

Sonneve ld型SOR法vs. 古典的SOR法

日下部, 雄三
九州大学大学院システム情報科学府情報工学専攻 : 修士課程

藤野, 清次
九州大学情報基盤研究開発センター

春松, 正敏
本田技術研究所

<https://doi.org/10.15017/1517963>

出版情報 : 九州大学大学院システム情報科学紀要. 14 (2), pp.71-76, 2009-09-25. 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 :

Sonneveld型SOR法 vs. 古典的SOR法

日下部雄三*・藤野清次**・春松正敏***

Sonneveld Typed SOR Method vs. Classical SOR Method

Yuzo KUSAKABE, Seiji FUJINO and Masatoshi HARUMATSU

(Received June 12, 2009)

Abstract: The classical SOR (Successive Over-Relaxation) method is originated from the dissertation by D. Young in 1950. After that, the SOR method has been often used for the solution of problems which stem from various applications. The SOR method, however, has many issues on possibility of the solution. Since the SOR method greatly depends on spectrum of iteration matrix, applicability of the SOR method is not robust. In this paper, we extend IDR (Induced Dimension Reduction) Theorem proposed by Sonneveld and van Gijzen to designing of the residual of the SOR method, and accelerate its convergence rate and stability. Through numerical experiments, we make reveal significant effect of accelerated residual of the proposed Sonneveld typed SOR method.

Keywords: Iterative methods, IDR Theorem, Gauss-Seidel method, SOR method

1. はじめに

大規模な連立一次方程式の解法として反復法がある。古典的逐次過剰緩和法(以下, SOR(Successive Over-Relaxation)と略す)は1950年 David Youngの学位論文⁹⁾に端を発する反復解法である。発表後, 多くの分野でSOR法は利用されてきたが, その収束性は反復行列のスペクトラムの大きさに強く依存するため, BiCG法系統の反復法などに比べると, その利用範囲は限定されたものになっていた。そのため, 他の解法と組み合わせての使用に止まり, SOR法単独で使用されることは少ないのが現状であった。

2008年, P. SonneveldとM. van GijzenらによりIDR定理とそれに基づく反復解法IDR(s)法が発表され, その画期的な着想と収束性のすばらしさから, 非常に注目されている^{4) 6)}。そして, IDR定理を応用した新しいタイプのGauss-Seidel(以下, GSと略す)法に関する素朴な雛形のままの反復法IDR-based GS法が提案された^{1) 3)}。そこで, 我々はその方法の拡張および古典的なGS法との関連性を調べた。その結果, 一般形の行列に適用可能な形にIDR-based GS法を拡張することができた⁷⁾。そこで, 本研究では, GS法の加速版であるSOR法にIDR定理を適用し, その結果新しいIDR-based SOR法が導出できたので評価結果を報告する。

本稿の構成は次のとおりである。第2節で, IDR-based

GS法の概要とその算法を記述する⁷⁾。第3節で, IDR-based SOR法の導出および算法を記述する。また, 算法中の係数の選択について考察する。第4節で, IDR-based SOR法と古典的なSOR法やBiCG法系統の積型反復法との演算量を比較する。第5節で, 数値実験にて, IDR-based GS法および同SOR法の収束特性を明らかにする。最後に, 第6節でまとめる。

2. IDR-based Gauss-Seidel法

2.1 一般化反復法

解くべき線形方程式を

$$Ax = b \quad (1)$$

とする。ここで, 行列 A は大きさ $n \times n$ の実数行列, x と b は n 次元の解ベクトルと右辺ベクトルと各々とする。一般的な場合へ拡張を考え係数行列 A を

$$A = M - N \quad (2)$$

と分離する。ここで, 行列 M, N は正則とする。このとき, 解くべき方程式(1)は次のように表わされる。

$$Mx = Nx + b. \quad (3)$$

さらに,

$$\begin{aligned} x &= M^{-1}Nx + M^{-1}b \\ &= Bx + b' \end{aligned} \quad (4)$$

と変形する。ここで,

$$B = M^{-1}N, \quad b' = M^{-1}b \quad (5)$$

平成21年6月12日受付

* 情報工学専攻修士課程

** 九州大学情報基盤研究開発センター

*** 本田技術研究所

である。

ここで、次の漸化式を考える。ここで、 \mathbf{x}_0 は初期近似解ベクトルである。

$$\mathbf{x}_{k+1} = B\mathbf{x}_k + \mathbf{b}', \quad k = 0, 1, 2, \dots \quad (6)$$

