient method developed by Hestenes and Stiefel [2], for instead of iterating with the original matrix A, the approximate inverse of A is used. In the ICCG method, the approximate inverse is obtained by incomplete Cholesky LL^T decomposition where a pre-selected sparsity pattern, usually that of A, is forced upon the L and L^T matrices. Kershaw [3] further generalized the ICCG method to treat non-symmetric systems by using a general LU decomposition of matrix A, and this is known as the Incomplete LU decomposition–Conjugate Gradient (ILUCG) method.

In the derivation of the algorithm for treating non-symmetric matrices, Kershaw [3] transforms the system

$$A\mathbf{x} = \mathbf{y} \tag{1}$$

into

$$M\mathbf{x}' = \mathbf{y}' \tag{2}$$

where

$$M = L^{-1}AU^{-1};\ \mathbf{x}' = U\mathbf{x}$$

and

$$\mathbf{y}' = L^{-1}\mathbf{y} \tag{3}$$

He then constructs a version of the Conjugate Gradient algorithm for the system (2) which minimizes $\mathbf{x}_i' - \mathbf{x}'$ in the Euclidean norm:

$$\| \mathbf{x}_i' - \mathbf{x}' \|_2 = (\mathbf{x}_i' - \mathbf{x}', \mathbf{x}_i' - \mathbf{x}')$$

* This work was supported by United States Department of Energy Contract No. EY-76-C-02-3073 (M.P.) and Lawrence Livermore Laboratory (Livermore, Calif. 94550) Contract No. W-7405-ENG-48 (G.K.-P.).

0021-9991/79/080263–07$02.00/0

where $\mathbf{x}_i'$ is the solution vector after i iterations. As a result, $\mathbf{x}_i - \mathbf{x}$ of system (1) is minimized in the N norm

$$\| \mathbf{x}_i - \mathbf{x} \|_N = (\mathbf{x}_i - \mathbf{x}, N(\mathbf{x}_i - \mathbf{x}))^{1/2}$$

where $N = U^T U$, as is clear from the transformation $\mathbf{x}' = U\mathbf{x}$.

It is however quite easy to construct a conjugate gradient algorithm with incomplete LU decomposition which minimizes $\| \mathbf{x}_i - \mathbf{x} \|_2$ instead of $\| \mathbf{x}_i' - \mathbf{x}' \|_2$. We have constructed and tested such an algorithm and have found its properties sufficiently interesting to merit further investigations.

In order to compare the two algorithms formally we find it best to use the nomenclature of Hestenes [4]. Hestenes provides a scheme for constructing various conjugate gradient algorithms for non-symmetric matrices:

To solve a system $A\mathbf{x} = \mathbf{y}$ where A is a square non-singular matrix, choose a pair of positive Hermitian matrices H and K and define another positive Hermitian matrix N through $N = A^*HA$, where A^* is complex conjugate of A. Then a conjugate gradient algorithm which minimizes $\| \mathbf{x}_i - \mathbf{x} \|_N$ for all i among the algorithms of the form

$$\mathbf{x}_i = \mathbf{x}_0 + P_{i-1}(T)T(\mathbf{x} - \mathbf{x}_0),$$

where P is a polynomial in $T = KN$ of degree $i - 1$, is given by:

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{y} - A\mathbf{x}_0 \\ \mathbf{g}_0 &= A^*H \\ \mathbf{p}_0 &= K\mathbf{g}_0 \end{aligned} \tag{4.1}$$

and the following recursive relationships:

$$\alpha_i = \frac{(\mathbf{g}_i\,, K\mathbf{g}_i)}{(\mathbf{p}_i\,, N\mathbf{p}_i)} = \frac{(\mathbf{r}_i\,, H^*AKA^*H\mathbf{r}_i)}{(\mathbf{p}_i\,, N\mathbf{p}_i)} \tag{4.2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i \tag{4.3}$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A\mathbf{p}_i \tag{4.4}$$

$$\mathbf{g}_{i+1} = A^*H\mathbf{r}_{i+1} \tag{4.5}$$

$$\beta_i = \frac{(\mathbf{g}_{i+1}\,, K\mathbf{g}_{i+1})}{(\mathbf{g}_i\,, K\mathbf{g}_i)} = \frac{(\mathbf{r}_{i+1}\,, H^*AKA^*H\mathbf{r}_{i+1})}{(\mathbf{r}_i\,, H^*AKA^*H\mathbf{r}_i)} \tag{4.6}$$

