お話：数値解析 第1回
収束の種類

長田直樹

1 はじめに

今日では、科学・技術にとどまらず、日常生活に密着する多くの分野でも、数値計算が重要な役割を果たしている。

数値計算とは、モデル化などにより得られた数学の問題の近似解 (数値解) を計算機によって求めることをいう。数値計算を支える数学的理論が数値解析である。

数値解析の研究や数値計算を行う上で役に立ちそうな数値解析の話題を、毎号読み切りで 1 年間連載する。

本連載のサポートページを

`http://www.cis.twcu.ac.jp/~osada/rikei/`

におく。数値例に使ったプログラムのいくつかは、このページで公開していく。

2 数値計算と数列

数値計算において、数列の極限を求めることに帰着させる場合がしばしばある。数列は、級数の部分和、反復法、離散化法、などにより得られる。表 1 に反復法の例、表 2 に離散化法の例を掲げる。離散化法では、刻み幅を規則的に小さくする (例えば $1, 1/2, 1/4, 1/8, \ldots$) ことにより、数列が得られる。

> 数列とは自然数全体の集合 $\mathbb{N}$ から集合 X への写像である。X が実数全体の集合 $\mathbb{R}$ のときは実数列、複素数全体の集合 $\mathbb{C}$ のときは複素数列、$\mathbb{R}^n$ のときは実ベクトル列などという。

近似値を数列の極限値として求めるアルゴリズムでは、何らかの条件を満たしたときに収束したものと見なし、計算を終了する。これらのアルゴリズムを比較する場合、収束の速さは重要な要素になる。

今回は収束の速さの分類とそれぞれの特徴を述べる。

表 1: 反復法の例

問題	方法
非線型方程式	ニュートン法
n 次代数方程式	ワイエルシュトラス法
連立 1 次方程式	ヤコビ法
固有ベクトル	ベキ乗法

表 2: 離散化法の例

問題	方法
微分	差分法
定積分	複合台形公式
常微分方程式の初期値問題	オイラー法

3 数列の収束の速さ

ここでは、数列 $\{x_\nu\}$ は実数列または複素数列とする。$\mathbb{R}^n$、$\mathbb{C}^n$、あるいはバナッハ空間の数列に対しても、絶対値を適当なノルムに置き換えることにより、ほぼ同様の議論を展開できる。

3.1 物理的速さとの対比

物理的な速さは、単位時間当りの移動距離で定義される。数列の収束の場合に当てはめると、移動距離に相当するのは得られた有効桁数 (誤差の絶対値の常用対数をとり符号を $-$ にしたもの)、単位時間は反復 1 回と考えられる。

近似値が x で真値が α のときの有効桁数は $-\log_{10}|x-\alpha|$ なので、x_ν における収束の速さを

$$\begin{aligned} v_\nu &= -\log_{10}|x_{\nu+1}-\alpha| + \log_{10}|x_\nu-\alpha| \\ &= -\log_{10}\left|\frac{x_{\nu+1}-\alpha}{x_\nu-\alpha}\right| \end{aligned} \tag{1}$$

により定義する [1, p.51]。

3.2　数列の収束の分類

数列 $\{x_\nu\}$ は α に収束し、誤差の比の極限値

$$\rho = \lim_{\nu\to\infty} \frac{x_{\nu+1}-\alpha}{x_\nu-\alpha} \tag{2}$$

が存在するとする。このとき、$|\rho| \le 1$ が成り立つ。[証明せよ。] $0 < |\rho| < 1$ のとき $\{x_\nu\}$ は α に収束率 (あるいは縮小率)ρ で**線型収束**するという。$\rho = 0$ のときは**超線型収束**、$\rho = 1$ のときは**対数収束**あるいは劣線型 (sublinear) 収束するという。

> 数値計算は多くの分野にまたがっており、用語も分野によって異なることがしばしばある。対数収束は収束の加速法、劣線型収束は主として非線型方程式の反復解法の分野で用いられている。
>
> 日本語として定着してない用語には、括弧書きで英語をつけてある。

