

## IDR(s)-Residual Reduction法に対する収束安定化法

中島, 真司  
九州大学大学院システム情報科学府情報学専攻

藤野, 清次  
九州大学情報基盤研究開発センター

<https://doi.org/10.15017/20054>

---

出版情報：九州大学大学院システム情報科学紀要. 16 (2), pp.57-62, 2011-09-26. 九州大学大学院システム情報科学研究所

バージョン：

権利関係：



# IDR( $s$ )-Residual Reduction 法に対する収束安定化法

中島 真司\* · 藤野 清次\*\*

## Advancement of Convergence Stability for IDR( $s$ ) Based Residual Reduction Method

Shinji NAKASHIMA\* and Seiji FUJINO\*\*

(Received July 22, 2011)

**Abstract:** IDR( $s$ )-Residual Reduction (abbreviated as R2) method as an extension of IDR( $s$ ) method have been proposed by the authors. Due to IDR( $s$ )-R2 method, a rich variety of preconditioning can be chosen in place of iteration matrix. In this article, we consider reduction of computational cost of IDR( $s$ )-R2 method from the viewpoint of building of dense matrix  $P$ . Slim Double Dense and a variant of Slim Double Dense for matrix  $P$  are proposed. Through numerical experiments, we clarify that preconditioned IDR( $s$ )-R2 method with the proposed structure of matrix  $P$  has preferable convergence stability.

**Keywords:** IDR( $s$ )-Residual Reduction method, Slim Double Dense, variant of Slim Double Dense

### 1. はじめに

IDR(Induced Dimension Reduction) 定理<sup>9)</sup>に基づく新しい反復法として, 2008 年 Sonneveld らにより IDR( $s$ ) 法<sup>6)</sup>が提案された。さらに, IDR 定理を定常反復法に適用した反復法も新しく提案された<sup>7)</sup>。さらに, 後者の反復法の収束性を一層改善させた IDR( $s$ )-Residual Reduction (以下では, IDR( $s$ )-R2 と略す) 法<sup>5)</sup>も提案された。IDR( $s$ )-R2 法では, 前処理について, 定常反復法の反復行列を云々枠組みから脱して, Krylov 部分空間法で広く用いられている様々な前処理が使用可能になった。

最近, IDR 定理に基づく反復法の算法中に現れる密行列  $P$  を Slim Dense 化することにより, 計算量削減を図る手法が提案された<sup>3)</sup>。行列  $P$  の Slim Dense 化は, IDR 定理に基づく反復法の算法中に現れる密行列  $P$  の転置とベクトルの積計算が全体の計算コストの中で大きな割合を占めることに着目した計算量削減法の一つである。しかし, Slim Dense 化は行列  $P$  の要素の大部分を零要素とするため, 収束性が不安定になる行列例も報告された<sup>10)</sup>。

本論文では, Slim Dense 行列  $P$  の構造に着目し, 非零要素数を増やした Slim Double Dense<sup>10)</sup>と Slim Double Dense を Gram-Schmidt 法<sup>4)</sup>にて直交させた Slim Double Dense variant を提案する。次に, それらを IDR( $s$ )-R2 法に適用し, その収束性について比較検討する。数値実験を通して, 元の密行列, Slim Double Dense, Slim Double Dense variant の 3 種類の行列  $P$  で収束性を比較検討する。

本論文の構成は次の通りである。第 2 節では, Slim Double Dense, Slim Double Dense variant 行列  $P$  の構造と直交化について記述する。第 3 節では, 数値実験によって, Slim Double Dense variant 行列  $P$  を使ったときの IDR( $s$ )-R2 法の収束性のよさを実証する。最後に, 第 4 節でまとめを行う。

### 2. Slim Double Dense, Slim Double Dense variant 行列 $P$ の構造と直交化

本節では, Slim Dense(以後, SD と略す) に対し非零要素数を増やした Slim Double Dense<sup>10)</sup>(以後, SDD と略す), また, Slim Double Dense を Gram-Schmidt 法にて直交させランダム性を増した Slim Double Dense variant(以後, SDD-v と略す) の構造と行列  $P$  の直交化について記述する。

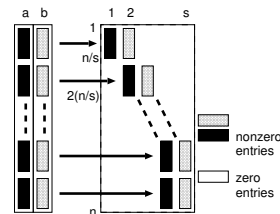


Fig. 1 Structure of Slim Double Dense matrix  $P$  of dimension  $n$ .

