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

https://doi.org/10.15017/17886

出版情報：九州大学大学院システム情報科学紀要. 15 (1), pp.1-6, 2010-03-26. 九州大学大学院システム情報科学研究院
バージョン：

九州大学大学院
システム情報科学紀要
第15巻　第1号　平成22年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.15, No.1, March, 2010

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 ω_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[8] is known to be a simple one of stationary iterative methods for solving a linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$. Here, $A \in R^{N\times N}$ is a nonsymmetric coefficient matrix, and $\boldsymbol{x}$, $\boldsymbol{b}$ 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 Relaxation) 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[7] 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[6]. He referred to AGS (Accelerated Gauss-Seidel) method. However, it was too ingenuous. 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[3) 5)]. Moreover we propose also relaxed and adaptive version of IDR-based Jacobi method as for parameter ω_n. We abbreviate relaxed, adaptive and cyclically adaptive IDR(s)-Jacobi methods, respectively.

2. IDR Theorem

We focus on iterative solution of linear systems $A\boldsymbol{x} = \boldsymbol{b}$. Starting from initial guess $\boldsymbol{x}_0$ of the solution, initial residual $\boldsymbol{r}_0$ is defined as $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. We define the residual vector at nth iteration as $\boldsymbol{r}_n$.

Let $\mathcal{G}_0 = \mathrm{span}\{\boldsymbol{r}_0, B\boldsymbol{r}_0, ..., B^{N-1}\boldsymbol{r}_0\}$, where B is any matrix in $R^{N\times N}$. So $\mathcal{G}_0$ is the Krylov space corresponding to B and $\boldsymbol{r}_0$. Let $\mathcal{G}_{j+1} = B(\mathcal{G}_j \cap \mathrm{Null}(P^T))$ for $j = 0, 1, 2, ...$, $\boldsymbol{r}_{n+1} \in \mathcal{G}_{j+1}$, and $\boldsymbol{v} \in \mathcal{G}_j \cap \mathrm{Null}(P^T)$, where $P = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ ...\ \boldsymbol{p}_s)$. Then, the residual converges by IDR theorem if the spectral radius $\rho(B) < 1$.

In IDR(s) method[6], the matrix B is defined as $(I - \omega_n A)$. The parameter ω_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[6], 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:

$$M = D, \tag{1}$$
$$N = A - D. \tag{2}$$

We devise acceleration of IDR(s)-Jacobi method in which the diagonal matrix D is modified by param-

* Department of Computer Science and Communication Engineering, Graduate student
** Research Institute for Information Technology

eter ω as (3), (4). We refer to this modification as relaxed IDR(s)-Jacobi method.

$$M = \frac{1}{\omega}D, \tag{3}$$

$$N = A - \frac{1}{\omega}D. \tag{4}$$

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

$$\boldsymbol{r}_{n+1} = (A - \frac{1}{\omega}D)(\frac{1}{\omega}D)^{-1}\boldsymbol{v},$$
$$\boldsymbol{v} = \boldsymbol{r}_n + \sum_{i=1}^{s}\gamma_i \Delta \boldsymbol{r}_{n-i}. \tag{5}$$

We show an algorithm of relaxed IDR(s)-Jacobi method with parameter s as below. $\boldsymbol{e}_k$ is s-dimension vector that kth value is 1 and the others are 0. A given ω is used at the equations (7)-(10).

