

第9章 連立一次方程式の解法2 — 反復法

本章では、主に疎行列に対して用いられている反復法 (Jacobi 反復法, Gauss-Seidel 法, SOR 法) の解説と、数値的性質について述べる。

内容

1. 反復法の原理 (Theory of Iteration Method) —
Jacobi 反復法, Gauss-Seidel 法, SOR 法に共通する理論を解説する。
2. Jacobi 反復法 (Jacobi's Iteration Method) —
Jacobi 反復法のアルゴリズムについて述べる。
3. Gauss-Seidel 法 (Gauss-Seidel Method) —
Gauss-Seidel 法のアルゴリズムについて述べる。
4. SOR 法 (SOR Method) —
SOR 法の収束条件, 最適な加速係数の決定。

9.1 反復法の原理

連立一次方程式

$$A\mathbf{x} = \mathbf{b} \quad (9.1)$$

の解の近似値 \mathbf{x}_{k+1} を漸化式

$$\mathbf{x}_{k+1} := M\mathbf{x}_k + \mathbf{c} \quad M \in M_n(\mathbb{R}), \mathbf{c} \in \mathbb{R}^n \quad (9.2)$$

で計算する方法を、連立一次方程式に対する反復法 (iteration method) と呼ぶ。この反復式は (9.1) と同値な系となるようにして作られる。即ち、

$$\boxed{\mathbf{x} = M\mathbf{x} + \mathbf{c} \text{ の解}} = \boxed{A\mathbf{x} = \mathbf{b} \text{ の解}} \quad (9.3)$$

となるようにする。

数値計算の常道として、漸化式はなるべく前の近似値に修正項を加えていくような形になるようにする。そうすることで、収束する場合は修正項の大きさも減少していくため情報落ちを起こし、近似値の変化も小さくなってついには全く変化しなくなることが期待出来、収束の判断がしやすくなる。

この常道に従えば、(9.2)は

$$\mathbf{x}_{k+1} := \mathbf{x}_k + (M - I)\mathbf{x}_k + \mathbf{c} \quad (9.4)$$

となる。

次に、(9.2)式によって得られる列 $\{\mathbf{x}_0, \mathbf{x}_1, \dots\}$ が収束するための条件を見ていくことにする。

定理 9.1.1

任意の初期値 $\mathbf{x}_0 \in \mathbb{R}^n$ に対して漸化式 (9.2) を用いて得られる列 $\{\mathbf{x}_i\}_0^\infty$ が収束するためには、行列 M の全ての固有値の絶対値が 1 より小さいことが必要十分条件である。

(証)

(\Rightarrow)
 $\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}$ とする。さすれば

$$\left. \begin{aligned} \mathbf{x} &= M\mathbf{x} + \mathbf{c} \\ \mathbf{x}_k &= M\mathbf{x}_{k-1} + \mathbf{c} \end{aligned} \right\}$$

より

$$\begin{aligned} \mathbf{x} - \mathbf{x}_k &= M(\mathbf{x} - \mathbf{x}_{k-1}) \\ &= \dots = M^k(\mathbf{x} - \mathbf{x}_0) \end{aligned}$$

から

$$\|\mathbf{x} - \mathbf{x}_k\| \leq \|M\|^k \|\mathbf{x} - \mathbf{x}_0\|$$

を得る。これより $\|M\| < 1$ であるためには、固有値は全て 1 未満でなくてはならない。

(\Leftarrow)
 仮定より、 $\|M\| < 1$ であるから

$$\begin{aligned} \mathbf{x}_k &= M\mathbf{x}_{k-1} + \mathbf{c} \\ &= M(M\mathbf{x}_{k-2} + \mathbf{c}) + \mathbf{c} = \dots \\ &= M^k \mathbf{x}_0 + \left(\sum_{i=1}^{k-1} M^i \right) \mathbf{c} \end{aligned}$$

を得る。よって、 $\lim_{k \rightarrow \infty} \mathbf{x}_k = (I - M)^{-1} \mathbf{c}$ となる。

(証明終)

この定理は Banach 空間における縮小写像原理の応用になっている。

またこの定理により、収束の早さについての目安をつけることができる。即ち、

$$\|\mathbf{x} - \mathbf{x}_k\| \leq \|M\|^k \|\mathbf{x} - \mathbf{x}_0\| \quad (9.5)$$

