

逐次的最小残差法の提案

中嶋, 徳正
九州大学大学院システム情報科学府情報学部門

藤野, 清次
九州大学情報基盤研究開発センター

<https://doi.org/10.15017/20058>

出版情報：九州大学大学院システム情報科学紀要. 16 (2), pp.83-88, 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：

逐次的最小残差法の提案

中嶋徳正*・藤野清次**

A Proposal of Successive Minimal Residual Method

Norimasa NAKASHIMA and Seiji FUJINO

(Received August 8, 2011)

Abstract: Convergence property of the successive over-relaxation (SOR) method depends on a relaxation parameter whose optimal value can be hardly computed. This paper determines the optimal value under a minimal condition for an intermediate residual in the successive computation of the SOR method. Through the organizations of formulas, the authors eliminate the relaxation parameter and propose a new iterative solver: successive minimal residual (SMR) method. Numerical tests for over 700 practical and real nonsymmetric matrix problems comparably check the performances of convergence, computation time and accuracy for the SMR method and other classical iterative methods.

Keywords: Iterative methods, Successive over-relaxation method, Relaxation parameter, Intermediate residual, Successive minimal residual method

1. はじめに

係数行列 $A \in \mathbb{R}^{N \times N}$, 解ベクトルおよび右辺ベクトル $x, b \in \mathbb{R}^N$ により構成される N 元連立1次方程式: $Ax = b$ の数値解法は直接法と反復法に大別され, 後者はさらに古典的反復法と Krylov 部分空間法に分類される. 古典的反復法の例としては Jacobi 法¹⁾ と Gauss-Seidel (GS) 法¹⁾, さらに両解法に緩和係数を取り入れて収束性の改善を図った relaxed Jacobi (RJ) 法²⁾ と逐次的過剰緩和 (Successive over-relaxation: SOR) 法¹⁾ が知られている. さらに, SOR法の算法に対称性を持たせた対称逐次的過剰緩和 (Symmetric SOR: SSOR) 法¹⁾ も古典的反復法の一つである.

RJ法, SOR法およびSSOR法では緩和係数は静的に与えることが多く, その場合の収束性は緩和係数に依存することが知られている. しかし, 最も速く収束させる最適な緩和係数の与え方には課題がある. 書籍等で周知のように¹⁾³⁾⁴⁾, SOR法に関しては緩和係数は0から2の間でなければ収束性は保証されず (Kahanの定理), property Aと呼ばれる特殊な行列に対してのみ最適な緩和係数に対する計算式を得る (Young-Vargaの定理). しかし, 最適な緩和係数は限られた行列にのみ計算可能であること, そして, その計算式には行列のスペクトル半径が現れ, この計算に膨大な計算量を必要とすることから実用性には乏しい. 以上のことから, 静的な緩和係数の最適値を決めることは非常に困難であり, 現状では

ユーザの経験と勘に頼るしかない.

著者らは最適な緩和係数を反復毎に動的に決定する手法に注目している. この場合, 前述の3つの解法はパラメータフリーとなるため, ユーザは緩和係数設定に関する煩わしさから解放される. 著者らはRJ法の算法において残差ベクトルを導入し, 最小残差条件に基づいて最適な緩和係数を動的に決定する手法を提案した⁵⁾. ただし, このアイデアをそのままSOR法へ適用することはできるが, SOR特有の逐次性のため実装は不可能であることが分かった. そこで, 本論文ではこの逐次性をうまく利用するため, 中間残差ベクトルを導入し, これに対する最小条件に基づいて最適な緩和係数を決定する. さらに, 数式を整理すると最終的には緩和係数が陽に現れない新解法: 逐次的最小残差 (Successive minimal residual: SMR) 法を得る.

本論文の構成は以下のとおりである. 第2節ではRJ法, SOR法およびSSOR法を対象に最小残差条件に基づいて最適な緩和係数を動的に決定する方法を示す. そして, SOR法およびSSOR法において実装上の問題点を指摘する. 第3節では, この問題点を解消するためSOR法の逐次計算において中間残差ベクトルを導入し, これに対する最小条件に基づいて最適な緩和係数を動的に決定する方法を述べる. さらに, SMR法の導出過程を示す. 第4節では700種類以上の実用的な実数非対称問題を対象にSMR法とRJ法および従来の古典的反復法 (Jacobi法, GS法) の収束性, 計算時間, 精度を比較する. 最後に, 第5節でまとめを行なう.