$$\mathbf{p}_{i+1} = K\mathbf{g}_{i+1} + \beta_i \mathbf{p}_i \tag{4.7}$$

or

$$\mathbf{p}_{i+1} = KA^*H\mathbf{r}_{i+1} + \beta_i \mathbf{p}_i \tag{4.8}$$

Here $\mathbf{x}_0$ is the initial guess for $\mathbf{x}$.

In view of (4.5), the following identity holds:

$$(\mathbf{g}_i\,, K\mathbf{g}_i) = (\mathbf{r}_i\,, H^*AKA^*H\,\mathbf{r}_i) \tag{4.9}$$

For our choices of H and K, it will not be possible to use (4.5), so (4.8) will be used instead of (4.5) and (4.7) and (4.9) will be substituted into (4.2) and (4.6). We choose also A to be real so that A^* and H^* will be replaced by A^T and H^T, the transposes of A and H respectively. We shall also make use of the lower and upper triangular matrices L and U where it is implied that their product LU is a reasonable approximation of A.

The algorithm of Kershaw, equations (9a)–(9e) of [3], now results from the choice of

$$H = A^{-T}U^TUA^{-1}$$

and

$$K = (U^TU)^{-1}A^T(LL^T)^{-1}A(U^TU)^{-1} \tag{5.1}$$

From which follows:

$$N = A^THA = U^TU$$

and

$$T = KN = (U^TU)^{-1}A^T(LL^T)^{-1}A \tag{5.2}$$

and the subsidiary relationships:

$$KA^TH = (U^TU)^{-1}A^T(LL^T)^{-1} \tag{5.3}$$

and

$$H^TAKA^TH = (LL^T)^{-1}$$

to be used in (4.8) and (4.9).

With the assumption $LU \simeq A$, it follows from (5.2) that $T \simeq I$, where I is the unit matrix.

We now try to construct another algorithm which maintains the $T \simeq I$ relationship but which also results in $N = I$. This can be achieved with the choice:

$$H = (AA^T)^{-1}$$

and

$$K = A^T(U^TL^T)^{-1}(LU)^{-1}A \tag{6.1}$$

resulting in:

$$N = A^THA = I; \quad T = KN = K \tag{6.2}$$

$$KA^TH = A^T(LU)^{-T}(LU)^{-1}; \quad H^TAKA^TH = (LU)^{-T}(LU)^{-1} \tag{6.3}$$

where

$$(LU)^{-T} = [(LU)^{-1}]^T$$

The algorithm (4) then becomes:

$$\mathbf{r}_0 = \mathbf{y} - A\mathbf{x}_0 ; \qquad \mathbf{p}_0 = A^T(LU)^{-T} (LU)^{-1} \mathbf{r}_0 \tag{7.1}$$

$$\alpha_i = \frac{[(LU)^{-1} \mathbf{r}_i , (LU)^{-1} \mathbf{r}_i]}{(\mathbf{p}_i , \mathbf{p}_i)} \tag{7.2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i \tag{7.3}$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A \mathbf{p}_i \tag{7.4}$$

$$\beta_i = \frac{[(LU)^{-1} \mathbf{r}_{i+1} , (LU)^{-1} \mathbf{r}_{i+1}]}{[(LU)^{-1} \mathbf{r}_i , (LU)^{-1} \mathbf{r}_i]} \tag{7.5}$$

$$\mathbf{p}_{i+1} = A^T(LU)^{-T} (LU)^{-1} \mathbf{r}_{i+1} + \beta_i \mathbf{p}_i \tag{7.6}$$

In following the procedure of Hestenes [4], we were able to choose minimization in the Euclidean norm by forcing $N = I$ in (6.2), from which H followed uniquely, while K had to be composed in such a way that it both approximated I and resulted in simple and manageable expressions for KA^TH and H^TAKA^TH in (6.3). One may wonder, however, if it is possible to arrive at the same algorithm by simply transforming the original system as Kershaw has done. [See our Eqs. (1) through (3)]. The answer is yes, and the required transformation is given by $M\mathbf{x}' = \mathbf{y}'$, where $M = (LU)^{-1}A$; $\mathbf{x}' = \mathbf{x}$; and $\mathbf{y}' = (LU)^{-1}\mathbf{y}$, which substituted into Eqs. (3'a) through (3'e) of Ref. [3] leads to our algorithm. Since in this transformation $\mathbf{x}$ is identical to $\mathbf{x}'$ and Kershaw's algorithm (3') minimizes $\mathbf{x}'$ in the Euclidean norm, it follows that in our algorithm $\mathbf{x}$ must also be minimized in the same norm.

