数理解析研究所講究録
第1791巻 2012年 11-20

近似解の精度を改善するIDRstab法

A variant of the IDRstab method for improving the accuracy of approximate solutions

相原 研輔 *, 阿部 邦美 †, 石渡 恵美子 **

* 東京理科大学大学院, † 岐阜聖徳学園大学経済情報学部, ** 東京理科大学理学部

Kensuke Aihara*, Kuniyoshi Abe† and Emiko Ishiwata**

* Graduate School of Science, Tokyo University of Science

† Faculty of Economics and Information, Gifu Shotoku University

** Department of Mathematical Information Science, Tokyo University of Science

1 はじめに

大規模非対称行列を係数に持つ線形方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を近似的に解く反復法として，帰納的次元縮小 (Induced Dimension Reduction, IDR) 定理に基づく IDR(s) 法 [13] が提案されている．ここで，A は $n \times n$ 行列，右辺項 $\boldsymbol{b}$ は n 次元ベクトルである．IDR(s) 法は，$s = 1$ のとき，Bi-Conjugate Gradient STABilized (BiCGSTAB) 法 [15] と数学的に等価になり，$s > 1$ のとき，BiCGSTAB 法における初期シャドウ残差を s 本に拡張したものであると解釈できる [9].

IDR(s) 法の反復過程では，BiCGSTAB 法と同様に，1 次の安定化多項式を用いて残差ノルムの最小化を行う．従って，非対称性の強い実行列に対して，IDR(s) 法は不安定な収束性を示す場合がある．この問題を解決するため，IDR(s) 法の安定化多項式を ℓ 次に拡張した IDRstab 法 [12] が提案されている．IDRstab 法は，パラメータ s, ℓ に対して，$s = 1$ のとき，BiCGstab(ℓ) 法 [8] と，$\ell = 1$ のとき，IDR(s) 法とそれぞれ数学的に等価である．また，IDR(s) 法に L 次の安定化多項式を付加した GBi-CGSTAB(s, L) 法 [14] も提案されている．この解法は，IDRstab 法と数学的に等価であるが，実装法が異なる．本論文は IDRstab 法に限って議論する．

$s > 1$, $\ell > 1$ を用いる IDRstab 法は，しばしば IDR(s) 法や BiCGstab(ℓ) 法よりも優れた収束性を示す．しかし，丸め誤差の影響により，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが異なる場合がある．このとき，真の残差ノルムは漸化式から求まる残差ノルムよりも大きい値で停滞し，近似解の精度が劣化する．そこで，我々は残差を更新する漸化式を修正することで，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いを一致させ，近似解の精度の改善を試みる．しかし，この漸化式の修正によって，行列ベクトル積の計算回数が増加する．一方，IDRstab 法は s や ℓ が大きくなるにつれて，行列ベクトル積に比べ，ベクトルを更新する AXPY の計算回数が相対的に多くなる．よって，計算時間は AXPY の計算回数に大きく左右される．AXPY とは，スカラー a と n 次元ベクトル $\boldsymbol{x}$, $\boldsymbol{y}$ に対して，$a\boldsymbol{x} + \boldsymbol{y}$ の形式の演算を表す．本論文では，残差を更新する漸化式を修正すると同時に，AXPY の計算回数を削減する工夫を行う．結果として，近似解の精度を改善し，かつ計算効率の良い新たな IDRstab 法の実装法を提案する．

さらに，従来の BiCG 系統の解法と同様に，IDRstab 法の残差ノルムは振動する．BiCG 系統の解法に対して，残差ノルムの振る舞いを滑らかにする擬似最小残差 (Quasi-Minimal Residual, QMR) アプローチの解法が開発されている．その代表的な解法として，QMR 法 [5] や QMRCGSTAB 法 [2] などがある．これらの解法が BiCG 法 [4] や BiCGSTAB 法の QMR 化によって得られるのと同様に，IDR(s) 法の残差ノルムの振る舞いを滑らかにした QMRIDR(s) 法 [3] が近年では提案されている．また，文献 [16] では，数値的に安定な基底の生成過程に基づいて，Flexible QMRIDR 法および Multi-Shift QMRIDR 法が提案されており，いずれも残差ノルムの振る舞いが滑らかである．しかし，高次の安定化多項式を用いる IDRstab 法の QMR 化は，文献 [3, 16] では扱われていない．

一方，残差ノルムの振る舞いを滑らかにするための異なる手法として，スムージング [7,17,18] が提案されてきた．特に，BiCG 法に QMR スムージング [18] を適用すると，QMR 法によって生成される残差と数学的に等価な残差を生成できることが知られている [18]．我々は，この QMR スムージングを本論文で提案する IDRstab 法の変形アルゴリズムに適用する．このとき，我々の提案する IDRstab 法の変形アルゴリズムは，残差を更新する漸化式を修正したため，計算量を大きく増加させることなく適用できる．一般に，スムージングによって得られる近似解の精度は，スムージング前の近似解の精度に比べて，向上こそしないものの，同程度の精度が維持されることが知られている [6]．すなわち，我々の提案する IDRstab 法の変形アルゴリズムを利用することで，近似解の精度が改善されるという利点を引き継いだまま，効率良く IDRstab 法の QMR 化が実現できる．数値実験において，近似解の精度が改善され，残差ノルムの振る舞いが滑らかになることを示す．

2 IDRstab 法

本節では，IDRstab 法の概略を述べる．IDR 定理に基づく反復法は，次元が縮小する空間列 $\{\mathcal{G}_j\}$ に対して，$\mathcal{G}_j$, $j = 0, 1, 2, \ldots$ に属する残差と対応する近似解を構築する．ただし，$\mathcal{G}_0 \equiv \mathcal{K}_n(A, \boldsymbol{r}_0)$, $\mathcal{G}_{j+1} \equiv (I - \omega_{j+1}A)(\mathcal{G}_j \cap \tilde{R}_0^{\perp})$ であり，$\mathcal{K}_n(A, \boldsymbol{r}_0)$ は行列 A と初期残差 $\boldsymbol{r}_0 \equiv \boldsymbol{b} - A\boldsymbol{x}_0$ で張る n 次の Krylov 部分空間，$\tilde{R}_0^{\perp}$ は $n \times s$ 行列 $\tilde{R}_0$ の像空間に対する直交補空間，ω_j は非零のスカラーを表す．