平成23年8月8日受付

* 情報知能工学部門

** 情報基盤研究開発センター

2. 最小残差条件に基づく緩和係数の動的決定法

RJ法, SOR法およびSSOR法の反復公式は次式で与えられる¹⁾.

$$\mathbf{x}^{(n)} = \mathcal{M}^{-1}(\omega \mathbf{b} - \mathcal{N} \mathbf{x}^{(n-1)}) \quad (1)$$

$$\mathcal{M} = \begin{cases} \mathcal{D} & (\text{RJ}) \\ \mathcal{D} + \omega \mathcal{L} & (\text{SOR, SSOR}(n : \text{odd})) \\ \mathcal{D} + \omega \mathcal{U} & (\text{SSOR}(n : \text{even})) \end{cases} \quad (2)$$

$$\mathcal{N} = \begin{cases} \omega \mathcal{A} - \mathcal{D} & (\text{RJ}) \\ \omega \mathcal{U} - (1 - \omega) \mathcal{D} & (\text{SOR, SSOR}(n : \text{odd})) \\ \omega \mathcal{L} - (1 - \omega) \mathcal{D} & (\text{SSOR}(n : \text{even})) \end{cases} \quad (3)$$

ここで, \mathcal{D} , \mathcal{L} および \mathcal{U} は係数行列 \mathcal{A} の対角成分, 狭義下三角成分および狭義上三角成分である. 本節では最小残差条件に基づいて緩和係数 ω を動的に決定する.

(1)および(3)より解ベクトル $\mathbf{x}^{(n)}$ と残差ベクトル $\mathbf{r}^{(n)} = \mathbf{b} - \mathcal{A} \mathbf{x}^{(n)}$ に対する漸化式は

$$\mathbf{x}^{(n)} = \mathbf{x}^{(n-1)} + \omega \mathcal{M}^{-1} \mathbf{r}^{(n-1)} \quad (4)$$

$$\mathbf{r}^{(n)} = \mathbf{r}^{(n-1)} - \omega \mathcal{A} \mathcal{M}^{-1} \mathbf{r}^{(n-1)} \quad (5)$$

で与えられる. (5)より残差 L_2 ノルム $\|\mathbf{r}^{(n)}\|_2$ を最小化させる ω は次式で与えられる.

$$\omega = \omega^{(n)} = \frac{(\mathcal{A} \mathcal{M}^{-1} \mathbf{r}^{(n-1)}, \mathbf{r}^{(n-1)})}{\|\mathcal{A} \mathcal{M}^{-1} \mathbf{r}^{(n-1)}\|_2^2} \quad (6)$$

ここで, (6)の分子にある括弧は内積を表し, その定義は本論文においては次式とする.

$$(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^N a_i b_i \quad (7)$$

(4), (5)および(6)より最小残差条件に基づいて緩和係数を動的に決定するRJ法, SOR法およびSSOR法が完成する. しかし, SOR法およびSSOR法では(6)中の \mathcal{M} に $\omega^{(n)}$ が存在する. このため, $\omega^{(n)}$ が不確定のまま $\mathcal{M}^{-1} \mathbf{r}^{(n-1)}$ の計算を進めるという不具合が生じることから, 上記の算法をそのまま利用することができない.

