

ベキ級数演算と常微分方程式の精度保証

柏木 雅英

kashi@waseda.jp

<http://verifiedby.me/>

早稲田大学 基幹理工学部 応用数理学科

常微分方程式の精度保証

大きく分けて **2 種類の方法**がある。

Taylor 展開を利用する方法

初期値問題を Taylor 展開をもとに精度保証付きで解く方法。境界値問題等は射撃法と非線形方程式の精度保証法を組み合わせで解く。

関数解析的方法

境界条件を満たす適当な関数空間を設定し、解全体を関数方程式と見て何らかの不動点定理を用いて解の存在保証と誤差評価を行う。

Taylor 展開ベースの方法について解説する。

初期値問題の精度保証

一階連立常微分方程式

$$\begin{aligned}\frac{dx}{dt} &= f(x, t), \quad x \in \mathbb{R}^l, t \in \mathbb{R} \\ x(t_0) &= x_0\end{aligned}$$

両辺を積分して Picard 型の不動点形式に

$$\boldsymbol{x}(t) = x_0 + \int_{t_0}^t f(\boldsymbol{x}(t), t) dt$$

Picard-Lindelöf の定理 (1)

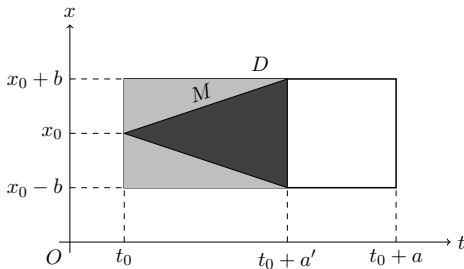
Picard-Lindelöf の定理

$$\begin{aligned}\frac{dx}{dt} &= f(x, t), \quad x \in \mathbb{R}^l, t \in \mathbb{R} \\ x(t_0) &= x_0\end{aligned}$$

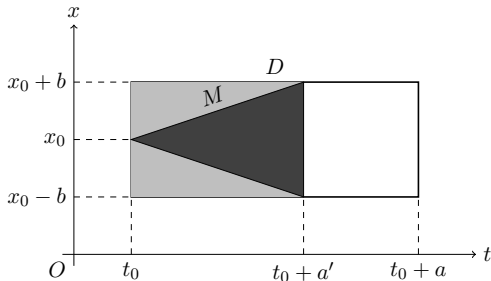
について、 f がある領域 $D = \{ (x, t) \mid t_0 \leq t \leq t_0 + a, \|x - x_0\| \leq b \}$ で Lipschitz 条件

$$\|f(x_1, t) - f(x_2, t)\| \leq L\|x_1 - x_2\|$$

を満たすとする。このとき、 $M = \sup_{(x,t) \in D} \|f(x, t)\|$ 、 $a' = \min(a, \frac{b}{M})$ とすると、 $[t_0, t_0 + a']$ でただ一つの解を持つ。



Picard-Lindelöf の定理 (2)



Picard-Lindelöf の定理の証明の概略

$$P : x(t) \mapsto x_0 + \int_{t_0}^t f(x(t), t) dt$$

とし、 $D' = \{ (x, t) \mid t_0 \leq t \leq t_0 + a', \|x - x_0\| \leq b \}$ とすると、

- P は D' 内の全ての連続関数を D' 内に移す。
- P は重み付き最大値ノルム $\|x(t)\| = \sup_{t_0 \leq t \leq t_0+a'} e^{-2L(t-t_0)} \|x(t)\|$ を入れると縮小写像となる。
- よって、 P の不動点は D' に唯一存在する。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}$$

$$P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \dots$$

解に近づいていく。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}$$

$$P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \cdots$$

解に近づいていく。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}$$

$$P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \dots$$

解に近づいていく。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}$$

$$P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \dots$$

解に近づいていく。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}\qquad P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \cdots$$

