World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

Improved IDR(s) method for gaining very accurate solutions

Yusuke ONOUE, Seiji FUJINO, and Norimasa NAKASHIMA

Abstract—The IDR(s) method based on an extended IDR theorem was proposed by Sonneveld and van Gijzen. The original IDR(s) method has excellent property compared with the conventional iterative methods in terms of efficiency and small amount of memory. IDR(s) method, however, has unexpected property that relative residual 2-norm stagnates at the level of less than 10^{-12}. In this paper, an effective strategy for stagnation detection, stagnation avoidance using adaptively information of parameter s and improvement of convergence rate itself of IDR(s) method are proposed in order to gain high accuracy of the approximated solution of IDR(s) method. Through numerical experiments, effectiveness of adaptive tuning IDR(s) method is verified and demonstrated.

Keywords—Krylov subspace methods, IDR(s), Adaptive tuning, Stagnation of relative residual

I. Introduction

We consider to solve a nonsymmetric linear system of equations,

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where A is a given nonsymmetric matrix in $R^{N\times N}$, and $\boldsymbol{x}$ is a solution vector in R^N, and $\boldsymbol{b}$ is a right-hand side vector in R^N. Krylov subspace methods are effective for solving linear systems of equations [4]. Krylov subspace is defined as follows:

$$K_n(A;\boldsymbol{r}_0) \;:=\; \mathrm{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{n-1}\boldsymbol{r}_0\}. \tag{2}$$

Here, $\boldsymbol{r}_0 := \boldsymbol{b} - A\boldsymbol{x}_0$ is an initial residual vector. The members of Krylov subspace methods, product-type Bi-Conjugate Gradient (BiCG) methods are often used for solving nonsymmetric linear systems of equations. BiCG stabilized (BiCGStab) method [4], and Generalized Product BiCG (GPBiCG) method [6], BiCGSafe method [2] and so on, are some versions of product-type Bi-Conjugate Gradient (BiCG) methods.

In 2008, one of Krylov subspace method, IDR(s) method is proposed by P. Sonneveld and M. B. van Gijzen [3]. IDR(s) method is based on the IDR theorem. IDR(s) method is competitive with or superior to most product-type BiCG methods, and outperforms BiCGStab method when $s > 1$.

However, we meet with a phenomenon that relative residual 2-norm of original IDR(s) method stagnates approximately between 10^{-12} and 10^{-15}. Therefore, we should consider adaptive tuning IDR(s) method on parameter s (abbreviated as AT_IDR(s) method) for avoidance the stagnation. We will demonstrate effectiveness of AT_IDR(s) method for avoidance stagnation of residual by means of some numerical experiments.

This paper is organized as follows. In section 2, we introduce outline of IDR(s) method. In particular, we describe IDR theorem and algorithm of IDR(s) method. In section 3, we present algorithm of AT_IDR(s) method in detail. We describe two strategies in order to build AT_IDR(s) method and algorithm of AT_IDR(s) method, and consider issues on implementation. In section 4, robustness of AT_IDR(s) method is demonstrated by numerical experiments. That is, we make clear that AT_IDR(s) method can solve systems with high accuracy. Finally, in section 5, we draw concluding remarks.

July 15, 2009

II. IDR(s) method

A. IDR theorem

IDR(s) method is based on the IDR theorem [5][3]. Let A be any matrix in $R^{N\times N}$, and $\boldsymbol{v}_0$ be any vector in R^N, and $\mathcal{G}_0$ be the complete Krylov space $K_N(A, \boldsymbol{v}_0)$. Let S denote any space in R^N, and define the sequence spaces $\mathcal{G}_j (j = 1, 2, \ldots)$ as

$$\mathcal{G}_j := (I - \omega_j A)(\mathcal{G}_{j-1} \cap S). \tag{3}$$

Here ω_j's are non-zero scalars. Then, the next two theorems holds.

(i) $\mathcal{G}_j \subseteq \mathcal{G}_{j-1}$ for all $j > 0$,
(ii) $\mathcal{G}_j = \{\boldsymbol{0}\}$ for some $j \leq N$.