IDRstab 法では，$\mathcal{G}_k$ に属する残差 $\boldsymbol{r}_k$ を $\mathcal{G}_{k+\ell}$ に属する残差 $\boldsymbol{r}_{k+\ell}$ へと更新する．ただし，整数 k は ℓ の倍数である．この残差の更新過程を “1 サイクル” と数える．この 1 サイクルは，$\mathcal{G}_k$ に属するベクトルを $\mathcal{G}_k \cap \tilde{R}_0^{\perp}$ へ写す “IDR step” と，IDR step で得られた $\mathcal{G}_k \cap \tilde{R}_0^{\perp}$ に属するベクトルに ℓ 次の多項式を作用させ，$\mathcal{G}_{k+\ell}$ へと写す “polynomial step” の 2 つのステップによって構成される．以下に各ステップの計算過程を示す．

2.1 IDR step

まず，1 サイクルの始まりにおいて，残差 $\boldsymbol{r}_k \in \mathcal{G}_k$ と対応する近似解 $\boldsymbol{x}_k$，および $n \times s$ 行列 AU_k と対応する行列 U_k が得られているとする．ただし，AU_k の各列ベクトルは $\mathcal{G}_k$ に属するものとする．ここで，行列 $\Pi_i^{(j)}$ を

$$\Pi_i^{(j)} \equiv I - A^i U_k^{(j-1)} \sigma_j^{-1} \tilde{R}_0^* A^{j-i},\ \sigma_j \equiv \tilde{R}_0^* A^j U_k^{(j-1)},\ i = 0, 1, \ldots, j,\ j = 1, 2, \ldots, \ell$$

と定義すると，$\tilde{R}_0^* \Pi_j^{(j)} = O$ かつ $\Pi_{i+1}^{(j)} A = A \Pi_i^{(j)}$ を満たす．

IDR step では，$\Pi_1^{(j)}$ を用いて残差が ℓ 回更新される．この更新に伴う計算過程を ℓ 回の “繰り返し” と呼び，j 回目 ($j = 1, 2, \ldots, \ell$) の繰り返しを添え字 “(j)” で表す．いま，$j-1$ 回の繰り返しによって，残差 $\boldsymbol{r}_k^{(j-1)}$ と対応する近似解 $\boldsymbol{x}_k^{(j-1)}$，およびベクトル $A^i \boldsymbol{r}_k^{(j-1)}$, $i = 1, 2, \ldots, j-1$ と行列 $A^i U_k^{(j-1)}$, $i = 0, 1, 2, \ldots, j$ が得られたとする．ただし，$\boldsymbol{x}_k^{(0)} \equiv \boldsymbol{x}_k$, $\boldsymbol{r}_k^{(0)} \equiv \boldsymbol{r}_k$, $U_k^{(0)} \equiv U_k$ である．このとき，j 回目の繰り返しは，以下のように行う．

まず，ベクトル $A^i \boldsymbol{r}_k^{(j)}$, $i = 0, 1, \ldots, j-1$ は，$\Pi_{i+1}^{(j)}$ による写像によって，

$$A^i \boldsymbol{r}_k^{(j)} \equiv \Pi_{i+1}^{(j)} A^i \boldsymbol{r}_k^{(j-1)} = A^i \boldsymbol{r}_k^{(j-1)} - A^{i+1} U_k^{(j-1)} \vec{\alpha}^{(j)},\ \vec{\alpha}^{(j)} \equiv \sigma_j^{-1} (\tilde{R}_0^* A^{j-1} \boldsymbol{r}_k^{(j-1)}) \tag{1}$$

と更新される．また，対応する近似解 $\boldsymbol{x}_k^{(j)}$ の漸化式は，

$$\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_k^{(j-1)} + U_k^{(j-1)} \vec{\alpha}^{(j)} \tag{2}$$

となる．

次に，行列 $A^iU_k^{(j)}$ の各列ベクトルは，$\Pi_i^{(j)}$ による写像と行列ベクトル積を用いて，Krylov 部分空間 $\mathcal{K}_s(\Pi_i^{(j)}A, \Pi_i^{(j)}A^i\boldsymbol{r}_k^{(j)})$ の基底となるように構築される．具体的には，まずベクトル $A^{j-1}\boldsymbol{r}_k^{(j)}$ に A を掛けることで，$A^j\boldsymbol{r}_k^{(j)}$ が得られる．そして，$A^iU_k^{(j)}$, $i=0,1,\ldots,j$ の 1 列目は，$\Pi_i^{(j)}$ を用いて，

$$A^iU_k^{(j)}\boldsymbol{e}_1 \equiv \Pi_i^{(j)}A^i\boldsymbol{r}_k^{(j)} = A^i\boldsymbol{r}_k^{(j)} - A^iU_k^{(j-1)}\vec{\beta}_1^{(j)},\ \vec{\beta}_1^{(j)} \equiv \sigma_j^{-1}\vec{\rho}_1^{(j)},\ \vec{\rho}_1^{(j)} \equiv \tilde{R}_0^*A^j\boldsymbol{r}_k^{(j)} \tag{3}$$

と計算される．同様にして，$q<s$ に対して，ベクトル $A^jU_k^{(j)}\boldsymbol{e}_q$ に A を掛けることで，$A^{j+1}U_k^{(j)}\boldsymbol{e}_q = \boldsymbol{c}_q^{(j)} \equiv A(A^jU_k^{(j)}\boldsymbol{e}_q)$ が計算され，$A^iU_k^{(j)}$, $i=0,1,\ldots,j$ の $q+1$ 列目は，$\Pi_i^{(j)}$ を用いて，

$$A^iU_k^{(j)}\boldsymbol{e}_{q+1} = \Pi_i^{(j)}A^{i+1}U_k^{(j)}\boldsymbol{e}_q = A^{i+1}U_k^{(j)}\boldsymbol{e}_q - A^iU_k^{(j-1)}\vec{\beta}_{q+1}^{(j)},\ \vec{\beta}_{q+1}^{(j)} \equiv \sigma_j^{-1}\vec{\rho}_{q+1}^{(j)},\ \vec{\rho}_{q+1}^{(j)} \equiv \tilde{R}_0^*\boldsymbol{c}_q^{(j)} \tag{4}$$

