

SOR における加速定数の決定について

東京教育大・理 高橋正子

§1. 序

2次元のラプラス方程式

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \\ \text{境界条件 } u(x, y) = f(x, y) \end{cases} \quad (1)$$

を数値的に解くことを考える。xy 平面を適当な格子に分割し、格子点 (x_i, y_j) についての差分方程式

$$u_{ij} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \quad (2)$$

によって①式を近似する。但し u_{ij} は $u(x_i, y_j)$ に対応する数値解を表わし、 (x_i, y_j) は領域内の全ての格子点に亘るものとする。この連立一次方程式を解くのに Gauss-Seidel 法は

$$u_{ij}' = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}') \quad (3)$$

の形の逐次代入を行うが、SOR (Successive OverRelaxation) 法では③における右辺と u_{ij} との差 d_{ij} に適当な定数 ω をか

けた値を修正量として

$$\begin{aligned} u'_{ij} &= u_{ij} + \omega d_{ij} \\ &= (1-\omega) u_{ij} + \frac{\omega}{4} (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \end{aligned} \quad (4)$$

による逐次代入を実行する。

u_{ij} を計算する順に並べてベクトル u で表わすとき、③の右辺は $Bu + C$ の形に表わされる。但し B は領域の形とその分割によって決まる行列であり、 C は更に境界条件にも関連するベクトルである。また、 B は対角要素が 0 であるから、その上、下三角行列を各々 U, L で表わすと ④は

$$u' = (1-\omega)u + \omega(L \cdot u + U \cdot u + C)$$

となる。 u' について解いて

$$u' = L_{\omega} \cdot u + (I - \omega L)^{-1} \omega C$$

$$\text{但し } L_{\omega} = (I - \omega L)^{-1} (\omega U + (1-\omega)I).$$

u の第 n ステップにおける修正量を $d^{(n)}$ とすると

$$d^{(n+1)} = L_{\omega} \cdot d^{(n)}.$$

ここで、 $0 < \omega < 2$ のとき ④ による逐次近似は収束するが、特に ω として、 L_{ω} の固有値の最大絶対値を最小にするような値を選べば

$$\omega_0 = \frac{2}{1 + \sqrt{1 - \mu^2}}$$

を選ぶとき、収束は最も速くなることが知られている。但し μ は B の絶対値最大の固有値で、 $1 \leq \omega_0 < 2$ である。

§2. ω の決定法

SOR の最適な加速定数 ω_0 は上で述べたように、 B の固有値を使って計算される値であるため、場合によってはこれをあらかじめ知ることが困難である。

ところで L_ω の固有値 λ_i は、 B の固有値 μ_i を用いて

$$\lambda_i = \frac{1}{2} \{ (\mu_i \omega)^2 - 2(\omega - 1) + \mu_i \omega \sqrt{(\mu_i \omega)^2 - 4(\omega - 1)} \}$$

と表わされる。従って

i) $1 \leq \omega < \omega_0$ のとき

λ_i は全て実数であり、そのうち絶対値最大のものは唯一つでしかも正であることが証明される。すなわち

$$\lambda_1 > |\lambda_2| \geq |\lambda_3| > \dots \geq |\lambda_N|$$

と並べることが出来る。

$\therefore$ B は Varga による cyclic matrix of index 2 である故、その固有値は絶対値等しく符号が逆な2つおつで対になっているため。