解に近づいていく。

Picard 型の不動点形式による近似解の生成

例えば、次の問題を Picard 型に直して、

$$\begin{aligned}\frac{dx}{dt} &= -x^2 \\ x(0) &= 1, \quad t \in [0, 0.1]\end{aligned}$$

$$P : x(t) \mapsto 1 + \int_0^t -x(t)^2 dt$$

初期値を定数関数 $x_0(t) = 1$ として $x_{i+1}(t) = P(x_i(t))$ のように反復すると、

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \dots$$

解に近づいていく。解の Taylor 展開が 1 次ずつ得られる。

Picard 型の不動点形式による解の包含

2 次の Taylor 展開

$$X_2 = 1 - t + t^2$$

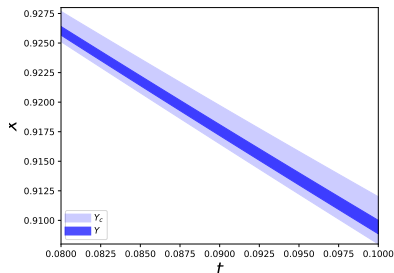
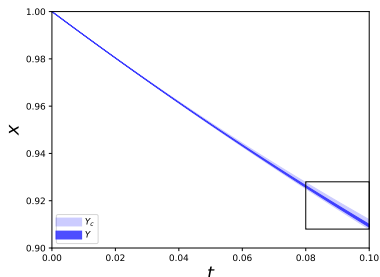
の最終項の係数を膨らませた区間関数

$$Y_c = 1 - t + [0.8, 1.2]t^2$$

に対して、

$$P(Y_c) \subset Y = 1 - t + [0.886, 1]t^2$$

となる。最終項の係数の比較で $Y \subset Y_c$ なので、 Y に解が存在する。



ベキ級数演算の導入

PSA = Power Series Arithmetic

Type-I PSA

$$x_0(t) = 1$$

$$x_1(t) = 1 - t$$

$$x_2(t) = 1 - t + t^2 - \frac{1}{3}t^3$$

$$x_3(t) = 1 - t + t^2 - t^3 + \frac{2}{3}t^4 - \frac{1}{3}t^5 + \frac{1}{9}t^6 - \frac{1}{63}t^7$$

$$x_4(t) = \dots$$

の、Taylor 展開の級数を決めるのに不要な部分を計算したくない!

Type-II PSA

$$Y_c = 1 - t + [0.8, 1.2]t^2$$

に対する包含

$$P(Y_c) \subset Y = 1 - t + [0.886, 1]t^2$$

を計算したい!

ベキ級数演算 (psa)

- 常微分方程式の初期値問題や数値積分の精度保証に使う。高階微分の計算にも使える。
- Type-I と Type-II の 2 種類がある。 n を整数として、
Type-I PSA 単に n 次より高次の項を捨てる。
Type-II PSA n 次より高次の項の影響を最高次の項 t^n の区間係数に含めさせる。

PSA の例 (積)

$1 + 2t - 3t^2$ と $1 - t + t^2$ の積	
Type-I PSA	Type-II PSA
定義域は決めなくてよい	定義域 = $[0, 0.1]$
$1 + t - 4t^2$	$1 + t + [-4, -3.5]t^2$

$$\begin{aligned}(1 + 2t - 3t^2)(1 - t + t^2) &= 1 + t - 4t^2 + 5t^3 - 3t^4 \\ &= 1 + t + (-4 + 5t - 3t^2)t^2 \in 1 + t + [-4, -3.5]t^2\end{aligned}$$

ベキ級数型

$$x_0 + x_1t + x_2t^2 + \cdots + x_nt^n$$

- ベキ級数型同士の加減乗除。
- ベキ級数型に対する $\exp$, $\log$, $\int$ などの数学関数。
- 演算結果の n 次までの項を残し、 $n+1$ 次以降は切り捨てる。
- 以下とほとんど同じ：
 - Mathematica の 'Series'。
 - Intlab の 'taylor'。
 - 1 変数関数の高階微分を計算する自動微分法。

Type-I PSA の演算規則 (1/4)

$$x(t) = x_0 + x_1t + x_2t^2 + \cdots + x_nt^n$$

$$y(t) = y_0 + y_1t + y_2t^2 + \cdots + y_nt^n$$

加減算

$$x(t) \pm y(t) = (x_0 \pm y_0) + (x_1 \pm y_1)t + \cdots (x_n \pm y_n)t^n$$

加算の例

$$x(t) = 1 + 2t - 3t^2$$

$$y(t) = 1 - t + t^2$$

$$x(t) + y(t) = 2 + t - 2t^2$$

Type-I PSA の演算規則 (2/4)

乗算

$$x(t) \times y(t) = z_0 + z_1 t + \cdots + z_n t^n$$

$$z_k = \sum_{i=0}^n x_i y_{k-i}$$

(n 次までで止め、 $n+1$ 次以降の項は計算しない。)