となる．以上の計算過程によって，j 回目の繰り返しでは，$A^i\boldsymbol{r}_k^{(j)} \in \mathcal{G}_k \cap \tilde{R}_0^{\perp}$, $i=0,1,\ldots,j-1$ かつ $A^iU_k^{(j)}\boldsymbol{e}_q \in \mathcal{G}_k \cap \tilde{R}_0^{\perp}$, $q=1,2,\ldots,s$, $i=1,2,\ldots,j$ を満たす．

2.2 polynomial step

IDR step の ℓ 回の繰り返しによって，残差 $\boldsymbol{r}_k^{(\ell)}$ と対応する近似解 $\boldsymbol{x}_k^{(\ell)}$，ベクトル $A^i\boldsymbol{r}_k^{(\ell)}$, $i=1,2,\ldots,\ell$ および $\ell+2$ 個の $n\times s$ 行列 $A^iU_k^{(\ell)}$, $i=0,1,\ldots,\ell+1$ が得られる．polynomial step では，$\boldsymbol{r}_k^{(\ell)}$ と $AU_k^{(\ell)}$ は，以下のように更新される．

$$\boldsymbol{r}_{k+\ell} = \boldsymbol{r}_k^{(\ell)} - \gamma_{1,k}A\boldsymbol{r}_k^{(\ell)} - \cdots - \gamma_{\ell,k}A^\ell\boldsymbol{r}_k^{(\ell)},\ AU_{k+\ell} = AU_k^{(\ell)} - \gamma_{1,k}A^2U_k^{(\ell)} - \cdots - \gamma_{\ell,k}A^{\ell+1}U_k^{(\ell)}.$$

ただし，係数 $\gamma_{1,k},\gamma_{2,k},\ldots,\gamma_{\ell,k}$ は，ノルム $\|\boldsymbol{r}_{k+\ell}\|_2$ の最小化から決定される．このとき，$\boldsymbol{r}_{k+\ell} \in \mathcal{G}_{k+\ell}$ かつ $AU_{k+\ell}\boldsymbol{e}_i \in \mathcal{G}_{k+\ell}$, $i=1,2,\ldots,s$ を満たす [12]．近似解 $\boldsymbol{x}_k^{(\ell)}$ および行列 $U_k^{(\ell)}$ に対しても，それぞれ $\boldsymbol{r}_k^{(\ell)}$, $AU_k^{(\ell)}$ に対応する更新を行うことで，$\boldsymbol{x}_{k+\ell}$, $U_{k+\ell}$ が得られる．以上，IDRstab 法の 1 サイクルを終了する．

3 IDRstab 法の変形アルゴリズム

従来の IDRstab 法では，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが異なり，近似解の精度が劣化する場合がある．そこで，まず漸化式から求まる残差ノルムと真の残差ノルムの振る舞いを一致させるため，残差を更新する漸化式を修正する．さらに，AXPY の計算回数を削減する工夫を行うことで，効率良く近似解の精度を改善する IDRstab 法の変形アルゴリズムを導出する．

3.1 残差を更新する漸化式の見直し

漸化式から求まる残差ノルムと真の残差ノルムの振る舞いの違いは，近似解と残差を更新する漸化式に深い関わりがある [10,11]．我々は，近似解と残差を更新する漸化式に対して，BiCG 系統の解法に近い以下の形式を考える．

$$\boldsymbol{x}_{m+1} = \boldsymbol{x}_m + \boldsymbol{u}_m,\ \boldsymbol{r}_{m+1} = \boldsymbol{r}_m - A\boldsymbol{u}_m,\ m=0,1,2,\ldots. \tag{5}$$

ここで，$\boldsymbol{u}_m$ は適当な探索方向ベクトルであり，ベクトル $A\boldsymbol{u}_m$ は $\boldsymbol{u}_m$ に A を陽に掛けることで得る．このように A を陽に掛ける操作は，漸化式から求まる残差ノルム $\|\boldsymbol{r}_{m+1}\|_2$ と真の残差ノルム $\|\boldsymbol{b}-A\boldsymbol{x}_{m+1}\|_2$ の振る舞いを一致させるために重要である．

一方，IDRstab 法の残差を更新する漸化式 (1) では，行列 $AU_k^{(j-1)}$ は $U_k^{(j-1)}$ に対して (5) のように A を陽に掛けるのではなく，$\Pi_1^{(j-1)}$ による写像で構築される．このため，計算されたベクトル $AU_k^{(j-1)}\vec{\alpha}^{(j)}$

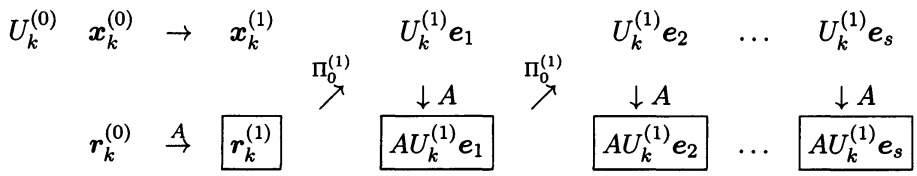

図 1: 変形した IDR step の 1 回目の繰り返し.

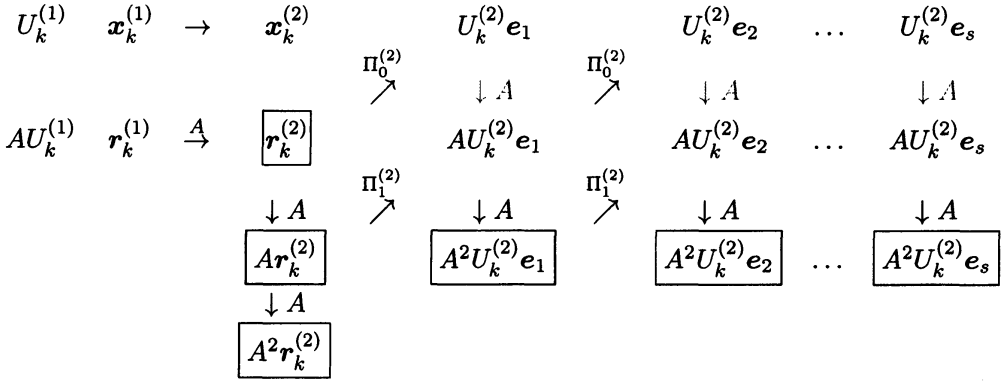

図 2: 変形した IDR step の 2 回目の繰り返し.