Our algorithm contains the same amount of computational work as that of equations (9') of Kershaw [3] so in this sense, the two are strictly comparable. As for the rate of convergence, we have compared the two algorithms on one problem only: a calculation of 2-dimensional transport of Tokamak plasmas using a dynamical grid method [5]. In this calculation a moving non-orthogonal grid system produced a simple 9 diagonal matrix A with variations in magnitude of not more than 10^4 among the matrix coefficients and the immediate sub and super diagonals having values around 0.5 after the diagonal elements have been normalized to 1.0. The dimension of the matrix A was 15×40. Both Kershaw's algorithm (K) and ours (P) were generated by an ILUCG generator program [6] and the values of $\epsilon = \| \mathbf{x}_i - \mathbf{x} \|_2 / \| \mathbf{x} \|_2$ compared. Here $\mathbf{x}$ is the 'exact' solution obtained after a large number of iterations when $\epsilon < 10^{-20}$ is satisfied, $\| \mathbf{x} \|_2 = (\sum_{j=1}^{J} x_j^2)^{1/2}$ where J is the dimension of the matrix A and

$$\| \mathbf{x}_i - \mathbf{x} \|_2 = (\mathbf{x}_i - \mathbf{x}, \mathbf{x}_i - \mathbf{x})^{1/2}$$

$\mathbf{x}_i$ being the 'solution' vector after i iterations.

Figures 1a and 1b show examples of two comparisons: the initial guess for $\mathbf{x}$ being much less accurate in (1a). In both cases, our algorithm (P) appears to give