B. Algorithm of IDR(s) method

The IDR theorem can be applied by generating residual vectors $\boldsymbol{r}_n$ that are forced to be in space $\mathcal{G}_j (j \leq N)$. Then, under assumptions of the IDR theorem, a linear system of equations will be solved after at most N dimension reduction steps. Algorithm of IDR(s) method is written as follows:

1. Let $\boldsymbol{x}_0$ be an initial guess, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
2. For $n = 0, \ldots, s-1$ Do
3. $\quad \boldsymbol{v}_n = A\boldsymbol{r}_n$
4. $\quad \omega_n = \dfrac{(\boldsymbol{v}_n, \boldsymbol{r}_n)}{(\boldsymbol{v}_n, \boldsymbol{v}_n)}$
5. $\quad \boldsymbol{q}_n = \omega_n \boldsymbol{r}_n, \quad \boldsymbol{e}_n = -\omega_n \boldsymbol{v}_n,$
6. $\quad \boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \boldsymbol{e}_n, \; \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{q}_n$
7. End Do
8. $E_s = (\boldsymbol{e}_{s-1} \cdots \boldsymbol{e}_0), \;\; Q_s = (\boldsymbol{q}_{s-1} \cdots \boldsymbol{q}_0)$
9. Do $n = s, s+1, \ldots$
10. $\quad$ Solve $\boldsymbol{c}_n$ from $P^T E_n \boldsymbol{c}_n = P^T \boldsymbol{r}_n$
11. $\quad \boldsymbol{v}_n = \boldsymbol{r}_n - E_n \boldsymbol{c}_n$
12. $\quad$ If $\mathrm{mod}(n, s+1) = s$ then
13. $\quad\quad \boldsymbol{t}_n = A\boldsymbol{v}_n$
14. $\quad\quad \omega_n = \dfrac{(\boldsymbol{t}_n, \boldsymbol{v}_n)}{(\boldsymbol{t}_n, \boldsymbol{t}_n)}$

15. $\boldsymbol{e}_n = -E_n\boldsymbol{c}_n - \omega_n\boldsymbol{t}_n$
16. $\boldsymbol{q}_n = -Q_n\boldsymbol{c}_n + \omega_n\boldsymbol{v}_n$
17. Else
18. $\boldsymbol{q}_n = -Q_n\boldsymbol{c}_n + \omega_n\boldsymbol{v}_n,\ \boldsymbol{e}_n = -A\boldsymbol{q}_n$
19. End If
20. $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \boldsymbol{e}_n,\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{q}_n$
21. if $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 \leq \epsilon$ then stop
22. $E_n = (\boldsymbol{e}_{n-1}\cdots\boldsymbol{e}_{n-s}),\ Q_n = (\boldsymbol{q}_{n-1}\cdots\boldsymbol{q}_{n-s})$
23. End Do

C. How to make matrix P

We discuss how to make matrix P. The matrix P is defined as $P = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s)$. The every entries of $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s$ are random numbers between 0.0 and 1.0. Then, matrix P is orthonormalized by modified Gram-Schmidt mathod as

$$(\boldsymbol{p}_i, \boldsymbol{p}_j) = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j). \end{cases}$$

We note that it is necessary to make matrix only once before the iteration process of IDR(s) method.

III. AT_IDR(s) METHOD

In this section, we discuss how to build AT_IDR(s) method which improves convergence property by tuning parameter s of IDR(s) method.

A. To build AT_IDR(s) method

For the larger parameter s, computation time of IDR(s) method per one iteration becomes longer. Therefore we adopt two significant strategies as follows:

1) **The first strategy**: Detect stagnation of residual.
2) **The second strategy**: Reset parameter s as the original value.

We describe the first strategy. First, we compute variation rate of residual norm $\sigma_n := \frac{|||\boldsymbol{r}_n||_2 - ||\boldsymbol{r}_{n-1}||_2|}{||\boldsymbol{r}_{n-1}||_2}$. Second, we regard it as occurence of stagnation, when the index "$sentinel$" of $\sigma_n < \delta$ in consecutive times. We remark that one must give parameters δ and $sentinel$ before iteration process of AT_IDR(s) method as $sentinel = 5$, $\delta = 0.1$. The second strategy is clear in trivial, so we omit description of the second strategy.