は，丸め誤差の影響で正しい値からかけ離れる可能性があり，漸化式から求まる残差ノルムと真の残差ノルムの間に差が生じる場合がある．そこで，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いを一致させるため，(5) の形式に基づく以下の漸化式を導入する．

$$\boldsymbol{r}_k^{(j)} = \boldsymbol{r}_k^{(j-1)} - A(U_k^{(j-1)}\vec{\alpha}^{(j)}),\ j = 1, 2, \ldots, \ell. \tag{6}$$

(6) では，探索方向ベクトル $U_k^{(j-1)}\vec{\alpha}^{(j)}$ に A を陽に掛ける．また，polynomial step においても，同様のアプローチを用いる．すなわち，残差 $\boldsymbol{r}_{k+\ell}$ は以下のように更新する．

$$\boldsymbol{r}_{k+\ell} = \boldsymbol{r}_k^{(\ell)} - A(\gamma_{1,k}\boldsymbol{r}_k^{(\ell)} - \gamma_{2,k}A\boldsymbol{r}_k^{(\ell)} - \cdots - \gamma_{\ell,k}A^{\ell-1}\boldsymbol{r}_k^{(\ell)}). \tag{7}$$

ここで，ベクトル $A(\sum_{i=1}^{\ell}\gamma_{i,k}A^{i-1}\boldsymbol{r}_k^{(\ell)})$ は $\sum_{i=1}^{\ell}\gamma_{i,k}A^{i-1}\boldsymbol{r}_k^{(\ell)}$ に A を陽に掛けることで得る．以上の (6) と (7) の導入によって，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが一致し，結果として近似解の精度の劣化を回避できると考えられる．

3.2 計算量の削減

漸化式 (6)，(7) を用いる場合，従来の方法に比べて行列ベクトル積が余分に必要となるため，計算量が増大する．一方，IDRstab 法は s や ℓ が大きくなるにつれて，行列ベクトル積に比べ，AXPY の計算回数が相対的に多くなる．そこで，全体の計算量を軽減するために，AXPY の計算回数を削減する工夫を行う．

まず，行列 $A^*\tilde{R}_0$ を事前に計算し，保持する．そして，文献 [1] の記述に従い，IDR step の σ_j, $\vec{\alpha}^{(j)}$, $\vec{\rho}_1^{(j)}$ および $\vec{\rho}_{q+1}^{(j)}$ $(q < s)$ を，それぞれ従来とは異なる式 $(A^*\tilde{R}_0)^*A^{j-1}U_k^{(j-1)}$, $\sigma_j^{-1}((A^*\tilde{R}_0)^*A^{j-2}\boldsymbol{r}_k^{(j-1)})$, $(A^*\tilde{R}_0)^*A^{j-1}\boldsymbol{r}_k^{(j)}$, $(A^*\tilde{R}_0)^*A^jU_k^{(j)}\boldsymbol{e}_q$ によって計算する．これらの計算方法の変更によって，IDR step の j 回目 $(j = 1, 2, \ldots, \ell)$ の繰り返しは，行列 $A^jU_k^{(j-1)}$ を用いずに実行できる．

IDR step の 1 回目の繰り返しでは，行列 $U_k^{(1)}$ の各列ベクトルを $\Pi_0^{(1)}$ による写像で構築し，$U_k^{(1)}$ に A を掛けることで，行列 $AU_k^{(1)}$ を得る．このとき，行列 $AU_k^{(0)} \equiv AU_k$ が無ければ，本来は (1) の更新

アルゴリズム 1. Quasi-Minimal Residual スムージング [18]

1. Set $\mathbf{s}_0 = \mathbf{r}_0$, $\mathbf{y}_0 = \mathbf{x}_0$, $\hat{\mathbf{u}}_0 = \hat{\mathbf{v}}_0 = \mathbf{0}$, $\tau_0 = \|\mathbf{r}_0\|_2$ and $m = 1$.
2. While $\|\mathbf{s}_m\|_2 > tol$
3. Compute $\mathbf{p}_m = \mathbf{x}_m - \mathbf{x}_{m-1}$ and $\mathbf{A}\mathbf{p}_m$
4. $\hat{\mathbf{v}}_m = \hat{\mathbf{v}}_{m-1} + \mathbf{p}_m$, $\hat{\mathbf{u}}_m = \hat{\mathbf{u}}_{m-1} + \mathbf{A}\mathbf{p}_m$
5. $\rho_m = \|\mathbf{s}_{m-1} - \hat{\mathbf{u}}_m\|_2$ and define τ_m by $1/\tau_m^2 = 1/\tau_{m-1}^2 + 1/\rho_m^2$
6. $\mathbf{y}_m = \mathbf{y}_{m-1} + (\tau_m^2/\rho_m^2)\hat{\mathbf{v}}_m$, $\mathbf{s}_m = \mathbf{s}_{m-1} - (\tau_m^2/\rho_m^2)\hat{\mathbf{u}}_m$
7. $\hat{\mathbf{v}}_m = (1 - (\tau_m^2/\rho_m^2))\hat{\mathbf{v}}_m$, $\hat{\mathbf{u}}_m = (1 - (\tau_m^2/\rho_m^2))\hat{\mathbf{u}}_m$
8. $m = m + 1$
9. End while

が行えず，残差 $\boldsymbol{r}_k^{(1)}$ は得られない．しかし，我々は (6) によって残差を更新するため，$AU_k^{(0)}$ は不要である．2 回目以降の繰り返しでは，$i = 1, 2, \ldots, j-2$ に対して，(1) の更新を行い，ベクトル $A^{j-2}\boldsymbol{r}_k^{(j)}$ に A を掛けることで，$A^{j-1}\boldsymbol{r}_k^{(j)}$ を得る．さらに，$i = 0, 1, \ldots, j-1$ に対して，(3) および (4) の更新を行う．そして，ℓ 回目の繰り返しの最後に，$A^{\ell-1}\boldsymbol{r}_k^{(\ell)}$ に A を掛け，$A^{\ell}\boldsymbol{r}_k^{(\ell)}$ を得る．各繰り返しにおける近似解と残差は，常に (2) と (6) によって更新する．以上の計算過程により，IDR step の j 回目の繰り返しでは，行列 $A^{j+1}U_k^{(j)}$ を構築する必要がなくなり，また polynomial step では，行列 $AU_k^{(\ell)}$ から $AU_{k+\ell}$ への更新が不要となる．この結果，従来の方法に比べ，1 サイクルあたり $\ell(s^2+3s)+\frac{1}{2}(\ell-1)-s$ 回の AXPY を削減できる．計算量の詳細は，4.2 節で述べる．

