令和 5 年度

卒業論文

radii polynomial approach における
零点探索手順の削除

2024 年 1 月 24 日

指導教員　関根 晃太　准教授

千葉工業大学　情報科学部　情報工学科

学生番号　2031116

西窪 壱華

目次

1　はじめに

今日の理工学では, 理想化された物理モデルを微分方程式の数学モデルで表し解くことによって, 未知の現象を予測したり, 新しい製品を設計することを可能としている [2]. そのモデルや微分方程式の複雑性から, 一般的に解の導出は計算機で行われるが, 同時に誤差が生じる. 計算機による計算過程を階層的に見た場合, 次のような誤差の分類ができる [1].

1. 科学計算のもとの数学モデルが, 正しく記述できていないことによる誤差 (**モデル化誤差**)
2. 無限次元の問題を有限次元の非線形方程式に近似することによって生じる誤差 (**離散化誤差**)
3. 問題の解法手順が計算機内で本質的に実現不能なことに起因する誤差 (**打切り誤差**)
4. 計算機内で扱うために有限桁の浮動小数点数に近似することによって生じる誤差 (**丸め誤差**)

1 は例えば実際の流体の流れと, それを記述する偏微分方程式という数学モデルとの間の差異である. 2 は偏微分方程式の無限次元解を計算機で実装可能にするために有限次元に近似するときの誤差である. 3 は本来無限回数繰り返すことで厳密解を与えられるアルゴリズムを計算機演算で実現することは不可能であるために有限回数で中断しなければならず, このとき生じる誤差を指す. 4 はコンピュータが実数を浮動小数点数で保持するために, 切り捨てや切り上げによって近似するときの誤差をいう. これらの誤差は, 例えば数値計算により品質保証されている製品の信頼性を損なう要因となる. 故に, 数値計算結果の精度を保証することで製品の信頼性向上したり, 理論解析が困難な問題の解の存在を数値計算により立証するために必要となる. ここで精度保証とは, 誤差 (真の解の存在範囲) の把握, 解の存在や一意性の数学的な保証を行うことを意味する.

2,3,4 にあたる解の精度保証付き数値計算に, Newton-Kantorovich の定理を精度保証付き数値計算に利用した radii polynomial approach(Newton-Kantorovich 型定理) [6] がある. この定理は有限次元や無限次元を問わず, 非線形方程式や偏微分方程式など殆どの微分方程式に用いることができる. しかし, 与式とは別に非線形不等式である radii polynomial の零点探索を行い, 精度保証付き数値計算を行う必要がある.

本研究では, radii polynomial approach における零点探索の手順を省略することを目的とする. van der Pol 方程式

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$$

に対してフーリエ・スペクトル法による数値計算で出力された周期解の精度保証を行う.

本論文の構成は次のとおりである. 第 2 章では, 準備として関数解析の基本事項をまとめる. 第 3 章では, 既存手法である解の存在を保証する Newton-Kantorovich の定理を用いた定理とその証明を示す. 第 4 章では, 既存の radii polynomial approach を計算機で実装する方法を提示する. 第 5 章では, radii polynomial の零点探索を削除する手法を提案する. 第 6 章では, 既存, 提案手法の数値実験を行い, 精度と実行時間の比較をする. 第 7 章では, 本論文のまとめと今後の課題を示す.

2 準備

2.1 ノルムと Banach 空間

定義 1 (線形空間の公理)**.** 空でない集合 X が係数体 $\mathbb{K}$ 上の線形空間であるとは, 任意の $u, v \in X$ とスカラー $\alpha \in \mathbb{K}$ に対して, 加法 $u + v \in X$ とスカラー乗法 $\alpha u \in X$ が定義されていて, 任意の $u, v, w \in X$ とスカラー $\alpha, \beta \in \mathbb{K}$ に対して次のことが成り立つことである.

1. $(u + v) + w = u + (v + w)$
2. $u + v = v + u$
3. $u + 0 = u$ となる $0 \in X$ が一意に存在
4. $u + (-u) = 0$ となる $-u \in X$ が一意に存在
5. $\alpha(u + v) = \alpha u + \beta v$
6. $(\alpha + \beta)u = \alpha u + \beta u$
7. $(\alpha\beta)u = \alpha(\beta u)$
8. $1u = u, \quad 1 \in \mathbb{K}$

定義 2 ($\mathit{ノルムとノルム空間の定義}$)**.** X を係数体 $\mathbb{K}$ 上の線形空間とする. X で定義された関数 $\|\cdot\| : X \to \mathbb{R}$ が X のノルムであるとは

1. $\|u\| \geq 0, \quad u \in X$
2. $\|u\| = 0 \Leftrightarrow u = 0$
3. $\|\alpha u\| = |\alpha|\|u\|, \quad (\alpha \in \mathbb{K}, u \in X)$
4. $\|u + v\| \leq \|u\| + \|v\|$

が成立することである. さらに X に一つのノルムが指定されているとき, X はノルム空間という.