B. Diagram of adaptive tuning AT_IDR(s) method on parameter s

In this section, we discuss adaptive tecnique of AT_IDR(s) method using a diagram. Fig. 1 shows the diagram of adaptive tuning AT_IDR(s) method on parameter s. At first, we compute the variation rate of residual 2-norm σ_n. Next, we check whether $\sigma_n < \delta$ or not. If $\sigma_n < \delta$, we increase $count$ by one, and regard it as occurence of stagnation of residual when $count = sentinel$. On the other hand, if $\sigma_n \geq \delta$, we reset $count$ as 0, and regard it as avoidance of stagnation of residual. When stagnation of residual occurs, we increase parameter s by one, and reset $count$ as 0. When stagnation of residual is avoided, we reset parameter s as the original value.

C. Algorithm of AT_IDR(s) method

We present algorithm of AT_IDR(s) method as follows: At lines number 9 and between 23 and 31, we detect the stagnation of the residual norm, and tune parameter s.

1. Let $\boldsymbol{x}_0$ be an initial guess, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
2. For $n = 0, \ldots, s-1$ Do
3. $\boldsymbol{v}_n = A\boldsymbol{r}_n$
4. $\omega_n = \dfrac{(\boldsymbol{v}_n, \boldsymbol{r}_n)}{(\boldsymbol{v}_n, \boldsymbol{v}_n)}$
5. $\boldsymbol{q}_n = \omega_n\boldsymbol{r}_n, \quad \boldsymbol{e}_n = -\omega_n\boldsymbol{v}_n,$
6. $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \boldsymbol{e}_n,\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{q}_n$
7. End Do
8. $E_s = (\boldsymbol{e}_{s-1}\cdots\boldsymbol{e}_0), \quad Q_s = (\boldsymbol{q}_{s-1}\cdots\boldsymbol{q}_0)$
9. $s_{min} = s$
10. Do $n = s, s+1, \ldots$
11. Solve $\boldsymbol{c}_n$ from $P^T E_n\boldsymbol{c}_n = P^T\boldsymbol{r}_n$
12. $\boldsymbol{v}_n = \boldsymbol{r}_n - E_n\boldsymbol{c}_n$
13. If $\mathrm{mod}(n, s+1) = s$ then
14. $\boldsymbol{t}_n = A\boldsymbol{v}_n$
15. $\omega_n = \dfrac{(\boldsymbol{t}_n, \boldsymbol{v}_n)}{(\boldsymbol{t}_n, \boldsymbol{t}_n)}$
16. $\boldsymbol{e}_n = -E_n\boldsymbol{c}_n - \omega_n\boldsymbol{t}_n$
17. $\boldsymbol{q}_n = -Q_n\boldsymbol{c}_n + \omega_n\boldsymbol{v}_n$
18. Else
19. $\boldsymbol{q}_n = -Q_n\boldsymbol{c}_n + \omega_n\boldsymbol{v}_n,\ \boldsymbol{e}_n = -A\boldsymbol{q}_n$
20. End If
21. $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \boldsymbol{e}_n,\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{q}_n$
22. If $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 \leq \epsilon$ then stop
23. $\sigma_n = \dfrac{|||\boldsymbol{r}_{n+1}||_2 - ||\boldsymbol{r}_n||_2|}{||\boldsymbol{r}_n||_2}$
24. If $\sigma_n < \delta$ then
25. $count = count + 1$
26. If $count = sentinel$ and $s < s_{max}$ then
27. $count = 0, s = s + 1$
28. End If
29. Else
30. $count = 0, s = s_{min}$
31. End If
32. $E_n = (\boldsymbol{e}_{n-1}\cdots\boldsymbol{e}_{n-s}),\ Q_n = (\boldsymbol{q}_{n-1}\cdots\boldsymbol{q}_{n-s})$
33. End Do

D. Issues on implementation

In the above algorithm of AT_IDR(s) method, there are issues on implementation. That is, they are to build matrices P, Q_n and E_n. Number of columns of matrices P, Q_n and E_n is s. Therefore we have to change number of columns of matrices P, Q_n and E_n, if parameter s increases. However it is difficult to implement matrices with dynamic allocation. We show how to build matrices P, Q_n and E_n. First, we set s_{max} as upper limit of parameter s. Second, matrices P, Q_n and E_n are built as an $N \times s_{max}$ matrix of constant size. Third, from $s+1$ to s_{max} columns of matrices P, Q_n and E_n are ignored when $s < s_{max}$ for simplicity.