だから、少なくとも M の最大固有値の絶対値倍以上の早さで収束していくことになる。

以後の節で述べる幾つかの反復法は、漸化式 (9.2) における M , \mathbf{c} の与え方によって決まる。ここでは反復の十分条件を A の性質から導くこととなるが、その全てはこの定理を用いて示される。

9.2 Jacobi 反復法

実対称行列の固有値問題の解法, Jacobi 法と区別するために, 「Jacobi 反復法」と呼ぶことにする。
この Jacobi 反復法を成分ごとに書くと次のようなアルゴリズムになる。

アルゴリズム 13 (Jacobi 反復法)

1. 初期値 $\mathbf{x}_0 \in \mathbb{R}^n$ を与える。
2. for $k = 1, 2, \dots$ に対して以下を計算する。

$$\begin{cases} x_1^{(k+1)} & := x_1^{(k)} + \frac{1}{a_{11}}(b_1 - \sum_{j=1}^n a_{1j}x_j^{(k)}) \\ x_2^{(k+1)} & := x_2^{(k)} + \frac{1}{a_{22}}(b_2 - \sum_{j=1}^n a_{2j}x_j^{(k)}) \\ & \vdots \\ x_n^{(k+1)} & := x_n^{(k)} + \frac{1}{a_{nn}}(b_n - \sum_{j=1}^n a_{nj}x_j^{(k)}) \end{cases}$$

これを (9.2) の形で書けば,

$$\mathbf{x}_{k+1} := J\mathbf{x}_k + \mathbf{c} \quad (9.6)$$

ここで

$$J = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{a_{n-1,n}}{a_{n-1,n-1}} \\ -\frac{a_{n1}}{a_{nn}} & \cdots & -\frac{a_{nn-1}}{a_{nn}} & 0 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{bmatrix}.$$

この漸化式の持つ収束の条件を調べてみる。

$J \in M_n(\mathbb{R})$ の固有多項式 $p_J(\lambda)$ は

$$\begin{aligned} p_J(\lambda) &= |J - \lambda I| \\ &= \begin{vmatrix} -\lambda & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & -\lambda & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & -\lambda \end{vmatrix} \\ &= (-1)^n a_{11} a_{22} \cdots a_{nn} \begin{vmatrix} a_{11}\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}\lambda & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}\lambda \end{vmatrix} \end{aligned} \quad (9.7)$$

となる。よって $a_{ii} \neq 0$ ($i = 1, 2, \dots, n$) であれば

$$p_J(\lambda) = \begin{vmatrix} a_{11}\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}\lambda & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}\lambda \end{vmatrix} \quad (9.8)$$

と考えてよい。

この固有多項式を解析的に解くことは一般に不可能である。そこで、Jacobi 反復法が収束する必要十分条件を求めることはあきらめて、十分条件を得ることで我慢する。そのために以下の Gerschgorin の定理を用いる。

定理 9.2.1 (Gerschgorin の定理)

正方行列 $A = [a_{ij}]$ ($i, j = 1, 2, \dots, n$) に対し、複素平面 \mathbb{C} において中心 a_{ii} 、半径 $\sum_{j=1, j \neq i}^n |a_{ij}|$ の円 S_i ($i = 1, 2, \dots, n$) を定義した時、全ての A の固有値 λ は $\cup_{i=1}^n S_i$ 内に存在する。これは任意の A の固有値 λ に対し

$$|a_{ii} - \lambda| \leq \sum_{j=1, j \neq i}^n |a_{ij}|$$

を満たす $i \in \{1, 2, \dots, n\}$ が必ず存在することを意味する。

さすれば任意の J の固有値 λ に対して

$$|\lambda| \leq \max_i \sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| \quad (9.9)$$

だから

$$\max_j \sum_{i=1, i \neq j}^n \left| \frac{a_{ij}}{a_{ii}} \right| \leq 1 \quad (9.10)$$

が収束のための十分条件となる。これは同時に $\|M\|_1 \leq 1$ なる条件も満たしている。

また、 $\|\cdot\|_\infty$ を用いて $\|M\|_\infty \leq 1$ なる条件を満たすためには、

$$\max_i \sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| \leq 1 \quad (9.11)$$

となればよい。

実用的には上の条件で十分であろうが、理論的にはこれをもう少しゆるめることができる。[→ Varga 参照]