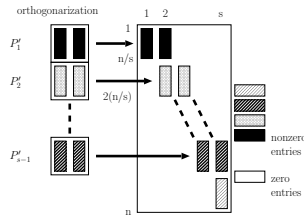
#### 2.1 Slim Double Dense

Fig. 1 に SDD 行列  $P$  の構造を示す<sup>10)</sup>。SDD は,  $[N/s]$

平成 23 年 7 月 22 日受付

\*情報学専攻博士後期課程

\*\*情報基盤研究開発センター



**Fig. 2** Structure of Slim Double Dense variant matrix  $P$  of dimension  $n$ .

の列ベクトルを2本ずつ階段状に配置しスリム化したものである。互いに直交する  $N$  次元のベクトル  $\mathbf{a} = (a_i)$ ,  $\mathbf{b} = (b_i)$ , ( $i = 1, \dots, N$ ) を  $s$  等分した階段状の配置で, 行列  $P$  の各列の直交性を実現する。ただし, ベクトル  $\mathbf{b}$  は以下の式にてベクトル  $\mathbf{a}$  に直交させる。

$$b_i = (-1)^i a_{i-(-1)^i}. \quad (1)$$

要素数  $\lfloor N/s \rfloor$  の列ベクトルは合計  $2s$  個となる。以下に, SDD 行列  $P$  の構成手順を示す。ただし, 行列  $P$  の非零要素を1次元配列  $\tilde{\mathbf{p}} = (\tilde{p}_1, \dots, \tilde{p}_{N_p})$  への格納でメモリの節約を図る。すなわち,  $\mathbf{a} = (\tilde{p}_1, \dots, \tilde{p}_N)$ ,  $\mathbf{b} = (\tilde{p}_{N+1}, \dots, \tilde{p}_{2N})$  で済む。

#### Composition of SDD matrix $P$

1.  $N_p = 2N$
2. Do  $i = 1, \dots, N$
3.  $\tilde{p}_i = \text{random number}$
4. End Do
5.  $t_1^{\text{ind}} = 1, t_2^{\text{ind}} = 1$
6.  $t_1^{\text{ptr}} = 1, t_2^{\text{ptr}} = t_1^{\text{ptr}} + \lfloor N/s \rfloor$
7. Do  $i = 3, \dots, s-1$
8.  $t_i^{\text{ind}} = t_{i-1}^{\text{ind}} + \lfloor N/s \rfloor$
9.  $t_i^{\text{ptr}} = t_{i-1}^{\text{ptr}} + 2 \times \lfloor N/s \rfloor$
10. End Do
11.  $t_s^{\text{ind}} = t_{s-1}^{\text{ind}} + \lfloor N/s \rfloor$
12.  $t_s^{\text{ptr}} = t_{s-1}^{\text{ptr}} + \lfloor N/s \rfloor + (N - (s-1)\lfloor N/s \rfloor)$
13.  $t_{s+1}^{\text{ptr}} = N_p + 1$
14. Do  $i = 1, \dots, N$
15.  $\tilde{p}_{n+i} = (-1)^i a_{i-(-1)^i}$
16. EndDo

次に, SDD 行列  $P$  のときの  $P^T$  とベクトルの積計算の手順を以下に示す。

#### Matrix-vector multiplication in SDD matrix $P$

1. Do  $i = 1, \dots, s$
2.  $y_i = 0, k = t_i^{\text{ind}}$
3. Do  $j = t_i^{\text{ptr}}, \dots, t_{i+1}^{\text{ptr}} - 1$
4.  $y_i = y_i + \tilde{p}_j * u_k, k = k + 1$
5. End Do
6. End Do

SDD は SD に対して, 非零要素の増加により安定性が増すと期待できる。一方, 非零要素を増やしたため, それに伴い計算時間が長くなる心配がある。また, SD は最初から直交系になっている利点があるのに対して, SDD は直交化の必要がある。