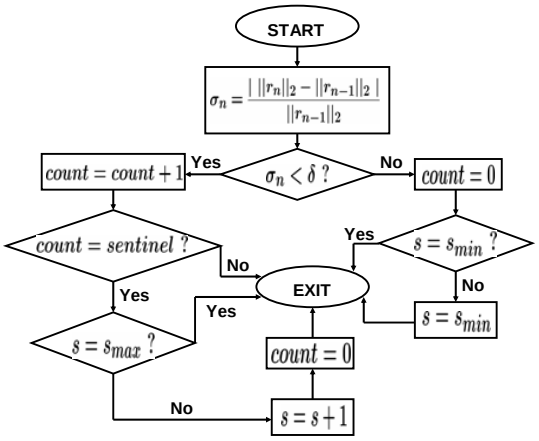


Fig. 1. Diagram of adaptive tuning AT_IDR(s) method on parameter s.

IV. Numerical Experiments

In this section we discuss numerical experiments of comparing performance of AT_IDR(s) method with IDR(s) method. All computations are carried out in double precision floating-point arithmetic on a PC with a POWER5 processor (1.9GHz). Intel Fortran Compiler90 ver 7.1 and compile option -O3 -qtune=power5 -qarch=pwr5 -qhot was used. In all cases the iteration was started with the initial guess solution $\boldsymbol{x}_0 = \mathbf{0}$. The maximum iterations was fi xed as 10000. We set parameters of AT_IDR(s) method as $sentinel = 5,\ \delta = 0.1$. Four test matrices are from University of Florida Sparse Matrix Collection[1]. Description of test matrices is shown in Table I. In this Table, "nnz" means number of nonzero entries, and "$ave.\ nnz$" means number of nonzero entries per single row.

TABLE I
Specifications of test matrices.

group/matrix	dimension	nnz	$ave.$ nnz
Watson/ChemMaster1	40,401	201,201	4.98
Watson/Baumann	112,211	748,331	6.67
FEMLAB/Sme3Da	12,504	874,887	69.97
FEMLAB/Sme3Db	29,067	2,081,063	71.59
FEMLAB/Sme3Dc	42,930	3,148,656	73.34
Quaglino/ViscoPlastic1	4,326	61,166	14.14
Quaglino/ViscoPlastic2	32,769	381,326	11.64

A. Numerical Results

Tables 2–8 for matrices Sme3Dc, ChemMaster1, ViscoPlastic2, Baumann, Sme3Da, Sme3Db and Viscoplastic1 show iterations and CPU time in seconds of IDR(s) and AT_ IDR(s) methods when the stopping criterion, i.e., $\frac{||\boldsymbol{r}_n||_2}{||\boldsymbol{r}_0||_2}$ is less than 10^{-12}, 10^{-13} and 10^{-14}, respectively. In Tables, "max" means that iterative methods did not converge until maximum iterations. "break" means also that all computations were halted because of huge numeical errors during iteration process. "itr." means also number of iterations. Some observations are gaind from Tables 2–8.

- AT_IDR(s) method performs well compared with the original IDR(s) method.
- IDR(s) method does not converge often when ϵ for convergence criterion is set as 10^{-13}, 10^{-14} and s is more than 4.
- On the other hand, AT_IDR(s) method converges for almost cases.

In order to make out how robust AT_IDR(s) is for analysis with high accuracy, we made stopping criterion more degree by degree. When the stopping criterion is $\frac{||\boldsymbol{r}_n||_2}{||\boldsymbol{r}_0||_2} \leq 10^{-15}$ which is almost as same as the so-called machine epsilon of 2.2×10^{-16}, IDR(s) method doesn't converged for all cases. On the other hand, AT_IDR(s) method converged for all cases. We can understand effectiveness of AT_IDR(s) method for analysis with high accuracy.

Figs.2–3 display relative residual history of IDR(s) and AT_IDR(s) methods for matrices ChemMaster1 and ViscoPlastic2. In these Figures, we show relative residual history of IDR(s) method in red solid line and AT_IDR(s) method in green dashed line, and variation of parameter s of AT_IDR(s) method in blue plot. From Figs.2–3, the following observations can be made as below.