L_ω の固有値 λ_i に対応する固有ベクトルを t_i とし、 $d^{(1)}$ の固有ベクトル展開を

$$d^{(1)} = c_1 t_1 + c_2 t_2 + \dots + c_N t_N$$

とするとき

$$d^{(k+1)} = L_\omega \cdot d^{(k)}$$

$$= \lambda_1^k c_1 t_1 + \lambda_2^k c_2 t_2 + \dots + \lambda_N^k c_N t_N$$

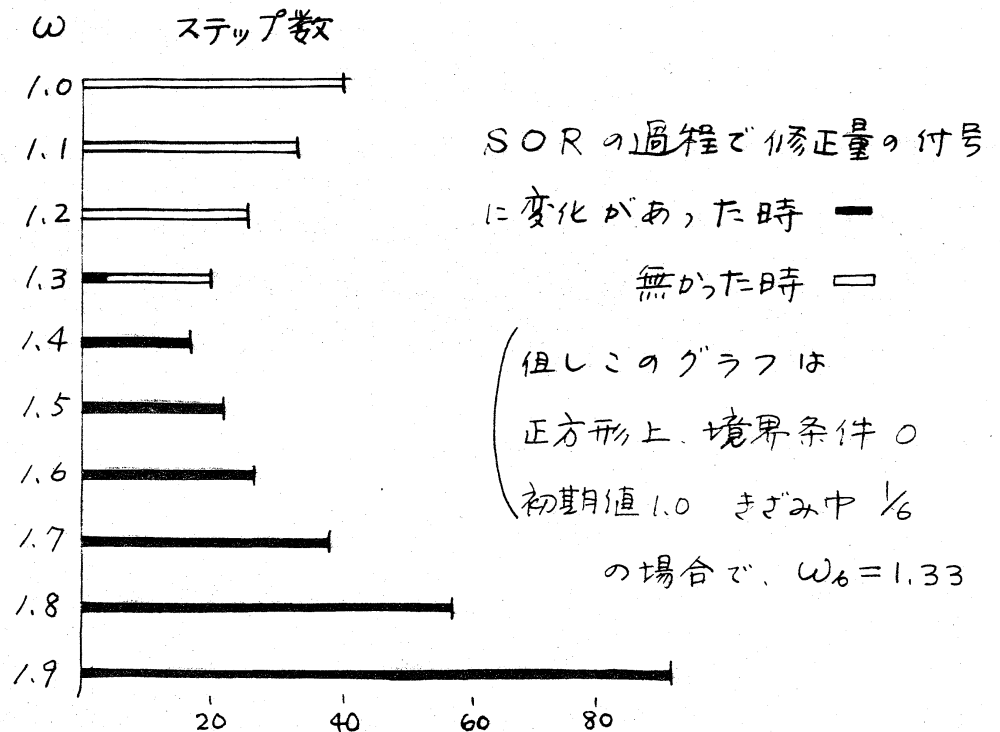
$$= \lambda_1^k \left\{ c_1 \pi_1 + \left(\frac{\lambda_2}{\lambda_1} \right)^k c_2 \pi_2 + \dots + \left(\frac{\lambda_N}{\lambda_1} \right)^k c_N \pi_N \right\}.$$

故に k が十分大なるとき $d^{(k+1)}$ と $d^{(k)}$ は各要素毎に同符号であることが言える。

ii) $\omega_0 < \omega < 2$ のとき

λ_i は一般に実数でない。(但し $\mu_i = 0$ に対応する λ_i だけは例外で、 $1 - \omega$ なる負の実数である。) 従って $d^{(k+1)}$ と $d^{(k)}$ は必ずしも要素毎に同符号にならない。

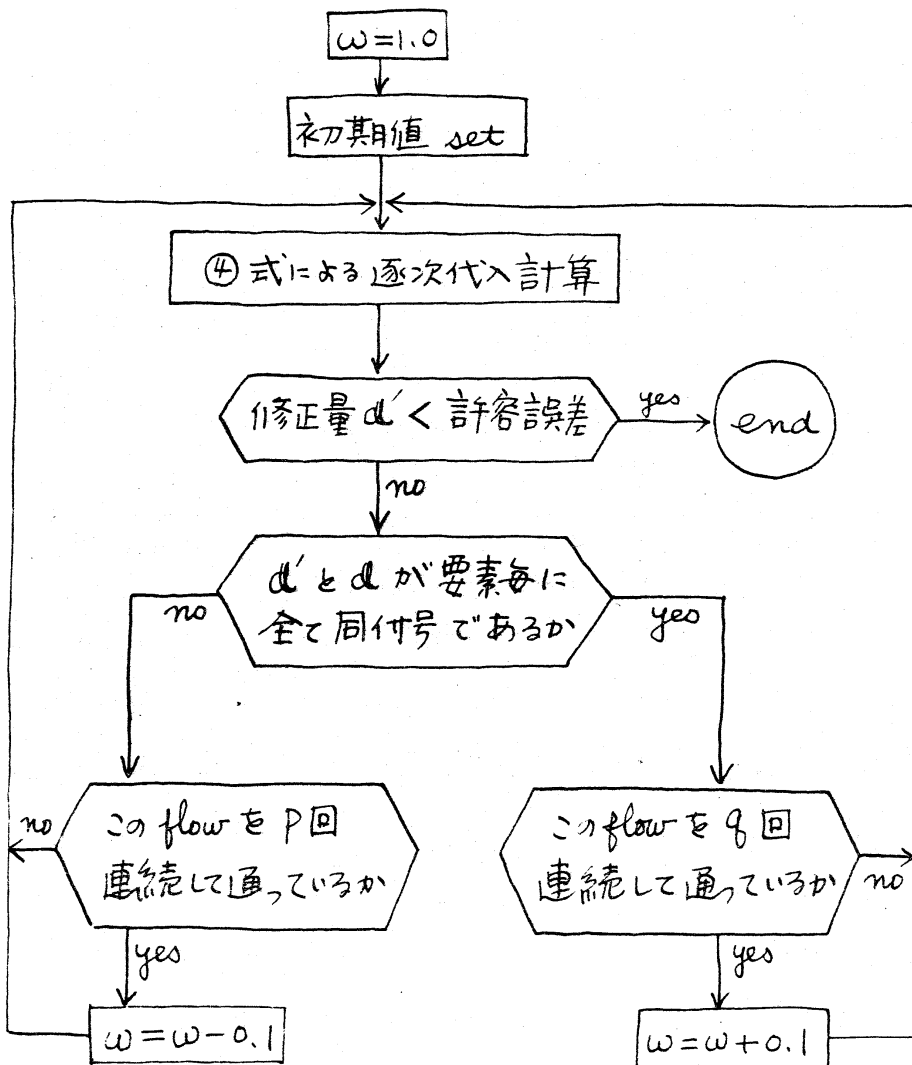
実際、初期値や格子の中を変えて試みた実験例では、いずれも次図のような顕著な傾向が見られた。