乗算の例

$$x(t) = 1 + 2t - 3t^2$$

$$y(t) = 1 - t + t^2$$

$$x(t) \times y(t) = 1 + t - 4t^2 + 5t^3 - 3t^4$$

を 2 次の項までで打ち切った、

$$x(t) \times y(t) = 1 + t - 4t^2$$

を計算結果とする。

Type-I PSA の演算規則 (3/4)

sin などの数学関数

その関数を g として、

$$\begin{aligned} & g(x_0 + x_1 t + \cdots + x_n t^n) \\ &= g(x_0) + \sum_{i=1}^n \frac{1}{i!} g^{(i)}(x_0) (x_1 t + \cdots + x_n t^n)^i \end{aligned}$$

のように g の点 x_0 での **Taylor 展開** に代入することによって得る。この計算中に現れる加算や乗算は **Type-I PSA** で行う。

Type-I PSA の演算規則 (4/4)

除算

$x \div y = x \times (1/y)$ と乗算と逆数関数に分解

不定積分

$$\int_0^t x(t)dt = x_0t + \frac{x_1}{2}t^2 + \cdots \frac{x_n}{n+1}t^{n+1}$$

ベキ級数型

$$x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n$$

- 固定された有限閉区間 $D = [t_1, t_2]$ 上で定義される。
- 演算結果は n 次までしか保持しないが、 $n + 1$ 次以降の項の影響は n 次の項の係数を区間にすることで吸収する。
- 係数 $x_0, \cdots, x_n$ は区間。
- ただし多くの場合、 $x_0, \cdots, x_{n-1}$ は幅の狭い区間、 x_n は幅の広い区間。

Type-II PSA の演算規則 (1/5)

$$x(t) = x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n$$

$$y(t) = y_0 + y_1 t + y_2 t^2 + \cdots + y_n t^n$$

加減算

$$x(t) \pm y(t) = (x_0 \pm y_0) + (x_1 \pm y_1)t + \cdots + (x_n \pm y_n)t^n$$

加算の例

$$x(t) = 1 + 2t - 3t^2$$

$$y(t) = 1 - t + t^2$$

$$x(t) + y(t) = 2 + t - 2t^2$$

(加減算は Type-I PSA とまったく同じ)

Type-II PSA の演算規則 (2/5)

乗算

- ① まず、打ち切り無しで乗算を行う。

$$x(t) \times y(t) = z_0 + z_1 t + \cdots + z_{2n} t^{2n}$$

$$z_k = \sum_{i=\max(0, k-n)}^{\min(k, n)} x_i y_{k-i}$$

- ② $2n$ 次から n 次に減次する。

m 次から n 次への減次

$$x_0 + x_1 t + x_2 t^2 + \cdots + x_m t^m \implies z_0 + z_1 t + \cdots + z_n t^n$$

$$z_i = x_i \quad (0 \leq i \leq n-1)$$

$$z_n = \left\{ \sum_{i=n}^m x_i t^{i-n} \mid t \in D \right\}$$

乗算の例

定義域を $D = [0, 0.1]$ とする。

$$x(t) = 1 + 2t - 3t^2$$

$$y(t) = 1 - t + t^2$$

$$\begin{aligned}x(t) \times y(t) &= 1 + t - 4t^2 + 5t^3 - 3t^4 \\&= 1 + t + (-4 + 5t - 3t^2)t^2 \\&\in 1 + t + \{-4 + 5t - 3t^2 \mid t \in [0, 0.1]\} t^2 \\&= 1 + t + [-4, -3.5]t^2\end{aligned}$$

Type-II PSA の演算規則 (4/5)

sin などの数学関数

その関数を g として、

$$\begin{aligned} & g(x_0 + x_1 t + \cdots + x_n t^n) \\ &= g(x_0) + \sum_{i=1}^{n-1} \frac{1}{i!} g^{(i)}(x_0) (x_1 t + \cdots + x_n t^n)^i \\ &+ \frac{1}{n!} g^{(n)} \left(\text{hull} \left(x_0, \left\{ \sum_{i=0}^n x_i t^i \mid t \in D \right\} \right) \right) (x_1 t + \cdots + x_n t^n)^n \end{aligned}$$

のように g の点 x_0 での剰余項付きの Taylor 展開に代入することによって得る。この計算中に現れる加算や乗算は Type-II PSA で行う。

Type-II PSA の演算規則 (5/5)

