2008－HPC－118 (2)
2008／12／16

IDR定理に基づく新しい反復法群の概観と性能評価

尾上勇介　藤野清次†　中嶋徳正*
九州大学大学院システム情報科学府　†九州大学情報基盤研究開発センター
*九州大学大学院システム情報科学研究院

IDR定理に基づくIDR(s)法の発表後，その改良版であるMR_IDR(s)法，Bi_IDR(s)法などが続々と同一著者らにより発表されてきた．最初のMR_IDR(s)法では，中間残差ノルムの最小化により，収束安定性が補強された．次のBi_IDR(s)法では，双直交条件の採用により，収束安定性が一層向上し，より洗練された解法になった．本稿では，これらの新生IDR(s)法ファミリーの概観を述べ，数値実験を通して，IDR(s)法ファミリーの収束特性を比較し議論する．

An overview of a family of new iterative methods based on IDR theorem and its performance evaluation

Yusuke Onoue　Seiji Fujino†　Norimasa Nakashima*
Graduate School of Information Science and Electrical Engineering, Kyushu University
†Research Institute for Information Technology, Kyushu University
*Faculty of Information Science and Electrical Engineering, Kyushu University

After birth of IDR(s) method based on IDR Theorem, two variants of MR_IDR(s) and Bi_IDR(s) methods were proposed one after another. The former method gained stability by adoptation of strategy of minimizing intermediate residual norm with extra computational cost. The latter method became sophiscated and elegant variant with stability by means of adoptation of bi-orthogonalization conditions. In this article, we overview a family of IDR(s) methods, and evaluate performance of these variants through several numerical experiments.

1　はじめに

2007年，P. Sonneveld と M. van Gijzen により，旧版のIDR定理[8]をさらに発展させた拡張IDR(s)定理に基づく反復法IDR(s)法が発表された[6]．IDRとは，Induced Dimension Reduction (数学的帰納法による次元縮小法) の略である．そして，論文[6]の中で，IDR(s)法がBiCGStab法，BiCGStab(L)法，GMRES法などと比べて，収束性が優れかつ必要メモリ量が少ないことが数値実験で示された．さらに，IDR(s)法の変形版として，Minimum Residual IDR(s) (最小残差IDR(s)，以下MR_IDR(s)と略す)法[7]とBi-orthogonalized IDR(s) (双直交IDR(s)，以下Bi_IDR(s)と略す)法[5]が次々と発表されてきた．

そこで，本稿では，MR_IDR(s)法，Bi_IDR(s)法の概要を述べ，数値実験を通して，これら2種類の反復法の有効性を調べる．本稿の構成は次のとおりである．第2節では，IDR(s)法の概要と算法を記述する．第3節で，MR_IDR(s)法の概要と算法，第4節でBi_IDR(s)法の概要と算法を記述する．第5節で，数値実験を通して，MR_IDR(s)法とBi_IDR(s)法の有効性を検証する．最後に，第6節でまとめを行う．

2　IDR(s)法

IDR(s)法の主な特徴について以下に述べる．

IDR定理

$A \in R^{N\times N}$とし，行列Pを一次独立なベクトル群$\boldsymbol{p}_i (i = 1, 2, \ldots, s)$を列ベクトルに持つ行列$P = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s)$，空間$\mathcal{G}_0$を完全Krylov空間$K_N(A, \boldsymbol{r}_0)$とする．さらに，一連の空間$\mathcal{G}_j (j = 1, 2, \ldots)$を，

$$\mathcal{G}_j := (I - \omega_j A)(\mathcal{G}_{j-1} \cap \mathrm{Null}(P^T)) \qquad (1)$$

と定義する．ここで，パラメータω_jは非零のスカラー値，$\mathrm{Null}(P^T)$は行列Pの転置P^Tの零空間

とする．このとき，次の定理が成り立つ．

(i) $\mathcal{G}_j \subseteq \mathcal{G}_{j-1}$ for all $j > 0$

(ii) $\mathcal{G}_j = \{\mathbf{0}\}$ for some $j \leq N$

空間 $\mathcal{G}_{j+1}$ に属する最初の残差 $\boldsymbol{r}_{n+1}$ の計算

IDR(s) 法では，IDR 定理で定義される空間 $\mathcal{G}_j$ に $s+1$ 個の残差ベクトルが属するように構成する．残差ベクトル $\boldsymbol{r}_{n+1}$ を，空間 $\mathcal{G}_{j+1}$ に属する最初の残差ベクトルとすると，残差ベクトル $\boldsymbol{r}_{n+1}$ は次式で計算される．

$$\boldsymbol{r}_{n+1} = (I - \omega_j A)\boldsymbol{v}_n,\ \boldsymbol{v}_n \in \mathcal{G}_j \cap \mathrm{Null}(P^T). \quad (2)$$