数列 $\{x_\nu\}$ が収束率 ρ で線型収束するとき、収束の速さの極限は

$$\lim_{\nu\to\infty} v_\nu = -\log_{10}|\rho| = \log_{10}\frac{1}{|\rho|}$$

となる。したがって、ρ が 0 に近いとき収束は速く、1 に近いとき収束は遅い。

3.3　収束次数と漸近誤差定数

数列 $\{x_\nu\}$ が α に線型収束または超線型収束するものとする。実数 $p \ge 1$ と零でない極限値

$$C = \lim_{\nu\to\infty} \frac{|x_{\nu+1}-\alpha|}{|x_\nu-\alpha|^p} \tag{3}$$

が存在するとき、$\{x_\nu\}$ は α に $\boldsymbol{p}$ **次収束**するという。C は**漸近誤差定数**と呼ばれる。

数列 $\{x_\nu\}$ が α に p 次収束するとき、正数 M と自然数 N が存在して

$$|x_{\nu+1}-\alpha| \le M|x_\nu-\alpha|^p \quad (\nu \ge N) \tag{4}$$

となる。

収束次数 p の近似値は、連続する 3 項 $x_{\nu-1}, x_\nu, x_{\nu+1}$ から

$$\frac{\log_{10}|(x_{\nu+1}-\alpha)/(x_\nu-\alpha)|}{\log_{10}|(x_\nu-\alpha)/(x_{\nu-1}-\alpha)|} \tag{5}$$

により与えられ、**計算収束次数** (computational order of convergence) と呼ばれる。[(3) から (5) を導け。]

4　1 点反復列の収束

α に収束する数列 $\{x_\nu\}$ が、関数 $\phi(x)$ によって、$x_{\nu+1} = \phi(x_\nu)$ と表される場合を考える。$\phi(x)$ を **1 点反復関数** [3, p.8] という。

$\phi(x)$ が α で微分可能であれば

$$\lim_{\nu\to\infty} \frac{x_{\nu+1}-\alpha}{x_\nu-\alpha} = \phi'(\alpha)$$

となる。このとき、数列 $\{x_\nu\}$ が α に収束率 $\phi'(\alpha)$ で線型収束することと $0 < |\phi'(\alpha)| < 1$ は同値である。

定理 1 反復関数 $\phi(x)$ は α の近傍で C^p 級 (複素関数の場合は正則) であるとする。$x_{\nu+1} = \phi(x_\nu)$ によって生成される数列 $\{x_\nu\}$ に対し、次は同値である。

1. $\phi'(\alpha) = \ldots = \phi^{(p-1)}(\alpha) = 0,\ \phi^{(p)}(\alpha) \ne 0$
2. α に p 次収束し、漸近誤差定数が $|\phi^{(p)}(\alpha)|/p!$ となる。

証明テーラーの定理

$$\begin{aligned}\phi(x_\nu) &= \alpha + \sum_{k=1}^{p-1} \frac{\phi^{(k)}(\alpha)}{k!}(x_\nu-\alpha)^k + R_\nu \\ & \quad R_\nu \to 0 \quad (\nu\to\infty)\end{aligned}$$

より導ける。[証明を完成せよ。]　□

定理 1 より、1 点反復列の収束次数は自然数になる。

5　数値例

関数 $f(x) = (x-1)^2(x-2) = x^3 - 4x^2 + 5x - 2$ の零点 ($f(\alpha) = 0$ となる α を $f(x)$ の零点という。) を、収束の種類の異なる 4 通りの反復法で求める。$|f(x_\nu)| < 1.0 \times 10^{-15}$ となったときに反復を終了する。

コンパイラは gcc 4.1.2 を用い、倍精度 (10 進 16 桁弱) 計算を行った。非線型方程式の反復解法については、回を改めて説明する。

5.1　3 次収束の例

例 1 初期値を $x_0 = 2.1$ にとりハレー法

$$x_{\nu+1} = x_\nu - \frac{\dfrac{f(x_\nu)}{f'(x_\nu)}}{1 - \dfrac{f(x_\nu)f''(x_\nu)}{2(f'(x_\nu))^2}}, \quad \nu = 0, 1, \ldots \tag{6}$$