#### 2.2 Slim Double Dense variant

**Fig. 2** に SDD-v 行列  $P$  の構造を示す。SDD-v は, SDD<sup>10)</sup> を Gram-Schmidt の直交化法にて直交させたものである。また,  $P_{ij}((s-1)\lfloor N/s \rfloor < i \leq N, j = s-1) = 0$  とした。SDD-v は要素数  $\lfloor N/s \rfloor$  の列ベクトルを  $s$  個対角線上に,  $s-1$  個右副対角線上に並べた行列にスリム化することである。要素数  $\lfloor N/s \rfloor$  の列ベクトルは合計で  $(2s-1)$  個となる。SDD-v は SD のように対角線上の要素が非零であることに加え, 副対角線上の要素が非零となるため, (2) 式にて行列  $P = (p_{i,j})(i = 1, \dots, N, j = 1, \dots, s)$  を初期化する。ただし, 乱数は区間  $[0, 1]$  の一様乱数である。

$$P_{ij} = \begin{cases} r & (\lfloor N/s \rfloor \times (j-2) < i \leq \lfloor N/s \rfloor \times j) \\ 0 & (\text{otherwise}). \end{cases} \quad (2)$$

上記の初期化を行った時点では, 行列  $P$  の列ベクトルは直交系ではないため, **Fig. 2** 示すように, 行で重なる列ベクトル同士を直交化を施す必要がある。本論文の数値実験においては, Gram-Schmidt の直交化法を用いる。直交化後, 各列ベクトルで正規化を施すことにより,  $P$  の各列ベクトルは正規直交系となる。以下に, SDD-v 行列  $P$  の構成手順を示す。SDD-v 行列  $P$  の構成は, SDD 行列  $P$  の構成手順の1,2行目と11-16行目を以下のように書き換えることで得られる。ただし, 行列  $P$  の非零要素を1次元配列  $\tilde{\mathbf{p}} = (\tilde{p}_1, \dots, \tilde{p}_{N_p})$  に格納することでメモリの節約を図る。

#### Composition of SDD-v matrix $P$

1.  $N_p = N + \lfloor N/s \rfloor \times (s-1)$
2. Do  $i = 1, \dots, N_p$
- ⋮
11.  $t_{s+1}^{\text{ptr}} = N_p + 1$
12. Orthogonalize  $\tilde{\mathbf{p}}(t_1^{\text{ptr}} : t_2^{\text{ptr}} - 1),$   
 $\tilde{\mathbf{p}}(t_2^{\text{ptr}} : t_2^{\text{ptr}} + \lfloor N/s \rfloor - 1)$

13. Do  $i = 2, \dots, s - 1$
14. Orthogonalize  $\tilde{\mathbf{p}}(t_i^{\text{ptr}} + \lfloor N/s \rfloor : t_{i+1}^{\text{ptr}} - 1)$ ,
15.  $\tilde{\mathbf{p}}(t_{i+1}^{\text{ptr}} : t_{i+1}^{\text{ptr}} + \lfloor N/s \rfloor - 1)$
16. End Do

次に, SDD-v 行列  $P$  のときの  $P^T$  とベクトルの積計算の手順を以下に示す. **Matrix-vector multiplication in SDD-v matrix  $P$**

1. Do  $i = 1, \dots, s$
2.  $y_i = 0, k = t_i^{\text{ind}}$
3. Do  $j = t_i^{\text{ptr}}, \dots, t_{i+1}^{\text{ptr}} - 1$
4.  $y_i = y_i + \tilde{p}_j * u_k, k = k + 1$
5. End Do
6. End Do

3 種類の行列  $P$  には以下に示す特徴がある.

- SDD-v は SD に対して, 非零要素の増加により安定性が強化されると期待できる. 一方, 非零要素の増加により計算時間が長くなる心配がある.
- SD は最初から直交系であるのに対して, SDD-v は直交化の必要がある.
- SDD-v は SD に対して実装がやや複雑になる.
- SDD-v は SDD に対して, Gram-Schmidt の直交化法により安定性が増すと期待できる.

### 2.3 各行列 $P$ の特徴

Table 1 に各行列  $P$  の特徴をまとめたものを示す. 計算量, 収束の安定性, そして実装の容易さの 3 点から各行列  $P$  の特徴を抽出しまとめた.