ここで，パラメータ ω_j は，残差ノルム $||\boldsymbol{r}_{n+1}||_2$ が最小になるように求める．

残差ベクトルの線形結合: ベクトル $\boldsymbol{v}_n$ の計算

$\boldsymbol{v}_n \in \mathcal{G}_j$ より，ベクトル $\boldsymbol{v}_n$ は空間 $\mathcal{G}_j$ に属する残差ベクトルの線形結合で表される．残差の差分ベクトル $\boldsymbol{e}_n := \boldsymbol{r}_{n+1} - \boldsymbol{r}_n$ を定義する．$\boldsymbol{r}_{n-i}(i = 0, \ldots, s) \in \mathcal{G}_j$ より，$\boldsymbol{e}_{n-i}(i = 1, \ldots, s) \in \mathcal{G}_j$ なので，ベクトル $\boldsymbol{v}_n$ は，

$$\boldsymbol{v}_n = \boldsymbol{r}_n - \sum_{i=1}^{s} c_i \boldsymbol{e}_{n-i} \quad (3)$$

と表せる．$\boldsymbol{v}_n \in \mathrm{Null}(P^T)$ より $P^T\boldsymbol{v}_n = 0$ が成り立つため，係数群 $c_i(i = 1, \ldots, s)$ を未知数とする $s \times s$ の連立一次方程式の求解から，(3) 式中の係数群 c_i が定まり，ベクトル $\boldsymbol{v}_n$ が求まる．

空間 $\mathcal{G}_{j+1}$ に属する k 番目の残差 $\boldsymbol{r}_{n+k}$ の計算

空間 $\mathcal{G}_{j+1}$ に属する $k(2 \leq k \leq s+1)$ 番目の残差 $\boldsymbol{r}_{n+k}$ は，以下の一連の計算で求められる．

IDR(s) 法における $\boldsymbol{r}_{n+k}$ の計算

1. Solve c_i from $P^T \sum_{i=1}^{s} c_i \boldsymbol{e}_{n+k-1-i} = P^T \boldsymbol{r}_{n+k-1}$
2. $\boldsymbol{v}_{n+k-1} = \boldsymbol{r}_{n+k-1} - \sum_{i=1}^{s} c_i \boldsymbol{e}_{n+k-1-i}$
3. $\boldsymbol{r}_{n+k} = (I - \omega_j A)\boldsymbol{v}_{n+k-1}$

ここで，$1 \leq i < k$ のとき，$\boldsymbol{r}_{n+k-1-i} \in \mathcal{G}_{j+1} \subseteq \mathcal{G}_j$ であり，かつ $k \leq i \leq s$ のとき，$\boldsymbol{r}_{n+k-1-i} \in \mathcal{G}_j$ であるので，$\boldsymbol{e}_{n+k-1-i} \in \mathcal{G}_j$ が成り立つ．すなわち，$\boldsymbol{r}_{n+k} \in \mathcal{G}_{j+1}$ である．

IDR(s) 法の算法を以下に示す．

算法 1: IDR(s) 法の算法

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

ϵ は収束判定のための微小な値とする．

3 MR_IDR(s) 法

MR_IDR(s) 法の主な特徴について以下に述べる．

空間 $\mathcal{G}_{j+1}$ に属する k 番目の残差 $\boldsymbol{r}_{n+k}$ の計算

空間 $\mathcal{G}_{j+1}$ に属する $k(1 \leq k \leq s+1)$ 番目の残差 $\boldsymbol{r}_{n+k}$ は，以下の一連の計算で求められる．