除算

$x \div y = x \times (1/y)$ と乗算と逆数関数に分解

不定積分

$$\int_0^t x(t)dt = x_0t + \frac{x_1}{2}t^2 + \cdots \frac{x_n}{n+1}t^{n+1}$$

常微分方程式の初期値問題

一階連立常微分方程式

$$\begin{aligned}\frac{dx}{dt} &= f(x, t), \quad x \in \mathbb{R}^l, t \in \mathbb{R} \\ x(t_0) &= x_0\end{aligned}$$

初期値問題の精度保証アルゴリズム

$t_0 < t_1 < t_2 < \dots$ に対して、

- **ベキ級数演算 (PSA)** を用いた、 $x(t_i)$ を元に $x(t_{i+1})$ を精度保証付きで計算する方法 (**短い区間**での精度保証)
- **Affine Arithmetic** を用いて区間幅の膨らみを抑制しながら、短い区間での精度保証で得られた解を**長い区間**に渡って接続する方法

短い区間での精度保証 (1/3)

$v = x(t_s)$ の値を初期値として、 $x(t_e)$ の値を計算する。

平行移動 & 両辺を積分で不動点形式に

$$\begin{aligned} x(t) &= v + \int_0^t f(x(t), t + t_s) dt \\ (v &= x(t_i), \quad t \in [0, t_e - t_s]) \end{aligned}$$

解の Taylor 展開の生成

Type-I PSA 型の変数 $X_0 = v$, $T = t$ を用いて、 $k = 0$ とし、

① 次数 k の Type-I PSA で以下を計算

$$X_{k+1} = v + \int_0^t f(X_k, T + t_s) dt$$

② 次数 $k = k + 1$ とする。

を n 回繰り返すと、 X_n として解の n 次の Taylor 展開が得られる。

短い区間での精度保証 (2/3)

解の存在保証

Type-II PSA の定義域を $D = [0, t_e - t_s]$ と設定し、Type-I PSA の反復で得られた n 次の Taylor 近似

$$X_n = x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n$$

と $T = t$ を用いて、

- ① X_n の最終項の係数を膨らませた候補者集合

$$Y_c = x_0 + x_1 t + x_2 t^2 + \cdots + V_c t^n$$

を作成する。

- ② $v + \int_0^t f(Y_c, T + t_s) dt$ を次数 n の Type-II PSA で計算し、 $n + 1$ 次から n 次に減次したものを

$$Y = x_0 + x_1 t + x_2 t^2 + \cdots + V t^n$$

とする。 $n - 1$ 次までの係数は X_n と全く同じになることに注意。

- ③ $V \subset V_c$ なら Y 内に解の存在が保証される。

短い区間での精度保証 (3/3)

解を含みそうな候補者集合の作成は、例えば次のように行なうことが出来る。

候補者集合の作成

- ① $v + \int_0^t f(X_n, T + t_s) dt$ を次数 n の Type-II PSA で計算し、 $n + 1$ 次から n 次に減次したものを $Y_0 = x_0 + x_1 t + \cdots + V_0 t^n$ とする。
- ② $r = \|V_0 - x_n\|$ とし、

$$V_c = x_n + 2r ([-1, 1], \dots, [-1, 1])^T$$

とする。(半径を **Newton 法の修正量の 2 倍**にする。)

短い区間での精度保証の例

$$\frac{dx}{dt} = -x^2$$

$$x(0) = 1, \quad t \in [0, 0.1]$$

展開の次数: $n = 2$

10 進 3 桁演算。

(Type-I PSA による Taylor 展開の生成)

$$X_0 = \boxed{1}$$

$$X_1 = 1 + \int_0^t (-X_0^2) dt$$

$$= 1 + \int_0^t (-1) dt$$

$$= \boxed{1 - t}$$

$$X_2 = 1 + \int_0^t (-X_1^2) dt$$

$$= 1 + \int_0^t (-(1 - t)^2) dt$$

$$= 1 + \int_0^t (-(1 - 2t)) dt$$

$$= \boxed{1 - t + t^2}$$

(候補者集合の生成)

$$1 + \int_0^t (-X_2^2) dt$$

$$= 1 + \int_0^t (-(1 - t + t^2)^2) dt$$

$$= 1 + \int_0^t (-(1 - 2t + [2.8, 3]t^2)) dt$$

$$= 1 - t + t^2 + [-1, -0.933]t^3$$

2 次に減次して、

$$Y_0 = 1 - t + [0.9, 1]t^2$$

$$r = |[0.9, 1] - 1| = 0.1 \text{ なので、}$$

$$Y_c = \boxed{1 - t + [0.8, 1.2]t^2}$$

(Type-II PSA による精度保証)

$$1 + \int_0^t (-Y_c^2) dt$$

$$= 1 - t + t^2 + [-1.14, -0.786]t^3$$

2 次に減次して、

$$Y = \boxed{1 - t + [0.886, 1]t^2}$$

$[0.886, 1] \subset [0.8, 1.2]$ なので、 Y 内に真の解が存在する。

Lohner 法 (1/3)