Algorithm of relaxed IDR(s)-Jacobi method

Let $\boldsymbol{x}_0$ be an initial solution, put $\boldsymbol{r}_0 = \boldsymbol{b} - \mathrm{A}\boldsymbol{x}_0$,
Let $P = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ ...\ \boldsymbol{p}_s) \in R^{N\times s}$, set $\gamma_0 = 0$,
ω is given, (6)
{initial loop : build matrices
$E = (d\boldsymbol{r}_1\ d\boldsymbol{r}_2\ ...\ d\boldsymbol{r}_s), Q = (d\boldsymbol{x}_1\ d\boldsymbol{x}_2\ ...\ d\boldsymbol{x}_s)$
by relaxed IDR(1) Jacobi method.}
for $n = 0, 1, ..., s-1$
  $\boldsymbol{s}_n = (\frac{1}{\omega}D)^{-1}(\boldsymbol{r}_n - \gamma_n d\boldsymbol{r}_n),$ (7)
  $d\boldsymbol{x}_{n+1} = \boldsymbol{s}_n - \gamma_n d\boldsymbol{x}_n,$
  $d\boldsymbol{r}_{n+1} = -(\frac{1}{\omega}D - A)\boldsymbol{s}_n - \boldsymbol{r}_n,$ (8)
  $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_{n+1},$
  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_{n+1},$
  if $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 \le \epsilon$ stop
  $\gamma_{n+1} = (\boldsymbol{p}_1, \boldsymbol{r}_{n+1})/(\boldsymbol{p}_1, d\boldsymbol{r}_{n+1}),$
  $E\boldsymbol{e}_{n+1} = d\boldsymbol{r}_{n+1}, Q\boldsymbol{e}_{n+1} = d\boldsymbol{x}_{n+1},$
end for,
$M = P^T E, \boldsymbol{f} = P^T \boldsymbol{r}_s,$
$n = s, k = 1,$
{main loop}
while $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 > \epsilon$
  solve $\boldsymbol{c}$ from $M\boldsymbol{c} = \boldsymbol{f}$,
  $\boldsymbol{s}_n = (\frac{1}{\omega}D)^{-1}(\boldsymbol{r}_n - E\boldsymbol{c}),$ (9)
  $d\boldsymbol{x}_{n+1} = \boldsymbol{s}_n - Q\boldsymbol{c},$
  $d\boldsymbol{r}_{n+1} = -(\frac{1}{\omega}D - A)\boldsymbol{s}_n - \boldsymbol{r}_n,$ (10)
  $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_{n+1},$
  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_{n+1},$
  $E\boldsymbol{e}_k = d\boldsymbol{r}_{n+1}, Q\boldsymbol{e}_k = d\boldsymbol{x}_{n+1},$
  $M\boldsymbol{e}_k = P^T d\boldsymbol{r}_{n+1}, \boldsymbol{f} = \boldsymbol{f} + M\boldsymbol{e}_k,$
  $n = n + 1, k = k + 1,$
  if $k > s$ then $k = 1$,
end while.

4. IDR(s)-Jacobi with adaptive tuning

In the iteration procedure of the relaxed IDR(s)-Jacobi method, the residual $\boldsymbol{r}_{n+1}$ is updated adjustively as the following equation.

$$\boldsymbol{r}_{n+1} = (\frac{1}{\omega_n}D - A)(\frac{1}{\omega_n}D)^{-1}(\boldsymbol{r}_n - E\boldsymbol{c})$$
$$= (I - \omega_n D^{-1}A)(\boldsymbol{r}_n - E\boldsymbol{c}). \tag{11}$$

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

$$\omega_n = \frac{(D^{-1}A\boldsymbol{t}_n, \boldsymbol{t}_n,)}{(D^{-1}A\boldsymbol{t}_n, D^{-1}A\boldsymbol{t}_n)}, \quad \boldsymbol{t}_n = \boldsymbol{r}_n - E\boldsymbol{c}. \tag{12}$$

As well known, the optimum parameter ω_n of the conventional SOR method is estimated at the range of [0.0, 2.0]. However, ω_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 ω_n is computed at the equations (13), (16).

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

Let $\boldsymbol{x}_0$ be an initial solution, put $\boldsymbol{r}_0 = \boldsymbol{b} - \mathrm{A}\boldsymbol{x}_0$,
Let $P = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ ...\ \boldsymbol{p}_s) \in R^{N\times s}$, set $\gamma_0 = 0$,
{initial loop : build matrices
$E = (d\boldsymbol{r}_1\ d\boldsymbol{r}_2\ ...\ d\boldsymbol{r}_s), Q = (d\boldsymbol{x}_1\ d\boldsymbol{x}_2\ ...\ d\boldsymbol{x}_s)$
by relaxed IDR(1) Jacobi method.}
for $n = 0, 1, ..., s-1$
  compute ω_n that minimizes $||\boldsymbol{r}_{n+1}||_2$, (13)
  $\boldsymbol{s}_n = (\frac{1}{\omega_n}D)^{-1}(\boldsymbol{r}_n - \gamma_n d\boldsymbol{r}_n),$ (14)
  $d\boldsymbol{x}_{n+1} = \boldsymbol{s}_n - \gamma_n d\boldsymbol{x}_n,$
  $d\boldsymbol{r}_{n+1} = -(\frac{1}{\omega_n}D - A)\boldsymbol{s}_n - \boldsymbol{r}_n,$ (15)
  $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_{n+1},$
  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_{n+1},$