MR_IDR(s) 法における $\boldsymbol{r}_{n+k}$ の計算

1. Solve c_i from $P^T \sum_{i=1}^{s} c_i \boldsymbol{g}_{n+k-1-i} = P^T \boldsymbol{r}_{n+k-1}$
2. $\boldsymbol{v}_{n+k-1} = \boldsymbol{r}_{n+k-1} - \sum_{i=1}^{s} c_i \boldsymbol{g}_{n+k-1-i}$
3. $\boldsymbol{r}_{n+k} = (I - \omega_j A)\boldsymbol{v}_{n+k-1}$

ここで，補助ベクトル $\boldsymbol{g}_n$ は，次式で定義される．

$$\boldsymbol{g}_n := (-1) * \boldsymbol{e}_n = \boldsymbol{r}_n - \boldsymbol{r}_{n+1}. \quad (4)$$

$1 \leq i < k$ のとき，$\boldsymbol{r}_{n+k-i} \in \mathcal{G}_{j+1} \subseteq \mathcal{G}_j$ であり，かつ $k \leq i \leq s$ のとき，$\boldsymbol{r}_{n+k-i} \in \mathcal{G}_j$ であるので，

$\boldsymbol{g}_{n+k-i} \in \mathcal{G}_j$ が成り立つ．すなわち，$\boldsymbol{r}_{n+k+1} \in \mathcal{G}_{j+1}$ である．

中間残差の最小化

ベクトル $\boldsymbol{g}_i (i = n+k-s, \ldots, n+k-1, k = 0, \ldots, s)$ を列ベクトルとする行列 $G_{n+k} := (\boldsymbol{g}_{n+k-1} \cdots \boldsymbol{n+k-s})$ を定義する．行列 G_{n+k} の最初の k 列の列ベクトルを互いに直交させ，かつ空間 $\mathcal{G}_{j+1}$ に属する k 番目の中間残差 $\boldsymbol{r}_{n+k}$ を行列 G_{n+k} の最初の k 列の列ベクトルに直交させることで，中間残差 $\boldsymbol{r}_{n+k}$ を最小化できる．

以下に，MR_IDR(s) 法の算法を示す．

算法 2:　MR_IDR(s) 法の算法

Let $\boldsymbol{x}_0$ be a random vector, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
$G_{-1}, U_{-1} = O \in R^{N \times s}, M = I, \omega_0 = 1$
$n = 0, j = 0$
While $||\boldsymbol{r}_n||_2 / ||\boldsymbol{r}_0||_2 > \epsilon$ Do
　Do $k = 1, \ldots, s$
　　$\boldsymbol{m} = P^T \boldsymbol{r}_n$
　　Solve $\boldsymbol{c}$ from $M_{j-1}\boldsymbol{c} = \boldsymbol{m}$
　　$\boldsymbol{v} = \boldsymbol{r}_n - G_{j-1}\boldsymbol{c}, \bar{\boldsymbol{u}} = U_{j-1}\boldsymbol{c} + \omega_j \boldsymbol{v}$
　　$\bar{\boldsymbol{g}} = A\bar{\boldsymbol{u}}$
　　Do $i = 1, \ldots, k-1$
　　　$\alpha = (\boldsymbol{g}_{n-i}, \bar{\boldsymbol{g}})$
　　　$\bar{\boldsymbol{g}} = \bar{\boldsymbol{g}} - \alpha \boldsymbol{g}_{n-i}, \bar{\boldsymbol{u}} = \bar{\boldsymbol{u}} - \alpha \boldsymbol{u}_{n-i}$
　　End Do
　　$\alpha = \sqrt{(\bar{\boldsymbol{g}}, \bar{\boldsymbol{g}})}$
　　$\boldsymbol{g}_n = \frac{1}{\alpha}\bar{\boldsymbol{g}}, \boldsymbol{u}_n = \frac{1}{\alpha}\bar{\boldsymbol{u}}$
　　$\beta_n = (\boldsymbol{r}_n, \boldsymbol{g}_n)$
　　$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \beta_n \boldsymbol{g}_n, \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \beta_n \boldsymbol{u}_n$
　　$n = n + 1$
　End Do
　$G_j = (\boldsymbol{g}_{n-1} \cdots \boldsymbol{g}_{n-s}), U_j = (\boldsymbol{u}_{n-1} \cdots \boldsymbol{u}_{n-s})$
　$M_j = P^T G_j, \boldsymbol{m} = P^T \boldsymbol{r}_n$
　Solve $\boldsymbol{c}$ from $M_{j-1}\boldsymbol{c} = \boldsymbol{m}$
　$\boldsymbol{v} = \boldsymbol{r}_n - G_j \boldsymbol{c}, \boldsymbol{t} = A\boldsymbol{v}$
　$\omega_{j+1} = \dfrac{(\boldsymbol{t}, \boldsymbol{v})}{(\boldsymbol{t}, \boldsymbol{t})}$
　$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + U_j \boldsymbol{c} + \omega_{j+1}\boldsymbol{v}$
　$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - G_j \boldsymbol{c} - \omega_{j+1}\boldsymbol{t}$
　$n = n + 1, j = j + 1$
End While