(以下は未定係数法など、任意の別の方法でもよい。)

平行移動 & 両辺を積分で不動点形式に

$$\begin{aligned}x(t) &= v + \int_0^t f(x(t), t + t_i) dt \\(v &= x(t_s), \quad t \in [0, t_e - t_s])\end{aligned}$$

解の Taylor 展開の生成

Type-I PSA 型の変数 $X_0 = v$, $T = t$ を用いて、 $k = 0$ とし、

① 次数 k の Type-I PSA で以下を計算

$$X_{k+1} = v + \int_0^t f(X_k, T + t_s) dt$$

② 次数 $k = k + 1$ とする。

を n 回繰り返すと、 X_n として解の n 次の Taylor 展開が得られる。求まった X_n の t^i の係数を α_i とする。

$$X_n = v + \alpha_1 t + \alpha_2 t^2 + \cdots + \alpha_n t^n$$

大雑把な解の包含 V

$[0, t_e - t_s]$ における解 $x(t)$ を包含する候補者区間 $V_c \subset \mathbb{R}^l$ を考える。

$$\begin{aligned} P(V_c) &\subset v + \int_0^t f(V_c, [0, t_e - t_s] + t_i) dt \\ &\subset v + f(V_c, [t_s, t_e])t \\ &\subset v + f(V_c, [t_s, t_e])[0, t_e - t_s] \end{aligned}$$

により、 $V = v + f(V_c, [t_e, t_s])[0, t_e - t_s] \subset V_c$ が成立すれば V 内に $x(t)$ が包含される。

反復改良

$V_1 = V, k = 1$ とし、

$$V_{k+1} = V_k \cap (v + f(V_k, [t_e, t_s])[0, t_e - t_s])$$

で更に精度を上げることも出来る。

区間初期値による解の Taylor 展開の生成

初期値を v の代わりに V 、初期時刻を t_s の代わりに $[t_s, t_e]$ として、再度 Taylor 展開の生成を行う。すなわち、Type-I PSA 型の変数 $X_0 = V$, $T = t$ を用いて、 $k = 0$ とし、

- ① 次数 k の Type-I PSA で以下を計算

$$X_{k+1} = V + \int_0^t f(X_k, T + [t_s, t_e]) dt$$

- ② 次数 $k = k + 1$ とする。

を n 回繰り返すと、 X_n として解の n 次の Taylor 展開が得られる。求まった X_n の t^i の係数を β_i とする。

$$X_n = V + \beta_1 t + \beta_2 t^2 + \cdots + \beta_n t^n$$

精度保証解

このとき、

$$v + \alpha_1 t + \alpha_2 t^2 + \cdots + \alpha_{n-1} t^{n-1} + \beta_n t^n$$

に真の解が存在する。

短い区間での Lohner 法の例

$$\frac{dx}{dt} = -x^2$$

$$x(0) = 1, \quad t \in [0, 0.1]$$

展開の次数: $n = 2$

10 進 3 桁演算。

(Type-I PSA による Taylor 展開の生成) (大雑把な解の包含の生成)

$$X_0 = \boxed{1}$$

$$X_1 = 1 + \int_0^t (-X_0^2) dt$$

$$= 1 + \int_0^t (-1) dt$$

$$= \boxed{1 - t}$$

$$X_2 = 1 + \int_0^t (-X_1^2) dt$$

$$= 1 + \int_0^t (-(1 - t)^2) dt$$

$$= 1 + \int_0^t (-(1 - 2t)) dt$$

$$= \boxed{1 - t + t^2}$$

$$r = ||\text{mag}((-1^2)[0, 0.1])|| = 0.1$$

$$V_c = 1 + 2 \times 0.1 \times [-1, 1] = \boxed{[0.8, 1.2]} \quad X_0 = [0.856, 1]$$

$$1 + (-[0.8, 1.2]^2)[0, 0.1] = \boxed{[0.856, 1]}$$

$$X_1 = [0.856, 1] + [-1, -0.732]t$$

$$X_2 = \boxed{[0.856, 1] + [-1, -0.732]t + [0.627, 1]t^2}$$

$[0.856, 1] \subset [0.8, 1.2]$ なので

$[0.856, 1]$ 内に真の解が存在する。

(解の包含の生成)