次に、このアルゴリズムの計算量を求めてみよう。

反復1回につき、

乗除算 — $n(n+1)$ 回

加減算 — $n(n-1+2) = n(n+1)$ 回

だから、 k 回反復すると

乗除算 — $kn(n+1)$ 回

加減算 — $kn(n+1)$ 回

となる。

問題 9.2.1

枢軸選択なしの LU 分解と Jacobi 反復法を使って n 次元の連立一次方程式を解く時、Jacobi 反復法が計算量の面で優位となる最大反復回数を求めよ。

9.3 Gauss-Seidel 法

Jacobi 反復法にちょっとした改良を加えてできたのが Gauss-Seidel 法である。

アルゴリズム 13 より, \mathbf{x}_{k+1} の第 i 成分 $x_i^{(k+1)}$ を計算するときには既に $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ は求められている。そこで, $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$ の代わりに $k+1$ 近似値であるこれらを用いることで, 収束のスピードアップを期待できる。これが Gauss-Seidel 法の原理である。

アルゴリズム 14 (Gauss-Seidel 法)

1. 初期値 $\mathbf{x}_0 \in \mathbb{R}^n$ を決める。
2. $k = 1, 2, \dots$ に対して, 以下を計算する。

$$\begin{cases} x_1^{(k+1)} & := x_1^{(k)} + \frac{1}{a_{11}}(b_1 - \sum_{j=1}^n a_{1j}x_j^{(k)}) \\ x_2^{(k+1)} & := x_2^{(k)} + \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k+1)} - \sum_{j=2}^n a_{2j}x_j^{(k)}) \\ & \vdots \\ x_n^{(k+1)} & := x_n^{(k)} + \frac{1}{a_{nn}}(b_n - \sum_{j=1}^{n-1} a_{nj}x_j^{(k+1)} - a_{nn}x_n^{(k)}) \end{cases}$$

である。Jacobi 反復法に比べて, 新旧の \mathbf{x}_k を取っておく必要がない分, 記憶容量が節約できる。

この Gauss-Seidel 法が Jacobi 反復法より収束が本当に早くなるかどうかを見てみよう。

行列とベクトルで上のアルゴリズムを記述すると

$$D = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}, L = \begin{bmatrix} & a_{21} & & \\ & \vdots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & \end{bmatrix}, U = \begin{bmatrix} & a_{12} & \cdots & a_{1n} \\ & & \ddots & \vdots \\ & & & a_{n-1,n} \end{bmatrix} \quad (9.12)$$

とおけば

$$\mathbf{x}_{k+1} := D^{-1}(\mathbf{b} - L\mathbf{x}_{k+1} - U\mathbf{x}_k) \quad (9.13)$$

となり, これを右辺に \mathbf{x}_{k+1} を含まない形にすれば,

$$\mathbf{x}_{k+1} := -(I + D^{-1}L)^{-1}D^{-1}U\mathbf{x}_k + (I + D^{-1}L)^{-1}D^{-1}\mathbf{b}$$

から

$$\mathbf{x}_{k+1} := -(D + L)^{-1}U\mathbf{x}_k + (D + L)^{-1}\mathbf{b} \quad (9.14)$$

を得る。ここで $G = -(D + L)^{-1}U$ と置くことにする。

このとき次の定理が成立する。

定理 9.3.1

$$\sum_{i=1, i \neq j}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1 \quad (j = 1, 2, \dots, n)$$

ならば Gauss-Seidel 法は収束し,

$$\|G\|_{\infty} \leq \|J\|_{\infty} < 1$$

となる。

(証)

仮定より, $\|J\|_\infty < 1$ は自明。 $\|G\|_\infty \leq \|J\|_\infty$ を言えばよい。

今, $J = (a_{ij}) \in M_n(\mathbb{R})$ とする。これは行列式ではなく, 元の行列の成分の絶対値を成分として持つ実行列である。このとき $e^T = [1 \ 1 \ \cdots \ 1]$ とすれば

$$|J|e \leq \|J\|_\infty e < e$$

となる。この不等号は, ベクトルの全ての成分で成り立っていることを意味する。

ここで $J := L^J + U^J$, L^J, U^J は J の上, 下三角行列, とすると,

$$|U^J|e \leq (\|J\|_\infty I - |L^J|)e$$

となる。更に, L^J は対角成分が全て 0 の下三角行列だから,

$$(L^J)^n = |L^J|^n = 0$$

より

$$\begin{aligned} 0 \leq |(I - L^J)^{-1}| &= |I + L^J + (L^J)^2 + \cdots + (L^J)^{n-1}| \\ &\leq I + |L^J| + |L^J|^2 + \cdots + |L^J|^{n-1} \\ &= (I - |L^J|)^{-1} \end{aligned}$$