表 3: ハレー法：$x_0 = 2.1$

ν	x_ν	$f(x_\nu)$
0	2.10	1.2×10^{-1}
1	2.0021	2.1×10^{-3}
2	2.000000026	2.6×10^{-8}
3	2.000000000000000	0.0

表 4: ニュートン法：$x_0 = 2.1$

ν	x_ν	$f(x_\nu)$
0	2.10	0.12
1	2.015	0.016
2	2.00045	0.00045
3	2.00000041	4.1×10^{-7}
4	2.00000000000033	3.3×10^{-13}
5	2.000000000000000	0.0

で計算した結果を表 3 に示す。

収束は非常に速く x_3 が真値を与えている。倍精度で計算しているため、表 3 からは $x_{\nu+1} - 2$ と $x_\nu - 2$ の関係は読み取れない。

5.2　2 次収束の例

例 2 初期値を $x_0 = 2.1$ としたニュートン法

$$x_{\nu+1} = x_\nu - \frac{f(x_\nu)}{f'(x_\nu)}, \quad \nu = 0, 1, \ldots \tag{7}$$

による結果は表 4 である。

$$|x_{\nu+1} - 2| \leq 2|x_\nu - 2|^2, \quad \nu = 0, 1, 2, 3.$$

を満たしている。反復を 1 回繰り返すごとに有効桁数 ($= -\log_{10} |$ 誤差 $|$) が 2 倍になっている。

5.3　線型収束の例

例 3 初期値を $x_0 = 0.9$ としてニュートン法により計算した結果を表 5 に示す。

$$(x_{\nu+1} - 1) \fallingdotseq (1/2)(x_\nu - 1), \quad \nu = 0, \ldots, 20$$

となっているので線型収束である。反復を 1 回繰り返すごとに誤差が半分になっている。

表 5: ニュートン法：$x_0 = 0.9$

ν	x_ν	$f(x_\nu)$
0	0.90	-1.1×10^{-1}
1	0.95	-2.9×10^{-3}
2	0.97	-7.3×10^{-4}
	中略	
21	0.999999948	-2.7×10^{-15}
22	0.999999974	-6.8×10^{-16}

表 6: 簡易ニュートン法：$x_0 = 0.9$

ν	x_ν	$f(x_\nu)$
0	0.900	-1.1×10^{-2}
1	0.947	-2.8×10^{-3}
2	0.960	-1.6×10^{-3}
	中略	
1000000	0.99999977	-5.3×10^{-14}
2000000	0.99999989	-1.3×10^{-14}
4000000	0.999999943	-3.3×10^{-15}
	中略	
7273219	0.999999968	-9.9996×10^{-16}

5.4　対数収束の例

例 4 初期値を $x_0 = 0.9$, 係数 $A = 1/f'(x_0)$ として簡易ニュートン法

$$x_{\nu+1} = x_\nu - Af(x_\nu), \quad \nu = 0, 1, \ldots \tag{8}$$

を適用すると表 6 のようになる。

$|f(x_{7273219})| < 1.0 \times 10^{-15}$ となり $\nu = 7273219$ で反復が終了する。収束は極めて遅い。反復回数が 2 倍になると誤差が半分になっている。つまり誤差 $x_\nu - 1$ が $1/\nu$ の定数倍で近似できることが推察できる。

6　3 次収束

6.1　二宮市三先生

パソコンが普及する以前、数値計算はプログラム言語 (主として FORTRAN) でプログラムを書いたり、数値計算ライブラリを利用することによりなされていた。プログラム言語で利用する標準関数や数値計算ライブラリは数学ソフトウェアと呼ばれていた。

数学ソフトウエア開発の大御所に二宮市三先生 (名古屋大名誉教授) がおられる。名大在職中に NUMPAC という汎用数値計算ライブラリーを作られた方である。二宮先生は定年退職後、Borland C++ 上に IEEE754 規格を拡張した 4 倍精度と 8 倍精度のシステムを完成させた。([2, p.6] 参照)