初期値を $[0.856, 1]$ として Taylor 展開を行う。

初期値を 1 とした Taylor 展開と合成した、

$$\boxed{1 - t + [0.627, 1]t^2}$$

内に、真の解が存在する。

長い区間に渡って接続する (1/2)

flow map

常微分方程式において、値 $x(t_s)$ ($t = t_s$ における初期値) に対して値 $x(t_e)$ (時刻 $t = t_e$ における解の値) を対応させる写像

$$\phi_{t_s, t_e} : \mathbb{R}^l \rightarrow \mathbb{R}^l, \quad \phi_{t_s, t_e} : x(t_s) \mapsto x(t_e)$$

を flow map と呼ぶ。

初期値に関する変分方程式

$x^*(t)$ を v を初期値とした常微分方程式の解とすると、

$$\begin{aligned} \frac{d}{dt}y(t) &= f_x(x^*(t), t)y(t), \quad y \in \mathbb{R}^{l \times l} \\ y(t_s) &= I, \quad t \in [t_s, t_e] \end{aligned}$$

を解くことによって、**flow map の微分 (ヤコビ行列)** を $\phi'_{t_s, t_e}(v) = y(t_e)$ で得られる。

長い区間に渡って接続する (2/2)

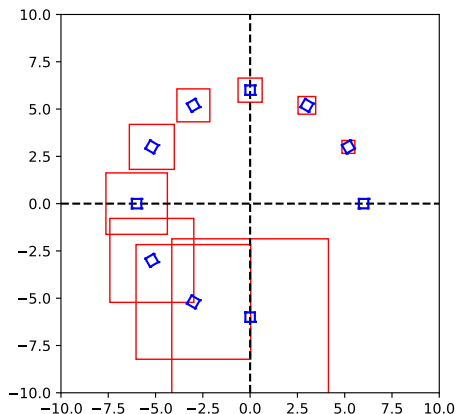
時刻 $t_0 < t_1 < t_2 < \dots$ における解の包含を J_i , $c_i = \text{mid}(J_i)$ とする。

平均値形式

$$\phi_{t_i, t_{i+1}}(x) \in \phi_{t_i, t_{i+1}}(c_i) + \phi'_{t_i, t_{i+1}}(J_i)(x - c_i)$$

- 単純にこのまま計算すると、wrapping effect によって区間幅が激しく増大する。
- wrapping effect を抑えるため、**affine arithmetic** を使って接続する

Wrapping Effect



成分が区間であるような「区間ベクトル」が $\mathbb{R}^n$ の超直方体領域しか表現できないため、真の像が直方体で包含されることにより区間幅が増大する現象。

区間 $I_0 = \begin{pmatrix} [5.75, 6.25] \\ [-0.25, 0.25] \end{pmatrix}$ に対

して $I_{k+1} = \begin{pmatrix} \sqrt{3}/2 & -0.5 \\ 0.5 & \sqrt{3}/2 \end{pmatrix} I_k$

のように行列を作用させた例。

青が真の像、赤が区間演算の結果。

Affine Arithmetic とは

- 区間演算の過大評価を抑制できる。その代わり計算時間がかかる。
- 全ての変数について、入力変数またはノイズに関する依存性を保持するため、区間幅の爆発を防げる。

- 全ての数値は

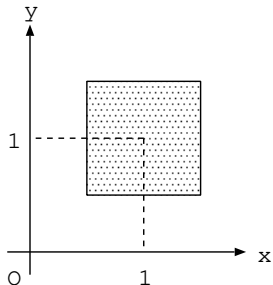
$$x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \cdots + x_n\varepsilon_n$$

のような **Affine 形式** で表現される。 ε_i は $-1 \leq \varepsilon_i \leq 1$ を動く **ダミー変数** であり、その係数により依存性を表現する。