となり,

$$\begin{aligned} G = -(D + L)^{-1} &= -(I + D^{-1}L)^{-1}D^{-1}U \\ &= (I - L^J)^{-1}U^J \end{aligned}$$

だから, $(I - |L^J|)^{-1}$ が非負成分からなる行列であることから,

$$\begin{aligned} |G|e &\leq (I - |L^J|)^{-1}|U^J|e \\ &\leq (I - |L^J|)^{-1}(I - |L^J| + (\|J\|_\infty - 1)I)e \\ &= \{I + (I - |L^J|)^{-1}(\|J\|_\infty - 1)\}e \end{aligned}$$

ここで, $I \leq (I - |L^J|)^{-1}$, $\|J\|_\infty < 1$ より

$$\leq (I + (\|J\|_\infty - 1)I)e = \|J\|_\infty e$$

よって, $\|G\|_\infty \leq \|J\|_\infty$ を得る。

(証明終)

更に条件を絞ることによって, 次の定理を得る。

定理 9.3.2 (Stein-Rosenberg の定理)

$J = L^J + U^J \in M_n(\mathbb{R})$ が非負要素しか持たないとき, $G = (I - L^J)^{-1}U^J$ は

- (1) $|\lambda_n(G)| = |\lambda_n(J)| = 0$
- (2) $0 < |\lambda_n(G)| < |\lambda_n(J)| < 1$
- (3) $|\lambda_n(G)| = |\lambda_n(J)| = 1$
- (4) $|\lambda_n(G)| > |\lambda_n(J)| > 1$

のうち, どれか一つを満たす。ここで $\lambda_n(A)$ は $A \in M_n(\mathbb{R})$ の絶対値最大のものをさす。

よって, $A \in M_n(\mathbb{R})$, $a_{ij} \geq 0$ であれば, もし Jacobi 反復法が収束すれば, Gauss-Seidel 法はより速く収束し, 逆に Jacobi 反復法が発散するときは Gauss-Seidel 法も発散することが示された。

9.4 SOR 法

Gauss-Seidel 法にパラメータ ω を導入して更なる加速を図ったのが SOR (Successive Over-Relaxation) である。

アルゴリズム 15 (SOR 法)

1. 初期値 \mathbf{x}_0 を与える。
2. $k = 0, 1, 2, \dots$ に対して

$$\begin{cases} x_1^{(k+1)} & := x_1^{(k)} + \omega \frac{1}{a_{11}} (b_1 - \sum_{j=1}^n a_{1j} x_j^{(k)}) \\ x_2^{(k+1)} & := x_2^{(k)} + \omega \frac{1}{a_{22}} (b_2 - a_{21} x_1^{(k+1)} - \sum_{j=2}^n a_{2j} x_j^{(k)}) \\ & \vdots \\ x_n^{(k+1)} & := x_n^{(k)} + \omega \frac{1}{a_{nn}} (b_n - \sum_{j=1}^{n-1} a_{nj} x_j^{(k+1)} - a_{nn} x_n^{(k)}) \end{cases} \quad (9.15)$$

ここで, ω を加速パラメータと呼ぶ。

この漸化式をベクトルと, 先ほどの式 (9.12) で定義した行列を用いて表わせば,

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \omega D^{-1} \{ \mathbf{b} - L\mathbf{x}_{k+1} - (D + U)\mathbf{x}_k \} \quad (9.16)$$

より

$$\frac{1}{\omega} D(I + \omega D^{-1} L)\mathbf{x}_{k+1} := \frac{1}{\omega} D \{ I - \omega(I + D^{-1} U) \} \mathbf{x}_k + \mathbf{b} \quad (9.17)$$

となる。 $\omega = 1$ のときには Gauss-Seidel 法と一致する。

ここで $B(\omega) = \frac{1}{\omega} D(I + \omega D^{-1} L)$ という行列を定義する。さすれば

$$H(\omega) = I - B(\omega)^{-1} A$$

と $H(\omega) \in M_n(\mathbb{R})$ を定義することにより, SOR 法の反復式は

$$\mathbf{x}_{k+1} := H(\omega)\mathbf{x}_k + B(\omega)^{-1}\mathbf{b} \quad (9.18)$$

と表わせる。

定理 9.4.1 (Kahan の定理)

