On convergence of Inverse-based IC decomposition

Seiji FUJINO* and Akira SHIODE**

*Computing and Communications Center, Kyushu University
**Graduate School of Information Science and Electrical Engineering, Kyushu University
(E-mail: fujino@cc.kyushu-u.ac.jp)

The numerous variants of Cholesky decomposition and different dropping strategy to ensure sparsity and preferable cost of computation can be utilized to devise a number of preconditioners. Recently an efficient implementation of the incomplete LU decomposition derived from Crout version of incomplete LU decomposition (ILUC) for solving a linear system of equations with nonsymmetric matrix has been proposed by Na Li et al. [1] [2] [5]. ILUC decomposition is a useful preconditioner which estimates and utilizes the norms of the rows of $L^{-1}$ and the columns of $U^{-1}$ [3]. In addition, ILUC enables efficient implementation of rigorous dropping strategy based on estimating the norms of the inverse factors. Moreover ILUC decomposition allows the development of potentially more effective strategy for the usual incomplete Cholesky decomposition.

In this paper we extend Crout version of incomplete LU decomposition to that of incomplete Cholesky decomposition for solving linear systems with symmetric positive definite matrix.

The error due to the inverse of the factors is more significant than that of the factors themselves. Because when $A = U^T U$ and $A \approx U^T U$, using $U \approx U + X$ ($U^T = U^T + X^T$) with error matrix $X$, we can gain the following inverses

\[ U^{-1} = \tilde{U}^{-1} + X, \quad U^{-T} = \tilde{U}^{-T} + X^T. \]  

However, the preconditioned matrix is given by

\[ U^{-T} A U^{-1} = I + \tilde{U} X + X^T \tilde{U}^T + X^T A X. \]  

Here it is remarked that we cannot find the error matrix $\tilde{X}$ in the expression of preconditioned matrix. Accordingly we should estimate the error matrix $X$ of $U^{-1}$ instead of the error matrix $\tilde{X}$ of $U$ measured in the usual IC decomposition.

In case of the symmetric matrix, we use representation of $L = U^T$ for convenience. To minimize the error of $L^{-1}$ leads to the criterion that an entry $l_{j,k}$ should be dropped in the $k$th step if

\[ |l_{j,k}/\sigma_{j,k}| \|e_j e_k^T L^{-1}\|_\infty = |l_{j,k}|\xi_k/|\sigma_{j,k}| \leq \tau \]  

$e_k$ being the $k$th unit vector and $\tau$ a parameter for tolerance of dropping. Here $L^{-1}$ cannot be computed with low cost, so the idea is to estimate $\|e_j e_k^T L^{-1}\|_\infty$ by

\[ \|e_j e_k^T \tilde{L}^{-1} b\|_\infty \approx \|e_j e_k^T \tilde{L}^{-1} b\|_\infty /\|b\|_\infty , \]  

for a suitable vector $b$. For the purpose of recursive computation of $b$, we can obtain the following algorithm.

set $\xi_1 = 1/l_{1,1}$; $\nu_i = 0$ ($i = 1, \ldots n$)

for $k = 2, n$

$\text{temp}_+ = 1 - \nu_k$

$\text{temp}_- = -1 - \nu_k$

if $|\text{temp}_+| > |\text{temp}_-|$ then

$\xi_k = \text{temp}_+/l_{k,k}$

else

$\xi_k = \text{temp}_- /l_{k,k}$


end if
for \( j = k + 1, n \) \((l_{j,k} \neq 0)\)
\[ \nu_j = \nu_j + \xi_k l_{j,k} \]
end for
end for.

Here \( \xi_k \) is the approximation of \( e_j e_k^T \tilde{L}^{-1} \).

Numerical experiments will be presented. A linear system of equations were solved by the pre-conditioned CG method. As preconditioning, we adopt the IC decomposition with standard dropping technique and that with Inverse-based dropping technique. All computations were done in double precision floating point arithmetics, and performed on PC with CPU of 3.2GHz clock and main memory of two Gigabytes. Fortran compiler ver. 8.0, and optimum option -O3 was used. The right-hand side of vector \( b \) was imposed as all 1.0. The stopping criterion for successful convergence of the iterative method is less than \( 10^{-8} \) of the relative residual 2-norm \( \|r_{n+1}\|_2/\|r_0\|_2 \). The coefficient matrix \( A \) was normalized by diagonal scaling. In all cases the iteration of CG method was started with the initial guess solution \( x_0 = 0 \). The maximum iteration is set as same as the dimension of each matrix. Tolerance values \( \tau \) were varied from 0.005 up to 0.15 with the interval of 0.005. In total 30 cases were examined for dropping strategies and matrices, respectively. The shifted parameter \( \alpha \) of linear systems \( A + \alpha I \) was fixed as 0.10, where \( I \) denotes the unit matrix. Five test matrices were derived from Matrix-Market database of sparse matrices[4] and problem which stemmed from a realistic analysis. Table 1 shows description of four test matrices. In Table “total nnz” means the total number of nonzero entries of each matrix. Similarly in Table “ave. nnz” means average number of nonzero entries per one row of each matrix.