スペクトル半径 $\rho(B) < 1.0$ の関係が成り立つときこの漸化式は収束する。また、残差ベクトル \mathbf{r}_k は次の式で定義する。

$$\mathbf{r}_k = B\mathbf{x}_k + \mathbf{b}' - \mathbf{x}_k. \quad (7)$$

上記の式(7)により近似解ベクトル \mathbf{x}_{k+1} と残差ベクトル \mathbf{r}_{k+1} は次の漸化式で表すことができる。ただし、 $\mathbf{r}_0 = M^{-1}(\mathbf{b} - A\mathbf{x}_0)$ である。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k, \quad (8)$$

$$\mathbf{r}_{k+1} = B\mathbf{r}_k. \quad (9)$$

このとき、上記の2つの漸化式から構成される算法は定常反復法の漸化式を反復アルゴリズムに置き換えたものである。その収束性は反復行列 B に依存する。また、 B が単位行列のときRichardsonの算法になる²⁾。

次に、IDR定理に基づき、残差 \mathbf{r}_{k+1} が式(9)を拡張した次の関係式を満たすとす。

$$\mathbf{r}_{k+1} = B(\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})). \quad (10)$$

係数 γ_k は、原論文³⁾に従い、IDR定理で導入された任意ベクトル \mathbf{p} との直交関係、

$$\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1}) \perp \mathbf{p} \quad (11)$$

すなわち、

$$(\mathbf{p}, \mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})) = 0 \quad (12)$$

から求められる。式(10)から次の漸化式が導ける。

$$\begin{aligned} (B - I)(\mathbf{x}_{k+1} - \mathbf{x}_k) &= \mathbf{r}_{k+1} - \mathbf{r}_k \\ &= B(\mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1})) - \mathbf{r}_k \\ &= (B - I)(\mathbf{r}_k + \gamma_k B(\mathbf{x}_k - \mathbf{x}_{k-1})). \end{aligned} \quad (13)$$

共通因子 $(B - I)$ を両辺から除去すると、

$$\begin{aligned} \mathbf{x}_{k+1} - \mathbf{x}_k &= \mathbf{r}_k + \gamma_k B(\mathbf{x}_k - \mathbf{x}_{k-1}) \\ &= \mathbf{r}_k + \gamma_k(\mathbf{r}_k - \mathbf{r}_{k-1}) + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) \end{aligned} \quad (14)$$

となる。よって、

$$d\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_{k-1}, \quad (15)$$

$$d\mathbf{r}_k = \mathbf{r}_k - \mathbf{r}_{k-1} \quad (16)$$

とおくと、IDR定理に基づいた場合の近似解ベクトル \mathbf{x}_{k+1} と残差ベクトル \mathbf{r}_{k+1} は次の漸化式で表せる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k + \gamma_k d\mathbf{r}_k + \gamma_k d\mathbf{x}_k, \quad (17)$$

$$\begin{aligned} \mathbf{r}_{k+1} &= B(\mathbf{r}_k + \gamma_k d\mathbf{r}_k) \\ &= M^{-1}N(\mathbf{r}_k + \gamma_k d\mathbf{r}_k). \end{aligned} \quad (18)$$

ただし、上記の漸化式では $\mathbf{r}_k = M^{-1}(\mathbf{b} - A\mathbf{x}_k)$ となり、収束した相対残差と真の相対残差に相違が生じる場合がある。そこで、 \mathbf{r}_k を \mathbf{r}'_k と置き換え、 $\mathbf{r}'_k = M^{-1}\mathbf{r}_k$ とすると次の近似解ベクトル \mathbf{x}_{k+1} と残差ベクトル \mathbf{r}_{k+1} の漸化式が得られる。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + M^{-1}(\mathbf{r}_k + \gamma_k d\mathbf{r}_k) + \gamma_k d\mathbf{x}_k, \quad (19)$$

$$\begin{aligned} \mathbf{r}_{k+1} &= B(\mathbf{r}_k + \gamma_k d\mathbf{r}_k) \\ &= NM^{-1}(\mathbf{r}_k + \gamma_k d\mathbf{r}_k). \end{aligned} \quad (20)$$

このとき、反復行列は従来の $B = M^{-1}N$ ではなく $B = NM^{-1}$ となるが、スペクトル半径 $\rho(B) < 1.0$ の関係は同様に成り立ち上記の漸化式も収束する⁸⁾。一般化反復法の算法を以下に示す。

一般化反復法の算法

Let \mathbf{x}_0 be an initial solution,

put $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,

Let \mathbf{p} be a random vector,

set $\gamma_0 = 0$,

for $k = 0, 1, 2, \dots$

$$\mathbf{s}_k = M^{-1}(\mathbf{r}_k + \gamma_k d\mathbf{r}_k),$$

$$d\mathbf{x}_{k+1} = \mathbf{s}_k + \gamma_k d\mathbf{x}_k,$$

$$d\mathbf{r}_{k+1} = N\mathbf{s}_k - \mathbf{r}_k,$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k + d\mathbf{r}_{k+1},$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + d\mathbf{x}_{k+1},$$

if $\frac{\|\mathbf{r}_{k+1}\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$ stop

$$\gamma_{k+1} = \frac{(\mathbf{p}, \mathbf{r}_{k+1})}{(\mathbf{p}, d\mathbf{r}_{k+1})},$$

end for.