定義 3 ($\mathit{ノルム空間の収束と極限}$)**.** X をノルム空間とする. X の点列 $(u_n) \subset X$ は

$$\forall \epsilon > 0 \; \exists N \in \mathbb{N} \; \forall n \geq N \quad \text{に対して} \quad \|u_n - u\| < \epsilon$$

のとき, 点 $u \in X$ に収束するといい,

$$\|u_n - u\| \to 0, \quad (n \to \infty)$$

と表す. このとき, u を u_n の極限といい

$$u_n \to u, \quad (n \to \infty)$$

と表す.

定義 4 (Cauchy 列)**.** X をノルム空間とする. そのとき X の点列 (u_n) が Cauchy 列であるとは

$$u_n - u_m \to 0, \quad (n, m \to \infty)$$

が成立することである. 即ち

$$\|u_n - u_m\| \to 0, \quad (n, m \to \infty)$$

が成立することである.

定義 5 (完備)**.** X をノルム空間とする. X が完備であるとは, 任意の Cauchy 列 (u_n) が X の中で極限を持つことである. すなわち, 任意の Cauchy 列 $(u_n \subset X)$ が

$$\|u_n - u\| \to 0, \quad (n \to \infty)$$

となる極限 u を X 内に持つことである.

定義 6 (Banach 空間)**.** ノルム空間 X が Banach 空間であるとは, X が完備であることである.

定理 1 (逆三角不等式)**.** X をノルム空間とする. 任意の $u, v \in X$ について次の不等式を満たす.

$$|\|u\| - \|v\|| \leq \|u - v\|$$

証明. 任意の $u, v \in X$ について

$$\begin{aligned}\|u\| &= \|u - v + v\| \leq \|u - v\| + \|v\| \\ \|v\| &= \|v - u + u\| \leq \|v - u\| + \|u\| = \|u - v\| + \|u\|\end{aligned}$$

となる. よって

$$\begin{aligned}\|u\| - \|v\| &\leq \|u - v\| \\ \|v\| - \|u\| &\leq \|u - v\|\end{aligned}$$

となるため,

$$|\|u\| - \|v\|| \leq \|u - v\|$$

を持つ. ■

定義 7 (有界列)**.** X をノルム空間とする. そのとき X の点列 (u_n) が有界列とは任意の $n \in \mathbb{N}$ に対して

$$\|u_n\| \leq M$$

となる定数 $M > 0$ が存在することである.

定理 2 (Cauchy 列ならば有界列)**.** X をノルム空間とする. そのとき X の点列 (u_n) が Cauchy 列ならば有界列でもある.

証明. X の点列 (u_n) が Cauchy 列であるため, $\epsilon - N$ 論法を用いた表記で

$$\forall \epsilon > 0,\ \exists N \in \mathbb{N},\ \forall n, m \geq N \text{に対して} \|u_n - u_m\| < \epsilon$$

を満たす. $\epsilon = 1$ としても, それに対応した N が存在し, 任意の $n \geq N$ に対して

$$\|u_n - u_N\| < 1$$

を満たす.

任意の $n \geq N$ に対して $\|u_n\|$ が $\|u_N\|$ で評価できることを示す. 逆三角不等式である定理 (1) を用いると

$$|\|u_n\| - \|u_N\|| \leq \|u_n - u_N\| < 1$$

となる. 絶対値の性質より $|\|u_n - u_N\|| < 1$ は

$$\|u_N\| - 1 < \|u_n\| < \|u_N\| + 1$$

となる. よって

$$M = \max\{\|u_1\|, \|u_2\|, \cdots, \|u_{N-1}\|, \|u_N\| + 1\}$$

とすると, 任意の $n \in \mathbb{N}$ について

$$\|u_n\| \leq M$$

が成り立つため, 点列 (u_n) は有界列である. ■

定義 8 (線形部分空間)**.** 線形空間 X の空ではない集合 M が任意の元 $u, v \in M$ と任意の係数体 $\alpha \in \mathbb{K}$ に対して

$$u + v \in M$$
$$\alpha u \in M$$

を満たすとき, M は線形空間 X の線形部分空間と呼ぶ.