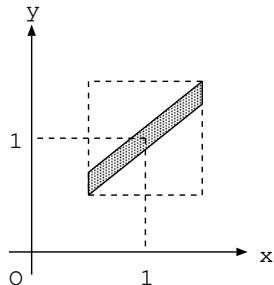
- 乗除算や数学関数などの非線形演算が出現する度に **ダミー変数の数が増え**、計算が遅くなる。

ダミー変数 ε は異なる変量間の相関性を表現する

$$\begin{aligned}x &= 1 + 0.5\varepsilon_1 \\y &= 1 + 0.5\varepsilon_2\end{aligned}$$



$$\begin{aligned}x &= 1 + 0.5\varepsilon_1 \\y &= 1 + 0.4\varepsilon_1 + 0.1\varepsilon_2\end{aligned}$$



x, y それぞれの変域は同じだが、"joint range" は異なる。

区間との相互変換

区間 $\rightarrow$ affine

$$\begin{pmatrix} [\underline{x_1}, \overline{x_1}] \\ [\underline{x_2}, \overline{x_2}] \\ \vdots \\ [\underline{x_n}, \overline{x_n}] \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{\overline{x_1} + \underline{x_1}}{2} + \frac{\overline{x_1} - \underline{x_1}}{2} \varepsilon_1 \\ \frac{\overline{x_2} + \underline{x_2}}{2} + \frac{\overline{x_2} - \underline{x_2}}{2} \varepsilon_2 \\ \vdots \\ \frac{\overline{x_n} + \underline{x_n}}{2} + \frac{\overline{x_n} - \underline{x_n}}{2} \varepsilon_n \end{pmatrix}$$

affine $\rightarrow$ 区間

$$x = a_0 + a_1 \varepsilon_1 + \cdots + a_n \varepsilon_n$$

$$\Rightarrow [a_0 - \delta, a_0 + \delta], \quad (\delta = \sum_{i=1}^n |a_i|)$$

線形計算は簡単

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n$$

$$y = y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n$$

加算、減算

$$x \pm y = (x_0 \pm y_0) + (x_1 \pm y_1)\varepsilon_1 + \cdots + (x_n \pm y_n)\varepsilon_n$$

$$x \pm \alpha = (x_0 \pm \alpha) + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \quad .$$

定数の乗算

$$\alpha x = (\alpha x_0) + (\alpha x_1)\varepsilon_1 + \cdots + (\alpha x_n)\varepsilon_n \quad .$$

非線形単項演算 (数学関数)

f : $\exp, \log, \dots$ などの単項演算

affine 変数 x に対して、 $z = f(x)$ を計算することを考える。

$$x = x_0 + x_1\varepsilon_1 + \dots + x_n\varepsilon_n$$

- ① x の変動区間 I を次のように計算する:

$$I = [x_0 - \delta, x_0 + \delta], \quad \delta = \sum_{i=1}^n |x_i| \quad ,$$

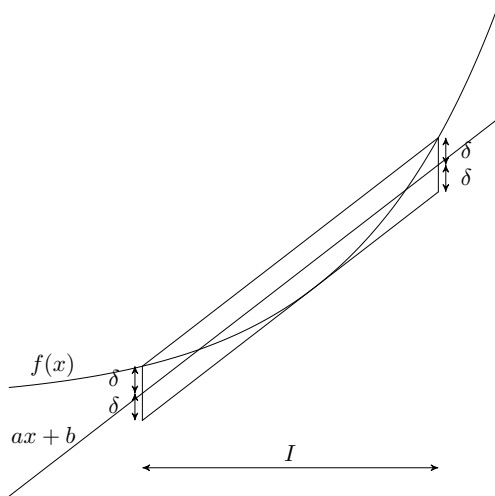
- ② f の I 上における線形近似 $ax + b$ と最大誤差 δ を計算する:

$$\delta = \max_{x \in I} |f(x) - (ax + b)|$$

- ③ 計算結果 z は次の式で計算する:

$$a(x_0 + x_1\varepsilon_1 + \dots + x_n\varepsilon_n) + b + \delta\varepsilon_{n+1}$$

線形近似 $ax + b$ と誤差 δ



非線形二項演算

二項演算子 $g(x, y)$ に対して、線形近似 $ax + by + c$ を考える。(単項演算の場合とほぼ同様。)

乗算

$$\begin{aligned} z &= y_0x + x_0y - x_0y_0 + \delta_x\delta_y\varepsilon_{n+1} \\ &= x_0y_0 + \sum_{i=1}^n (y_0x_i + x_0y_i)\varepsilon_i \\ &\quad + \left(\sum_{i=1}^n |x_i|\right)\left(\sum_{i=1}^n |y_i|\right)\varepsilon_{n+1} \end{aligned}$$