if $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 \le \epsilon$ stop
$\gamma_{n+1} = (\boldsymbol{p}_1, \boldsymbol{r}_{n+1})/(\boldsymbol{p}_1, d\boldsymbol{r}_{n+1})$,
$E\boldsymbol{e}_{n+1} = d\boldsymbol{r}_{n+1}, Q\boldsymbol{e}_{n+1} = d\boldsymbol{x}_{n+1}$,
end for,
$M = P^T E, \boldsymbol{f} = P^T \boldsymbol{r}_s$,
$n = s, k = 1$,
{main loop}
while $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 > \epsilon$
solve $\boldsymbol{c}$ from $M\boldsymbol{c} = \boldsymbol{f}$,
compute ω_n that minimizes $||\boldsymbol{r}_{n+1}||_2$, (16)
$\boldsymbol{s}_n = (\frac{1}{\omega_n} D)^{-1}(\boldsymbol{r}_n - E\boldsymbol{c})$, (17)
$d\boldsymbol{x}_{n+1} = \boldsymbol{s}_n - Q\boldsymbol{c}$,
$d\boldsymbol{r}_{n+1} = -(\frac{1}{\omega_n} D - A)\boldsymbol{s}_n - \boldsymbol{r}_n$, (18)
$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + d\boldsymbol{r}_{n+1},\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + d\boldsymbol{x}_{n+1}$,
$E\boldsymbol{e}_k = d\boldsymbol{r}_{n+1}, Q\boldsymbol{e}_k = d\boldsymbol{x}_{n+1}$,
$M\boldsymbol{e}_k = P^T d\boldsymbol{r}_{n+1}, \boldsymbol{f} = \boldsymbol{f} + M\boldsymbol{e}_k$,
$n = n+1, k = k+1$,
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 ω_n every step. However, that requires another matrix-vector product for computing ω_n. Moreover, the updated residuals may not be in $\mathcal{G}_{j+1}$ because of the matrix $B = I - \omega_n D^{-1} A$ varying .

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

if mod $(n, s+1) = 0$ then
compute ω_n that minimizes $||\boldsymbol{r}_{n+1}||_2$, (19)
else
$\omega_n = \omega_{n-1}$, (20)
end if.

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 $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2$. In all cases the iteration was started with the initial guess solution $\boldsymbol{x}_0 = \mathbf{o}$. All test matrices were normalized with diagonal scaling. The maximum iteration was fixed as 10000. We show the optimum parameter ω of SOR and relaxed IDR(s)-Jacobi method in which ω 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 1 The characteristics of test matrix.

matrix	N	nnz	ave. nnz	analytical field
epb1	14,734	95,053	6.45	thermal
epb2	25,228	175,027	6.94	
poisson3Da	13,514	352,762	26.10	structural
poisson3Db	85,623	2,374,949	27.74	
raefsky2	3,242	294,276	90.77	hydro-
raefsky3	21,200	1,488,768	70.22	dynamic
memplus	17,758	126,150	7.10	electrical
wang4	26,068	177,196	6.80	

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 $||\boldsymbol{b} - A\boldsymbol{x}_{n+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$ for the converged solutions $\boldsymbol{x}_{n+1}$. "max" and "div." also mean that the iterative method did not converge until maximum iteration, and that the residual $\boldsymbol{r}_{n+1}$ 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 ω 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 ω of three kinds of IDR(s)-Jacobi methods when $s = 4, 8$. The ω of adaptive and cyclically adaptive IDR(s)-Jacobi

Table 2 Numerical results of Jacobi, GS, SOR and three kinds of IDR(s)-Jacobi methods.

