On convergence of Inverse-based IC decomposition

Seiji FUJINO and Akira SHIODE  
(Kyushu University)

Outline of my talk

1. Difference between standard Incomplete Cholesky and Inverse-based IC decompositions

2. Extension of Crout version of ILU decomposition to Inverse-based IC decomposition for solving a linear systems with symmetric positive definite matrices

3. Numerical experiments

4. Concluding remarks
Background and Objectives

- ILUC(Crout) decomposition for solving a linear system of equations with nonsymmetric matrix has been proposed by Li et al. in 2003.

- ILUC decomposition enables efficient implementation of rigorous dropping strategy based on estimating the norms of the inverse factor $L^{-1}$.

- We extend ILUC decomposition to IC decomposition for solving linear systems with symmetric positive definite matrix.

Preconditioned equation

- Solved equation

\[ Ax = b \]  

- Preconditioned equation

\[ (U^{-T} A U^{-1})(U x) = U^{-T} b \]

$U$: upper triangular matrix, $T$: transpose

- The error due to the inverse of the factors will be more significant than that of the factors themselves.
Error due to inverse of factor

1. When \( A = \bar{U}^T \bar{U} \) and \( A \approx U^T U \), using \( U \equiv \bar{U} + \tilde{X} \)  
\((U^T = \bar{U}^T + \tilde{X}^T)\) with error matrix \( \tilde{X} \),
we can gain the following inverses

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

2. However, the preconditioned matrix is given by

\[
U^{-T} A U^{-1} = I + \bar{U} X + X^T \bar{U} + X^T A X,
\]

Here we cannot find the error \( \tilde{X} \).

3. Then we estimate the error \( X \) of \( U^{-1} \) instead of the error matrix \( \tilde{X} \) of \( U \) measured in the usual IC decomposition.

Inverse-based dropping

1. We use representation of \( L = U^T \) for convenience.

2. Entry \( \overline{a_{i,i}} \) is corresponded to the diagonal entry of the
   \( i \)th row of \( A \), and updated in the decomposition process.

\[
L_k = \begin{pmatrix}
0 & \cdots & 0 \\
l_{2,1} & \ddots & \cdots \\
\vdots & \ddots & \ddots \\
\vdots & \ddots & \ddots \\
l_{k,k} & \ddots & \ddots & \cdots & a_{k+1,k+1} \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & \cdots & l_{k+1,k+1} \\
\cdots & \ddots & \ddots & \ddots & \ddots \\
l_{n,1} & \cdots & l_{n,k} & 0 & \cdots & 0 \\
\end{pmatrix}
\]
Norm estimation of $L_k^{-1}$

1. We suppose that fill-in $l_{j,k}$ ($j > k$) is dropped when we update the $k$ column of matrix $L$.

2. Lower matrix $L_k$ is presented as below. $e_i$ ($1 \leq i \leq n$) is the unit vector with the $i$th row of 1.

$$L_k = \bar{L}_k - l_{j,k} e_j e_k^T.$$ \hfill (6)

3. For $j > k$, from $\bar{L}_k e_j = \overline{a_{j,j}} e_j$, we can gain

$$L_k = \bar{L}_k - l_{j,k} e_j e_k^T = \bar{L}_k (I - l_{j,k} e_j e_k^T / \overline{a_{j,j}}).$$ \hfill (7)

Norm estimation of $L_k^{-1}$ (contd.)

1. As a result, we can represent $L_k^{-1}$ as

$$L_k^{-1} = (I - l_{j,k} e_j e_k^T / \overline{a_{j,j}})^{-1} \bar{L}_k^{-1}$$

$$= \bar{L}_k^{-1} + l_{j,k} e_j e_k^T \bar{L}_k^{-1} / \overline{a_{j,j}}.$$ \hfill (8)

2. In dropping strategy, it is preferable for us to utilize

$$||l_{j,k} e_j e_k^T \bar{L}_k^{-1} / \overline{a_{j,j}}||_\infty = |l_{j,k} / \overline{a_{j,j}}| ||e_j e_k^T \bar{L}_k^{-1}||_\infty$$ \hfill (9)

3. This leads to rough norm estimation for the $k$th row of $L_k^{-1}$. 
**Maximum norm of $A$**

1. We define maximum norm of $A$ as

$$\|A\|_{\infty} = \max_{x \neq 0} \frac{\|Ax\|_{\infty}}{\|x\|_{\infty}} = \max_i \sum_{j=1}^{n} |a_{i,j}|$$

2. Moreover we can approximate $\|e_j e_k^T \bar{L}_k^{-1}\|_{\infty}$ with a certain vector $b$ ($\neq 0$)

$$\|e_j e_k^T \bar{L}_k^{-1}\|_{\infty} = \max_{b \neq 0} \frac{\|e_j e_k^T \bar{L}_k^{-1} b\|_{\infty}}{\|b\|_{\infty}}.$$ 

**Decision of vector $b$**

1. For estimating of $\|e_j e_k^T \bar{L}_k^{-1}\|_{\infty}$, it is sufficient that we decide a vector $b (\neq 0)$ which satisfies equation (11).

2. It is known that, at the definition of maximum norm, when at $i = m$

$$\sum_{j=1}^{n} |a_{i,j}|$$

is maximum, the $j$th component of vector $b$ which gives the maximum value, is written as

$$b_j = \text{sign}(a_{m,j}). \quad (j = 1, 2, \ldots, n)$$

3. Here, $\text{sign}$ means the sign function as

$$\text{sign}(t) = +1 \ (t \geq 0), \ \text{sign}(t) = -1 \ (t < 0)$$
1. To compute directly $\bar{L}^{-1}$ is too expensive.