以上のことを利用して、④による逐次近似計算を実行しながら

ら $\omega < \omega_b$ か、 $\omega > \omega_b$ かを判断し、 ω を ω_b に近づけるよう
に修正し、4 要素を速めて行くことが出来るのではないかと考
えて作ったのが次のプログラムである。

§3. プログラム



§4. 計算の結果と比較

下に示す各条件の下でラプラスの方程式を (1) ω_b を用いた

SOR法 (prog B) (2) §3 に示したプログラム (prog A) および (3) Gauss-Seidel 法 によって解いたところ、下表のような結果が得られた。

但し 境界条件 $f(x, y) \equiv 0$

領域 I : $0 \leq x \leq 1, 0 \leq y \leq 1$

II : $0 \leq x \leq 1, x \leq y \leq 1$

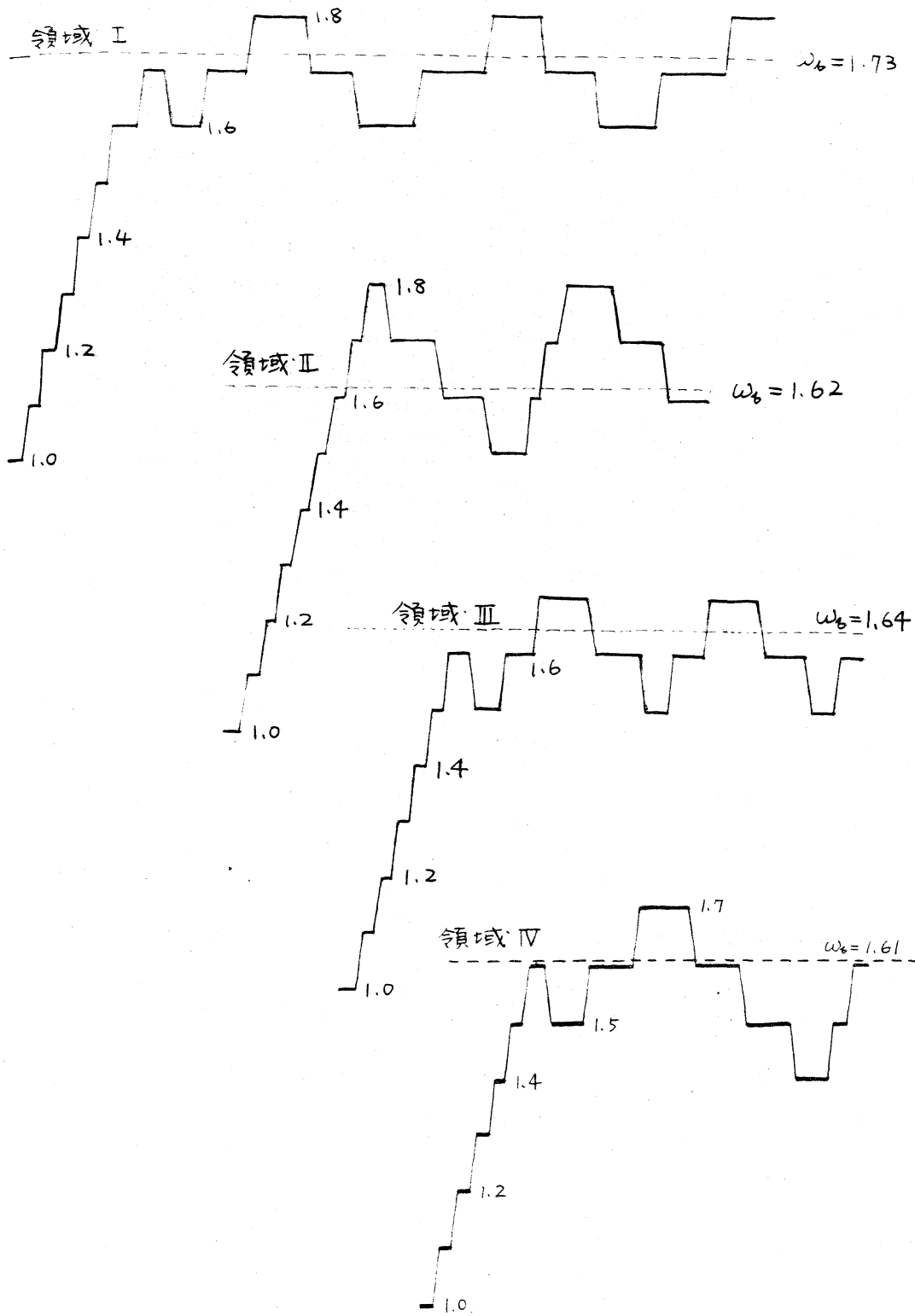
III : $\begin{cases} 0 \leq x \leq \frac{1}{2}, 0 \leq y \leq 1 \\ \frac{1}{2} \leq x \leq 1, \frac{1}{2} \leq y \leq 1 \end{cases}$

IV : $\begin{cases} 0 \leq x \leq \frac{1}{2}, -1+2x \leq y \leq 1 \\ \frac{1}{2} \leq x \leq 1, -1+2x \leq y \leq 1 \end{cases}$

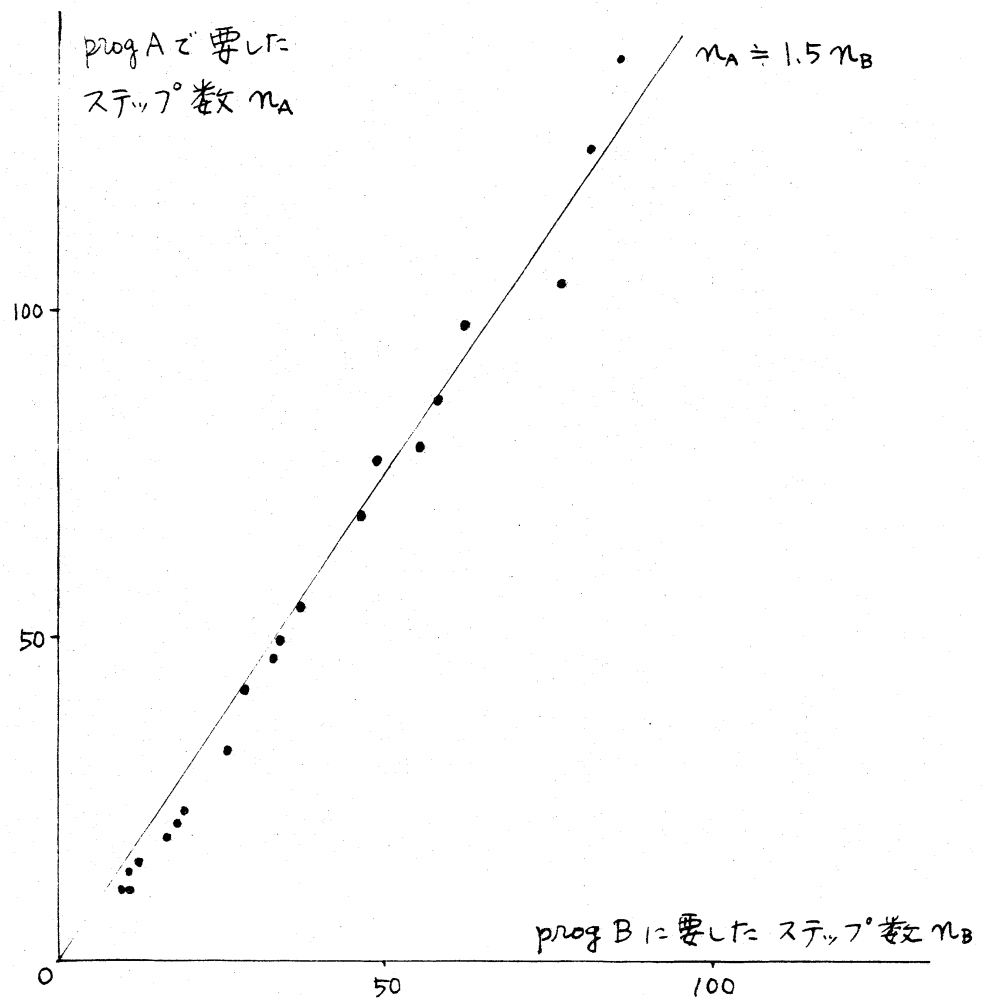
初期値 $u(x, y) \equiv 1.0$, $|d^{(k)}|$ の許容誤差 10^{-5}