matrix	method	s	itr.	time	TRR
	Jacobi	-	max	-	-4.91
	GS	-	max	-	-7.05
	SOR, ω =1.5	-	4184	1.84	-8.00
	IDR(s)-	1	max	-	-6.31
	Jacobi	2	2077	0.88	-8.03
	ω =1.2	4	1094	0.54	-8.00
epb1		8	693	0.45	-8.13
	adap.	1	2738	1.53	-8.33
	IDR(s)-	2	1387	0.84	-8.05
	Jacobi	4	895	0.59	-8.39
		8	935	0.75	-4.89
	cyclically	1	2565	1.23	-8.02
	adap.	2	1423	0.69	-8.10
	IDR(s)-	4	884	0.46	-8.25
	Jacobi	8	599	**0.38**	-7.99
	Jacobi	-	2830	0.32	-8.00
	GS	-	1427	1.17	-8.00
	SOR, ω =1.3	-	779	0.82	-8.00
	IDR(s)-	1	574	0.43	-8.12
	Jacobi	2	358	**0.29**	-8.00
	ω =1.1	4	311	0.30	-8.10
epb2		8	308	0.46	-8.11
	adap.	1	429	0.47	-8.13
	IDR(s)-	2	341	0.40	-8.05
	Jacobi	4	338	0.45	-8.16
		8	503	0.81	-8.13
	cyclically	1	445	0.41	-8.24
	adap.	2	327	0.30	-8.03
	IDR(s)-	4	309	0.31	-8.23
	Jacobi	8	297	0.45	-8.01
	Jacobi	-	max	-	div.
	GS	-	2043	3.40	-8.00
	SOR, ω =1.9	-	176	0.44	-8.03
	IDR(s)-	1	708	0.74	-8.32
	Jacobi	2	406	0.45	-8.16
	ω =0.7	4	225	0.26	-8.01
poisson		8	186	**0.23**	-8.20
3Da	adap.	1	453	0.77	-8.09
	IDR(s)-	2	283	0.52	-8.19
	Jacobi	4	243	0.44	-8.10
		8	max	-	4.04
	cyclically	1	409	0.56	-8.00
	adap.	2	269	0.35	-8.20
	IDR(s)-	4	197	0.27	-8.04
	Jacobi	8	177	0.25	-8.12
	Jacobi	-	max	-	div.
	GS	-	6367	110.45	-8.00
	SOR, ω =1.8	-	769	12.08	-8.01
	IDR(s)-	1	max	-	div.
	Jacobi	2	1763	16.63	-8.01
	ω =0.7	4	730	7.39	-8.00
poisson		8	428	4.86	-8.05
3Db	adap.	1	1824	30.24	-8.01
	IDR(s)-	2	1033	17.37	-8.09
	Jacobi	4	722	12.60	-7.26
		8	max	-	-1.33
	cyclically	1	1478	19.00	-8.01
	adap.	2	908	10.79	-8.37
	IDR(s)-	4	450	5.25	-8.02
	Jacobi	8	404	**4.47**	-8.09

Table 3 Numerical results of Jacobi, GS, SOR and three kinds of IDR(s)-Jacobi methods.(cont'd)