3. 中間残差最小条件に基づく緩和係数の動的決定法

本節ではSOR法の緩和係数の動的決定法について再度検討する. SOR法に対する反復公式を成分で表記すると

$$x_i^{(n)} = a_{ii}^{-1} \left(\omega b_i - \omega \sum_{j=1}^{i-1} a_{ij} x_j^{(n)} - (\omega - 1) a_{ii} x_i^{(n-1)} - \omega \sum_{j=i+1}^N a_{ij} x_j^{(n-1)} \right) \quad (8)$$

となる. (8)を残差ベクトルの成分 $r_i^{(n)}$ を用いた式に変形すると

$$x_i^{(n)} = \omega a_{ii}^{-1} \left\{ r_i^{(n-1)} - \sum_{j=1}^{i-1} a_{ij} (x_j^{(n)} - x_j^{(n-1)}) \right\} + x_i^{(n-1)} \quad (9)$$

ここで, 解ベクトルの1番目から i 番目の成分までを更新したときの残差ベクトルを中間残差ベクトルと定義し, 次式のように $e_i^{(n)}$ で表す. なお, T は転置を意味する.

$$e_i^{(n)} = \mathbf{b} - \mathcal{A} \begin{bmatrix} x_1^{(n)} & \cdots & x_{i-1}^{(n)} & x_i^{(n)} & x_{i+1}^{(n-1)} & \cdots & x_N^{(n-1)} \end{bmatrix}^T \quad (10)$$

$e_i^{(n)}$ の添字 i は0から N までの整数値を取る. ここで, $i=0$ とは n 反復目の解ベクトルの成分は1つも更新されていないことを意味するので, $e_0^{(n)} = \mathbf{r}^{(n-1)}$ となる. 同様に $i=N$ では解ベクトルのすべての成分が更新されたため, $e_N^{(n)} = \mathbf{r}^{(n)}$ となる. (9)および(10)より $e_i^{(n)}$ の i に対する漸化式は

$$e_i^{(n)} = e_{i-1}^{(n)} - \omega a_{ii}^{-1} \left\{ r_i^{(n-1)} - \sum_{j=1}^{i-1} a_{ij} (x_j^{(n)} - x_j^{(n-1)}) \right\} \mathbf{a}_i \quad (11)$$

で与えられる. ここで, \mathbf{a}_i は係数行列 \mathcal{A} の i 列目の成分からなるベクトルである. したがって, 係数行列は \mathbf{a}_1 から \mathbf{a}_N を用いて次式で表現される.

$$\mathcal{A} = (\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_N), \quad \mathbf{a}_i = (a_{1i} \ a_{2i} \ \cdots \ a_{Ni})^T \quad (12)$$

簡便のため以下の作業用変数:

$$y_i^{(n)} = a_{ii}^{-1} \left\{ r_i^{(n-1)} - \sum_{j=1}^{i-1} a_{ij} (x_j^{(n)} - x_j^{(n-1)}) \right\} \quad (13)$$

を導入する. (11)より中間残差 L_2 ノルムを最小にする緩和係数 ω は

$$\omega_i^{(n)} = \frac{(y_i^{(n)} \mathbf{a}_i, e_{i-1}^{(n)})}{\|y_i^{(n)} \mathbf{a}_i\|_2^2} = \frac{(\mathbf{a}_i, e_{i-1}^{(n)})}{y_i^{(n)} \|\mathbf{a}_i\|_2^2} \quad (14)$$

となる. (9), (11)および(13)はそれぞれ次式のように変形される.

$$x_i^{(n)} = x_i^{(n-1)} + \omega_i^{(n)} y_i^{(n)} \quad (15)$$

$$e_i^{(n)} = e_{i-1}^{(n)} - \omega_i^{(n)} y_i^{(n)} \mathbf{a}_i \quad (16)$$

$$y_i^{(n)} = a_{ii}^{-1} \left\{ r_i^{(n-1)} - \sum_{j=1}^{i-1} a_{ij} \omega_j^{(n)} y_j \right\} \quad (17)$$