2.2 IDR-based GS 法

係数行列 A を以下のように分離する。

$$A = L + D + U. \quad (21)$$

ここで、 L は狭義下三角行列、 D は対角行列、 U は狭義上三角行列とする。GS法では反復行列 $B = NM^{-1}$ は

$$M = L + D, \quad (22)$$

$$N = -U \quad (23)$$

と定義し、それによりIDR-based GS法(以下、IGS法と略す)が得られる。

3. IDR-based SOR法

3.1 IDR-based SOR法の導出

第2節と同様に，一般化反復法の算法をSOR法に適用することを考える．SOR法は，GS法の近似解ベクトル \mathbf{x}_{k+1} の更新の際に，近似解ベクトルの修正量を加速係数 ω で緩和することにより収束を速める反復法である．SOR法の反復行列 $B = NM^{-1}$ は加速係数 ω を含む

$$M = L + \frac{1}{\omega}D, \quad (24)$$

$$N = \left(\frac{1}{\omega} - 1\right)D - U \quad (25)$$

で定義される．以下に，IDR-based SOR法の算法を示す．

IDR-based SOR法の算法

Let \mathbf{x}_0 be an initial solution,
 put $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,
 Let \mathbf{p} be a random vector,
 set $\gamma_0 = 0$,
 for $k = 0, 1, 2, \dots$
 $\mathbf{s}_k = (L + \frac{1}{\omega}D)^{-1}(\mathbf{r}_k + \gamma_k \mathbf{d}\mathbf{r}_k)$,
 $d\mathbf{x}_{k+1} = \mathbf{s}_k + \gamma_k d\mathbf{x}_k$,
 $d\mathbf{r}_{k+1} = -((1 - \frac{1}{\omega})D + U)\mathbf{s}_k - \mathbf{r}_k$,
 $\mathbf{r}_{k+1} = \mathbf{r}_k + d\mathbf{r}_{k+1}$,
 $\mathbf{x}_{k+1} = \mathbf{x}_k + d\mathbf{x}_{k+1}$,
 if $\frac{\|\mathbf{r}_{k+1}\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$ stop
 $\gamma_{k+1} = -\frac{(\mathbf{p}, \mathbf{r}_{k+1})}{(\mathbf{p}, d\mathbf{r}_{k+1})}$,
 end for.

3.2 係数 γ_k の選択

残差ベクトル \mathbf{r}_{k+1} の更新の式は中間ベクトルを用いない場合，

$$\mathbf{r}_{k+1} = NM^{-1}(\mathbf{r}_k + \gamma_k d\mathbf{r}_k) \quad (26)$$

と書き表せる．このとき任意ベクトル \mathbf{p} と $\mathbf{r}_k + \gamma_k d\mathbf{r}_k$ との直交条件から，

$$\gamma_k = -\frac{(\mathbf{p}, \mathbf{r}_k)}{(\mathbf{p}, d\mathbf{r}_k)} \quad (27)$$

と係数 γ_k が決まる．これを**選択1**と呼ぶ．

一方，残差ベクトル \mathbf{r}_{k+1} の更新の式において，括弧の中の式のノルム： $\|\mathbf{r}_k + \gamma_k d\mathbf{r}_k\|_2$ の最小化から係数 γ_k を決める．すなわち，

$$\begin{aligned} & (\mathbf{r}_k + \gamma_k d\mathbf{r}_k, \mathbf{r}_k + \gamma_k d\mathbf{r}_k) \\ & = (\mathbf{r}_k, \mathbf{r}_k) + 2\gamma_k(\mathbf{r}_k, d\mathbf{r}_k) + \gamma_k^2(d\mathbf{r}_k, d\mathbf{r}_k) \end{aligned}$$

を係数 γ_k の2次式と考えて γ_k で偏微分する．そして，得られた式 $(\mathbf{r}_k, d\mathbf{r}_k) + \gamma_k(d\mathbf{r}_k, d\mathbf{r}_k)$ の値が0であることから，

$$\gamma_k = -\frac{(d\mathbf{r}_k, \mathbf{r}_k)}{(d\mathbf{r}_k, d\mathbf{r}_k)} \quad (28)$$