上記の算法の 16 行目の $M_j = P^T G_j$ は，次式を用いることにより，低コストで計算できる．

$$P^T \boldsymbol{g}_{n-i} = P^T (\boldsymbol{r}_{n-i} - \boldsymbol{r}_{n-i+1}) / \beta_{n-i}.$$

4　Bi_IDR(s) 法

Bi_IDR(s) 法の主な特徴について以下に述べる．

Bi_IDR(s) 法の設計方針

残差ベクトル $\boldsymbol{r}_{n+1}$ を空間 $\mathcal{G}_{j+1}$ に属する最初の中間残差とする．Bi_IDR(s) 法では，以下の 2 つの直交条件が成立するように算法を設計する．

$$\boldsymbol{g}_{n+i} \perp \boldsymbol{p}_j \quad (i = 2 \cdots s, j = 1 \cdots i), \tag{5}$$

$$\boldsymbol{r}_{n+i+1} \perp \boldsymbol{p}_j \quad (i = 1 \cdots s, j = 1 \cdots i). \tag{6}$$

ここで，ベクトル $\boldsymbol{p}_i$ は，IDR 定理の行列 P の列ベクトルである．上記の 2 条件を用いることで，演算コストダウンと収束性の安定化が可能である．

空間 $\mathcal{G}_{j+1}$ に属する最初の残差 $\boldsymbol{r}_{n+1}$ の計算

(6) 式の直交条件は，最初の中間残差はベクトル $\boldsymbol{p}_1$ に直交することを意味する．最後の中間残差は，ベクトル $\boldsymbol{p}_1 \sim \boldsymbol{p}_s$ に直交する．すなわち，空間 G_j の最後の中間残差 $\boldsymbol{r}_n$ は，ベクトル $\boldsymbol{p}_1 \sim \boldsymbol{p}_s$ に直交する．したがって，次式が成り立つ．

$$\boldsymbol{r}_n \in \mathcal{G}_j \cap \mathrm{Null}(P^T). \tag{7}$$

残差ベクトル $\boldsymbol{r}_{n+1}$ は，次式で計算できる．

$$\boldsymbol{r}_{n+1} = (I - \omega_j A)\boldsymbol{r}_n. \tag{8}$$

空間 $\mathcal{G}_{j+1}$ に属する k 番目の残差 $\boldsymbol{r}_{n+k}$ の計算

空間 $\mathcal{G}_{j+1}$ に属する $k (2 \le k \le s+1)$ 番目の残差 $\boldsymbol{r}_{n+k}$ は，MR_IDR(s) 法と同様に求められる．$\boldsymbol{r}_{n+k}$ を計算するには，以下の連立一次方程式を解く必要がある．

$$P^T \sum_{i=1}^{s} c_i \boldsymbol{g}_{n+k-1-i} = P^T \boldsymbol{r}_{n+k-1},$$

$$\sum_{i=1}^{s} c_i \boldsymbol{p}_j^T \boldsymbol{g}_{n+k-1-i} = \boldsymbol{p}_j^T \boldsymbol{r}_{n+k-1} \ (j = 1, \ldots, s). \tag{9}$$