(15)から(17)を見ると, $\omega_i^{(n)} y_i^{(n)}$ はワンセットになってい

ることが分かる．そして，(14)より $\omega_i^{(n)}$ の算定において母に $y_i^{(n)}$ がある．残差ベクトルが収束または停滞傾向にある場合は(15)より $y_i^{(n)} \rightarrow 0$ となり， $\omega_i^{(n)}$ の算定に影響することが考えられる．そこで， $\omega_i^{(n)} y_i^{(n)}$ を一つの変数 $\Delta x_i^{(n)}$ と見なして算法を書き換える．結果，式(14)から(17)は以下の式に変形される．

$$\Delta x_i^{(n)} = \frac{\left(\mathbf{a}_i, \mathbf{e}_{i-1}^{(n)} \right)}{\|\mathbf{a}_i\|_2^2} \quad (18)$$

$$x_i^{(n)} = x_i^{(n-1)} + \Delta x_i^{(n)} \quad (19)$$

$$\mathbf{e}_i^{(n)} = \mathbf{e}_{i-1}^{(n)} - \Delta x_i^{(n)} \mathbf{a}_i \quad (20)$$

$$y_i^{(n)} = a_{ii}^{-1} \left\{ r_i^{(n-1)} - \sum_{j=1}^{i-1} a_{ij} \Delta x_j^{(n)} \right\} \quad (21)$$

これにより実際の計算において0またはそれに非常に近い数値による割り算が回避できる．また，(21)の計算は事実上不要となるため演算量が削減できる．

以上の過程において緩和係数は(14)より一旦対角行列に拡張されており，GSOR法⁶⁾に類似する．しかし，最終的な算法は式(18)から(20)で構成され，ここには緩和係数やGS法の算法より得た一時的な解ベクトルは見当たらない．むしろ，解ベクトルの各成分を中間残差最小条件の下で修正するだけの単純な反復法であるといえる．この特徴から本論文では新規解法を逐次的最小残差 (Successively minimal residual : SMR) 法と命名する．SMR法の算法を以下に示す．

1. Let $\mathbf{x}^{(0)}$ be an initial vector and $\mathbf{e}_0^{(0)} = \mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$
2. For $n = 1, 2, \dots$ Do
3. For $i = 1, 2, \dots, N$ Do
4. $\Delta x_i^{(n)} = \frac{\left(\mathbf{a}_i, \mathbf{e}_{i-1}^{(n)} \right)}{\|\mathbf{a}_i\|_2^2}$
5. $x_i^{(n)} = x_i^{(n-1)} + \Delta x_i^{(n)}$
6. $\mathbf{e}_i^{(n)} = \mathbf{e}_{i-1}^{(n)} - \Delta x_i^{(n)} \mathbf{a}_i$
7. End Do
8. $\mathbf{r}^{(n)} = \mathbf{e}_N^{(n)}$
9. if $\frac{\|\mathbf{r}^{(n)}\|_2}{\|\mathbf{r}^{(0)}\|_2} < \epsilon$ stop
10. End Do

紙面の制限のため詳細は略すが，SMR法の算法に対称性を持たせることでSSOR法に対応した対称逐次的最小残差 (Symmetric SMR : SSMR) 法が構築できる．

RJ法およびSMR法の1反復あたりの演算量およびメモリ量をTable 1に示す．参考としてJacobi法，GS法およびSOR法の結果も示す．表中の N は係数行列の次元数， N' は非零要素数をそれぞれ意味する．なお，メモリ量については係数行列，解ベクトルおよび右辺ベクトル分は除外する．Table 1よりRJ法の演算量はJacobi法よりも

Table 1 The floating-point operations per one iteration and memory requirement of iterative methods. The value in brackets means the ratio to the results for the Jacobi method.

Method	Operations (ratio)	Memories (ratio)
Jacobi	$2N' + 4N$ (= 1.0)	N (= 1.0)
GS	$4N' + N$ (< 2.0)	0 (0)
SOR	$4N' + N$ (< 2.0)	$N + 1$ ($\simeq 1.0$)
RJ	$2N' + 10N - 1$ (1.0 to 2.0)	$3N + 1$ ($\simeq 3.0$)
SMR	$4N' + 3N$ (< 2.0)	$2N + 1$ ($\simeq 2.0$)

多く，特に係数行列の疎行列度が高い (N'/N が1に近い) 場合は両者の差が顕著となる．このようにRJ法では緩和係数を動的に決定できることでJacobi法よりも優れた収束性が期待できる反面，同法に対する追加の演算量は多い．一方，SMR法の演算量はGS法よりも多いがSOR法より少ない．前述のようにSMR法はSOR法を基本とし，GS法の収束性向上という目的を持つ．したがって，SMR法はGS法より優れた収束性がSOR法よりも高速に実現できることが期待される．