と係数 γ_k を決定する．これを**選択2**と呼ぶ．

前者の**選択1**の場合，任意ベクトル \mathbf{p} と \mathbf{r}_{k+1} を直交させるには，式(27)中の \mathbf{p} は $\mathbf{p} = (NM^{-1})^{-T}\mathbf{p}$ とする必要があるが， \mathbf{p} は元々任意であるためその必要性がない．一方，後者の**選択2**の場合， $\|\mathbf{r}_{k+1}\|_2$ を最小化するためには，式(28)中の $\mathbf{r}_k, d\mathbf{r}_k$ は， $\mathbf{r}_k = NM^{-1}\mathbf{r}_k, d\mathbf{r}_k = NM^{-1}d\mathbf{r}_k$ と各々おく必要がある．しかし，演算量が増加するため，あまり得策ではないと思われる．

4. 反復1回あたりの演算量

Table 1に，SOR法，ISOR法，積型反復法の反復1回あたりの演算量を示す．積型反復法は，実非対称行列を解くときよく使用されるBiCGStab法，GPBiCG法，BiCGSafe法を選んだ．表中の“NNZ”は非零要素数， $A\mathbf{v}$ は行列・ベクトル積計算， (\mathbf{u}, \mathbf{v}) はベクトルの内積計算， $\mathbf{u} \pm \mathbf{v}$ はベクトル同士の加減算， $\alpha\mathbf{v}$ は定数とベクトルの積を各々意味する．また，各計算の右側の括弧は必要な演算量の単位を意味する．ただし，SOR法では行列・ベクトル積およびベクトル同士の加減算(表中の“*”つき)は各1回，3回である．ただし，収束判定のためには残差 \mathbf{r}_k の更新が必要であるが，SOR法の算法中にそれが存在しない．そのため，実際には行列・ベクトル積が2回，ベクトル同士の加減算は4回に増えることを付記する．

Table 1 Computational cost per one iteration of each method.

method	$A\mathbf{v}$ ($\times 2NNZ$)	(\mathbf{u}, \mathbf{v}) ($\times 2n$)	$\mathbf{u} \pm \mathbf{v}$ ($\times n$)	$\alpha\mathbf{v}$ ($\times n$)
IGS	1	2	5	2
SOR	1*	0	3*	2
ISOR	1	2	5	2
BiCGStab	2	4	6	6
GPBiCG	2	7	16	13
BiCGSafe	2	7	14	13

5. 数値実験

5.1 テスト問題

Table 2にテスト行列(12個)の仕様を示す．テスト行列はフロリダ大学の疎行列データベースから選出した⁵⁾．各行列の解析分野は，構造解析，流体力学，回路解析，音響解析とし，これらの行列でISOR法の性能を評価する．

5.2 計算機環境と計算条件

計算機環境と計算条件は次の通りである．計算はすべて倍精度浮動小数点演算で行った．計算機は

Table 2 Specifications of test matrices.

matrix	dimension	NNZ	ave. NNZ	analytical field
epb1	14,734	95,053	6.45	structural
epb2	25,228	175,027	6.94	
epb3	84,617	463,625	5.48	
poisson3Da	13,514	352,762	26.10	
poisson3Db	85,623	2,374,949	27.74	
xenon1	48,600	1,181,120	24.30	
raefsky2	3,242	294,276	90.77	hydro- dynamic
raefsky3	21,200	1,488,768	70.22	
memplus	17,758	126,150	7.10	electrical
wang3	26,064	177,168	6.80	
wang4	26,068	177,196	6.80	
k3plates	11,107	378,927	34.12	

HP Workstation xw4200(CPU: Intel(R) Pentium(R) 4, クロック周波数: 3.8GHz, メモリ: 3Gbytes, OS: Suse Linux version 9.2, ホスト名: mizar)を用いた. コンパイラはIntel Fortran Compiler ver.8.1を用い, 最適化オプションは-O3を使用した. 時間の計測には, Fortranの時間計測関数etimeを用いた. 計算時間の単位は秒とする. 右辺項ベクトル**b**は, 厳密解が $\hat{x} = (1, \dots, 1)^T$ となるように, $\mathbf{b} = \mathbf{A}\hat{x}$ と定めた. 調べた反復法は, 古典的GS法, IGS法, 古典的SOR法, ISOR法, BiCGStab法, GPBiCG法, BiCGSafe法の計7種類である. IGS法, ISOR法の補助ベクトル**p**および BiCGStab法, GPBiCG法, BiCGSafe法の初期シャドウ残差は, 初期残差 $\mathbf{r}_0 (= \mathbf{b} - \mathbf{A}\mathbf{x}_0)$, 区間[0, 1]の一樣乱数, 定数1.0の3通りを与えた. SOR法, ISOR法の加速係数 ω は区間[1.0, 2.0]で0.1刻みで11通り与え, 最速のものを表に示した. 収束判定条件は, 相対残差の2ノルム: $\|\mathbf{r}_{k+1}\|_2 / \|\mathbf{r}_0\|_2 \leq 10^{-6}$ とした. 初期近似解 \mathbf{x}_0 はすべて0とした. 行列は予め対角スケールリングによって対角項をすべて1.0に正規化した. また, 最大反復回数は10000回とした.