図 1-2 に，変形した IDR step の計算過程を示す．ただし，$\ell = 2$ とし，表記は文献 [12] に倣う．ここで，□で囲まれたベクトルは，A に関する行列ベクトル積を陽に行うことで得られる．特に，従来の IDRstab 法では，$\boldsymbol{r}_k^{(j)}$ は $\Pi_1^{(j)}$ による写像で更新されるのに対し，図 1-2 では，(6) のように行列ベクトル積を陽に行う．以後，本節で述べた我々が提案する IDRstab 法の変形アルゴリズムを，“変形 1” と呼ぶ．IDRstab 法の変形 1 は，従来の IDRstab 法と数学的に等価である．

4 QMR スムージングによる IDRstab 法の QMR 化

従来の BiCG 系統の解法と同様に，IDRstab 法の残差ノルムは振動する．一方，我々が提案した IDRstab 法の変形 1 は，計算量を大きく増加させることなくスムージングを適用できる．そこで本節では，QMR スムージングを IDRstab 法の変形 1 に適用することで，効率良く IDRstab 法の QMR 化を実現し，残差ノルムの振る舞いを滑らかにする．

4.1 QMR スムージングの適用

文献 [18] で提案された QMR スムージングを，アルゴリズム 1 に示す．いま，$\{\boldsymbol{r}_m\}$ と $\{\boldsymbol{x}_m\}$ を適当な反復法によって生成される残差列と対応する近似解列とする．このとき，近似解 $\boldsymbol{x}_m$ に対する探索方向ベクトル $\boldsymbol{p}_m$ とベクトル $A\boldsymbol{p}_m$ に対して，アルゴリズム 1 の 4-6 行の計算を行うことで，スムージングされた残差 $\boldsymbol{s}_m$ と対応する近似解 $\boldsymbol{y}_m$ が得られる．このようにして得られた残差ノルム $\|\boldsymbol{s}_m\|_2$ の振る舞いは，QMR アプローチから得られる残差ノルムのように滑らかである [18].

アルゴリズム 1 におけるベクトル $A\boldsymbol{p}_m$ は，数値的安定性の理由から，探索方向ベクトル $\boldsymbol{p}_m$ に A を陽に掛けることで得る．近似解 $\boldsymbol{x}_m$ と残差 $\boldsymbol{r}_m$ が (5) の形式で更新される場合は，行列ベクトル積を陽に行って得られたベクトル $A\boldsymbol{p}_m$ を再利用できるため，行列ベクトル積を増加させることなく QMR

表 1: 1サイクルあたりの行列ベクトル積数，ベクトル更新回数，内積数.

	MVs	AXPYs	DOTs
IDRstab	$\ell(s+1)$	$\frac{1}{2}\ell(\ell+1)(s+s^2)+\ell(s^2+3s)+2\ell$	$\ell(s+2s^2)+\frac{1}{2}\ell(\ell+3)$
変形 1	$\ell(s+1)+\ell+1$	$\frac{1}{2}\ell(\ell+1)(s+s^2)+s+\frac{3}{2}\ell+\frac{1}{2}$	$\ell(s+2s^2)+\frac{1}{2}\ell(\ell+3)$
変形 2	$\ell(s+1)+\ell+1$	$\frac{1}{2}\ell(\ell+1)(s+s^2)+s+5\ell+4$	$\ell(s+2s^2)+\frac{1}{2}\ell(\ell+5)+1$

スムージングを適用できる．ここで，従来の IDRstab 法は，残差を (5) とは異なる形式で更新する．そのため，アルゴリズム 1 を実装するには，ベクトル $A\boldsymbol{p}_m$ を得るために行列ベクトル積を余分に行う必要があり，計算効率が悪い．そこで本論文では，QMR スムージングを我々の提案した IDRstab 法の変形 1 に適用する．IDRstab 法の変形 1 では，IDR step と polynomial step において，近似解と残差を (5) の形式で更新する．従って，IDR step と polynomial step のそれぞれにアルゴリズム 1 の計算過程を組み込むことで，行列ベクトル積を増加させることなく QMR 化が実現できる．また，一般にスムージングによって得られる近似解の精度は，スムージング前の近似解の精度と同程度となることが知られている [6]．このため，QMR 化した IDRstab 法によって得られる近似解の精度は，変形 1 によって得られる近似解の精度が維持されると考えられる．

以後，IDRstab 法の変形 1 に QMR スムージングを適用したアルゴリズムを “変形 2” と呼び，アルゴリズム 2 に示す．ただし，アルゴリズムの表記は MATLAB コードに倣う．特に，$q \leq s$ に対して，行列 $W=[\boldsymbol{w}_1,\ldots,\boldsymbol{w}_s]$ の部分行列 $[\boldsymbol{w}_1,\ldots,\boldsymbol{w}_q]$ および列ベクトル $\boldsymbol{w}_q$ は，それぞれ $W_{(:,1:q)}$, $W_{(:,q)}$ と記述する．また，$[W_0;\ldots;W_j]\equiv[W_0^{\mathsf{T}},\ldots,W_j^{\mathsf{T}}]^{\mathsf{T}}$ である．さらに，アルゴリズム 2 において，$i=0,1,\ldots,j$ のとき，$\mathbf{U}_i$, $\mathbf{V}_i$, $\mathbf{r}_i$ および $\mathbf{u}_i$ は，それぞれ $\mathbf{U}$, $\mathbf{V}$, $\mathbf{r}$, $\mathbf{u}$ の成分を表し，$\mathbf{U}=[\mathbf{U}_0;\ldots;\mathbf{U}_j]$, $\mathbf{V}=[\mathbf{V}_0;\ldots;\mathbf{V}_j]$, $\mathbf{r}=[\mathbf{r}_0;\ldots;\mathbf{r}_j]$, $\mathbf{u}=[\mathbf{u}_0;\ldots;\mathbf{u}_j]$ である．