**Table 1** Summary of characteristics of dense and slim dense matrices  $P$ .

matrix $P$	computational cost	convergence stability	implementation
D	large	sound	simple
SDD	small	moderate	complicated
SDD-v	small	moderate	complicated

## 3. 数 値 実 験

### 3.1 ILU(0) 前処理つき IDR( $s$ )-R2 法

ILU(0) 前処理つき IDR( $s$ )-R2 法において, 行列  $P$  を乱数からなる密行列とした場合と, SDD とした場合と, SDD-v とした場合の 3 通りについて実験を行った. 前処理つき

IDR( $s$ )-R2 法の算法を以下に示す.

#### 前処理つき IDR( $s$ )-R2 法の算法

1. Let  $\mathbf{x}_0$  be an initial guess
2. put  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
3.  $P = (\mathbf{p}_1 \mathbf{p}_2 \dots \mathbf{p}_s) \in \mathbb{R}^{N \times s}$ , set  $\gamma_0 = 0$
4. {initial loop : build  $E = (\mathbf{d}\mathbf{r}_1 \dots \mathbf{d}\mathbf{r}_s)$
5.  $Q = (\mathbf{d}\mathbf{x}_1 \mathbf{d}\mathbf{x}_2 \dots \mathbf{d}\mathbf{x}_s)$  by IDR-R2}
6. for  $k = 0, 1, \dots, s - 1$
7.  $\mathbf{v}_k = \mathbf{r}_k - \gamma_k \mathbf{d}\mathbf{r}_k$
8. Compute  $C^{-T} C^{-1} \mathbf{v}_k$
9.  $\mathbf{d}\mathbf{x}_{k+1} = C^{-T} C^{-1} \mathbf{v}_k - \gamma_k \mathbf{d}\mathbf{x}_k$
10.  $\mathbf{d}\mathbf{r}_{k+1} = A\mathbf{d}\mathbf{x}_k$
11.  $\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{d}\mathbf{r}_{k+1}, \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}\mathbf{x}_{k+1}$
12. if  $\|\mathbf{r}_{k+1}\|_2 / \|\mathbf{r}_0\|_2 \leq \epsilon$  then exit
13.  $\gamma_{k+1} = \frac{(\mathbf{p}, \mathbf{r}_{k+1})}{(\mathbf{p}, \mathbf{d}\mathbf{r}_{k+1})}$
14.  $E\mathbf{e}_{k+1} = \mathbf{d}\mathbf{r}_{k+1}, Q\mathbf{e}_{k+1} = \mathbf{d}\mathbf{x}_{k+1}$
15. end for
16.  $G = P^T E, \mathbf{f} = P^T \mathbf{r}_s, n = 1$
17. {main loop}
18. for  $k = s, s + 1, \dots,$
19. Solve  $\mathbf{c}$  from  $G\mathbf{c} = \mathbf{f}$
20.  $\mathbf{v}_k = \mathbf{r}_k - E\mathbf{c}$
21. Compute  $C^{-T} C^{-1} \mathbf{v}_k$
22.  $\mathbf{d}\mathbf{x}_{k+1} = C^{-T} C^{-1} \mathbf{v}_k - Q\mathbf{c}$
23.  $\mathbf{d}\mathbf{r}_{k+1} = A\mathbf{d}\mathbf{x}_k$
24.  $\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{d}\mathbf{r}_{k+1}, \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}\mathbf{x}_{k+1}$
25. if  $\|\mathbf{r}_{k+1}\|_2 / \|\mathbf{r}_0\|_2 \leq \epsilon$  then exit
26.  $E\mathbf{e}_n = \mathbf{d}\mathbf{r}_{k+1}, Q\mathbf{e}_n = \mathbf{d}\mathbf{x}_{k+1}$
27.  $G\mathbf{e}_n = P^T \mathbf{d}\mathbf{r}_{k+1}, \mathbf{f} = \mathbf{f} + G\mathbf{e}_n, n = n + 1$
28. if  $n > s$  then  $n = 1$
29. end for

ILU(0) 前処理は, 係数行列を,

$$A = (\tilde{L} + I)(\tilde{U} + \tilde{D}) + R \quad (3)$$

と分解し,

$$K = (\tilde{L} + I)(\tilde{U} + \tilde{D}) \quad (4)$$

とする前処理である. ここで,  $\tilde{L}, \tilde{D}, \tilde{U}$  は, フィルインを考慮しない ILU(0) 分解によって生成された狭義下三角行

列, 対角行列, 狭義上三角行列を各々意味し,  $R$  は  $\tilde{L}$ ,  $\tilde{D}$ ,  $\tilde{U}$  の疎性を保つための誤差行列を意味する. ILU(0) 前処理では, 不完全分解過程で非零要素を除いた位置に現れるフィルインを考慮しない.