毎年 6 月頃に数値解析シンポジウム (今年は、`http://nas2008.akita-pu.ac.jp/`) が開催される。数値計算の実務家や数値解析の研究者など 100 人以上が、二泊三日で泊まり込み熱心な討論を行う。本誌の読者にはお馴染の一松信先生も、常連の参加者である。

昨年 (2007 年) のシンポジウムで二宮先生と歓談のおり、先生が筆者に 8 倍精度システムを勧められたので、喜んで頂くことにした。数日後に圧縮されたシステムが郵送で届いた。インストールと動作確認だけして、そのままにしてあったのだが、本原稿の執筆に 8 倍精度システムが役立つことになった。

6.2 8 倍精度システム

二宮先生の 8 倍精度システムは、8 倍精度浮動小数点型 `octo` を符号なし整数 (32 ビット)8 個からなる配列 (32 バイト =256 ビット) により構成してある。符号に 1 ビット、指数に 15 ビット、仮数に 240 ビット (仮数部の先頭ビットが 1 になるように指数を決める) で、有効桁数は 10 進約 72 桁 ($\fallingdotseq \log_{10} 2^{240}$) である。

実務上のほとんどの数値計算は倍精度で十分であるが、3 次以上の高次収束の様子を見るときなどには、多倍長計算は重宝である。

6.3 再び 3 次収束の例

例 5 例 1 と例 2 の数列を 8 倍精度で計算すると表 7 のようになる。同じ数字 n が d 個続くとき、n_d と書くことにする [3, p.271]。例えば、2.00045 を 2.0_345 と書く。本例では、$f'(2) = 1$ なので、$f(x_\nu) \fallingdotseq x_\nu - 2$ となるため、関数値 $f(x_\nu)$ は省略した。

x_2, x_3, x_4 によるニュートン法の計算収束次数は、2.0044、ハレー法は 3.00043 である。

表 7 よりハレー法では

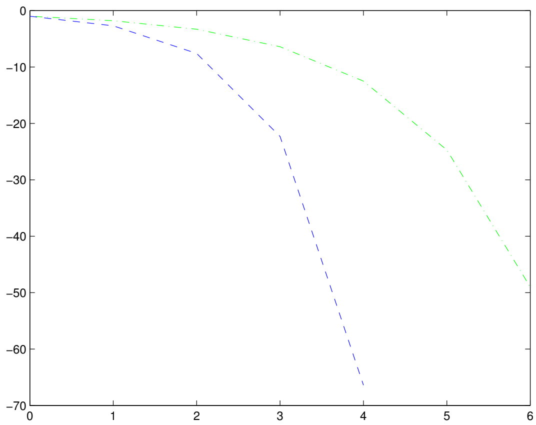
$$|x_{\nu+1} - 2| \leq 3.1|x_{\nu+1} - 2|^3 \quad \nu = 0, 1, 2, 3$$

表 7: 8 倍精度計算：$x_0 = 2.1$

	ニュートン法	ハレー法
ν	x_ν	x_ν
0	2.10	2.10
1	2.015	2.0021
2	$2.0_{3}45$	$2.0_{7}26$
3	$2.0_{6}41$	$2.0_{22}52$
4	$2.0_{12}33$	$2.0_{66}41$
5	$2.0_{24}22$	
6	$2.0_{48}10$	

となることが見て取れる。

x 軸に反復回数、y 軸に $\log_{10}|x_\nu - 2|$ をとり、折れ線で結んだグラフを示す。ハレー法は破線、ニュートン法は一点鎖線で表している。



8 倍精度での計算では、2 次収束 (ニュートン法) と 3 次収束 (ハレー法) の違いがよく分かる。

参考文献

[1] C. Brezinski and R. Zaglia, Extrapolation methods theory and practice, North-Holland, 1991.

[2] 二宮市三編、数値計算のわざ、共立出版、2006

[3] J.F. Traub, Iterative methods for the solution of equations, Chelsea, 1964

(おさだなおき/東京女子大学)