$\forall A \in M_n(\mathbb{R})$ に対し,

$$\max |\lambda(H(\omega))| \geq |\omega - 1| \text{ for } \forall \omega$$

(証)

$$\det(I + \omega D^{-1} L) = \det \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = 1$$

から

$$\begin{aligned} \det(H(\omega)) &= \det\{(I + \omega D^{-1}L)H(\omega)\} \\ &= \det\{I - \omega(I + D^{-1}U)\} \\ &= \det\{(1 - \omega)I - \omega D^{-1}U\} \\ &= (1 - \omega)^n = \prod_{i=1}^n \lambda_i(H(\omega)) \end{aligned}$$

よって

$$\max_i |\lambda_i(H(\omega))| \geq |\omega - 1|$$

(証明終)

この定理により、少なくとも $0 < \omega < 2$ でなければ SOR 法の収束性は保証されない。しかし、 ω がこの範囲であっても、必ず収束するわけではない。

係数行列 A の条件を絞ることで収束条件がわかりやすくなる。

定理 9.4.2 (Ostrowski-Reich の定理)

Hermite 行列 A が正定値のとき、 $0 < \forall \omega < 2$ に対して、

$$\max_i \lambda_i(H(\omega)) < 1$$

(証)

今 A を式 (9.12) で定義したように

$$A := L + D + U$$

とする。

A が Hermite であることから

$$L^* = U$$

が成立する。

$B(\omega) = \frac{1}{\omega}D(I + \omega D^{-1}L)$ であるから、

$$\begin{aligned} B(\omega) + B(\omega)^* - A &= \frac{1}{\omega}D(I + \omega D^{-1}L) + \frac{1}{\omega}(I + \omega L^*D^{-1})D - A \\ &= \frac{1}{\omega}D + L + \frac{1}{\omega}D + U - A \\ &= \left(\frac{2}{\omega} - 1\right)D \end{aligned}$$

となる。

今 λ を $A^{-1}(2B(\omega) - A) \in \mathbb{C}$ の固有値、対応する固有ベクトルを $\mathbf{x} \in \mathbb{C}^n$ とすると

$$A^{-1}(2B(\omega) - A)\mathbf{x} = \lambda\mathbf{x}$$

より、両辺に \mathbf{x}^*A をかけて

$$\mathbf{x}^*(2B(\omega) - A)\mathbf{x} = \lambda\mathbf{x}^*A\mathbf{x}。$$

A は Hermite であるから

$$\mathbf{x}^*(2B^*(\omega) - A)\mathbf{x} = \bar{\lambda}\mathbf{x}^*A\mathbf{x}。$$

また

$$\mathbf{x}^*(B(\omega) + B^*(\omega) - A)\mathbf{x} = \operatorname{Re}(\lambda)\mathbf{x}^*A\mathbf{x}$$

となる。よって A の正定値性より

$$\operatorname{Re}(\lambda) > 0$$

がわかる。更に

$$Q := A^{-1}(2B(\omega) - A) = 2A^{-1}B(\omega) - I$$

とおくと

$$(Q - I)(Q + I)^{-1} = I - B^{-1}(\omega)A = H(\omega).$$

μ, x を $H(\omega)$ の固有値, 固有ベクトルとすると

$$(Q - I)(Q + I)^{-1}x = H(\omega)x = \mu x.$$

$y := (Q + I)^{-1}x$ とすると

$$(Q - I)y = \mu(Q + I)y \text{ より}$$

$$(\mu - 1)Qy = (1 + \mu)y.$$

$y \neq 0$ と $\mu \neq 1$ から

$$Qy = \frac{1 + \mu}{1 - \mu}y.$$

よって $\lambda = (1 + \mu)/(1 - \mu)$ は $A^{-1}(2B(\omega) - A)$ の固有値になる。

$$|\mu|^2 = \frac{|\lambda|^2 + 1 - 2\operatorname{Re}(\lambda)}{|\lambda|^2 + 1 + 2\operatorname{Re}(\lambda)}$$

から

$$\mu < 1$$

が示された。

(証明終)

$\max |\lambda(H(\omega))|$ を最小にする ω を決定するには次の性質を使う。

定義 9.4.3 (property A)

$A \in M_n(\mathbb{C})$ が property A であるとは, ある基本行列 P が存在して

$$PAP^{-1} = \begin{bmatrix} D_1 & M_1 \\ M_2 & D_2 \end{bmatrix} \text{ (ここで } D_1, D_2 \text{ は対角行列)}$$