ここで，(5) 式の直交条件より，$j < k-1-i$ のとき，$\boldsymbol{p}_j^T \boldsymbol{g}_{n+k-1-i} = 0$ となる．また，(6) 式の直交条件より，$j < k-1$ のとき，$\boldsymbol{p}_j^T \boldsymbol{r}_{n+k-1} = 0$ となる．すなわち，$j < k-1-i$ の場合，$\boldsymbol{p}_j^T \boldsymbol{g}_{n+k-1-i}$ は計算不要で，$j < k-1$ の場合，$\boldsymbol{p}_j^T \boldsymbol{r}_{n+k-1}$ は計算不要である．したがって，MR_IDR(s) 法や，IDR(s) 法に比べて，演算コストが低く抑えられる．

以下に，Bi_IDR(s) 法の算法を示す．

算法 3:　Bi_IDR(s) 法の算法

Let $\boldsymbol{x}_0$ be a random vector, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
$\boldsymbol{g}_i = \boldsymbol{u}_i = \boldsymbol{0}, i = 1 \cdots s, M = I, \omega = 1$

3. $n = 0$
4. While $||\boldsymbol{r}_n||_2/||\boldsymbol{r}_0||_2 > \epsilon$ Do
5. $\boldsymbol{f} = P^T\boldsymbol{r}_n, \boldsymbol{f} = (\phi_1, \ldots, \phi_s)$
6. Do $k = 1, \ldots, s$
7. Solve $\boldsymbol{c}$ from $M\boldsymbol{c} = \boldsymbol{f}, \boldsymbol{c} = (\gamma_1, \ldots, \gamma_s)$
8. $\boldsymbol{v} = \boldsymbol{r}_n - \sum_{i=k}^{s} \gamma_i \boldsymbol{g}_i, \boldsymbol{u}_k = \omega\boldsymbol{v} + \sum_{i=k}^{s} \gamma_i \boldsymbol{u}_i$
9. $\boldsymbol{g}_k = A\boldsymbol{u}_k$
10. Do $i = 1, \ldots, k-1$
11. $\alpha = \dfrac{(\boldsymbol{q}_i, \boldsymbol{g}_k)}{\mu_{i,i}}$
12. $\boldsymbol{g}_k = \boldsymbol{g}_k - \alpha\boldsymbol{g}_i, \boldsymbol{u}_k = \boldsymbol{u}_k - \alpha\boldsymbol{u}_i$
13. End Do
14. $\mu_{i,k} = (\boldsymbol{q}_i, \boldsymbol{g}_k), i = k \cdots s, M_{i,k} = \mu_{i,k}$
15. $\beta = \dfrac{\phi_k}{\mu_{k,k}}$
16. $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \beta\boldsymbol{g}_k, \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \beta\boldsymbol{u}_k$
17. If $k < s$ then
18. $\phi_i = 0, i = 1 \cdots k, \phi_i = \phi_i - \beta\mu_{i,k}, i = k+1 \cdots s$
19. $\boldsymbol{f} = (\phi_1, \ldots, \phi_s)$
20. End If
21. $n = n + 1$
22. End Do
23. $\boldsymbol{t} = A\boldsymbol{r}_n$
24. $\omega = \dfrac{(\boldsymbol{t}, \boldsymbol{r})}{(\boldsymbol{t}, \boldsymbol{t})}$
25. $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \omega\boldsymbol{r}_n, \boldsymbol{r}_{n+1} = \boldsymbol{r}_n - \omega\boldsymbol{t}$
26. $n = n + 1$
27. End While

5 数値実験

5.1 計算機環境と計算条件

数値実験は，九州大学情報基盤研究開発センターに設置されたIBM eServer p5モデル595ですべて行った．OSはIBM AIX 5L，主メモリは64Gbyte，CPUはPOWER5（クロック周波数1.9GHz）を使用した．プログラムはFortran90で実装し，計算はすべて倍精度浮動小数点演算で行い，コンパイラはIBM XL Fortran version 9.1，最適化オプションは，-O3 -qarch=pwr5 -qtune=pwr5 -qhotを使用した．反復法の初期近似解 $\boldsymbol{x}_0$ はすべて零とし，最大反復数は10000回，収束判定条件は，相対残差の2ノルム $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2$ の値が 10^{-12} 以下になったときとした．

