An Adaptive Strategy of Improving Convergence of IDR(s)-Jacobi Method

Kusakabe, Yuzo
Department of Computer Science and Communication Engineering, Kyushu University

Fujino, Seiji
Research Institute for Information Technology, Kyushu University

日下部, 雄三
九州大学大学院 システム情報科学府情報工学専攻

藤野, 清次
情報基盤研究開発センター

https://doi.org/10.15017/17886
An Adaptive Strategy of Improving Convergence of IDR(s)-Jacobi Method

Yuzo KUSAKABE* and Seiji FUJINO**

(Received December 11, 2009)

Abstract: The conventional Jacobi method is well known to be a simple one of stationary iterative methods for solving a linear system of equations, but it converges slowly and lacks of robustness of convergence. Therefore, we improve this Jacobi method by means of Induced Dimension Reduction (IDR) Theorem proposed by Sonneveld et al. in 2008 in order to gain robustness of convergence. That is, we devise the IDR-based Jacobi method with relaxed parameter $\omega_n$ and its adaptive and cyclically adaptive tuning. Many numerical experiments verifies effectiveness and robustness of the IDR-based Jacobi methods. Characteristics of convergence of some IDR-based Jacobi methods may be useful for a variety of analysis in the field of applications.

Keywords: IDR Theorem, Jacobi method, Relaxation, Cyclic adaptive tuning

1. Introduction

The conventional Jacobi method by Young is known to be a simple one of stationary iterative methods for solving a linear system of equations $Ax = b$. Here, $A \in \mathbb{R}^{N \times N}$ is a nonsymmetric coefficient matrix, and $x, b$ are solution and right-hand side vectors of order $N$, respectively. As well known, the conventional Jacobi method converges more slowly than other stationary iterative methods, i.e., so-called Gauss-Seidel and SOR (Successive Over Reluation) methods. Moreover it lacks of robustness of convergence. However, it is very easy to parallelize the Jacobi method on parallel computations. Because its algorithm does not have dependency for entries of matrix at all.

In 2008, P. Sonneveld and M. van Gijzen proposed the IDR Theorem. Furthermore they verified theoretically and experimentally effectiveness of IDR(s) method based on the IDR Theorem with parameter $s$ which means the dimension of a pre-chosen subspace. Moreover one of the authors proposed a crucial and thought-provoking iterative method. He referred to AGS (Accelerated Gauss-Seidel) method. However, it was too ingenious. Therefore we improved his idea so as to be applied to solve practical problems. We referred to them as IDR-based Gauss-Seidel and SOR methods. We abbreviated IDR(s)-GS and IDR(s)-SOR methods, respectively.

In this paper, we examine validity of IDR-based Gauss-Seidel and SOR methods through a number of numerical experiments. Moreover we propose also relaxed and adaptive version of IDR-based Jacobi method as for parameter $\omega_n$. We abbreviate relaxed, adaptive and cyclically adaptive IDR(s)-Jacobi methods, respectively.

2. IDR Theorem

We focus on iterative solution of linear systems $Ax = b$. Starting from initial guess $x_0$ of the solution, initial residual $r_0$ is defined as $r_0 = b - Ax_0$. We define the residual vector at $n$th iteration as $r_n$.

Let $\mathcal{G}_0 = \text{span}\{r_0, Br_0, ..., B^{N-1}r_0\}$, where $B$ is any matrix in $\mathbb{R}^{N \times N}$. So $\mathcal{G}_0$ is the Krylov space corresponding to $B$ and $r_0$. Let $\mathcal{G}_{j+1} = B(\mathcal{G}_j \cap \text{Null}(P^T))$ for $j = 0, 1, 2, ..., r_{n+1} \in \mathcal{G}_{j+1}$, and $v \in \mathcal{G}_j \cap \text{Null}(P^T)$, where $P = (p_1, p_2, ..., p_s)$. Then, the residual converges by IDR theorem if the spectral radius $\rho(B) < 1$.

In IDR(s) method, the matrix $B$ is defined as $(I - \omega_n A)$. The parameter $\omega_n$ is computed every $(s + 1)$th step such that the updated residuals are in $\mathcal{G}_{j+1}$. On the other hand, in AGS method, the matrix $B$ is defined as the iteration matrix of Gauss-Seidel method.