4. 数 値 実 験

本節ではJacobi法，GS法，RJ法，SMR法を対象に収束性と計算時間を比較する．扱う問題はフロリダ大学疎行列コレクション⁷⁾にて2010年6月18日時点で公開された実数非対称行列問題 (全732種類) である．ただし，Table 2に示す5種については後述の計算機環境および計算条件において30日以上の日数を以ってしても計算が終了しないため，本実験において評価の対象外とした．

数値実験はCPUにIntel Core i7 950 (クロック : 3.06 GHz, L3キャッシュ : 8 MB), 24 GBの主メモリ量, OSとしてOpenSUSE 11.1を搭載したPC上で実施する．プログラムはC言語 (gcc v4.4.1, 最適化オプション : “-O3”) で実装し，演算はすべて倍精度浮動小数点演算で行なう．なお，gccの-O3オプションは自動並列化機能を持たないので，プロセッサの状態によらずコアは一つしか使わない．計算時間の測定にはC言語のgetrusage関数を利用する．反復計算では初期値を $\mathbf{x}^{(0)} = \mathbf{0}$ ，収束判定値を $\epsilon = 10^{-8}$ および 10^{-4} に設定し，前者を「高精度版」，後者を「低精度版」と呼ぶことにする．算法間の性能比較のため，最大反復回数は次元数の10倍とする．

Table 3およびTable 4に解法間の収束性の比較結果を示す．表中の“res”は反復計算終了時の相対残差 L_2 ノルム，“RES”は収束後の真の相対残差 L_2 ノルムをそれぞれ意味する．同表より計算が完了した全727種類に対して

Table 2 Specifications of matrices which are not utilized for the numerical test.

matrix	N	N'	N'/N
circuit5M	5,558,326	59,524,291	10.71
circuit5M_dc	3,523,317	14,865,409	4.22
Freescall1	3,428,755	17,052,626	4.97
memchip	2,707,524	13,343,948	4.93
rajat31	4,690,002	20,316,253	4.33

Jacobi法では約83%，GS法では約77%の問題において相対残差ノルムが不定 (NaN) または発散 (Div.) を示した。この結果に対し、動的な緩和係数を導入することで、RJ法では約23%，SMR法に至っては0%にまで改善された。以上の割合は低精度版においても同じであった。

未収束 (Not Conv.) および収束 (Conv.) に注目する。高精度版ではGS法が最も多くの問題を解いており、逆にSMR法は最も少ない結果となった。SMR法は不定や発散は生じないもののに90%以上の問題で未収束となった。同様の傾向はRJ法でも見られる。これに対して、低精度版においてはSMR法が全体の35%以上と他の解法の2倍以上の問題を解くことが可能となることを示した。これにともない、SMR法に対する未収束の問題数は高精度版の結果に対して30%程度の改善が見られた。このことから、SMR法における残差は非常に緩やかであるが確実に減少させるという優れた特徴を持つことが予想される。一方、RJ法における未収束の問題数は低精度版と高精度版で大差がなかった。その理由として、RJ法では残差の減少は起こるものの、その途中で停滞を起こしたためと考えられる。

Table 5では1反復あたりの計算時間比を示す。ここでは、次元数が500以上で4つの解法すべてが最大反復回数に達した454例に対する計算時間の平均値を基にJacobi法に対する比率を決定した。GS法とSMR法に対する結果はTable 1の比率の欄とほぼ一致した。また、両者の1反復あたりの演算量の差は $2N$ であるが、それによる比率への影響は小さい。RJ法の比率はTable 5に示した1から2の間であるが、1に近い数値となった。以上の結果の要因として、テスト問題が疎行列を対象としているものの非零要素数 (N') が次元数 (N) に比べ十分大きいためであると考えられる。

上記では727種類の問題に対する残差の傾向を解法ごとに示したが、これを解法間で調査する。Fig. 1およびFig. 2ではベン図を用いてTable 3およびTable 4の数値を場合分けでそれぞれ表示する。両図(a)において、Jacobi法において不定を示す110例はGS法およびRJ法においても不定を示す。この3つの解法に共通する演算に係数行列の対角成分を用いた割り算があることから、原因としては対角成分に0があることによるブレイクダウンと予想される。一方、SMR法についてはその算法より零割

Table 3 The number of problems for the states of residual in each iterative method with $\varepsilon = 10^{-8}$. The value in the parenthesis means the ratio to the total number of problems for the numerical tests.