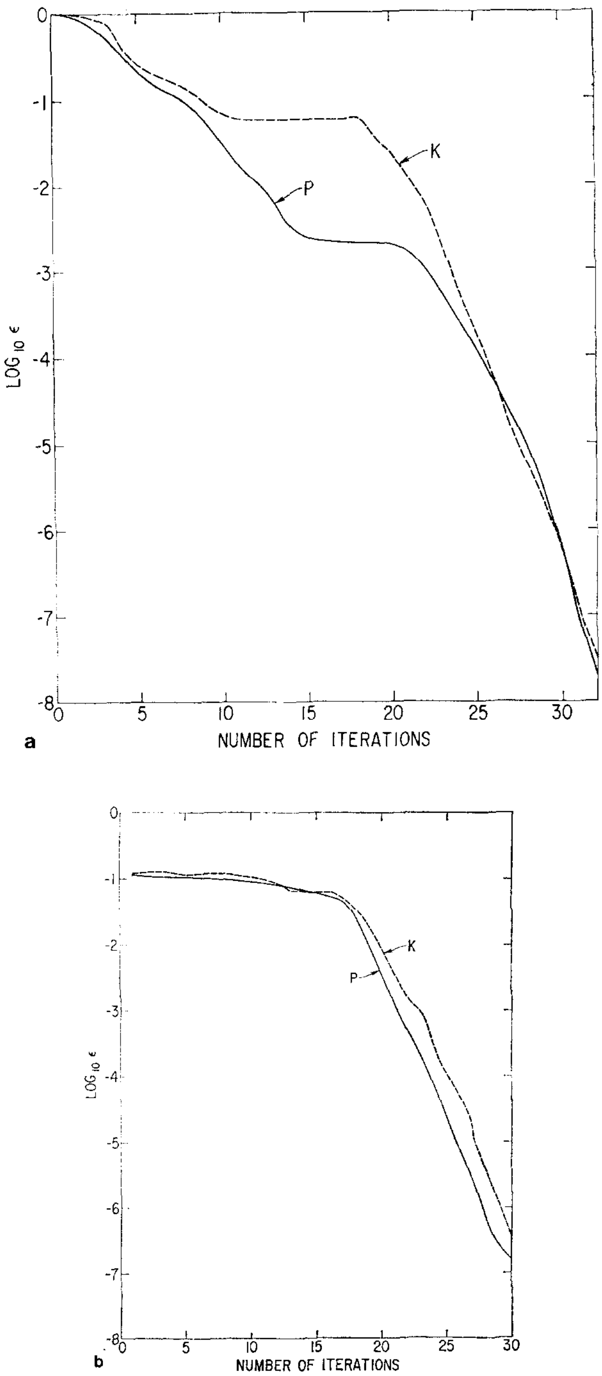

FIG. 1. Convergence curves using the Euclidean norm ε. Both the Kershaw algorithm (K) and our algorithm (P) (Eqs. (7.1)–(7.6) are shown. (a) applies to a case where the initial guess for the solution vector is less accurate than that of (b).

more accurate solutions down to $\epsilon \simeq 5 \times 10^{-5}$ beyond which the two are equally good. Furthermore, while the P-algorithm gives a monotonically decreasing ϵ, the K-algorithm does not.

Figure 2 gives a comparison of the maximum (infinity) norm given by:

$$\epsilon_{\max} = \frac{(\max \mid \mathbf{x}_i - \mathbf{x} \mid) * J}{\| \mathbf{x} \|_2}$$

In conclusion, we would like to emphasize that these results have only been tested on one problem only, hence it may or may not represent a typical behavior. The main purpose of this letter is to draw attention to our algorithm (7) and invite further comparisons.

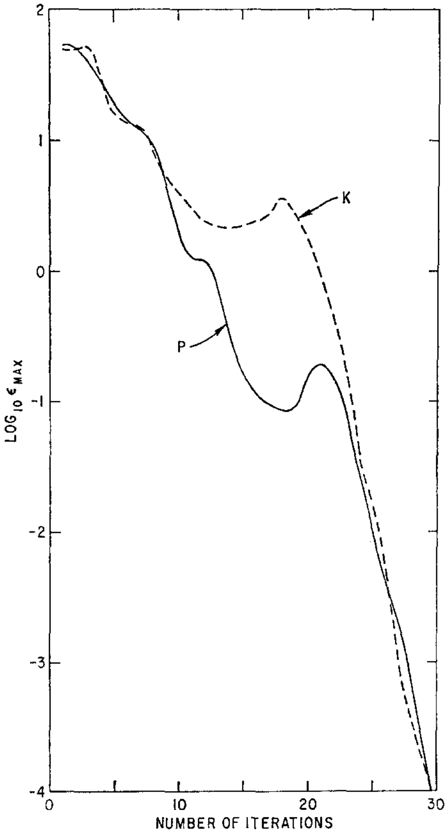

FIG. 2. Convergence curves using the maximum (infinity) norm

$$\epsilon_{\max} = \frac{(\max \mid \mathbf{x}_i - \mathbf{x} \mid) * J}{\| \mathbf{x} \|_2} .$$

REFERENCES

1. J. A. MEIJERINK AND H. A. VAN DER VORST, *Math. Comp.* **31**, 137 (1977), 148–162.
2. M. R. HESTENES AND E. STIEFEL, *J. Res. Nat. Bur. Standards* **49** (1952), 409.
3. D. S. KERSHAW, *J. Computational Physics* **26**, 1 (1978), 43–65.
4. M. R. HESTENES, "Process of Symposia on Applied Math. VI. Numerical Analysis," McGraw–Hill, New York, 1956.
5. S. C. JARDIN, S. P. HIRSHMAN, AND J. L. JOHNSON, Calculation of a two-dimensional transport of Tokamak plasmas using dynamical grid method, *in* "Proceedings of the 8th Conference on Numerical Simulation of Plasma, Monterey, June 1978."
6. G. KUO-PETRAVIC AND M. PETRAVIC, *Comput. Phys. Comm.* (1978).

RECEIVED: June 27, 1978; REVISED: October 31, 1978

M. PETRAVIC AND G. KUO-PETRAVIC

Plasma Physics Laboratory
Princeton University
Princeton, New Jersey 08540