3. Relaxed IDR(s)-Jacobi method

We consider giving so-called iteration matrix of Jacobi method to the matrix $B$. The so-called iteration matrix $B = NM^{-1}$ of conventional Jacobi and IDR(s)-Jacobi method is defined as follows:

$$M = D,$$

$$N = A - D.$$

We devise acceleration of IDR(s)-Jacobi method in which the diagonal matrix $D$ is modified by param-
eter $\omega$ as (3), (4). We refer to this modification as relaxed IDR$(s)$-Jacobi method.

\begin{align}
M &= \frac{1}{\omega} D, \\
N &= A - \frac{1}{\omega} D. 
\end{align}

Hence, the residual vector $r_{n+1}$ of relaxed IDR$(s)$-Jacobi method is defined as the below equation. The vector $v$ is a combination of the residuals to satisfy $v \in \mathcal{G}_j$. The coefficients $\gamma_i (i = 1, 2, \ldots, s)$ are computed so that $P^T v = 0$.

\begin{align}
r_{n+1} &= (A - \frac{1}{\omega} D)(\frac{1}{\omega} D)^{-1} v, \\
v &= r_n + \sum_{i=1}^{s} \gamma_i \Delta r_{n-i}.
\end{align}

We show an algorithm of relaxed IDR$(s)$-Jacobi method with parameter $s$ as below. $e_k$ is $s$-dimension vector that $k$th value is 1 and the others are 0. A given $\omega$ is used at the equations (7)-(10).

**Algorithm of relaxed IDR$(s)$-Jacobi method**

Let $x_0$ be an initial solution, put $r_0 = b - Ax_0$, $P = (p_1, p_2 \ldots p_s) \in \mathbb{R}^{N \times s}$, set $\gamma_0 = 0$, $\omega$ is given,

\begin{align}
\text{initial loop : build matrices } \\
E &= (dr_1, dr_2 \ldots dr_s), Q = (dx_1, dx_2 \ldots dx_s) \\
\text{by relaxed IDR(1) Jacobi method.}
\end{align}

for $n=0, 1, \ldots, s-1$

\begin{align}
s_n &= (\frac{1}{\omega} D)^{-1}(r_n - \gamma_n dr_n), \\
dx_{n+1} &= x_n - \gamma_n dx_n, \\
rd_{n+1} &= -(\frac{1}{\omega} D - A)s_n - r_n, \\
r_{n+1} &= r_n + dr_{n+1}, \\
x_{n+1} &= x_n + dx_{n+1}, \\
\text{if } ||r_{n+1}||_2 / ||r_0||_2 \leq \epsilon \text{ stop}
\end{align}

$\gamma_{n+1} = (p_1, r_{n+1})/(p_1, dr_{n+1}),$

\begin{align}
Ec_{n+1} &= dr_{n+1}, Qe_{n+1} = dx_{n+1} \\
\text{end for.}
\end{align}

$M = P^T E, f = P^T r_s, n = s, k = 1,$

\begin{align}
\text{end loop }
\end{align}

while $||r_{n+1}||_2 / ||r_0||_2 > \epsilon$

solve $c$ from $Mc = f$, 

\begin{align}
s_n &= (\frac{1}{\omega} D)^{-1}(r_n - \gamma_n dr_n), \\
dx_{n+1} &= x_n - Qc, \\
rd_{n+1} &= -(\frac{1}{\omega} D - A)s_n - r_n
\end{align}

$r_{n+1} = r_n + dr_{n+1}, x_{n+1} = x_n + dx_{n+1}, \\
Ec_k = dr_{n+1}, Qe_k = dx_{n+1}, \\
M e_k = P^T dr_{n+1}, f = f + Me_k,$

$n = n + 1, k = k + 1,$

end while.

**4. IDR$(s)$-Jacobi with adaptive tuning**

In the iteration procedure of the relaxed IDR$(s)$-Jacobi method, the residual $r_{n+1}$ is updated adaptively as the following equation.

\begin{align}
r_{n+1} &= (\frac{1}{\omega} D - A)(\frac{1}{\omega} D)^{-1}(r_n - Ec) \\
&= (I - \omega D^{-1} A)(r_n - Ec).
\end{align}

Therefore, we can compute the optimum $\omega_n$ which minimizes the residual 2-norm $||r_{n+1}||_2$ as below.

\begin{align}
\omega_n &= \frac{(D^{-1} A t_n, t_n)}{(D^{-1} A t_n, D^{-1} A t_n)}, t_n = r_n - Ec
\end{align}

As well known, the optimum parameter $\omega_n$ of the conventional SOR method is estimated at the range of $[0.0, 2.0]$. However, $\omega_n$ computed by eq. (12) does not have upper and lower limitations. Then we limited as $\omega_n \in [0.1, 2.0]$. We show an algorithm of IDR$(s)$-Jacobi method with adaptive tuning. The algorithm of IDR$(s)$-Jacobi method with adaptive tuning differs from that of relaxed IDR$(s)$-Jacobi method in that the optimum $\omega_n$ is computed at the equations (13), (16).