Tables 3-4に各行列における各解法の収束性を示す. ただし, 表中の“TRR”は真の相対残差(True Relative Residual)の常用対数(\log_{10})の値を意味し, 近似解 \mathbf{x}_{k+1} に対する $\|\mathbf{b} - \mathbf{A}\mathbf{x}_{k+1}\|_2 / \|\mathbf{b} - \mathbf{A}\mathbf{x}_0\|_2$ の値である. また, 表中の“rand”, “const”は一樣乱数(random number), 定数1.0(constant)を各々意味する. さらに, 表中の“max”, “break”は最大反復回数まで未収束, そして計算のオーバーフローのため計算が途中で強制終了したことを各々意味する. また, 各行列ごとに収束までの計算時間が最短のものを太字で表示した. **Table 5**に**Tables 3-4**より選択1のISOR法が $\omega = 1.0$ で最速となった行列epb3, raefsky2, raefsky3, k3platesにおける選択1のISOR法の ω による反復回数の変動の様子を示す.

Fig. 1に行列poisson3Dbにおける4つの解法の相対残差履歴を示す. **Fig. 2**に行列epb2における4つの解法の

Table 3 Convergence of SOR, ISOR and other methods.

matrix	method	γ_k	p (r_0^*)	ω	itr.	time	TRR
epb1	GS	-	-	-	7614	8.49	-6.00
	IGS	1	r_0	-	1600	1.62	-6.40
			rand	-	2143	2.18	-6.56
			const	-	1576	1.60	-6.30
	SOR	-	-	1.6	2049	2.29	-6.00
	ISOR	1	r_0	1.5	518	0.49	-6.01
			rand	1.5	464	0.42	-6.05
			const	1.5	466	0.46	-6.04
	2	-	1.1	586	0.54	-6.00	
	BiCGStab	-	(rand)	-	259	0.31	-6.17
GPBiCG	-	(rand)	-	260	0.44	-6.60	
BiCGSafe	-	(rand)	-	256	0.39	-6.16	
epb2	GS	-	-	-	951	1.89	-6.00
	IGS	1	r_0	-	205	0.38	-6.64
			rand	-	208	0.40	-6.44
			const	-	214	0.41	-6.61
	SOR	-	-	1.3	529	1.06	-6.00
	ISOR	1	r_0	1.2	157	0.30	-6.06
			rand	1.2	152	0.28	-6.22
			const	1.2	158	0.29	-6.30
	2	-	1.0	136	0.24	-6.01	
	BiCGStab	-	(rand)	-	116	0.28	-6.18
GPBiCG	-	(const)	-	114	0.37	-6.21	
BiCGSafe	-	(const)	-	116	0.35	-6.06	
epb3	GS	-	-	-	max	60.34	-4.59
	IGS	1	r_0	-	1563	11.24	-6.28
			rand	-	1373	9.83	-6.15
			const	-	1384	10.01	-6.09
	SOR	-	-	1.0	max	60.34	-4.59
	ISOR	1	r_0	1.0	1537	10.19	-6.29
			rand	1.0	1379	9.14	-6.00
			const	1.0	1406	9.24	-6.16
	2	-	1.0	max	63.52	-4.75	
	BiCGStab	-	(const)	-	1122	9.37	-6.02
GPBiCG	-	(all)	-	-	-	-	
BiCGSafe	-	(r_0)	-	1031	10.65	-6.01	
poisson-3Da	GS	-	-	-	2148	7.15	-6.00
	IGS	1	r_0	-	191	0.43	-6.44
			rand	-	174	0.39	-6.37
			const	-	237	0.53	-6.32
	SOR	-	-	1.9	134	0.47	-6.04
	ISOR	1	r_0	1.8	74	0.16	-6.01
			rand	1.8	76	0.18	-6.08
			const	1.8	74	0.17	-6.01
	2	-	1.9	124	0.28	-6.02	
	BiCGStab	-	(rand)	-	74	0.27	-6.02
GPBiCG	-	(rand)	-	70	0.28	-6.04	
BiCGSafe	-	(const)	-	70	0.28	-6.05	
poisson-3Db	GS	-	-	-	break	-	-
	IGS	1	r_0	-	819	15.49	-6.97
			rand	-	674	12.73	-6.66
			const	-	744	14.06	-6.98
	SOR	-	-	all	break	-	-
	ISOR	1	r_0	1.8	239	4.50	-6.03
			rand	1.8	249	4.69	-6.18
			const	1.8	253	4.78	-6.11
	2	-	1.1	514	9.53	-6.00	
	BiCGStab	-	(rand)	-	193	9.25	-6.02
GPBiCG	-	(all)	-	-	-	-	
BiCGSafe	-	(const)	-	175	7.19	-6.00	
xenon1	GS	-	-	-	break	-	-
	IGS	1	r_0	-	max	63.90	-5.50
			rand	-	5343	34.21	-4.10
			const	-	max	64.36	-5.27
	SOR	-	-	all	break	-	-
	ISOR	1	r_0	1.8	1246	8.24	-6.08
			rand	1.9	1377	9.11	-6.21
			const	1.6	1157	7.70	-4.44
	2	-	1.2	1020	6.60	-6.00	
	BiCGStab	-	(const)	-	436	4.76	-6.06
GPBiCG	-	(const)	-	418	5.36	-6.00	
BiCGSafe	-	(r_0)	-	361	4.40	-6.01	