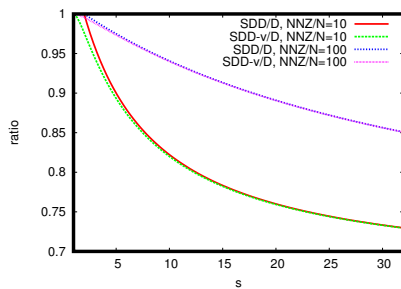
### 3.2 IDR(s)-R2 における SDD, SDD-v 行列 $P$ のときの計算量

SDD, SDD-v 行列  $P$  のときの IDR(s)-R2 における計算量を見積もる. 前処理つき IDR(s)-R2 法の算法の 18 行目から 29 行目がメインループとなり, その反復 1 回当りの計算量を Table 2 に示す.  $u, v$  は  $N$  次元ベクトルを表す.

**Table 2** Computational costs of IDR(s)-R2 method with both SDD and SDD-v matrices  $P$ .

$P$	$Au$	$u^T v$	$u \pm v$	total
D	$nnz$	$(3s + 1)N$	$4N$	$3sN + 5N + nnz$
SDD	$nnz$	$(2s + 3)N$	$4N$	$2sN + 7N + nnz$
SDD-v	$nnz$	$(2s + 3 - 1/s)N$	$4N$	$2sN + (7 - 1/s)N + nnz$

Fig. 3 に SDD と SDD-v の行列の反復 1 回当りの計算量の D の行列に対する同計算量の比率の変動の様子を示す. 平均非零要素数が  $nnz/N = 10$  の場合と  $nnz/N = 100$  の場合についてプロットした. 同様に, Fig. 4 に SDD-v の行列の同計算量の SDD の行列の同計算量に対する比率の変動の様子を示す. 平均非零要素数が  $nnz/N = 10$  と  $nnz/N = 100$  の 2 通りの場合をプロットした.

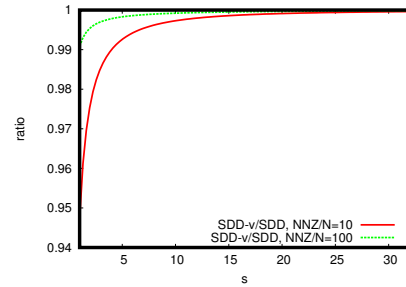


**Fig. 3** Variation of ratio of computational cost of SDD and SDD-v matrices to D matrix.

Fig. 3 より, SDD, SDD-v の行列のスリム化において, パラメータ  $s$  の値が大きくなり, 平均非零要素数  $nnz/N$  が少ないとき程, 計算量の削減の割合が大きくなる. また, Fig. 4 より, 同  $s$  の値が小さくなり, 平均非零要素数  $nnz/N$  が少ないとき程, SDD-v の行列は SDD の行列に対して, 計算量の削減の割合が大きくなる, ことがわかる.