matrix	method	s	itr.	time	TRR
	Jacobi	-	max	-	div.
	GS	-	max	-	111.41
	SOR, ω =0.9	-	8225	8.14	-8.00
	IDR(s)-	1	1112	0.62	-8.52
	Jacobi	2	559	0.31	-8.13
	ω =0.6	4	426	0.24	-8.25
raefsky2		8	378	**0.22**	-8.23
	adap.	1	856	0.79	-8.37
	IDR(s)-	2	512	0.47	-8.38
	Jacobi	4	462	0.43	-8.03
		8	493	0.48	-8.87
	cyclically	1	851	0.63	-8.01
	adap.	2	507	0.34	-9.09
	IDR(s)-	4	420	0.27	-8.78
	Jacobi	8	385	0.25	-8.54
	Jacobi	-	max	-	div.
	GS	-	max	-	div.
	SOR, ω =0.7	-	8875	53.31	-8.00
	IDR(s)-	1	max	-	div.
	Jacobi	2	max	-	div.
	ω =all	4	max	-	div.
raefsky3		8	max	-	div.
	adap.	1	9971	62.66	-8.27
	IDR(s)-	2	8457	77.39	-8.11
	Jacobi	4	max	-	-4.26
		8	max	-	-3.54
	cyclically	1	8303	40.30	-8.01
	adap.	2	6990	44.20	-8.22
	IDR(s)-	4	5089	21.49	-8.04
	Jacobi	8	4852	**20.62**	-7.68
	Jacobi	-	max	-	-5.84
	GS	-	max	-	-6.76
	SOR, ω =1.9	-	1993	1.21	-8.00
	IDR(s)-	1	max	-	-6.94
	Jacobi	2	max	-	32.80
	ω =0.9	4	2830	1.64	-8.12
memplus		8	453	0.37	-8.06
	adap.	1	9761	7.15	-8.20
	IDR(s)-	2	4786	3.69	-8.06
	Jacobi	4	2115	1.78	-8.02
		8	max	-	-3.08
	cyclically	1	8692	5.21	-7.87
	adap.	2	4587	2.74	-8.01
	IDR(s)-	4	1645	1.04	-8.01
	Jacobi	8	393	**0.34**	-7.94
	Jacobi	-	7076	0.62	-8.00
	GS	-	3464	2.77	-8.00
	SOR, ω =1.9	-	290	0.26	-8.01
	IDR(s)-	1	939	0.56	-8.08
	Jacobi	2	486	0.31	-8.18
	ω =0.9	4	276	0.22	-8.09
wang4		8	215	0.30	-8.41
	adap.	1	584	0.54	-8.15
	IDR(s)-	2	330	0.33	-8.02
	Jacobi	4	262	0.29	-8.21
		8	1119	1.71	-4.97
	cyclically	1	578	0.43	-8.08
	adap.	2	315	0.24	-8.31
	IDR(s)-	4	229	**0.20**	-8.73
	Jacobi	8	216	0.26	-8.05

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 $s = 1, 2, ..., 10$. 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.

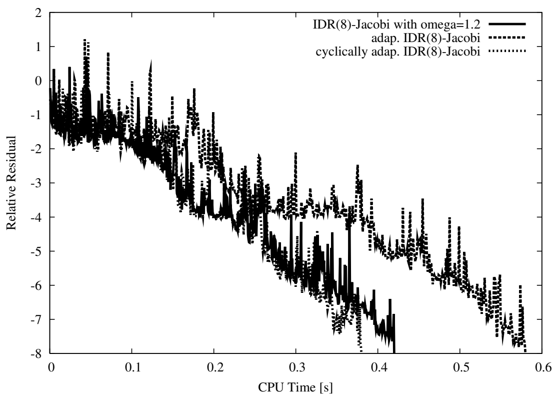

Fig. 1 History of relative residual 2-norm of three kinds of IDR(s)-Jacobi methods for epb1 when $s = 8$.

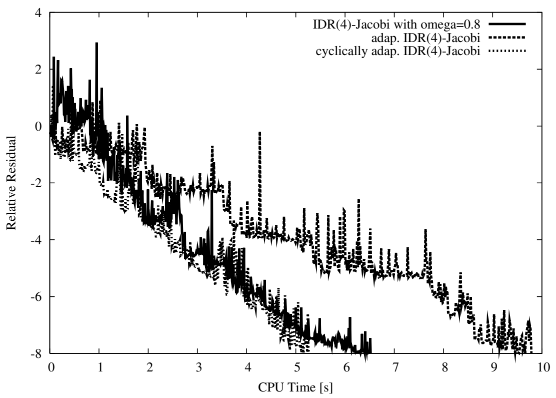

Fig. 2 History of relative residual 2-norm of three kinds of IDR(s)-Jacobi methods for poisson3Db when $s = 4$.

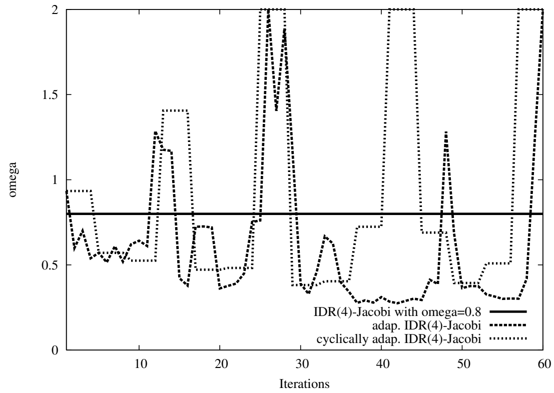

Fig. 3 Variation of ω of three kinds of IDR(s)-Jacobi methods for poisson3Db when $s = 4$, from 1st to 60th iteration.