Table 4 Convergence of SOR, ISOR and other methods.

matrix	method	γ_k	p (r_0^*)	ω	itr.	time	TRR
raefsky2	GS	-	-	-	max	19.74	111.36
	IGS	1	r_0	-	308	0.35	-5.78
			rand	-	321	0.36	-5.70
			const	-	329	0.38	-5.91
	SOR	-	-	1.0	max	19.74	111.36
	ISOR	1	r_0	1.0	339	0.40	-6.16
			rand	1.0	353	0.41	-6.02
			const	1.0	349	0.42	-6.06
	2	-	1.2	4101	4.73	-6.00	
	BiCGStab	-	(const)	-	304	0.65	-6.02
GPBiCG	-	(rand)	-	289	0.64	-6.15	
BiCGSafe	-	(r_0)	-	291	0.63	-6.17	
raefsky3	GS	-	-	-	break	-	-
	IGS	1	r_0	-	1832	9.81	-6.08
			rand	-	1979	10.52	-6.09
			const	-	1804	9.59	-6.12
	SOR	-	-	all	break	-	-
	ISOR	1	r_0	1.0	1915	10.77	-6.13
			rand	1.0	1907	10.71	-6.05
			const	1.0	1899	10.65	-6.01
	2	-	1.0	max	55.69	65.73	
	BiCGStab	-	(rand)	-	1435	15.00	-6.00
GPBiCG	-	(rand)	-	1395	15.92	-6.00	
BiCGSafe	-	(rand)	-	1407	15.61	-6.01	
memplus	GS	-	-	-	max	14.97	-5.11
	IGS	1	r_0	-	4954	5.95	-6.21
			rand	-	7164	8.67	-5.37
			const	-	6092	7.33	-4.99
	SOR	-	-	1.9	1782	2.69	-6.00
	ISOR	1	r_0	1.9	444	0.53	-6.05
			rand	1.9	472	0.56	-6.15
			const	1.9	431	0.53	-6.04
	2	-	1.2	1203	1.37	-6.00	
	BiCGStab	-	(r_0)	-	152	0.24	-6.08
GPBiCG	-	(r_0)	-	96	0.21	-6.01	
BiCGSafe	-	(r_0)	-	110	0.22	-6.01	
wang3	GS	-	-	-	4499	8.84	-6.00
	IGS	1	r_0	-	375	0.72	-6.31
			rand	-	551	1.06	-6.42
			const	-	531	1.01	-6.26
	SOR	-	-	1.9	221	0.45	-6.02
	ISOR	1	r_0	1.8	106	0.19	-6.36
			rand	1.8	114	0.21	-6.22
			const	1.8	111	0.22	-6.34
	2	-	1.9	222	0.39	-6.02	
	BiCGStab	-	(r_0)	-	96	0.23	-6.29
GPBiCG	-	(r_0)	-	97	0.35	-6.10	
BiCGSafe	-	(r_0)	-	98	0.30	-6.15	
wang4	GS	-	-	-	2473	4.87	-6.00
	IGS	1	r_0	-	448	0.85	-6.67
			rand	-	530	1.01	-6.30
			const	-	541	1.04	-6.43
	SOR	-	-	1.9	246	0.50	-6.01
	ISOR	1	r_0	1.8	134	0.26	-6.02
			rand	1.8	134	0.26	-6.00
			const	1.8	135	0.26	-6.29
	2	-	1.9	179	0.34	-6.04	
	BiCGStab	-	(r_0)	-	98	0.23	-6.29
GPBiCG	-	(r_0)	-	101	0.37	-6.24	
BiCGSafe	-	(rand)	-	96	0.30	-6.21	
k3plates	GS	-	-	-	max	28.18	-5.94
	IGS	1	r_0	-	max	16.43	-5.27
			rand	-	max	16.55	-5.82
			const	-	max	16.42	-5.45
	SOR	-	-	1.0	max	28.18	-5.94
	ISOR	1	r_0	1.0	6020	9.95	-5.51
			rand	1.3	5219	8.68	-5.13
			const	1.2	8387	14.18	-5.23
	2	-	1.0	786	1.33	-6.00	
	BiCGStab	-	(rand)	-	176	0.52	-6.00
GPBiCG	-	(const)	-	162	0.54	-6.00	
BiCGSafe	-	(const)	-	156	0.50	-6.00	