State	Jacobi	GS	RJ	SMR
NaN	110 (15.13)	122 (16.78)	126 (17.33)	0 (0.00)
Div. (res ≥ 1.0)	495 (68.09)	438 (60.25)	39 (5.36)	0 (0.00)
Not Conv. (res $> \varepsilon$)	51 (7.02)	67 (9.22)	490 (67.40)	664 (91.33)
Conv. (res $\leq \varepsilon$)	71 (9.77)	99 (13.62)	71 (9.77)	63 (8.67)

Table 4 The number of problems for the states of residual in each iterative method with $\varepsilon = 10^{-4}$.

State	Jacobi	GS	RJ	SMR
NaN	110 (15.13)	122 (16.78)	126 (17.33)	0 (0.00)
Div. (res ≥ 1.0)	490 (67.40)	432 (59.42)	39 (5.36)	0 (0.00)
Not. Conv. (res $> \varepsilon$)	45 (6.19)	47 (6.46)	474 (65.20)	469 (64.51)
Conv. (res $\leq \varepsilon$)	82 (11.28)	125 (17.19)	87 (11.97)	258 (35.49)

Table 5 The ratio of the computation time for each iterative method to that for the Jacobi method.

Method	Jacobi	GS	RJ	SMR
Ratio	1.00	1.94	1.21	1.95

による不定は係数行列の少なくとも1つの列の成分すべてが0のときに生じる。しかし、実用的な問題においてそのような連立1次方程式を構築することは考えにくい。したがって、実用上SMR法は零割によるブレイクダウンを起こさないという優れた特徴を持つ。他方、GS法およびRJ法に関してはそれぞれ12例および16例の問題で単独に不定を示す。これについては算法特有のものであることが考えられる。

残差が発散する場合 (Fig. 1 (b) および Fig. 2 (b)) を見ると、Jacobi法が発散する事例のほとんどはGS法でも発散することが分かった。この特徴はStein-Rosenbergの定理⁸⁾により説明できる。また、両図においてJacobi法とGS法の事例数に差が生じる。これは反復計算により残差が減少し低精度版では収束に達したが、高精度版では収束に達する前に残差が増加傾向となりそのまま発散したためと考えられる。

Fig. 1 (d) および Fig. 2 (d) よりSMR法が単独で収束する例は他の3つの解法に比べ倍以上と多い。このことは、従来の古典的反復法では太刀打ちできない問題に対して解を得る可能性を拡大させたといえる。しかし、他の3つの解法では収束を示すがSMR法では収束しない事例が多く見られる。また、Fig. 1 (c) および Fig. 2 (c) より他