領域 prog.	I	II	III	IV	きざみ中
prog B	16	12	11	10	$\frac{1}{6}$
prog A	19	15	14	11	
G-S	39	23	20	16	
prog B	49	34	37	33	$\frac{1}{20}$
prog A	77	49	54	48	
G-S	336	164	187	150	

上の計算の prog A における ω の値の変化の様子も次図に示す。(但しきざみ中 $\frac{1}{20}$ の場合のみ)



他のいくつかの実験結果も含めて、次の図に見られる通り
 prog A では prog B のほぼ 1.5 倍のステップを要した。



また、SOR で加速定数 ω として 1 と 2 の間のいろいろな
 値を用いた場合と、prog A とを比較したのが次表である。
 (領域は I で、その他の条件は 6 頁と同じ)

	ω	要したステップ数	誤差 $\times 10^5$	$\lim_{k \rightarrow \infty} \frac{\ x^{(k)}\ }{\ x^{(k-1)}\ }$	きまり
SOR	1.0	683	90	0.989	$\frac{1}{30}$
	1.1	573	74	0.987	
	1.2	479	60	0.984	
	1.3	397	48	0.980	
	1.4	324	37	0.974	
	1.5	259	28	0.967	
	1.6	199	19	0.955	
	1.7	142	12	0.934	
	1.8	81	4	0.854	
	1.9	122	0	-	
prog A		124	4	-	
SOR	1.0	39	2	0.750	$\frac{1}{6}$
	1.1	32	2	0.693	
	1.2	25	1	0.615	
	1.3	19	0	0.480	
	1.4	16	0	-	
	1.5	21	0	-	
	1.6	26	0	-	
	1.7	37	0	-	
prog A		19	0	-	

とち、誤差について

前頁の表の第4欄に見られるように、誤差(最終結果と真の解の差)は、 ω の選べ方によって非常に変わってくる。収束の遅い方法、すなわち

$$\lim_{k \rightarrow \infty} \frac{\|d^{(k)}\|}{\|d^{(k-1)}\|} = \lim_{k \rightarrow \infty} \frac{\max_j |(d^{(k)})_j|}{\max_j |(d^{(k-1)})_j|}$$

が大きい方法では、まだ真の解から程遠いにもかかわらず、 $d^{(k)}$ が許容限界以下になることがあり得るからである。従って ω の決定は、単に収束の速さのためばかりでなく、得られる結果の精度のためにも、十分慎重に検討されるべきであると思う。

参考文献

Varga(1962): Matrix Iterative Method

Todd(1962): Survey of Numerical Analysis

Young(1955): ORDVAC Solutions of the

Dirichlet Problem, J of ACM

(注) 9頁の表の誤差欄において、きざみ中が小さい時の方が精度が悪くなっているが、これはこゝで扱った方程式の場合、truncation-errorが小さく、round-off-errorが誤差の中心を占めているためと考えられる。