アルゴリズム 2 において，下線で記した 13 行，32 行は，3 節で導入した漸化式 (6)，(7) をそれぞれ表している．また，数字で記した 3 行，17–20 行，および 33–36 行が QMR スムージングの計算，すなわちアルゴリズム 1 の計算過程に対応する．本論文では，IDRstab 法の変形 1 のアルゴリズムを具体的に明記しないが，これはアルゴリズム 2 から QMR スムージングの計算を削除し，近似解の漸化式を導入することで得られる．なお，文献 [12] の記述に従い，アルゴリズム 2 の 6-7 行および 25-26 行において，アーノルディ過程による行列 U_0, $A^jU_k^{(j)}$ の各列ベクトルの正規直交化を行っている．

4.2 計算量の比較

本節では，従来の IDRstab 法と提案した変形 1，変形 2 の計算量を比較する．表 1 に，1 サイクルあたりの行列ベクトル積数 (MVs)，ベクトル更新回数 (AXPYs)，n 次元ベクトル同士の内積数 (DOTs) を示す．ただし，IDR step におけるアーノルディ過程の計算量は含めない．

表 1 より，次のことが言える．従来の IDRstab 法に比べ，変形 1 は 1 サイクルあたり $\ell+1$ 回の行列ベクトル積が増加する代わりに，AXPY は $\ell(s^2+3s)+\frac{1}{2}(\ell-1)-s$ 回少ない．従って，行列の疎性が高い場合，AXPY の削減によって，計算量の増加を十分に抑えることができる．また，変形 2 は変形 1 を利用しているため，行列ベクトル積の増加はなく，1 サイクルあたり $\frac{7}{2}(\ell+1)$ 回の AXPY と $\ell+1$ 回の DOT のみで QMR スムージングを適用している．

5 数値実験

本節では，従来の IDRstab 法と提案した変形 1，変形 2 の収束性を，数値実験によって比較する．特に，変形 1 によって，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが一致し，近似解の精度が改善されること，変形 2 によって，残差ノルムが滑らかに収束することをそれぞれ示す．

アルゴリズム 2. IDRstab 法の変形 2

1. Select an initial guess $\mathbf{x}$ and an $(n \times s)$ matrix $\tilde{\mathbf{R}}_0$.
2. Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}$, $\mathbf{r} = [\mathbf{r}_0]$
3. Set $\mathbf{s} = \mathbf{r}_0$, $\mathbf{y} = \mathbf{x}$, $\hat{\mathbf{u}} = \hat{\mathbf{v}} = \mathbf{0}$, $\tau = \|\mathbf{r}_0\|_2^2$
% *Generate an initial* $(n \times s)$ *matrix* $\mathbf{U} = [\mathbf{U}_0]$
4. For $q = 1, \ldots, s$
5. if $q = 1$, $\mathbf{u}_0 = \mathbf{r}_0$, else, $\mathbf{u}_0 = \mathbf{A}\mathbf{u}_0$
6. $\vec{\mu} = (\mathbf{U}_{0(:,1:q-1)})^* \mathbf{u}_0$, $\mathbf{u}_0 = \mathbf{u}_0 - \mathbf{U}_{0(:,1:q-1)}\vec{\mu}$
7. $\mathbf{u}_0 = \mathbf{u}_0 / \|\mathbf{u}_0\|_2$, $\mathbf{U}_{0(:,q)} = \mathbf{u}_0$
8. End for
9. While $\|\mathbf{s}\|_2 > tol$
10. For $j = 1, \ldots, \ell$
% *The IDR step*
11. $\sigma = (\mathbf{A}^*\tilde{\mathbf{R}}_0)^* \mathbf{U}_{j-1}$
12. if $j = 1$, $\vec{\alpha} = \sigma^{-1}(\tilde{\mathbf{R}}_0^* \mathbf{r}_0)$, else, $\vec{\alpha} = \sigma^{-1}((\mathbf{A}^*\tilde{\mathbf{R}}_0)^* \mathbf{r}_{j-2})$
13. $\mathbf{r}_0 = \mathbf{r}_0 - \mathbf{A}(\mathbf{U}_0\vec{\alpha})$
14. For $i = 1, \ldots, j-2$
15. $\mathbf{r}_i = \mathbf{r}_i - \mathbf{U}_{i+1}\vec{\alpha}$
16. End for
17. $\hat{\mathbf{v}} = \hat{\mathbf{v}} + \mathbf{U}_0\vec{\alpha}$, $\hat{\mathbf{u}} = \hat{\mathbf{u}} + \mathbf{A}(\mathbf{U}_0\vec{\alpha})$
18. $\rho = \|\mathbf{s} - \hat{\mathbf{u}}\|_2^2$, $\tau = 1/(1/\tau + 1/\rho)$, $\eta = \tau/\rho$
19. $\mathbf{y} = \mathbf{y} + \eta\hat{\mathbf{v}}$, $\mathbf{s} = \mathbf{s} - \eta\hat{\mathbf{u}}$
20. $\hat{\mathbf{v}} = (1-\eta)\hat{\mathbf{v}}$, $\hat{\mathbf{u}} = (1-\eta)\hat{\mathbf{u}}$
21. if $j > 1$, $\mathbf{r} = [\mathbf{r}; \mathbf{A}\mathbf{r}_{j-2}]$
22. For $q = 1, \ldots, s$
23. if $q = 1$, $\mathbf{u} = \mathbf{r}$, else, $\mathbf{u} = [\mathbf{u}_1; \ldots; \mathbf{u}_j]$
24. $\vec{\beta} = \sigma^{-1}((\mathbf{A}^*\tilde{\mathbf{R}}_0)^* \mathbf{u}_{j-1})$, $\mathbf{u} = \mathbf{u} - \mathbf{U}\vec{\beta}$, $\mathbf{u} = [\mathbf{u}; \mathbf{A}\mathbf{u}_{j-1}]$
25. $\vec{\mu} = (\mathbf{V}_{j(:,1:q-1)})^* \mathbf{u}_j$, $\mathbf{u} = \mathbf{u} - \mathbf{V}_{(:,1:q-1)}\vec{\mu}$
26. $\mathbf{u} = \mathbf{u} / \|\mathbf{u}_j\|_2$, $\mathbf{V}_{(:,q)} = \mathbf{u}$
27. End for
28. $\mathbf{U} = \mathbf{V}$
29. End for
30. $\mathbf{r} = [\mathbf{r}; \mathbf{A}\mathbf{r}_{\ell-1}]$
% *The polynomial step*
31. $\vec{\gamma} = [\gamma_1; \ldots; \gamma_\ell] = \arg\min_{\vec{\gamma}} \|\mathbf{r}_0 - [\mathbf{r}_1, \ldots, \mathbf{r}_\ell]\vec{\gamma}\|_2$
32. $\mathbf{r}_0 = \mathbf{r}_0 - \mathbf{A}([\mathbf{r}_0, \ldots, \mathbf{r}_{\ell-1}]\vec{\gamma})$
33. $\hat{\mathbf{v}} = \hat{\mathbf{v}} + [\mathbf{r}_0, \ldots, \mathbf{r}_{\ell-1}]\vec{\gamma}$, $\hat{\mathbf{u}} = \hat{\mathbf{u}} + \mathbf{A}([\mathbf{r}_0, \ldots, \mathbf{r}_{\ell-1}]\vec{\gamma})$
34. $\rho = \|\mathbf{s} - \hat{\mathbf{u}}\|_2^2$, $\tau = 1/(1/\tau + 1/\rho)$, $\eta = \tau/\rho$
35. $\mathbf{y} = \mathbf{y} + \eta\hat{\mathbf{v}}$, $\mathbf{s} = \mathbf{s} - \eta\hat{\mathbf{u}}$
36. $\hat{\mathbf{v}} = (1-\eta)\hat{\mathbf{v}}$, $\hat{\mathbf{u}} = (1-\eta)\hat{\mathbf{u}}$
37. $\mathbf{U} = [\mathbf{U}_0 - \sum_{j=1}^{\ell} \gamma_j \mathbf{U}_j]$
38. End while