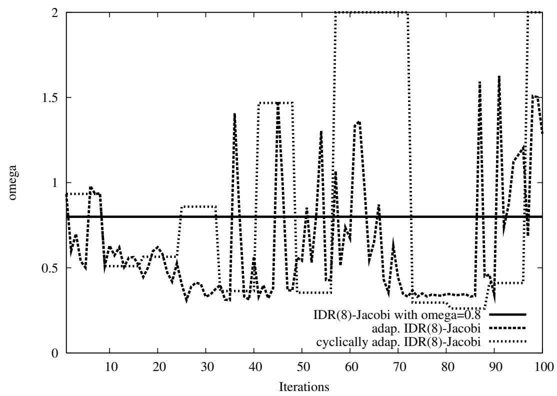

Fig. 4 Variation of ω of three kinds of IDR(s)-Jacobi methods for poisson3Db when $s = 8$, from 1st to 100th iteration.

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

1) University of Florida Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/index.html
2) Wesseling, P., Sonneveld, P., Numerical Experiments with a Multiple Grid- and a Preconditioned Lanczos Type Methods, Lecture Notes in Math., Springer, No.771, pp.543-562, 1980.
3) M. Harumatsu, Y. Kusakabe, S. Fujino, T. Fukushige, T. Arima, P., Sonneveld, A Proposal of Gauss-Seidel and Successive Over-Relaxation Methods based on IDR Theorem, Technical report of Information Processing Society of Japan, JAXA Chofu, June, 2009. (In Japanese)
4) Y. Kusakabe, S. Fujino, A consideration on convergence of some Jacobi methods based on the extended IDR Theorem, The Proc. of workshop on acceleration method, Tokyo, November, 2009.
5) Y. Kusakabe, S. Fujino, A proposal of Jacobi method based on extended Induced Dimension Reduction Theorem aimed at high convergence rate, The Proc. of EMAC 2009, Adelaide, 6-9, December, 2009.
6) P. Sonneveld, AGS - IDR - CGS - BiCGSTAB - IDR(s): The circle closed, A case of serendipity, The Proc. of Int. Kyoto Forum 2008 on Krylov subspace methods, pp.1-14, September, 2008.
7) P. Sonneveld, M. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Comput., Vol. 631, No.2, pp.1035-1062, 2008.
8) D. Young, Iterative methods for solving partial difference equations of elliptic type, Doctoral Thesis, Harvard University, 1950.

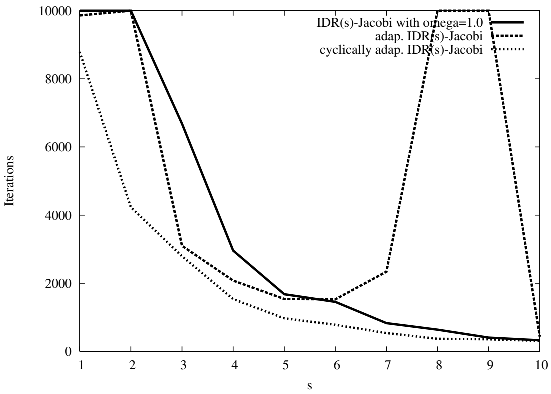

Fig. 5 Variation of iterations of three kinds of IDR(s)-Jacobi methods for memplus when $s = 1, 2, ..., 10$.

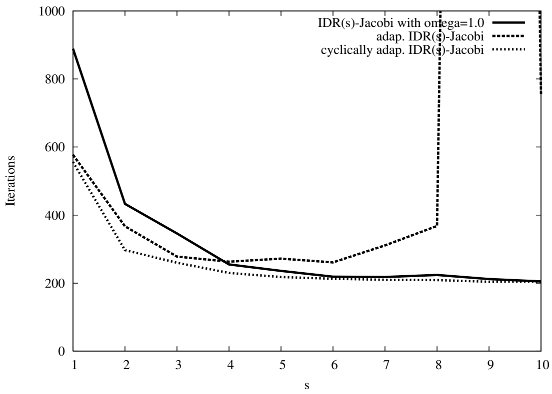

Fig. 6 Variation of iterations of three kinds of IDR(s)-Jacobi methods for wang4 when $s = 1, 2, ..., 10$.