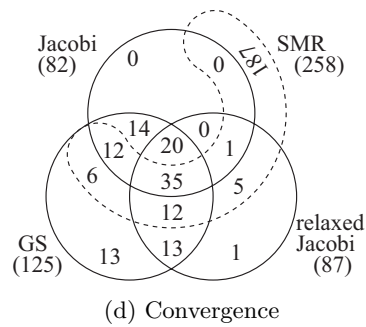
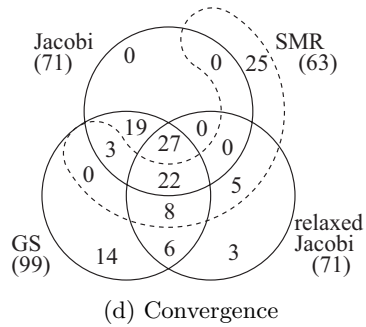
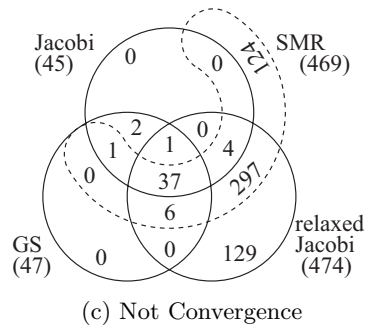
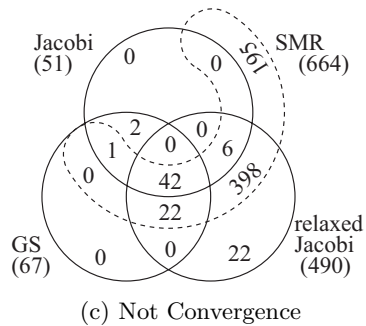
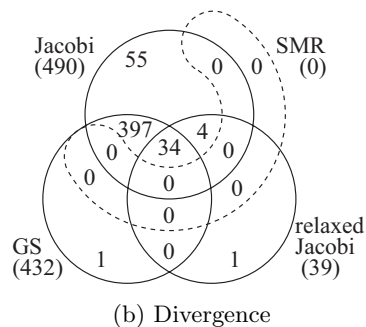
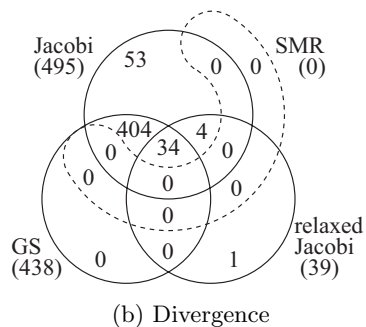
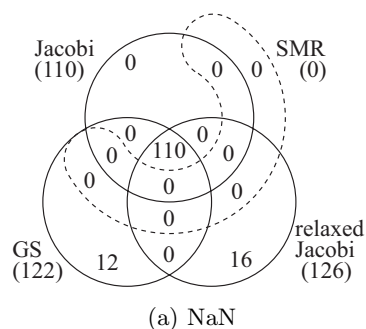
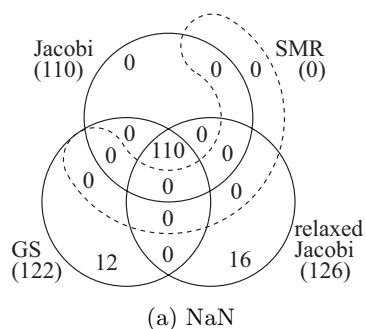


Fig. 1 Detailed classifications for Table 3.

Fig. 2 Detailed classifications for Table 4.

の3つの解法では未収束を示す事例の多くはSMR法においても未収束を示す。これらの理由としてはSMR法の収束性が他の3つの解法よりも低いことが考えられる。

高精度版においてSMR法の収束性を検証する。Table 6ではSMR法のみ収束した25例のうち8例を対象に収束までの反復回数と次元数に対する比率を示す。Table 6より高速に収束する事例はあるものの、多くの事例において収束までの反復回数が次元数を大きく上回る。Table 7ではSMR法と他の3つの解法のうち少なくとも1つが収束した38例のうち8例を対象に収束までの反復回数

を示す。なお、最短の計算時間で収束した解法には数値を太字で表す。Table 7より他の3つの反復法で収束するほとんどの問題ではSMR法の収束性は低いことが示された。

Fig. 3に各反復法に対する残差の履歴を示す。RJ法およびSMR法はそれぞれ算法中の残差および中間残差の最小化のために緩和係数を動的に決定した。このため、前述のようにJacobi法やGS法において残差の発散傾向にある多くの問題に対してこれを回避させることができた。しかし、RJ法においては残差の停滞による未収束の傾向が強い。他方、SMR法では解ベクトルの1つ1つの成分に

Table 6 The number of iterations to convergence for the case that only the SMR method is available. The stopping criterion is set at $\epsilon = 10^{-8}$. The value in the bracket denotes the ratio of the number of iterations to the that of dimension N .