Table 5 Variation of iterations of ISOR method with $\gamma_k : 1$ when ω varies as $\omega = 0.5, 0.6, \dots, 1.0$ for matrices epb3, raefsky2, raefsky3 and k3plates.

matrix	ω	itr.	time	TRR
epb3 ($p=\text{rand}$)	0.5	8308	53.79	-4.75
	0.6	3395	21.83	-6.02
	0.7	2485	16.08	-6.07
	0.8	1977	12.80	-6.03
	0.9	1730	11.22	-6.00
	1.0	1379	9.14	-6.00
raefsky2 ($p=r_0$)	0.5	945	1.08	-6.11
	0.6	726	0.83	-6.00
	0.7	652	0.76	-6.68
	0.8	477	0.55	-6.21
	0.9	389	0.45	-6.15
	1.0	339	0.40	-6.16
raefsky3 ($p=r_0$)	0.5	2941	16.77	-6.03
	0.6	2863	16.18	-5.80
	0.7	2200	12.44	-6.00
	0.8	1642	9.40	-6.10
	0.9	1543	8.78	-6.04
	1.0	1915	10.87	-6.13
k3plates ($p=\text{const}$)	0.5	7406	12.53	-6.13
	0.6	2522	4.24	-5.60
	0.7	3881	6.51	-5.14
	0.8	8776	14.73	-5.96
	0.9	3313	5.58	-5.56
	1.0	max	16.73	-4.91

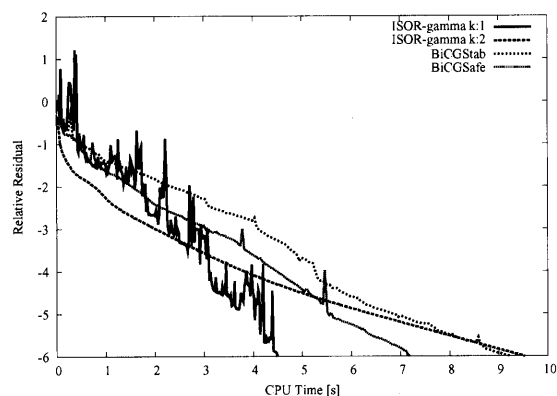


Fig. 1 History of relative residual 2-norm of four methods for matrix poisson3Db.

相対残差履歴を示す。Fig. 3に Tables 3-4より選択1の ISOR法が有効であった行列 poisson3Da, poisson3Db, wang3における選択1の ISOR法の ω による反復回数の変動の様子を示す。また, Fig. 4に Tables 3-4より選択1の ISOR法が $\omega = 1.0$ で最速となった行列 epb3, raefsky2, raefsky3, k3platesにおける選択1の ISOR法の ω による反復回数の変動の様子を示す。Figs. 1-4では各解法およびベクトル p の選択のうち, Tables 3-4で最も優れた収束性を示したものをを用いた。また, 加速係数 ω の範囲は区間 $[0.5, 2.0]$ で0.1刻みで計16通りを与えた。

Tables 3-5および Figs.1-4の観察から, 以下の知見を得ることができる。

- 古典的SOR法の収束性に比べて, ISOR法の収束性は優れている。

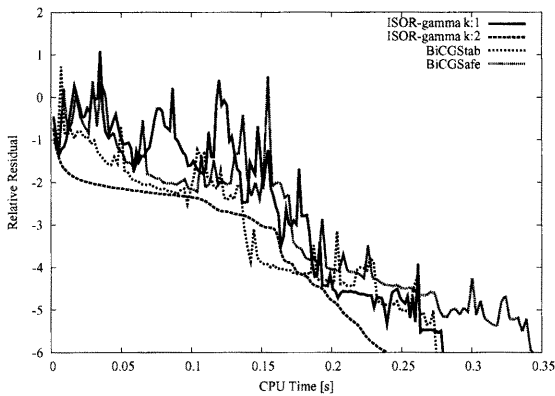


Fig. 2 History of relative residual 2-norm of four methods for matrix epb2.