表1にテスト行列の特徴を示す．これらの行列は，疎行列データベース [1] [3] から選出した．

表 1: テスト行列の特徴

行列	次元数	非零要素数	平均非零要素
big	13,209	91,465	6.92
epb1	14,734	95,053	6.45
epb2	25,228	175,027	6.94
garon2	13,535	373,235	27.58
memplus	17,758	126,150	7.10
poisson3da	13,514	352,762	26.10
poisson3db	85,623	2,374,949	27.74
raefsky2	3,242	293,551	90.55
sme3da	12,504	874,887	69.97
sme3db	29,067	2,081,063	71.60
xenon1	48,600	1,181,120	24.30
xenon2	157,464	3,866,688	24.56

5.2 実験結果

表2に，IDR(s)法，MR_IDR(s)法，Bi_IDR(s)法の最短計算時間を示す．表中の「$s_{\text{opt.}}$」は，各解法でパラメータ s を1 ～ 10まで1刻みで変えたとき，計算時間が最短となるパラメータ s の値を表す．「ratio」は，IDR(s)法の計算時間を1としたときの計算時間の比率を表す．また，3種類の解法のうち計算時間が最短のものを太字で表記した．

表2から，12個中10個の行列 (表2中で下線つき行列を除く) で，Bi_IDR(s)法が最も速く収束したことがわかる．また，行列epb1では，どの解法も $s = 2$ のとき，計算時間が最短だった．反復回数の多少は，MR_IDR(s)法 <IDR(s)法 <Bi_IDR(s)法の順であるのに対し，計算時間の長短は，Bi_IDR(s)法 <IDR(s)法 <MR_IDR(s)法の順で逆順である．以上から，演算コストは，Bi_IDR(s)法が最も低く，MR_IDR(s)法が最も高いことがわかる．

図1に，行列big, epb1におけるIDR(s)法およびMR_IDR(s)法，Bi_IDR(s)法の反復回数の変動を示す．図1では，実線がIDR(s)法，破線がMR_IDR(s)法，点線がBi_IDR(s)法の反復回数の変動の様子を各々示してある．図1から，反復回数の多さは，MR_IDR(s)法 <Bi_IDR(s)法 <IDR(s)法の順になるケースが多いことがわかる．

図2に，行列big, epb1におけるIDR(s)法およびMR_IDR(s)法，Bi_IDR(s)法の計算時間の変動を示す．図2から，計算時間の長短は，Bi_IDR(s)法 <IDR(s)法 <MR_IDR(s)法の順になるケースが多いことがわかる．

図3に，行列big, epb1におけるIDR(4)法およ

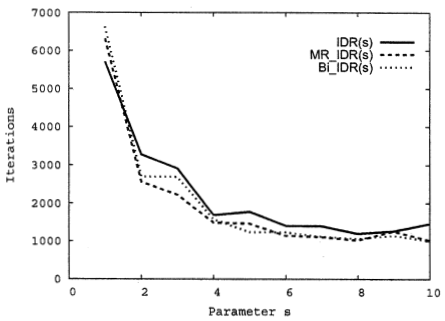
び MR_IDR(4) 法，Bi_IDR(4) 法の相対残差履歴を示す．図 3 より，MR_IDR(4) 法，Bi_IDR(4) 法は，IDR(s) 法に比べて，相対残差の振動が小さく，少ない反復回数で収束していることがわかる．

表 2: 3 種類の IDR(s) 法系統の反復法の最短の計算時間 [単位：秒]