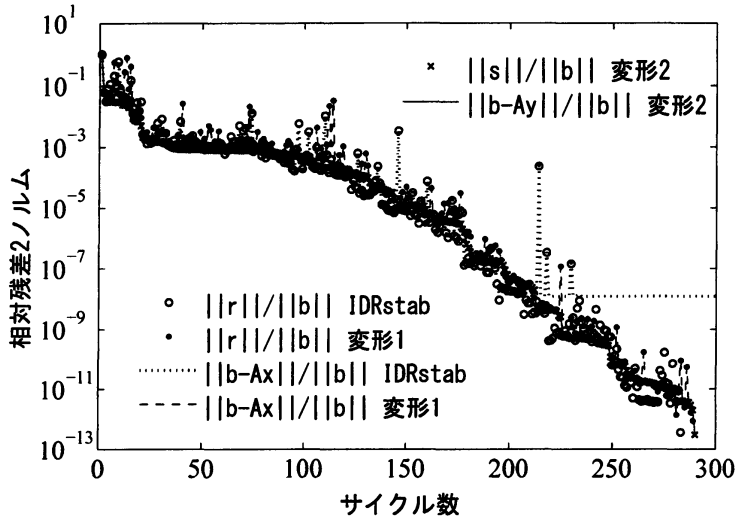

図 3: $(s, \ell) = (4, 2)$ に対する収束履歴.

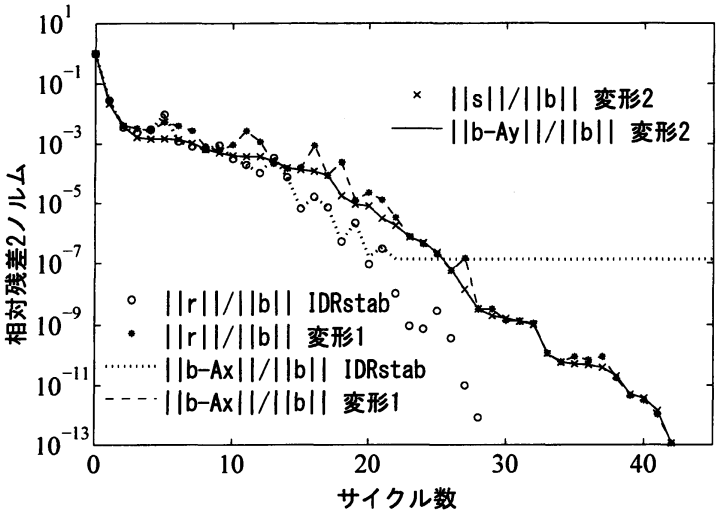

図 4: $(s, \ell) = (8, 8)$ に対する収束履歴.

数値実験は，HP の PC(Intel Core i7 2.67GHz CPU) で，Intel C++ 11.1.048 コンパイラの倍精度演算で行った．テスト行列は，文献 [12] と同様に，Matrix-Market に収納されている非対称疎行列より，SHERMAN5 を取り上げた．行列の次元数および非零要素数は，それぞれ 3312，20793 である．右辺項は，真の解 $\bar{\boldsymbol{x}} = (1, 1, \ldots, 1)^{\mathsf{T}}$ に対して，$\boldsymbol{b} = A\bar{\boldsymbol{x}}$ と設定した．初期値は $\mathbf{0}$，収束判定条件は $tol = 10^{-12}\|\boldsymbol{b}\|_2$ とした．行列 $\tilde{R}_0$ は，各成分を区間 $(0, 1)$ 上の乱数で生成し，各列ベクトルを正規直交化して与えた．(s, ℓ) の組み合わせは (4, 2), (4, 4), (8, 2), (4, 8), (8, 8) とした．

5.1 計算結果

図 3-4 に，(s, ℓ)=(4, 2), (8, 8) に対する各解法の相対残差 2 ノルム ($\|\boldsymbol{r}_k\|_2/\|\boldsymbol{b}\|_2$, $\|\boldsymbol{s}_k\|_2/\|\boldsymbol{b}\|_2$) および真の相対残差 2 ノルム ($\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2/\|\boldsymbol{b}\|_2$, $\|\boldsymbol{b} - A\boldsymbol{y}_k\|_2/\|\boldsymbol{b}\|_2$) の振る舞いを示す．グラフの横軸は