2. Then we search approximately for a vector $b$ so as to make $\|e_j e_k^T \bar{L}^{-1} b\|_\infty$ as large as possible in the vector $b$ as

$$b = (-b_1, -b_2, \ldots, -b_n)^T, \quad (b_1, b_2, \ldots, b_n = \pm 1) \quad (15)$$

On convergence of Inverse-basedIC decomposition – p.11/26

1. The following estimation holds approximately using the vector $b$ which makes $\|e_j e_k^T \bar{L}^{-1} b\|_\infty$ as large as possible.

$$\|e_j e_k^T \bar{L}^{-1}\|_\infty \approx \frac{\|e_j e_k^T \bar{L}^{-1} b\|_\infty}{\|b\|_\infty} \quad (16)$$

2. R.H.S. of this equation can be estimated the $k$th component of the solution $\bar{L} \xi = b$.

3. The $\xi_k$ component of vector $\xi$ is gained with the $k$th component of vector $b$ of R.H.S. of the above equation

$$\xi_k = (\xi_1, \xi_2, \ldots, \xi_k, 0, \ldots, 0)^T, \quad b_k = (b_1, b_2, \ldots, b_k, 0, \ldots, 0)^T$$

$$\xi_k = (b_k - e_k^T \bar{L}^{-1} \xi_{k-1})/l_{k,k} \quad (17)$$

On convergence of Inverse-basedIC decomposition – p.12/26
Estimation of $L^{-1}$

1. Finally, we can show the algorithm which estimates the norm of $||e_j e_k^T \bar{L}^{-1}||_\infty$.

2. $\xi_k$ corresponds to $||e_j e_k^T \bar{L}^{-1}||_\infty$. Also $\nu_k$ corresponds to $e_k^T \bar{L}_{k-1} \xi_{k-1}$.

   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

Inverse-based dropping

1. In inverse-based dropping, when the following equation holds, fill-in $l_{j,k}$ is dropped in the decomposition process.

   \begin{equation}
   |l_{j,k}/a_{j,j}||e_j e_k^T \bar{L}^{-1}||_\infty = |l_{j,k}|||\xi_k||/|a_{j,j}| \leq \tau
   \end{equation}
Numerical Experiments

1. A linear system of equations was solved by the preconditioned CG method.
2. We adopt the IC with standard dropping and that with Inverse-based dropping.
3. The right-hand side of vector $b$ was imposed as all 1.0.
4. The stopping criterion for convergence is less than $10^{-8}$ of the relative residual 2-norm $\frac{\|r_{n+1}\|_2}{\|r_0\|_2}$.
5. In all cases the iteration was started with $x_0 = 0$.
6. The maximum iteration is set as same as the dimension of each matrix.

Analytic conditions

1. Tolerance values $\tau$ were varied from 0.005 up to 0.15 with the interval of 0.005.
2. In total 30 cases were examined for dropping strategies and matrices, respectively.
3. The shifted parameter $\alpha$ of linear systems $A + \alpha I$ was fixed as 0.10, where $I$ denotes the unit matrix.
Test matrices

1. Four test matrices were derived from Matrix-Market database and matrix $T_{400000}$ was stemmed from a realistic analysis.

2. Table shows description of four test matrices.

3. In Table "total nnz" means the total number of nonzero entries of each matrix. Similarly "ave." means average number of nonzero entries per one row of each matrix.

<table>
<thead>
<tr>
<th>matrix</th>
<th>$n$</th>
<th>total nnz</th>
<th>ave.</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>

On convergence of Inverse-basedIC decomposition – p.18/26

Results for S3DKQ4M2

- Vertical axis means the total time preconditioning and CG iteration time, and horizontal axis means tolerance.
- Color bars in red and blue mean times of ICCG method with Inverse-based and standard dropping, respectively.
Results for S3DKQ4M2 (contd.)

- For larger tolerance of 0.095 ICCG method with the standard dropping could not converge.
- On the other hand, ICCG method with Inverse-based dropping converged over the whole range of tolerance.
- The much computation times of the former require sometimes as tolerances of 0.050, 0.070 and 0.095.

Results for S3DKT3M2

- At tolerances $\tau$ of .125, .13, .145 and .15, ICCG with standard dropping diverged. In contrast, ICCG with Inverse-based dropping converged.
- The former requires much computation times irregularly at tolerances of .095, .12, .135 and .14. In contrast, the latter converged smoothly over the whole range.
Figure (a) shows the results of CG method with usual IC decomposition at tolerances of .09 (red), .095 (green) and .1 (blue).

Similarly Figure (b) depicts the results of CG method with ib_IC decomposition at the above same tolerances.

In case of usual IC decomposition, much times are required at tolerance of 0.095 only compared with those of tolerance of .09 and .1. This means that error matrix $\tilde{X}$ of $U$ measured in the usual IC has possibility of misleading estimation.

In case of ib_IC decomposition, times for three tolerances are approximately as same as each other.
Figure (a) illustrates that at tolerances $\tau$ of .09 and .095 ICCG method with the standard dropping diverged.

Contrastively general behaviour of ICCG method with Inverse-based dropping converges nicely.

---

Figure (b) illustrates that the range of Inverse-based IC is larger than that of usual IC.
Results for T_400000

- Figures (a),(b) show times vs. tol. and history of residual of CG method with \_IC and ib\_IC decompositions at tol. = .09 for T_400000.
- Obviously we notice that ib\_IC decomposition yields better result than the usual IC decomposition.

(a)CPU times vs. tol.  (b)history of residual (tol.=.09)

Concluding remarks

- We concluded that Inverse-based dropping is superior to the standard dropping from the viewpoint of robustness for tolerance value and stability of convergence.
- It may be an alternative promising option if ICCG method with the standard dropping fails, though it is unlikely to be the fastest choice of convergence in general cases.