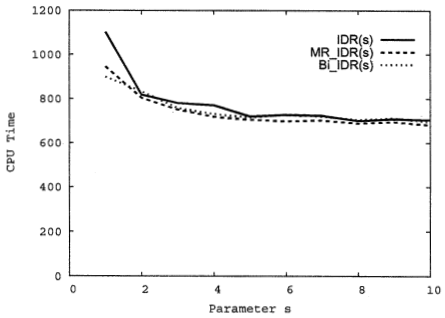
matrix	method	$s_{\mathrm{opt.}}$	itr.	time	ratio	mem. [MB]
big	IDR(s)	4	1687	1.36	1.00	2.91
	MR_IDR(s)	4	1489	1.77	1.30	3.82
	Bi_IDR(s)	7	1111	**0.98**	0.72	3.92
epb1	IDR(s)	2	818	0.58	1.00	2.49
	MR_IDR(s)	2	803	0.72	1.24	3.06
	Bi_IDR(s)	2	833	**0.49**	0.84	2.61
epb2	IDR(s)	3	450	0.68	1.00	4.99
	MR_IDR(s)	2	473	0.79	1.16	5.37
	Bi_IDR(s)	2	481	**0.54**	0.79	4.60
garon2	IDR(s)	2	777	1.26	1.00	5.76
	MR_IDR(s)	3	722	1.31	1.04	6.07
	Bi_IDR(s)	2	758	**1.12**	0.89	5.86
memplus	IDR(s)	5	574	0.67	1.00	4.36
	MR_IDR(s)	3	751	0.95	1.42	4.49
	Bi_IDR(s)	2	782	**0.62**	0.93	3.27
poisson-3da	IDR(s)	2	263	0.52	1.00	5.33
	MR_IDR(s)	4	232	0.54	1.04	6.87
	Bi_IDR(s)	4	238	**0.46**	0.88	6.05
poisson-3db	IDR(s)	5	528	15.31	1.00	41.22
	MR_IDR(s)	3	551	16.00	1.05	41.88
	Bi_IDR(s)	5	518	**14.77**	0.96	41.88
raefsky2	IDR(s)	7	420	0.40	1.00	4.05
	MR_IDR(s)	3	491	0.44	1.10	3.92
	Bi_IDR(s)	7	431	**0.38**	0.95	4.07
sme3da	IDR(s)	7	2272	**9.30**	1.00	12.64
	MR_IDR(s)	9	2088	10.01	1.08	15.02
	Bi_IDR(s)	7	2415	9.59	1.03	12.73
sme3db	IDR(s)	9	2668	45.47	1.00	31.25
	MR_IDR(s)	6	3441	56.62	1.25	32.13
	Bi_IDR(s)	4	3546	**44.53**	0.98	28.14
xenon1	IDR(s)	2	2240	12.12	1.00	18.15
	MR_IDR(s)	3	2015	12.98	1.07	21.86
	Bi_IDR(s)	4	1952	**10.74**	0.89	20.75
xenon2	IDR(s)	1	2725	**77.89**	1.00	55.66
	MR_IDR(s)	1	2847	84.88	1.09	59.27
	Bi_IDR(s)	1	2911	78.01	1.00	56.87

5.3 偽収束についての検証

IDR(s) では，パラメータ s が大きな値のとき，偽収束が起こる場合があった．本節では，実非対称 Toeplitz 行列 [2] において，MR_IDR(s) 法や Bi_IDR(s) 法において，偽収束が起こるか否かを検証した．本実験では，Toeplitz 行列の次元数を 2000, 行列中の要素の値を $\gamma = 1.5$ とした．