**Algorithm of IDR$(s)$-Jacobi method with adaptive tuning**

Let $x_0$ be an initial solution, put $r_0 = b - Ax_0$, $P = (p_1, p_2 \ldots p_s) \in \mathbb{R}^{N \times s}$, set $\gamma_0 = 0$, $\omega$ is given,

\begin{align}
\text{initial loop : build matrices } \\
E &= (dr_1, dr_2 \ldots dr_s), Q = (dx_1, dx_2 \ldots dx_s) \\
\text{by relaxed IDR(1) Jacobi method.}
\end{align}

for $n=0, 1, \ldots, s-1$

compute $\omega_n$ that minimizes $||r_{n+1}||_2$, 

\begin{align}
s_n &= (\frac{1}{\omega} D)^{-1}(r_n - \gamma_n dr_n), \\
dx_{n+1} &= x_n - \gamma_n dx_n, \\
rd_{n+1} &= -(\frac{1}{\omega} D - A)s_n - r_n, \\
r_{n+1} &= r_n + dr_{n+1}, x_{n+1} = x_n + dx_{n+1}.
\end{align}
the following equations.

\[ \begin{align*}
\gamma_{n+1} & = (p_1, r_{n+1})/(p_1, dr_{n+1}), \\
Ee_{n+1} & = dr_{n+1}, \ Qe_{n+1} = dx_{n+1},
\end{align*} \]

end for,

\[ M = P^T E, \ f = P^T r_s, \]

\[ n = s, k = 1, \ {\text{[main loop]}} \]

while \[ ||r_{n+1}||_2/||r_0||_2 > \epsilon \]

solve \( c \) from \( M c = f \),

compute \( \omega_n \) that minimizes \( ||r_{n+1}||_2 \), \( s_n = (\frac{1}{\omega_n}) D^{-1} (r_n - Ec), \)

\[ dx_{n+1} = s_n - Qe, \]

\[ dr_{n+1} = -(\frac{1}{\omega_n} D - A) s_n - r_n, \]

\[ r_{n+1} = r_n + dr_{n+1}, \ x_{n+1} = x_n + dx_{n+1}, \]

\[ Ee_k = dr_{n+1}, \ Qe_k = dx_{n+1}, \]

\[ M e_k = P^T dr_{n+1}, \ f = f + M e_k, \]

\[ n = n + 1, k = k + 1, \]

if \( k > s \) then \( k = 1 \),

end while.

5. IDR(s)-Jacobi with cyclically adaptive tuning

In IDR(s)-Jacobi method with adaptive tuning, we compute the optimum \( \omega_n \) every step. However, that requires another matrix-vector product for computing \( \omega_n \). Moreover, the updated residuals may not be in \( G_{j+1} \) because of the matrix \( B = I - \omega_n D^{-1} A \) varying.

Then, we compute \( \omega_n \) every \( (s+1) \)th step and update \( r_{n+1} \) as below in order to reducing matrix-vector products and to make the updated residuals in \( G_{j+1} \). That is, we replace the equation (16) by the following equations.

\[ \begin{align*}
\text{if mod} (n, s+1) & = 0 \text{ then} \\
& \text{compute } \omega_n \text{ that minimizes } ||r_{n+1}||_2, \\
\text{else} \\
& \omega_n = \omega_{n-1},
\end{align*} \]