Matrix	Jacobi	GS	RJ	SMR
adder_				
dcop_68	Div	Div	NC	16,230 (8.95)
trans_01	Div	Div	NC	1,638 (0.90)
bayer01	Div	Div	Div	119,249 (2.07)
bayer02	Div	Div	Div	72 (0.01)
circuit_1	Div	Div	NC	7,075 (2.70)
FA	Div	Div	NC	25,879 (2.44)
Kaufhold	Div	Div	NC	53 (0.01)
ns3Da	Div	Div	NC	86,898 (4.26)

Table 7 The number of iterations to convergence for the $\epsilon = 10^{-8}$ case. The bold script means that the corresponding solver converges the fastest.

Matrix	Jacobi	GS	RJ	SMR
add32	734	378	122	36,105
atmosmod1	8,096	4,042	8,085	2,835,200
cage13	260	11	24	62
cage14	230	10	18	26
ex37	Div	28	41	122
FEM_3D_				
thermal1	Div	112	158	3115
thermal2	Div	47	71	549
lung2	Div	Div	1545	434

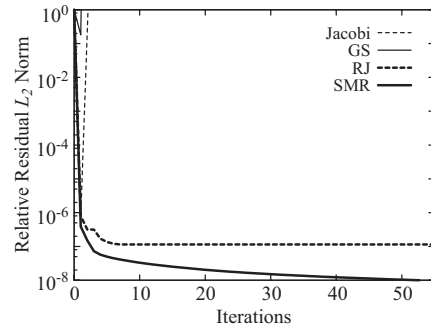
対して中間残差最小化条件を適用するため、残差の停滞が回避できたと考えられる。これらのことは同図(a)において確認できる。しかし、SMR法に対する残差減少の割合は非常に小さいため、同図(b)においてはGS法およびRJ法よりも多くの反復回数が必要となる。

5. おわりに

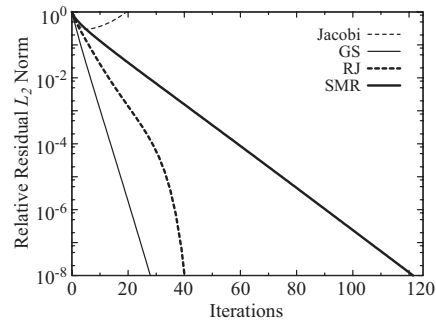
本論文ではSOR法の逐次計算部分に中間残差ベクトルを導入し、これに対する最小条件を加味することで緩和係数を動的に与える方法を考案した。そして、数式を整理することで緩和係数が陽に現れず、単に解ベクトルの各成分を中間残差最小条件に従って修正させる新しい反復法：逐次的最小残差 (SMR) 法を提案した。700種類を超える実用的な実数非対称行列問題を対象とする数値実験を通じて、SMR法は従来の古典的反復法では解けない問題が解けるようになることを示した。しかし、収束性については、残差は着実に減少するもののその割合は非常に緩やかであり、その改善が今後の課題である。

謝 辞

本研究の一部は平成22年度文部科学省科研費補助金(若手研究(B):20760058)の助成による。



(a) Matrix: Kaufhold



(b) Matrix: ex37

Fig. 3 Convergence history of iterative methods.

参 考 文 献

- 1) R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. M. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994.
- 2) Wathen A. J., On Relaxation of Jacobi Iteration for Consistent and Generalized Mass Matrices, *Comm. Appl. Numer. Meth.*, **7**(1991), 93–102.
- 3) R. S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.
- 4) Y. Saad, *Iterative Methods for Sparse Linear Systems*, 1st edition, PWS publishing, New York, 1996.
- 5) Nakashima N. and Tateiba M., A Comparative Study on Iterative Progressive Numerical Method for Boundary Element Analysis of Electromagnetic Multiple Scattering, *IEICE Trans. Electron.*, **E94-C** (2011), 865–873.
- 6) James K. R., Convergence of Matrix Iterations Subject to Diagonal Dominance, *SIAM J. Numer. Anal.*, **10**(1973), 478–484.
- 7) Davis T., University of Florida Sparse Matrix Collection, <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- 8) Stein P., Rosenberg R. L., On the Solution of Linear Simultaneous Equations by Iteration, *J. London Math. Soc.*, **23**(1948), 111–118.