- Results for matrix ChemMaster1 as shown in Fig.2:
  - Relative residual norm of IDR(s) method stagnates for all parameter s.
  - Relative residual norm of AT_IDR(1) method converges without stagnation.
  - Relative residual norm of AT_ IDR(4) and AT_ IDR(8) methods stagnate after 150 iterations, and converge after parameter s is tuned.
  - If parameter s is tuned at around 150 iterations, iterations of AT_IDR(4) and AT_IDR(8) may be lower.
- Results for matrix ViscoPlastic2 as shown in Fig.3:
  - Relative residual 2-norm of IDR(s) method diverges at s is equal to 1, and stagnates at s is equal to 4 or 8.
  - Relative residual 2-norm of AT_IDR(1) method stagnates after 3000 iterations, and converges at 3939 iterations.
  - Relative residual 2-norm of AT_ IDR(4) and AT_ IDR(8) methods converges without stagnation.
  - Parameter s of AT_IDR(s) is tuned too much, and CPU times of AT_IDR(s) per one iteration is longer than that of IDR(s) method.
  - When we set parameter δ for smaller value or $sentinel$ for larger value, parameter s is tuned more moderately, and AT_IDR(s) mathod converges faster.

From the above many observations, we can see that AT_IDR(s) method is more robust than IDR(s) mathod. However, we should fi nd out optimum parameters δ and $sentinel$ to improve convegence property of AT_IDR(s) method.

TABLE II
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX SME3DC.

		$\epsilon = 10^{-12}$		$\epsilon = 10^{-13}$		$\epsilon = 10^{-14}$	
method	s	itr.	time	itr.	time	itr.	time
IDR(s)	1	4048	292.5	max	-	max	-
	2	2096	153.5	2325	168.3	max	-
	4	1216	91.5	max	-	max	-
	8	1011	78.4	max	-	max	-
AT_IDR	1	6683	388.2	6690	386.9	7072	409.7
	2	3305	194.6	3318	194.8	3600	210.5
	4	1200	73.9	1239	76.3	1277	78.4
	8	986	62.4	995	63.2	1022	64.9

TABLE III
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX CHEMMASTER1.

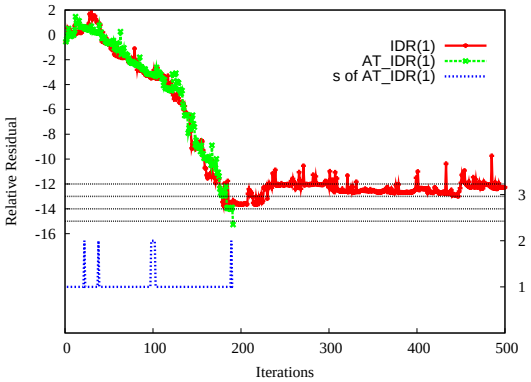
		$\epsilon = 10^{-12}$		$\epsilon = 10^{-13}$		$\epsilon = 10^{-14}$	
method	s	itr.	time	itr.	time	itr.	time
IDR(s)	1	176	0.81	179	0.82	max	-
	2	162	0.78	max	-	max	-
	4	295	1.46	max	-	max	-
	8	max	-	max	-	max	-
AT_IDR	1	176	0.85	184	0.90	188	0.89
	2	249	1.23	262	1.30	270	1.33
	4	151	0.85	330	1.82	341	1.86
	8	147	1.00	289	1.90	297	1.94

TABLE IV
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX VISCOPLASTIC2.

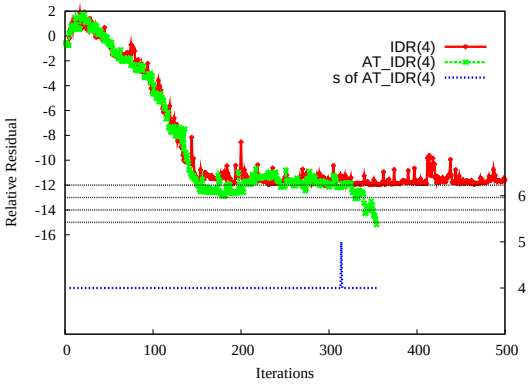
		$\epsilon = 10^{-12}$		$\epsilon = 10^{-13}$		$\epsilon = 10^{-14}$	
method	s	itr.	time	itr.	time	itr.	time
IDR(s)	1	max	-	max	-	max	-
	2	max	-	max	-	max	-
	4	1393	6.79	1520	7.46	max	-
	8	1288	7.24	1451	8.23	max	-
AT_IDR	1	1851	8.34	2650	11.90	3058	13.65
	2	1607	7.60	1834	8.69	1887	8.86
	4	1337	6.90	1490	7.72	1611	8.32
	8	1289	7.93	1460	9.09	1552	9.68

TABLE V
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX BAUMANN.

