

数值解析

2. 常微分方程式

$$\frac{dx}{dt} = f(t, x(t))$$

石渡哲哉 (芝浦工大数理)

常微分方程式の初期値問題 (IVP):

$$\frac{dx}{dt} = f(t, x(t)), \quad t > a, \quad (1)$$

$$x(a) = x_0 \quad (2)$$

の近似計算法を学び、誤差解析等について理解する。

対象:

- 上記のような 1 階の正規形 ODE に対する方法を紹介するが、1 階の連立系にしてもまったく同じように適用可能である。スカラー関数 x, f をベクトル値の関数に読み替えるだけで良い。また、高階 ODE は、1 階連立系に書きなおすことができることにも注意。

- f は十分滑らかとし、解の存在と一意性は保証されているとする。
また、このとき解 x も f の滑らかさに応じて十分に滑らかであることにも注意。

代表的な方法

- オイラー法 (Euler method)
- ホイン法 (Heun method), 中点法 (改良オイラー法)
- ルンゲ・クッタ法 (Runge-Kutta method)

2.1 オイラー法 (Euler method)

$$\frac{dx}{dt} = f(t, x(t))$$

の左辺を次で近似 ($h > 0$ とする) :

$$\frac{x(t+h) - x(t)}{h} \longleftarrow \frac{dx}{dt} = \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h}$$

式を整理すると,

$$x(t+h) = x(t) + h \cdot f(t, x(t))$$

となり, $x(t)$ の値から $x(t+h)$ の値を逐次的に計算することができる.

$t_n := a + nh$ とし, $x(t_n)$ の近似値を X_n と書くことにする.

(IVP) を次のように離散化する方法をオイラー法という.

オイラー法のスキーム

$$\begin{cases} X_{n+1} = X_n + h \cdot f(t_n, X_n) & (n = 0, 1, 2, \dots) \\ X_0 = x_0 \end{cases}$$

$h > 0$ はステップサイズ、刻み幅、離散化パラメータなどと呼ばれる.

近似の妥当性

Q $h \rightarrow 0$ としたとき, X_n は元の解 $x(t)$ に収束するだろうか? *1

*1 より詳しく, (t_n, X_n) をつないでできた折れ線が $x(t)$ の解曲線に収束するか, という観点もあり.

解の収束性について (誤差解析)

閉区間 $[a, b]$ ($T := b - a > 0$) 上で解 $x(t)$ が存在しているとする.

$h = T/N$ ($N \in \mathbb{R}$) とし, この区間で離散化誤差 $e_n := x(t_n) - X_n$ ($n = 0, 1, 2, \dots, N$) を調べる。

定理 (誤差解析)

離散化誤差に対して以下が成り立つ:

h や N に依存しない定数 $C > 0$ が存在して

$$\max_{0 \leq n \leq N} |e_n| \leq Ch.$$

Remark:

- 上の結果をランダウの記号を使って $|e_n| = O(h)$ と書くことも

ある.

- $h \searrow 0$ のとき, 誤差が h の 1 次のオーダーで 0 に収束する.
- 今回の誤差解析は離散化誤差のみ考慮している. 丸め誤差の影響についても考察する必要がある場合は, 初期値の設定, 各ステップの計算時に丸め誤差が混入するとして解析を行う.

オイラー法はシンプルであり、プログラミングも容易であるが、収束が遅く、実用上はあまり使われない。

次節以降において、オイラー法の改良や、ODE の数値計算でよく使われているルンゲ・クッタ法について紹介する。

2.2 次数と収束

以下の 1 段階法のスキームを考える:

$$X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots, N - 1) \quad (3)$$

$x(t)$ を真の解として、

$$x(t_{n+1}) = x(t_n) + h \cdot \Phi(t_n, x(t_n)) + O(h^{k+1})$$

を満たすとき、スキーム (3) は k 次の / 次数 k の (精度をもつ) 離散化である, という.

このとき, 以下の収束定理が成り立つ:

収束定理

Φ はリップシッツ連続とする. 1 段階法のスキーム (3) が k 次精度をもつとすると, 以下が成立する:

$$\max_{0 \leq n \leq N} |e_n| \leq O(h^k).$$

上の定理より, 近似の精度を上げれば収束も速くなることが分かる.

よって, 比較的シンプルなスキームで*2, なるべく高精度のものを作りたい, ということになる.

*2 計算の手間が少なければ, それだけ丸め誤差の混入の機会が減る.

2.3 オイラー法の改良

ODE $x'(t) = f(t, x(t))$ を t_n から t_{n+1} まで積分すると、

$$x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} f(s, x(s)) ds$$

i.e.