### 3.3 計算機環境と計算条件

計算はすべて倍精度浮動小数点演算で行った. 計算機



**Fig. 4** Variation of ratio of computational cost of SDD-v matrix to SDD matrix.

は Nehalem (CPU: Intel Xenon X5570, クロック周波数: 2.93GHz, メモリ: 24Gbytes, OS: RedHat Enterprise Linux 5.6) を用いた. プログラムは Fortran90 を用いて実装し, 最適化オプションは “-O3” を使用した. 右辺項ベクトル  $b$  は厳密解を  $\hat{x} = (1, 1, \dots, 1)^T$  とし,  $b = A\hat{x}$  で作成した. 収束判定条件は相対残差の 2 ノルム:  $\|r_{k+1}\|_2 / \|r_0\|_2 \leq 10^{-8}$  とした. 初期近似解  $x_0$  はすべて 0 とした. 行列は予め対角スケーリングによって対角項をすべて 1.0 に正規化した. 最大反復回数は行列の次元数が 10000 以下のとき 10000 回, それ以上のときは次元数と等しい回数とした. パラメータ  $s$  は, 2, 4, 8, 16, 32 の 5 通り変化させた. 同  $\omega$  は, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5 の 6 通り変化させた. 擬似乱数 (pseudorandom number) 生成には, 合同式 (congruence) 法  $r_{k+1} = d \times r_k + e \pmod f$  を用いた. ただし,  $d = 1229$ ,  $e = 351750$ ,  $f = 1664501$ ,  $r_0 = 1$  とし, そのとき周期  $f$  となる<sup>2)</sup>.

### 3.4 テスト行列

Table 3 に 16 個のテスト行列の特徴を示す. これらの行列は, Matrix Market<sup>11)</sup> および Florida 大学の疎行列データベース<sup>8)</sup> から選出した. 表中の “ $nnz$ ” は非零要素数, “ave.  $nnz$ ” は平均非零要素数を各々意味する.

### 3.5 実験結果

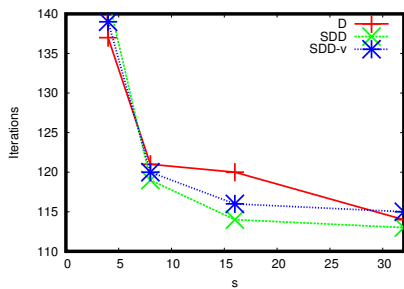
Table 4-Table 5 に行列  $P$  を SDD, SDD-v 化した ILU(0) 前処理つき IDR(s)-R2 法の収束性を示す. 表中の時間の単位はすべて秒とする. 表中の “TRR” は, 近似解  $x_{k+1}$  に対する真の相対残差 (True Relative Residual) 2 ノルムの常用対数の値である. 表中の “D” は行列  $P$  を乱数からなる密 (Dense) 行列, “SDD” は Slim Double Dense, “SDD-v” は Slim Double Dense variant 行列を各々意味する. また,  $s$  を変化させたとき Dense において計算時間が最短であるケースを示す. “ratio” は行列  $P$  を密行列とした場合の最短の合計時間に対する各行列の場合の合計時間の比を表す. Table 6 に, 各  $P$  における最速の度数を示す. また, Fig. 5-Fig. 8 に, 行列 epb1, epb3 における反復回数と  $s$  の関係, 合計時間と  $s$  の関係を各々示す.

**Table 3** Specification of test matrix.

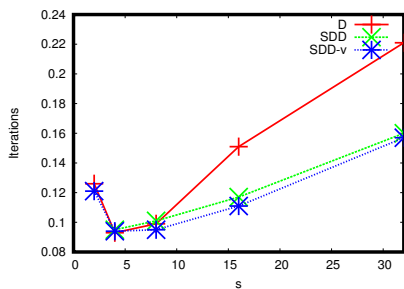
matrix	dimen- sion	$nnz$	ave. $nnz$	analytic field
language	399,130	1,216,334	3.05	language
epb3	84,617	463,625	5.48	heat exchanger
epb1	14,734	95,053	6.45	
epb2	25,228	175,027	6.94	
add20	2,395	13,151	5.49	electric circuit
memplus	17,758	99,147	5.58	
dc2	116,835	766,396	6.56	
ecl32	51,993	380,415	7.32	
matrix_9	103,430	1,205,518	11.66	
torso3	259,156	4,429,042	17.09	
big	13,209	91,465	6.92	structural
poisson3Da	13,514	352,762	26.10	
poisson3Db	85,623	2,374,949	27.74	
sme3Da	12,504	874,887	69.97	
sme3Db	29,067	2,081,063	71.60	
raefsky2	3,242	293,551	90.55	fluid