6. Numerical experiments

All computations were done in double precision floating point arithmetics of Fortran90, and performed on Nehalem with CPU of Intel Xeon X5570, clock of 2.93GHz, main memory of 24GB and OS of RedHat Enterprise Linux 5.2. Optimum option -O3 was used. Stopping criterion for successful convergence of conventional Jacobi, Gauss-Seidel, SOR and IDR(s)-Jacobi methods is less than \( 10^{-8} \) of the relative residual 2-norm \( ||r_{n+1}||_2/||r_0||_2 \). In all cases the iteration was started with the initial guess solution \( x_0 = 0 \). All test matrices were normalized with diagonal scaling. The maximum iteration was fixed as 10000. We show the optimum parameter \( \omega \) of SOR and relaxed IDR(s)-Jacobi method in which \( \omega \) varied from 0.5 until 2 at the interval of 0.1. The parameter \( s \) varied as 1, 2, 4 and 8. Table 1 shows the characteristics of test matrix derived from Florida sparse matrix collection.\(^1\)

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The characteristics of test matrix.</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix</td>
<td>( N )</td>
</tr>
<tr>
<td>epb1</td>
<td>14,734</td>
</tr>
<tr>
<td>epb2</td>
<td>25,228</td>
</tr>
<tr>
<td>poisson3Da</td>
<td>13,514</td>
</tr>
<tr>
<td>poisson3Db</td>
<td>85,623</td>
</tr>
<tr>
<td>rae5sky2</td>
<td>3,242</td>
</tr>
<tr>
<td>rae5sky3</td>
<td>21,200</td>
</tr>
<tr>
<td>memplus</td>
<td>17,758</td>
</tr>
<tr>
<td>wang4</td>
<td>26,068</td>
</tr>
</tbody>
</table>

Table 2 presents numerical results of conventional Jacobi, Gauss-Seidel, SOR, relaxed, adaptive and cyclically adaptive IDR(s)-Jacobi methods. In Table 2, “TRR” means values of True Relative Residual of \( ||b - Ax_{n+1}||_2/||b - Ax_0||_2 \) for the converged solutions \( x_{n+1} \). “max” and “div.” also mean that the iterative method did not converge until maximum iteration, and that the residual \( r_{n+1} \) diverged, respectively. The bold figure means the fastest case for each matrix.

From Table 2, we can see that relaxed IDR(s)-Jacobi and cyclically adaptive IDR(s)-Jacobi methods outperform with adaptive IDR(s)-Jacobi method in view of convergence rate. Besides, adaptive and cyclically adaptive IDR(s)-Jacobi methods converge more robustly than relaxed IDR(s)-Jacobi method.

Figures 1, 2 depict history of relative residual 2-norm of relaxed, adaptive and cyclically adaptive IDR(s)-Jacobi methods with the optimum \( \omega \) for matrices epb1 and poisson3Db. From Figs. 1, 2, we can observe that cyclically adaptive IDR(s)-Jacobi methods converge faster than the other methods by reducing computational cost and less iterations.

Figures 3, 4 exhibit variation of \( \omega \) of three kinds of IDR(s)-Jacobi methods when \( s = 4, 8 \). The \( \omega \) of adaptive and cyclically adaptive IDR(s)-Jacobi
methods varies much, not around that of relaxed IDR(s)-Jacobi method.

**Figures 5, 6** represent variation of iterations of three kinds of IDR(s)-Jacobi methods for memplus and wang4 when $s = 1, 2, \ldots, 10$. As the parameter $s$ is larger, relaxed and cyclically adaptive IDR(s)-Jacobi methods converge at less iterations. On the other hand, adaptive IDR(s)-Jacobi method doesn't converge when $s$ is large.
An Adaptive Strategy of Improving Convergence of IDR(s)-Jacobi Method

Fig. 1 History of relative residual 2-norm of three kinds of IDR(s)-Jacobi methods for epb1 when s = 8.

Fig. 2 History of relative residual 2-norm of three kinds of IDR(s)-Jacobi methods for poisson3D when s = 4.

7. Concluding remarks

We demonstrated that convergence of the proposed cyclically adaptive IDR(s)-Jacobi methods are superior to that of relaxed and adaptive IDR(s)-Jacobi methods in view of convergence rate and robustness.

References


Fig. 3 Variation of $\omega$ of three kinds of IDR(s)-Jacobi methods for poisson3D when s = 4, from 1st to 60th iteration.

Fig. 4 Variation of $\omega$ of three kinds of IDR(s)-Jacobi methods for poisson3D when s = 8, from 1st to 100th iteration.

Fig. 5 Variation of iterations of three kinds of IDR(s)-Jacobi methods for memplus when $s = 1, 2, ..., 10$.

Fig. 6 Variation of iterations of three kinds of IDR(s)-Jacobi methods for wang4 when $s = 1, 2, ..., 10$. 