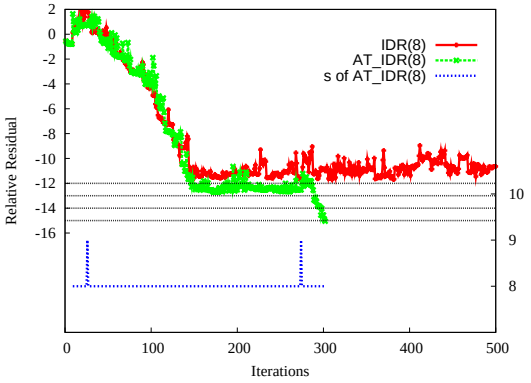
		$\epsilon = 10^{-12}$		$\epsilon = 10^{-13}$		$\epsilon = 10^{-14}$	
method	s	itr.	time	itr.	time	itr.	time
IDR(s)	1	break	-	break	-	break	-
	2	max	-	max	-	max	-
	4	669	11.80	max	-	max	-
	8	max	-	max	-	max	-
AT_IDR	1	2621	38.69	2695	39.64	2760	40.87
	2	1304	20.87	1418	22.72	1437	22.90
	4	569	10.61	581	10.90	613	11.48
	8	500	12.31	520	12.83	541	13.38



(a)$s = 1$

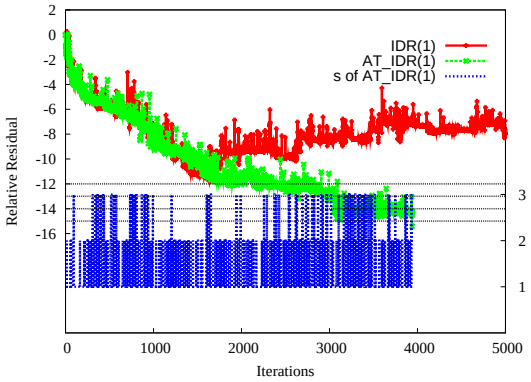


(b)$s = 4$

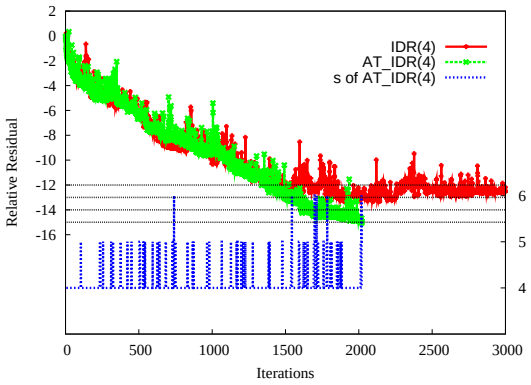


(c)$s = 8$

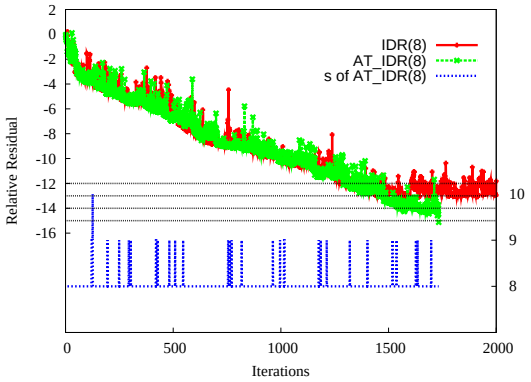
Fig. 2. Relative residual history of IDR(s) and AT_IDR(s) methods, and variation of parameter s of AT_IDR(s) method for matrix ChemMaster1.



(a)$s = 1$



(b)$s = 4$



(c)$s = 8$

Fig. 3. Relative residual history of IDR(s) and AT_IDR(s) methods, and variation of parameter s of AT_IDR(s) method for matrix ViscoPlastic2.