表 2: 各解法のサイクル数，行列ベクトル積数，計算時間，真の相対残差 2 ノルム．

(s, ℓ)	解法	Cycles	MVs	Time[sec]	True res.
(4, 2)	IDRstab	282	2825	2.57	1.21E−08
	変形 1	288	3748	2.37	8.72E−13
	変形 2	289	3761	2.47	2.97E−13
(4, 4)	IDRstab	116	2325	2.59	8.25E−10
	変形 1	118	2954	2.39	4.43E−13
	変形 2	118	2954	2.43	3.80E−13
(8, 2)	IDRstab	132	2385	3.40	1.50E−10
	変形 1	135	2843	2.96	8.90E−13
	変形 2	135	2843	3.01	4.66E−13
(4, 8)	IDRstab	56	2245	3.34	2.10E−07
	変形 1	72	3532	3.96	7.32E−13
	変形 2	72	3532	4.04	8.00E−13
(8, 8)	IDRstab	28	2025	4.74	1.32E−07
	変形 1	42	3410	6.55	1.15E−13
	変形 2	42	3410	6.57	1.15E−13

IDRstab 法におけるサイクル数，縦軸は相対残差 2 ノルムを表す．また表 2 に，各解法の収束までに要したサイクル数 (Cycles)，行列ベクトル積数 (MVs)，計算時間 (Time[sec])，および収束時の真の相対残差 2 ノルム (True res.) を示す．

図 3-4，および表 2 より，次のことが言える．まず従来の IDRstab 法では，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが異なり，近似解の精度が劣化している．これに対し変形 1 では，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが一致し，近似解の精度が改善されている．(s, ℓ)=(4, 2), (4, 4), (8, 2) の場合，変形 1 の収束までに要したサイクル数は，従来の IDRstab 法と同程度である．このとき，変形 1 の行列ベクトル積数は，従来の IDRstab 法に比べて増加しているが，計算時間は短い．これは，変形 1 で AXPY が十分に削減されたためと考えられる．一方，(s, ℓ)=(4, 8), (8, 8) の場合，変形 1 の収束までに要したサイクル数は，従来の IDRstab 法よりも多く，収束速度が低下している．このような収束速度の低下は，パラメータ s, ℓ が大きい場合に起きる傾向がある．

また，従来の IDRstab 法および変形 1 では，残差ノルムが振動している．これに対し変形 2 では，残差ノルムが滑らかに収束している．このとき，得られた近似解の精度は，変形 1 の場合と同程度であり，変形 1 に対する変形 2 の計算時間の増分は，平均して約 1.6%であった．従って，変形 2 は近似解の精度が改善されるという変形 1 の性質を引き継いだまま，計算量を大きく増加させることなく，残差ノルムの振る舞いが滑らかになったと言える．

6 まとめ

IDRstab 法に対して，近似解の精度を改善するため，漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが一致する変形アルゴリズム (変形 1) を提案した．この変形 1 は，従来の IDRstab 法に比べ，行列ベクトル積を余分に必要とするものの，AXPY を大幅に削減することで，計算量の増加を抑えることができる．さらに，変形 1 を利用することで，行列ベクトル積を増加させずに，効率良く IDRstab 法の QMR 化 (変形 2) を実現した．変形 2 の計算時間の増加はわずか 1.6%程度であり，得られる近似解の精度は，変形 1 によって得られる近似解の精度と同程度である．数値実験では，提案手

法によって近似解の精度が改善され，また残差ノルムの振る舞いが滑らかになることを示した．ただし，パラメータの s, ℓ が大きいとき，収束速度が低下する場合があった．この原因の究明と改善策については，今後の課題としたい．

謝辞 本研究について，貴重なご意見を頂いたUtrecht大学のGerard L. G. Sleijpen先生，Hamburg-Harburg工科大学のJens-Peter M. Zemke先生に深く御礼申し上げます．

参考文献

[1] K. Aihara, K. Abe and E. Ishiwata, An alternative implementation of the IDRstab method saving vector updates, *JSIAM Letters*, **3** (2011), 69–72.

[2] T. F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto and C .H. Tong, A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems, *SIAM J. Sci. Comput.*, **15** (1994), 338–347.

[3] L. Du, T. Sogabe and S.-L. Zhang, A variant of the IDR(s) method with the quasi-minimal residual strategy, *J. Comput. Appl. Math.*, **236** (2011), 621–630.

[4] R. Fletcher, Conjugate gradient methods for indefinite systems, *Lecture Notes in Mathematics*, **506** (1976), 73–89.

[5] R. W. Freund and N. M. Nachtigal, QMR: a quasi minimal residual method for non-Hermitian linear systems, *Numer. Math.*, **60** (1991), 315–339.

[6] M. H. Gutknecht and M. Rozložnik, Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers?, *BIT*, **41** (2001), 86–114.

[7] W. Schönauer, Scientific Computing on Vector Computers, Elsevier, Amsterdam, 1987.

[8] G. L. G. Sleijpen and D. R. Fokkema, BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, *Electron. Trans. Numer. Anal.*, **1** (1993), 11–32.

[9] G. L. G. Sleijpen, P. Sonneveld and M. B. van Gijzen, Bi-CGSTAB as an induced dimension reduction method, *Appl. Numer. Math.*, **60** (2010), 1100–1114.

[10] G. L. G. Sleijpen and H. A. van der Vorst, Reliable updated residuals in hybrid Bi-CG methods, *Comput.*, **2** (1996), 141–163.

[11] G. L. G. Sleijpen, H. A. van der Vorst and D. R. Fokkema, BiCGstab(ℓ) and other hybrid Bi-CG methods, *Numer. Algorithms*, **7** (1994), 75–109.

[12] G. L. G. Sleijpen and M. B. van Gijzen, Exploiting BiCGstab(ℓ) strategies to induce dimension reduction, *SIAM J. Sci. Comput.*, **32** (2010), 2687–2709.

[13] P. Sonneveld and M. B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **31** (2008), 1035–1062.

[14] M. Tanio and M. Sugihara, GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials, *J. Comput. Appl. Math.*, **235** (2010), 765–784.

[15] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, **13** (1992), 631–644.

[16] M . B. van Gijzen, G. L. G. Sleijpen and J.-P. M. Zemke, Flexible and multi-shift induced dimension reduction algorithms for solving large sparse linear system, Report 11-06, DIAM, TU Delft & Bericht 156, INS, TU Hamburg-Harburg (2011).

[17] R. Weiss, Convergence behavior of generalized conjugate gradient methods, Ph.D. thesis, Univ. of Karlsruhe, Germany, 1990.

[18] L. Zhou and H. F. Walker, Residual smoothing techniques for iterative methods, *SIAM J. Sci. Comput.*, **15** (1994), 297–312.