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(s, x(s)) ds$$

オイラー法などの1段階法: $X_{n+1} = X_n + h \cdot \Phi(t_n, X_n)$ では、積分 $\int_{t_n}^{t_{n+1}} f(s, x(s)) ds$ を $h \cdot \Phi(t_n, X_n)$ でどう近似するかということに相当する。

オイラー法では、この積分を1点 $t = t_n$ での値を利用して、高さ $f(t_n, X_n)$ 、幅 h の長方形で近似している。(区分求積)

この近似を、よりよいものに変える。

代表的なものを2つ紹介： ホイン法、中点法（修正オイラー法）

ホイン法： 積分を台形で近似：

$$\begin{aligned}\int_{t_n}^{t_{n+1}} f(s, x(s)) ds &\approx \frac{h}{2} (f(t_n, x(t_n)) + f(t_{n+1}, x(t_{n+1}))) \\ &\approx \frac{h}{2} (f(t_n, X_n) + f(t_{n+1}, X_{n+1}))\end{aligned}$$

ただし、今 X_{n+1} の値は未知なので、これをオイラー法を利用して近似値 $\tilde{X}_{n+1} = X_n + hf(t_n, X_n)$ で代用する。

ホイン法スキーム

$$\left\{ \begin{array}{l} X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots) \\ \text{where } \Phi(t_n, X_n) = \frac{1}{2}(k_1 + k_2) \\ \quad k_1 = f(t_n, X_n) \\ \quad k_2 = f(t_{n+1}, X_n + hk_1) \\ \\ X_0 = x_0 \end{array} \right.$$

中点法: 積分を区間の midpoint での値を用いた長方形で近似:

$$\int_{t_n}^{t_{n+1}} f(s, x(s)) ds \approx hf\left(t_n + \frac{h}{2}, x\left(t_n + \frac{h}{2}\right)\right) \approx hf\left(t_n + \frac{h}{2}, X_{n+1/2}\right)$$

ただし、今 $X_{n+1/2}$ の値は未知なので、これを刻み幅を $h/2$ としたオイラー法を利用して近似値 $\tilde{X}_{n+1/2} = X_n + \frac{h}{2}f(t_n, X_n)$ で代用する。

中点法のスキーム

$$\left\{ \begin{array}{l} X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots) \\ \text{where } \Phi(t_n, X_n) = k_2 \\ \quad k_1 = f(t_n, X_n) \\ \quad k_2 = f\left(t_n + \frac{h}{2}, X_n + \frac{h}{2}k_1\right) \\ X_0 = x_0 \end{array} \right.$$

以上の2つの方法は2次精度をもつ1段階法となる。(各自)

⇒ 先の定理より2次収束することも分かる。

これらの方法のように、 $[t_n, t_{n+1}]$ の複数の点を用い、それらの点での f の値を (何らかの意味での) 平均をとることで積分の近似精度を上げることができることがある。この方向での高次化の方法にルンゲ・クッタ法が知られている。(これまで紹介した方法は、すべてこの方法の一種となっている。)

2.4 ルンゲ・クッタ法 (Runge-Kutta method)

s 段のルンゲ・クッタ法

$$\left\{ \begin{array}{l} X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots) \\ \text{where } \Phi(t_n, X_n) = b_1 k_1 + b_2 k_2 + \dots + b_s k_s \\ k_1 = f(t_n, X_n) \quad (c_1 = 0, a_{11} = 0) \\ k_2 = f(t_n + c_2 h, X_n + h a_{21} k_1) \\ k_3 = f(t_n + c_3 h, X_n + h(a_{31} k_1 + a_{32} k_2)) \\ \vdots \\ k_s = f(t_n + c_s h, X_n + h(a_{s1} k_1 + a_{s2} k_2 + \\ \dots + a_{s,s-1} k_{s-1})) \\ X_0 = x_0 \end{array} \right.$$

a_{ij}, b_i, c_i : ルンゲ・クッタ法の係数パラメータ

Remarks:

- オイラー法は $s = 1$, ホイン法は $s = 2, a_{21} = 1, b_1 = b_2 = 1/2, c_2 = 1$ のルンゲクッタ法とみることができる.
- $s = 1, 2, 3, 4$ に対しては、係数パラメータをうまく選ぶことで s 段 s 次のルンゲクッタ法のスキームを作ることができる.
- $s \geq 5$ では、 s 段 s 次のルンゲクッタ法のスキームは構成できないことが示されている. ^{*3}

^{*3} 陰的ルンゲ・クッタ法であれば可能.

実用上は、次の4段のルンゲ・クッタ法のスキームがよく使われている。

$$s = 4, a_{21} = a_{32} = 1/2, a_{31} = a_{41} = a_{42} = 0, a_{43} = 1$$

$$b_1 = b_4 = 1/6, b_2 = b_3 = 1/3$$

$$c_2 = c_3 = 1/2, c_4 = 1$$

この設定にすると、4次精度を持つことを示すことができる。

4段4次RK法のスキーム

$$\left\{ \begin{array}{l} X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots) \\ \text{where } \Phi(t_n, X_n) = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(t_n, X_n) \\ k_2 = f(t_n + \frac{1}{2}h, X_n + \frac{1}{2}hk_1) \\ k_3 = f(t_n + \frac{1}{2}h, X_n + \frac{1}{2}hk_2) \\ k_4 = f(t_n + h, X_n + hk_3) \\ X_0 = x_0 \end{array} \right.$$

【参考】3段3次RK法のスキーム

$$\left\{ \begin{array}{l} X_{n+1} = X_n + h \cdot \Phi(t_n, X_n) \quad (n = 0, 1, 2, \dots) \\ \text{where } \Phi(t_n, X_n) = \frac{1}{6}(k_1 + 4k_2 + k_3) \\ \quad k_1 = f(t_n, X_n) \\ \quad k_2 = f(t_n + \frac{1}{2}h, X_n + \frac{1}{2}hk_1) \\ \quad k_3 = f(t_n + h, X_n - hk_1 + 2hk_2) \\ \\ X_0 = x_0 \end{array} \right.$$