QR 分解を用いた解の接続

$J_i, c_i = \text{mid}(J_i)$ に対する平均値形式

$$\phi_{t_i, t_{i+1}}(x) \in \phi_{t_i, t_{i+1}}(c_i) + \phi'_{t_i, t_{i+1}}(J_i)(x - c_i)$$

について、 $A_i = \phi'_{t_i, t_{i+1}}(J_i)$, $B_i = \phi_{t_i, t_{i+1}}(c_i)$ と書く。

J_0 を初期値、 $c_0 = \text{mid}(J_0)$, $K_0 = J_0 - c_0$, $Q_0 = I$, $i = 0$ とし、

① J_i を元に A_i, B_i を計算する。

② $c_{i+1} = \text{mid}(B_i)$

③ $A_i Q_i$ の中心を

$$\text{mid}(A_i Q_i) \simeq QR$$

のように (近似)QR 分解し、得られた Q を Q_{i+1} とする。

④ $K_{i+1} = (Q_{i+1}^{-1} A_i Q_i) K_i + Q_{i+1}^{-1} (B_i - c_i)$

⑤ $J_{i+1} = Q_{i+1} K_{i+1} + c_{i+1}$

⑥ $i = i + 1$

を繰り返す。ただし、 Q_{i+1}^{-1} は Q_{i+1} の真の逆行列またはそれを含む区間行列でなければならない。

区間演算と Affine Arithmetic と QR 法の比較 (1)

$$I_0 = \begin{pmatrix} [6 - 2^{-3}, 6 + 2^{-3}] \\ [-2^{-3}, 2^{-3}] \end{pmatrix}$$

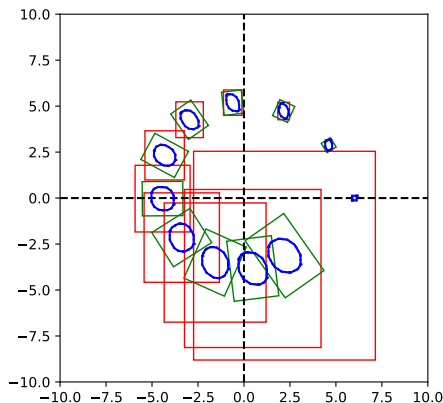
を初期区間とし、

$$I_{k+1} = \begin{pmatrix} \frac{7}{8} \sin(0.5) & -\cos(0.5) \\ \cos(0.5) & \sin(0.5) \end{pmatrix} I_k + \begin{pmatrix} [-2^{-5}, 2^{-5}] \\ [-2^{-3}, 2^{-3}] \end{pmatrix}$$

を計算させてみた。

線形変換 + ノイズ成分というこの形は、常微分方程式の初期値問題において各ステップ毎のノイズ (= 離散化誤差) の混入が避けられないことをモデル化している。

区間演算と Affine Arithmetic と QR 法の比較 (2)



青が Affine Arithmetic、緑が QR 分解法、赤が区間演算の結果。

ステップ幅 Δt の決定

ε_0 を 1 ステップで混入する誤差の目標値とする。例えば machine epsilon。

- ① Type-I PSA を用いて Taylor 展開を計算し、その係数を見て適切なステップ幅 Δt_0 を推定する。Type-I PSA で計算された Taylor 展開を

$$x_0 + x_1 t + x_2 t^2 + \cdots + x_{n-1} t^{n-1} + x_n t^n$$

として、

$$\Delta t_0 = \frac{\varepsilon_0^{\frac{1}{n}}}{\max(|x_{n-1}|^{\frac{1}{n-1}}, |x_n|^{\frac{1}{n}})}$$

とする。

- ② ステップ幅 Δt_0 を用いて Type-II PSA を使って候補者集合を作成する。
- ③ その候補者集合を使ったとき新たに混入する誤差を ε として、新しいステップ幅を

$$\Delta t_1 = \Delta t_0 \left(\frac{\varepsilon_0}{\varepsilon} \right)^{\frac{1}{n}}$$

で推定する。ただし、 n は Taylor 展開の次数。

- ④ ステップ幅 Δt_1 を用いて Type-II PSA を使って再度候補者集合を作成し、解の存在検証を行う。検証に失敗したら例えばステップ幅を半分にする。