という形の行列に相似変換が可能であることをいう。

このとき次の定理が成立する。

定理 9.4.4 (Young-Varga の定理)

A が property A であるとき,

$$\omega_b = \frac{2}{1 + \sqrt{1 - \max |\lambda(J)|^2}}$$

をおいたとき

$$\max |\lambda(H(\omega))| = \omega_b - 1 = \left(\frac{\max |\lambda(J)|}{1 + \sqrt{1 - \max |\lambda(J)|^2}} \right)^2$$

が最小となる。

即ち, この ω_b を用いれば, 最も収束を早めることが可能となる。が, 一般には最適な ω を得ることは簡単ではない。

問題 9.4.1

1. SOR 法 (9.15) の反復式が (9.16) のように表現できることを確認せよ。
2. (9.16) 式が $B(\omega)$ と $H(\omega)$ を用いて, (9.17) を経て (9.18) と表現できることを確認せよ。

9.5 数値的性質

係数行列 A と解 \mathbf{x} を

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ n \end{bmatrix}$$

とする。このような形の行列 A を3重対角 (Tridiagonal) 行列と呼ぶ。

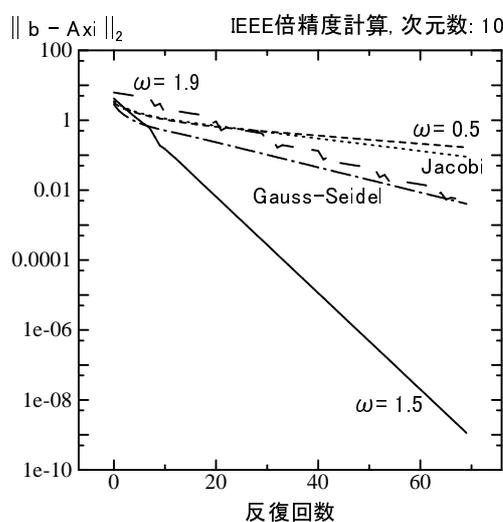


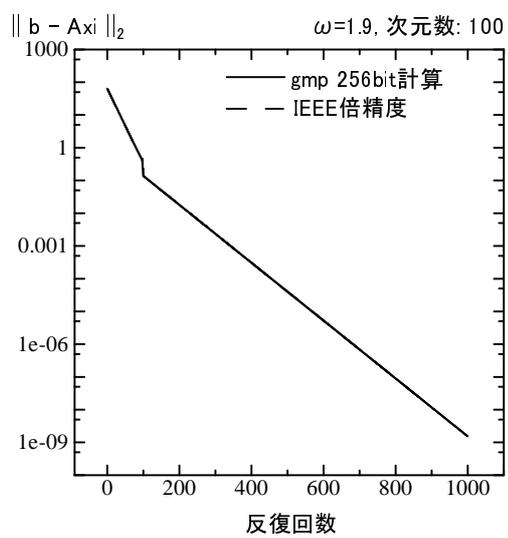
図 9.1: Jacobi 反復法, Gauss-Seidel 法, SOR 法の収束比較

この問題を $n = 10$ として, Jacobi 反復法, Gauss-Seidel 法, SOR 法 ($\omega = 0.5, 1.5, 1.9$) の残差ベクトルのユークリッドノルム $\|\mathbf{b} - A\mathbf{x}_k\|_2$ をプロットしたのが図 9.1 である。この問題では SOR 法 ($\omega = 1.5$), Gauss-Seidel 法, Jacobi 反復法の順で収束が遅くなっている。SOR 法でも加速係数 ω の選択次第で収束過程が全く変化してしまう。SOR 法ではなるべく最適な値に近いものを使用する必要がある。

図 9.2 は, 更に次元を上げて $n = 100$ とし, IEEE 倍精度, 多倍長計算した SOR 法の残差ベクトルのユークリッドノルムをプロットしたものである。全く重なっており, SOR 法が丸め誤差に対して頑健な解法であることが分かる。次章の図 10.1 と比較してみれば一目瞭然であろう。Jacobi 反復法, Gauss-Seidel 法についても同様の性質を持つ。

参考文献

反復法に関する邦文の解説書としては

図 9.2: 丸め誤差の影響 (SOR 法, 次元数 100, $\omega = 1.9$)**楽しい反復法**

仁木滉・河野敏行

共立出版

1998 年

がある。