**Table 4** Convergence of preconditioned IDR( $s$ )-R2 method with D, SDD and SDD-v matrix  $P$ .

matrix	$P$	$s$	itr.	total time[s]	ratio	TRR
big	D	8	377	0.320	1.00	-8.01
	SDD	16	280	0.283	0.88	-8.12
	SDD-v	16	283	<b>0.277</b>	0.87	-8.15
dc2	D	8	144	1.307	1.00	-8.09
	SDD	8	151	1.216	0.93	-8.19
	SDD-v	8	148	<b>1.205</b>	0.92	-8.18
ecl32	D	8	174	0.726	1.00	-8.07
	SDD	8	178	0.681	0.94	-8.10
	SDD-v	8	172	<b>0.663</b>	0.91	-8.15
epb1	D	4	137	<b>0.093</b>	1.00	-8.08
	SDD	4	142	0.095	1.02	-8.11
	SDD-v	4	139	0.094	1.01	-8.09
epb2	D	4	41	0.063	1.00	-8.06
	SDD	32	38	0.069	1.10	-8.15
	SDD-v	4	40	<b>0.060</b>	0.95	-8.05
epb3	D	4	159	0.781	1.00	-8.07
	SDD	8	118	<b>0.658</b>	0.84	-8.11
	SDD-v	4	144	0.700	0.90	-8.20
language	D	2	10	0.355	1.00	-8.77
	SDD	16	10	0.333	0.94	-8.48
	SDD-v	8	9	<b>0.324</b>	0.91	-8.49
matrix_9	D	8	99	1.452	1.00	-8.02
	SDD	8	97	1.330	0.92	-8.02
	SDD-v	8	96	<b>1.322</b>	0.91	-8.31
memplus	D	16	210	0.348	1.00	-8.01
	SDD	16	230	<b>0.314</b>	0.90	-8.07
	SDD-v	16	251	0.334	0.96	-8.03
poisson3Da	D	4	65	0.175	1.00	-8.09
	SDD	4	64	<b>0.171</b>	0.98	-8.31
	SDD-v	8	62	0.175	1.00	-8.55
poisson3Db	D	8	122	2.824	1.00	-8.15
	SDD	8	122	2.728	0.97	-8.04
	SDD-v	8	120	<b>2.686</b>	0.95	-8.13
raefsky2	D	8	46	0.097	1.00	-8.24
	SDD	4	48	0.098	1.01	-8.16
	SDD-v	16	44	<b>0.091</b>	0.94	-8.04
sme3Da	D	32	433	2.449	1.00	-8.05
	SDD	32	429	<b>2.255</b>	0.92	-8.35
	SDD-v	32	435	2.279	0.93	-8.74



**Fig. 5** Fluctuation of iterations of IDR( $s$ )-R2 method with three kinds of  $P$  for matrix epb1.



**Fig. 6** Fluctuation of total time of IDR( $s$ )-R2 method with three kinds of  $P$  for matrix epb1.

Table 4-Table 7 より以下の知見が得られる .

- 3 種類の行列の各最速度数は SDD-v の行列  $P$  が 10 回と最も多かった .
- SDD, SDD-v の行列  $P$  はどちらも元の Dense の場合より最速回数が多かった .
- SDD では 3 ケース, SDD-v では 1 ケースで比が 1.0 を

超えた .

- 比 ratio 0.95 以下の分布において, SDD と SDD-v の間に差は見られなかった .

**Table 5** Convergence of preconditioned IDR( $s$ )-R2 method with D, SDD and SDD-v matrix  $P$ . (cont'd)

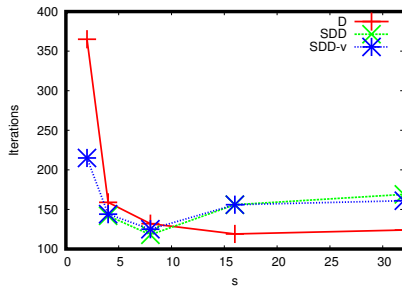
matrix	$P$	$s$	itr.	total time[s]	ratio	TRR
sme3Db	D	32	642	9.711	1.00	-8.08
	SDD	32	622	8.837	0.91	-8.17
	SDD-v	32	620	<b>8.796</b>	0.91	-8.06
torso3	D	2	27	0.980	1.00	-8.03
	SDD	32	27	0.949	0.97	-8.12
	SDD-v	32	27	<b>0.947</b>	0.97	-8.12
xenon1	D	4	979	6.181	1.00	-8.25
	SDD	16	334	<b>2.539</b>	0.41	-8.09
	SDD-v	8	458	3.075	0.50	-8.01

