

数値解析(C) (塩田)

— Jacobi 法・Gauss-Seidel 法 —

状況設定 $A = (a_{ij})$ を n 次行列、 $b = (b_i)$ を n 項ベクトルとすると、連立一次方程式 $Ax = b$ の解に収束する様なベクトルの列 $\{x^{(k)}\}_{k=0,1,\dots}$ を作る。

Jacobi 法の発想 $Ax = b$ を

$$\begin{cases} a_{11}x_1 = b_1 - (0 + a_{12}x_2 + \dots + a_{1n}x_n) \\ a_{22}x_2 = b_2 - (a_{21}x_1 + 0 + \dots + a_{2n}x_n) \\ \vdots \\ a_{nn}x_n = b_n - (a_{n1}x_1 + a_{n2}x_2 + \dots + 0) \end{cases}$$

と変形する。右辺に $x^{(k)}$ のデータを入れて、左辺から「新しい $x^{(k+1)}$ 」を計算しよう。

Jacobi 法のアルゴリズム $x^{(k)}$ の成分を $x^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix}$ と書いて、

$x^{(0)} :=$ 適当なベクトル ;

$k := 0$;

repeat

for $i := 1$ to n do

$$x_i^{(k+1)} := \left\{ b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right\} / a_{ii} ; \quad \dots \dots (*)$$

$k := k + 1$;

until (終了条件) ;

Gauss-Seidel 法の発想 (*) では $x_i^{(k+1)}$ を計算するときに全て古い $x_j^{(k)}$ を使っている。折角 $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ は計算してあるのだから、これを活かしてやろう。

Gauss-Seidel 法のアルゴリズム (*) の代わりに次の式を用いる :

$$x_i^{(k+1)} := \left\{ b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right\} / a_{ii} ; \quad \dots \dots (**)$$

反復行列 Jacobi 法、Gauss-Seidel 法で k ステップ目の誤差ベクトルを $e^{(k)} := x^{(k)} - x$ と置くと、

$$e^{(k+1)} = H e^{(k)} \quad e^{(k)} = H^k e^{(0)} \quad \dots \quad (\clubsuit)$$

が成り立つ。ただし H は「反復行列」と呼ばれる行列で

$$H = \begin{cases} H_J = -D^{-1}(L + U) & (\text{Jacobi 法}) \\ H_{GS} = -(D + L)^{-1}U & (\text{Gauss-Seidel 法}) \end{cases},$$

$$D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}, \quad L = \begin{pmatrix} 0 & \cdots & 0 \\ a_{21} & 0 & \\ \vdots & \ddots & \ddots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & \cdots & & 0 \end{pmatrix}.$$

ベクトルや行列に大きさ（ノルム）を定義すると収束性が議論できる。

定義 次の性質をもつ関数 $|v|$ を n 項ベクトル $v = (v_i)$ のノルムと言う。

- (1) $|v| \geq 0$ (等号は $v = 0$).
- (2) スカラー α に対して $|\alpha v| = |\alpha| \times |v|$.
- (3) $|v + w| \leq |v| + |w|$.

また $\|P\| := \max_{v \neq 0} \frac{|Pv|}{|v|}$ を n 次行列 $P = (p_{ij})$ のノルムと言う。

例 次の 3 種類がよく使われる。

- (1) $|v|_1 := \sum_{i=1}^n |v_i|$, $\|P\|_1 = \max_j \left(\sum_{i=1}^n |p_{ij}| \right)$.
- (2) $|v|_2 := \left(\sum_{i=1}^n |v_i|^2 \right)^{1/2}$, $\|P\|_2 = \max_k (\mu_k)^{1/2}$
(μ_k は $\overline{P}P$ の固有値).
- (∞) $|v|_\infty := \max_i |v_i|$, $\|P\|_\infty = \max_i \left(\sum_{j=1}^n |p_{ij}| \right)$.

命題 上の例のいずれのノルムについても次が成り立つ。

- (1) $|v_i| \leq |v|$. (2) $|Pv| \leq \|P\| \times |v|$.
- (3) $\|PQ\| \leq \|P\| \times \|Q\|$.

定理 各 i について

$$\sum_{j \neq i} |a_{ij}| < |a_{ii}|$$

が成り立てば Jacobi 法、Gauss-Seidel 法は収束する。(条件から $\|H_J\|_\infty < 1$ が言え、例えば Jacobi 法の収束性は (♣) と命題から直ちに示される。)

実行例 $\begin{cases} 3x - 2y = 1 \\ x + 3y = 4 \end{cases}$ を解く。

Jacobi 法

```
0 : x = ( 0.00000, 0.00000 )
1 : x = ( 0.33333, 1.33333 )
2 : x = ( 1.22222, 1.22222 )
3 : x = ( 1.14815, 0.92593 )
4 : x = ( 0.95062, 0.95062 )
5 : x = ( 0.96708, 1.01646 )
6 : x = ( 1.01097, 1.01097 )
7 : x = ( 1.00732, 0.99634 )
8 : x = ( 0.99756, 0.99756 )
9 : x = ( 0.99837, 1.00081 )
10 : x = ( 1.00054, 1.00054 )
11 : x = ( 1.00036, 0.99982 )
12 : x = ( 0.99988, 0.99988 )
13 : x = ( 0.99992, 1.00004 )
14 : x = ( 1.00003, 1.00003 )
15 : x = ( 1.00002, 0.99999 )
16 : x = ( 0.99999, 0.99999 )
```

Gauss-Seidel 法

```
0 : x = ( 0.00000, 0.00000 )
1 : x = ( 0.33333, 1.22222 )
2 : x = ( 1.14815, 0.95062 )
3 : x = ( 0.96708, 1.01097 )
4 : x = ( 1.00732, 0.99756 )
5 : x = ( 0.99837, 1.00054 )
6 : x = ( 1.00036, 0.99988 )
7 : x = ( 0.99992, 1.00003 )
8 : x = ( 1.00002, 0.99999 )
9 : x = ( 1.00000, 1.00000 )
```

収束しない例 同じ問題を $\begin{cases} x + 3y = 4 \\ 3x - 2y = 1 \end{cases}$ として解く。

Jacobi 法

```
0 : x = ( 0.00000, 0.00000 )
1 : x = ( 4.00000, -0.50000 )
2 : x = ( 5.50000, 5.50000 )
3 : x = ( -12.50000, 7.75000 )
4 : x = ( -19.25000, -19.25000 )
5 : x = ( 61.75000, -29.37500 )
6 : x = ( 92.12500, 92.12500 )
7 : x = ( -272.37500, 137.68750 )
8 : x = ( -409.06250, -409.06250 )
9 : x = ( 1231.18750, -614.09375 )
10 : x = ( 1846.28125, 1846.28125 )
11 : x = ( -5534.84375, 2768.92188 )
12 : x = ( -8302.76562, -8302.76562 )
13 : x = ( 24912.29688, -12454.64844 )
14 : x = ( 37367.94531, 37367.94531 )
15 : x = ( -112099.83594, 56051.41797 )
16 : x = ( -168150.25391, -168150.25391 )
17 : x = ( 504454.76172, -252225.88086 )
18 : x = ( 756681.64258, 756681.64258 )
19 : x = ( -2270040.92773, 1135021.96387 )
20 : x = ( -3405061.89160, -3405061.89160 )
```