<table>
<thead>
<tr>
<th>matrix</th>
<th>dimensions</th>
<th>total nnz</th>
<th>ave. nnz</th>
<th>analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3DKQ4M2</td>
<td>90,449</td>
<td>2,455,670</td>
<td>27.1</td>
<td>cylindrical shells</td>
</tr>
<tr>
<td>S3DKT3M2</td>
<td>90,449</td>
<td>1,921,955</td>
<td>21.2</td>
<td>cylindrical shells</td>
</tr>
<tr>
<td>CT20STIF</td>
<td>52,329</td>
<td>1,375,396</td>
<td>26.3</td>
<td>engine block</td>
</tr>
<tr>
<td>ENGINE</td>
<td>143,571</td>
<td>2,424,822</td>
<td>16.9</td>
<td>engine head</td>
</tr>
<tr>
<td>T_400000</td>
<td>417,524</td>
<td>6,053,860</td>
<td>14.5</td>
<td>structural analysis</td>
</tr>
</tbody>
</table>

In Figure 1(a),(b) we present the numerical results of the shifted ICCG methods with the standard dropping and that with Inverse-based dropping for matrices S3DKQ4M2 and S3DKT3M2, respectively. The vertical axis means the total computation time in seconds included preconditioning time and CG iteration time, and the horizontal axis means the tolerance values. Color bar in red means computation times of the shifted ICCG method with Inverse-based dropping, and color bar in blue means computation times of that with the standard dropping.

We restrict our attention to robustness for tolerance value and stability of convergence in place of computation times. From Figure 1(a), for larger tolerance of 0.095 the shifted ICCG method with the standard dropping could not converge. On the other hand, the shifted ICCG method with Inverse-based dropping converged over the whole range of tolerance. Furthermore the much computation times of the former require sometimes as tolerances of 0.050, 0.070 and 0.095.

From Figure 1(b), it can be seen that at tolerances \( \tau \) of 0.125, 0.130, 0.145 and 0.150, the shifted ICCG method with the standard dropping diverged. In contrast, the shifted ICCG method with Inverse-based dropping converges successfully. Moreover the former requires much computation times irregularly at tolerances of 0.095, 0.120, 0.135 and 0.140. In contrast, the latter converged smoothly over the whole range of tolerance. The Inverse-based dropping does not appear slightly to be competitive as for the computation times at the optimal tolerance \( \tau_{opt} \), because it often requires more fill-in in order to yield convergent iterations.
Figure 1 Computation times versus tolerance value of shifted ICCG method with the standard dropping and that with Inversed-based dropping for matrix S3DKQ4M2 and S3DKT3M2.

In Figure 2(a)-(b) we present history of relative residual of the shifted CG method with IC and ib IC decompositions for matrix S3DKT3M2. Figure 2(a) shows the results of the CG method with usual IC decomposition at tolerances of 0.09, 0.095 and 0.10. Similarly Figure 2(b) depicts the results of the shifted CG method with ib IC decomposition at the above same tolerances. From Figure 2 the observations are made as follows:

- In case of usual IC decomposition, much computation times are required at tolerance of 0.095 only compared with those of tolerance of 0.09 and 0.10. This means that error matrix $\tilde{X}$ of $U$ measured in the usual IC decomposition has possibility of misleading estimation.
- In case of ib IC decomposition, computation times for three tolerances are approximately as same as each other.

Additionally, in Figure 3 we exhibit the computation times of the shifted ICCG methods with two types of dropping when tolerance values are varied similarly. Figure 3 illustrates that at tolerances $\tau$ of 0.090 and 0.095 the shifted ICCG method with Inverse-based dropping converges quite nicely.

![Graphical representation of computation times and residual history for matrix S3DKT3M2 at different tolerances](image1.png)

(a) IC decomposition (b) ib IC decomposition

Figure 2 History of relative residual of the CG method with IC and ib IC decompositions for matrix S3DKT3M2 at tolerance values of 0.09, 0.095 and 0.10.

Figures 4(a),(b) show computation times versus tolerance and history of relative residual of the CG method with shifted IC and shifted ib IC decompositions at tolerance = 0.090 for matrix $T_{400000}$. Obviously we notice that ib IC decomposition yields better result than the usual IC decomposition.

Consequently, we concluded by presenting graphically the timing results for tested four matrices above when tolerance values are varied that Inverse-based dropping strategy is superior to the standard
(a) Computation times versus tolerance (tolerance = 0.09)
(b) History of relative residual

Figure 4 Computation times versus tolerance and history of relative residual of the CG method with shifted IC and shifted ib IC decompositions for matrix $T_{400000}$.

dropping from the viewpoint of robustness for tolerance value and stability of convergence. It may be an alternative promising option if the shifted ICCG method with the standard dropping fail, though it is unlikely to be the fastest choice of convergence in general cases.

References