**Table 6** Fastest cases for each matrix  $P$ .

structure of $P$	fastest case
D	1
SDD	5
SDD-v	10
total	16

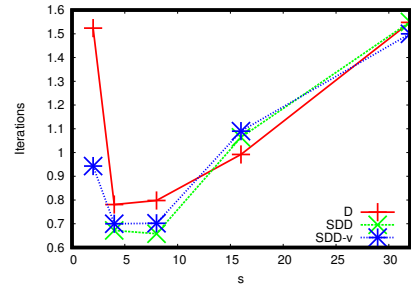
**Table 7** Distribution of ratio for each matrix  $P$ .

structure of $P$	$\sim 0.90$	$0.90 \sim 0.95$	$0.95 \sim 1.00$	$1.00 \sim$	Total
SDD	3	7	3	3	16
SDD-v	3	7	5	1	16



**Fig. 7** Fluctuation of iterations of IDR( $s$ )-R2 method with three kinds of  $P$  for matrix epb3.

Fig. 5-Fig. 6 より, 行列  $P$  の SDD 化, SDD-v 化により,  $P$  が密行列である場合よりも反復回数が安定しているケースがある.  $s$  が大きくなるに従って合計時間の減少率は大きくなる. Fig. 7-Fig. 8 より, 行列  $P$  の SDD 化, SDD-v 化により,  $P$  が密行列である場合よりも反復回数が増加し不安定となるケースがある. しかし, そのような場合でも, 計算量削減の効果により, 合計時間は元の Dense の場合との差は比較的小さい.



**Fig. 8** Fluctuation of total time of IDR( $s$ )-R2 method with three kinds of  $P$  for matrix epb3.

#### 4. まとめ

前処理つき IDR( $s$ )-R2 法の算法に行列  $P$  の SDD, SDD-v によりスリム化した行列  $P$  を適用し, その収束安定性を調べた. その結果, SDD-v 行列とした場合が最も高速であった. SDD 行列とした場合は, SDD-v 行列よりも計算時間はかかるが元の Dense 行列よりも計算時間が短かった.

#### 参考文献

- 1) 柿原正伸, 藤野清次: 緩和係数  $\omega$  を自動決定する対角緩和準口バスト ICCG 法の収束性, 情報処理学会論文誌, Vol.41, No.6, pp.1-11, 2005.
- 2) 森正武, 数値計算プログラミング, 岩波書店, 1990.
- 3) 尾上勇介, 藤野清次, IDR( $s$ ) 法系統の反復法に適用可能な計算量削減の工夫, 日本応用数学会論文誌, Vol.19, No.3, pp.329-350, 2009.
- 4) Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM, Philadelphia, 2003.
- 5) 関本幹, 尾上勇介, 藤野清次, 杉原正顯: IDR( $s$ ) 法ベースの残差縮小法の前処理について, 環瀬戸内応用数理研究部会第 14 回シンポジウム講演予稿集, 岡山理科大学, pp.35-40, 2011.
- 6) Sonneveld, P., van Gijzen, M. B., IDR( $s$ ): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., Vol.31, No.2, pp.1035-1062, 2008.
- 7) Sonneveld, P.: AGS - IDR - CGS - BiCGSTAB - IDR( $s$ ): the circle closed - A case of serendipity -, The Proc. of Int. Kyoto-Forum on Krylov subspace method, Kyoto, pp.1-14, 2008.
- 8) University of Florida Sparse Matrix Collection: <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- 9) Wesseling, P., Sonneveld, P.: Numerical Experiments with a Multiple Grid-and a Preconditioned Lanczos Type Methods, Lecture Notes in Math., Springer, 771, pp.543-562, 1980.
- 10) 尾上勇介, 藤野清次: IDR( $s$ ) 系統の反復法の行列  $P$  の適切な与え方について, 科研費分科会予稿集, 福岡建築会館, 2009.7.
- 11) Matrix Market の Home Page: <http://math.nist.gov/MatrixMarket/matrices.html>
- 12) Saad, Y., Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM, Philadelphia, 2003