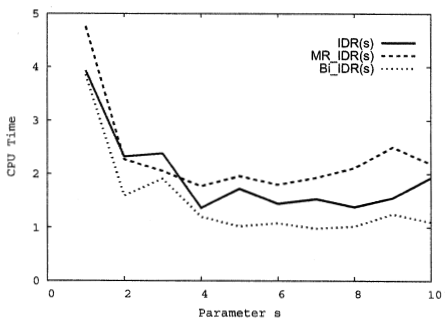


(a) 行列 big

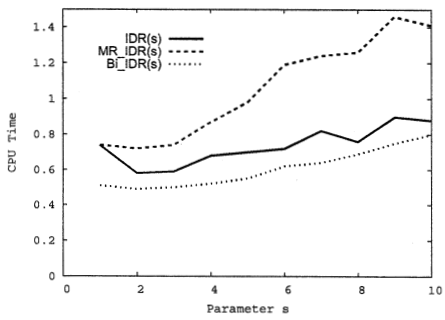


(b) 行列 epb1

図 1: 3 種類の IDR(s) 法系統の反復法の反復回数の変動

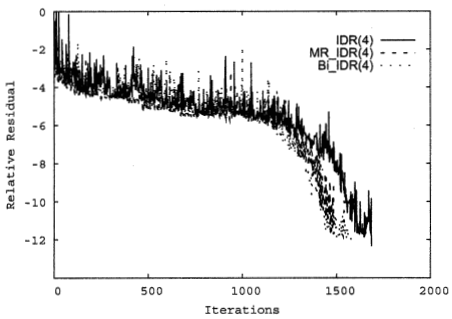


(a) 行列 big

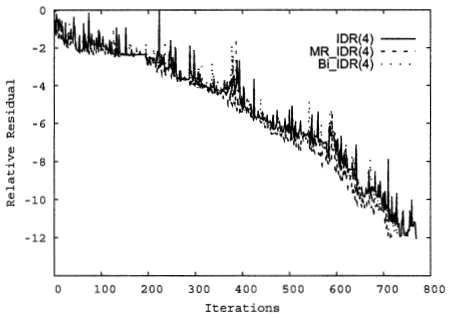


(b) 行列 epb1

図 2: 3 種類の IDR(s) 法系統の反復法の計算時間



(a) 行列 big



(b) 行列 epb1

図 3: 3 種類の IDR(s) 法系統の反復法の相対残差履歴

図 4 に Toeplitz 行列における IDR(s) 法および MR_IDR(s) 法，Bi_IDR(s) 法の真の相対残差 (TRR) の常用対数の値 ($\log_{10}$) の変動の様子を示す．真の相対残差とは，$||\boldsymbol{b} - A\boldsymbol{x}_n||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$ で定義される．図 4 では，実線が IDR(s) 法，一点鎖線が IDR(s) 法においてパラメータ ω を $\kappa = 0.7$ として修正した場合 [4]，破線が MR_IDR(s) 法，点線が Bi_IDR(s) 法の TRR の変動を表す．

図 4 から，IDR(s) 法では，$s \geq 18$ で偽収束が起こったことがわかる．また，IDR(s) 法ではパラメータ ω を修正することで，偽収束の問題はある程度改善され，$s \geq 39$ で偽収束が起こるようになる．一方，MR_IDR(s) 法，Bi_IDR(s) 法では，偽収束は全く起こらなかった．

6 まとめ

本稿では，IDR 定理から生まれた 2 種類の反復法 MR_IDR(s) 法，Bi_IDR(s) 法の概要を述べた．次に，数値実験により，両者の収束性能を比較検証した．その結果，MR_IDR(s) 法は，元の IDR(s) 法に比べて，中間残差ノルムの評価コストが高く，収束性は逆に僅かに遅くなるという評価が得られた．次に，Bi_IDR(s) 法は，元の IDR(s) 法に比べて，演算コストが低く抑えられ，収束性が向上することがわかった．さらに，IDR(s) 法では，パラメータ s が大きな値のとき偽収束が起こる場合があったのに対して，新しい MR_IDR(s) 法，Bi_IDR(s) 法では，偽収束が起こらないことがわかった．

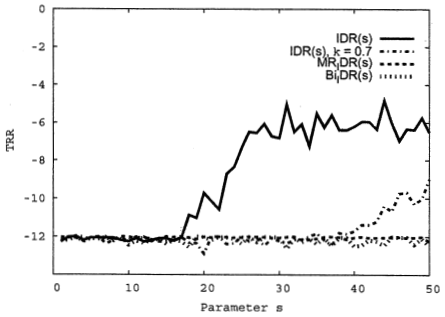


図 4: Toeplitz 行列における 4 種類の反復法の真の相対残差 (TRR) の変動の様子．

参考文献

[1] Davis, T. : Univ. of Florida sparse matrix collection, http://www.cise.ufl.edu/research/sparse /matrices /index.html

[2] 藤野清次, 張紹良 : 反復法の数理, 朝倉書店, 1996.

[3] Matrix Market の HP: http://math.nist.gov/MatrixMarket/matrices.html

[4] Sleijpen, G., van der Vorst, H.: Maintaining convergence properties of BiCGstab methods in finite precision arithmetic, Numerical Algorithms, Vol.10, pp.203-223, 1995.

[5] Sonneveld, P., van Gijzen, M.B. : An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties, TR 08-21, Math. Anal., Delft Univ. of Tech., 2008.

[6] Sonneveld, P., van Gijzen, M.B.: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, TR 07-07, Delft Univ. of Tech., 2007.

[7] van Gijzen, M.B., Sonneveld, P.: An IDR(s) variant with minimal intermediate residual norms, The Proc. of Int. Kyoto-Forum on Krylov Subspace method, pp.85-92, 2008.

[8] 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.