TABLE VI
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX SME3DA.

method	s	$\epsilon = 10^{-12}$ itr.	time	$\epsilon = 10^{-13}$ itr.	time	$\epsilon = 10^{-14}$ itr.	time
IDR(s)	1	1799	15.01	1976	16.19	2133	17.61
	2	891	7.77	max	-	max	-
	4	657	6.14	max	-	max	-
	8	max	-	max	-	max	-
AT_IDR	1	4127	31.37	4299	31.94	4299	31.94
	2	1020	8.37	1045	8.53	1087	8.74
	4	648	5.68	664	5.93	781	6.73
	8	523	5.03	529	5.02	542	5.18

TABLE VII
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX SME3DB.

method	s	$\epsilon = 10^{-12}$ itr.	time	$\epsilon = 10^{-13}$ itr.	time	$\epsilon = 10^{-14}$ itr.	time
IDR(s)	1	2250	85.21	2887	109.09	2934	110.01
	2	1293	50.11	max	-	max	-
	4	max	-	max	-	max	-
	8	max	-	max	-	max	-
AT_IDR	1	5429	191.56	5468	192.52	5499	193.44
	2	1304	47.79	1361	49.88	1393	51.76
	4	886	33.94	894	34.22	916	35.60
	8	684	27.86	703	28.63	721	29.05

V. CONCLUSION

We proposed AT_IDR(s) method for purpose of resolving stagnation of residual by tuning parameter s of IDR(s) method adaptively. We can conclude that AT_IDR(s) method converges when IDR(s) method doesn't converge because of stagnation of relative residual norm. AT_IDR(s) method is more robust than IDR(s) method.

As future work, we have two goals. The first goal is to find out optimum parameters δ and $sentinel$ to improve convegence rate of AT_IDR(s) method. The second goal is to devise more effective adaptive tuning technique for IDR(s) method.

REFERENCES

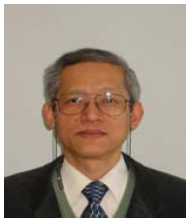
[1] T. Davis : Univ. of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices/index.html
[2] S. Fujino, M. Fujiwara, M. Yoshida : BiCGSafe method based on minimization of associate residual, Trans. of JSCES, Paper No.20050028, 2005. (In Japanese)
[3] P. Sonneveld, M. B. van Gijzen: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, SIAM J. Sci. Comput. Vol.31, No.2, pp.1035-1062, 2008.

TABLE VIII
CONVERGENCE OF IDR(s) AND AT_IDR METHODS FOR MATRIX VISCOPLASTIC1.

method	s	$\epsilon = 10^{-12}$ itr.	time	$\epsilon = 10^{-13}$ itr.	time	$\epsilon = 10^{-14}$ itr.	time
IDR(s)	1	1226	0.66	1388	0.76	max	-
	2	1098	0.65	1192	0.67	max	-
	4	985	0.64	1059	0.64	1152	0.71
	8	936	0.67	974	0.74	1268	0.89
AT_IDR	1	1367	0.78	1491	0.85	1697	1.01
	2	1120	0.66	1219	0.72	1366	0.81
	4	967	0.63	1062	0.69	1130	0.72
	8	932	0.69	987	0.75	1078	0.81

[4] H. A. van der Vorst: Iterative Krylov preciionditionings for large linear systems, Cambridge University Press, 2003.
[5] P. Wesseling, P. Sonneveld : Numerical Experiments with a Multiple Grid- and a Preconditioned Lanczos Type Methods, Lecture Notes in Math. Springer, No.771, pp.543-562, 1980.
[6] S.-L. Zhang: GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems, SIAM J. Sci. Comput., Vol.18, No.2, pp.537-551, 1997.

Yusuke Onoue received the B.E. and M.E. degrees from Kyushu University, Fukuoka, Japan in 2007 and 2009, respectively. He is currently working together the D.E. degree in computer science and communication engineering from Kyushu University, Fukuoka, Japan. His research interest is in iterative methods for linear systems and parallel computation. He is a member of Japan SIAM.



Seiji Fujino received the B.E. degree in Mathematics in 1974 from Kyoto University, Kyoto, Japan. Moreover He received the D.E. in 1993 from The University of Tokyo, Tokyo, Japan. He is currently a Professor of Research Institute for Information Technology, Kyushu University, Japan. His research interest is in iterative methods and its preconditioning for linear systems. He is a member of Japan SIAM, IPSJ, JSCES and JSST, respectively.

Norimasa Nakashima He received the B.E., M.E., and D.E. degrees in computer science and communication engineering from Kyushu University, Fukuoka, Japan in 1999, 2001, and 2004, respectively. From 2004 to 2008, he was a Research Associate at the Kyushu University. Currently, he is an Assistant Professor in the Department of Advanced Information Technology at Kyushu University. His research interest is in computational electromagnetics. He is a member of the IEICE of Japan, the IEEE, and Applied Computational Electromagnetics Society (ACES).