る反復回数の変動の幅が大きい。

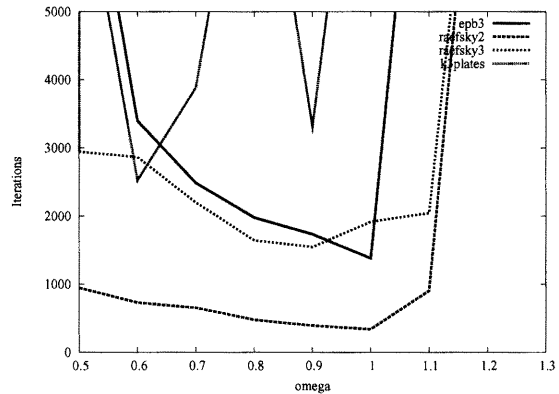


Fig. 4 Variation of iterations of ISOR method with $\gamma_k : 1$ when ω varies as $\omega = 0.5, 0.6, \dots, 1.3$ for matrices epb3, raefsky2, raefsky3 and k3plates.

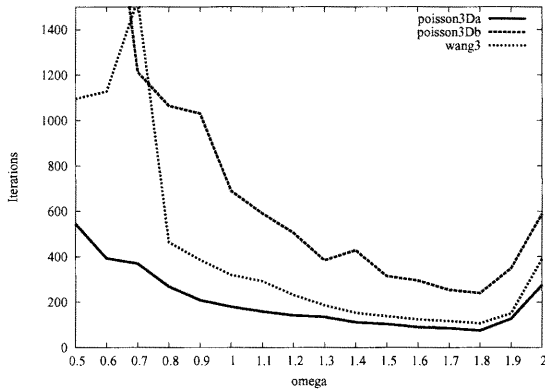


Fig. 3 Variation of iterations of ISOR method with $\gamma_k : 1$ when ω varies as $\omega = 0.5, 0.6, \dots, 2.0$ for matrices poisson3Da, poisson3Db and wang3.

- さらに、ISOR法は行列raefsky2, raefsky3を除く12個の行列においてIGS法よりも高速に収束した。
- IGS法とISOR法は、古典的GS法と同SOR法が収束しない行列に対しても収束した。
- ISOR法における係数 γ_k の選択では、行列epb2, xenon1, k3platesを除く11個の行列で選択1の方が良かった。一方、IGS法では選択1と選択2の性能は同程度であった⁷⁾。
- 最適 ω つきISOR法は、従来の積型反復法と遜色のない性能が得られた。
- Fig. 1に示した残差履歴から、選択2の係数 γ_k のISOR法の残差はIGS法のととき同様の単調減少性が観察された。ただし、数学的に残差ノルムを最小化した訳ではないことを付記する。
- ISOR法における加速係数 ω は、従来のように[1.0, 2.0]の範囲で有効な場合(Fig. 3参照)と、[0.5, 1.0]の範囲で有効な場合(Fig. 4参照)とがある。また、音響解析分野の行列k3platesにおいては加速係数によ

6. ま と め

本研究によって、ISOR法とIGS法は、古典的SOR法と同IGS法に比べて、圧倒的に優位な収束性を示すことがわかった。さらに、係数 γ_k は選択1の方が収束性がより良いことがわかった。また、最適加速係数 ω つきISOR法は、従来の積型反復法と遜色のない収束性を示すこともわかった。

参 考 文 献

- 1) 尾上勇介, ペーター ソネフェルト, 藤野清次, “IDR-AGS法の収束性と有効性の評価”, 情報処理学会 HPC研究会報告, No.125, pp.13-18, 2008.
- 2) Saad, Y., Iterative methods for sparse linear systems 2nd edition, SIAM, Philadelphia, 2003.
- 3) Sonneveld, P., 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.
- 4) Sonneveld, P., van Gijzen, M.B., IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Comput., Vol. 31, No.2, pp.1035-1062, 2008.
- 5) University of Florida Sparse Matrix Collection: <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- 6) 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.
- 7) 春松正敏, 日下部雄三, 藤野清次, 福重貴浩, 有馬敏幸, ペーター ソネフェルト, “IDR定理に基づくGS法とSOR法の提案と収束性評価”, 情報処理学会 HPC研究会報告, 調布, 2009.6.
- 8) 森 正武, “数値解析”, 共立出版, 東京, 2002.
- 9) Young, D., Iterative methods for solving partial difference equations of elliptic type, Doctoral Thesis, Harvard University, 1950.