定義 9 (ノルム空間の開球)**.** X をノルム空間とする. $x \in X$ とし, $r > 0$ を正実数とする. そのとき, 集合

$$B_X(x, r) := \{y \in X \mid \|x - y\|_X < r\}$$

を中心 x, 半径 r の開球という. X が明らかな場合は $B_X(x, r)$ を省略して $B(x, r)$ と表記する.

定義 10 (ノルム空間の開集合)**.** X をノルム空間とし, M を X の部分集合とする. 任意の $x \in M$ に対して, $B_X(x, r) \subset M$ となる $r > 0$ が存在する場合, M が開集合であるという.

定義 11 (ノルム空間の閉集合)**.** X をノルム空間とし, M を X の部分集合とする. M が閉集合であるとは, M の任意の点列 (u_n) の極限 $u \in X$ が M にも属することである. すなわち, 点列 $(u_n) \subset M$ について

$$u_n \to u, \quad (n \to \infty) \Rightarrow u \in M$$

であるとき, M は閉集合であるという.

定義 12 (閉部分空間)**.** X をノルム空間とし, M を X の線形部分空間が閉集合であるとき, M を閉部分空間であるという.

2.2 作用素

定義 13 (作用素)**.** ある線形空間 X から別の線形空間 Y への作用素 A とは,

$$\mathcal{D}(A) := \{u \in X \mid Au \in Y\}$$

としたとき, $\mathcal{D}(A)$ のどんな元に対しても, それぞれ集合 Y のただ一つの元を指定する規則ことである. また, $\mathcal{D}(A)$ は A の定義域と呼ばれ

$$\mathcal{R}(A) := \{Au \in Y \mid u \in \mathcal{D}(A)\}$$

を値域と呼ぶ.

定義 14 (単射)**.** 線形空間 X から線形空間 Y への作用素 A が

$$u_1 \neq u_2, \quad \forall u_1, u_2 \in \mathcal{D}(A) \Rightarrow A(u_1) \neq A(u_2)$$

を満たすときに作用素 A は単射であるという.

定義 15 (全射)**.** 線形空間 X から線形空間 Y への作用素 A が

$$Y = \mathcal{R}(A)$$

を満たすときに作用素 A は全射であるという.

定義 16 (逆作用素)**.** 線形空間 X から線形空間 Y への作用素 A とし, その定義域を $\mathcal{D}(A) \subset X$, 値域を $\mathcal{R}(A) \subset Y$ とする. そのとき,

$$\begin{aligned} A^{-1}(A(u)) &= u, \quad u \in \mathcal{D}(A) \\ A(A^{-1}(v)) &= v, \quad v \in \mathcal{R}(A) \end{aligned}$$

かつ

$$\begin{aligned} \mathcal{D}(A^{-1}) &= \mathcal{R}(A) \\ \mathcal{R}(A^{-1}) &= \mathcal{D}(A) \end{aligned}$$

となる Y から X への作用素 A^{-1} を A の逆作用素と呼ぶ.

定理 3 (単射と逆作用素の関係)**.** 線形空間 X から線形空間 Y への作用素 A とすると,

$$A \text{ が逆作用素を持つ} \Leftrightarrow A \text{ が単射である}$$

証明.「A が逆作用素を持つ $\Rightarrow A$ が単射である」の証明

単射の定義 (14) の対偶「任意の $u_1, u_2 \in \mathcal{D}(A)$ に対し $A(u_1) = A(u_2) \Rightarrow u_1 = u_2$」を満たすことを確かめる. A の逆作用素を A^{-1} とすると, 任意の $u_1, u_2 \in \mathcal{D}(A)$ に対し

$$\begin{aligned} & A(u_1) = Au_2 \\ \Rightarrow\ & A^{-1}(A(u_1)) = A^{-1}(A(u_2)) \\ \Rightarrow\ & u_1 = u_2 \end{aligned}$$

となる. 最後の変形は逆作用素 A^{-1} の定義に由来する.

「A が単射である $\Rightarrow A$ が逆作用素 A^{-1} を持つ」の証明

A の値域の定義 $\mathcal{R}(\mathcal{A}) = \{A(u) \in Y \mid u \in \mathcal{D}(A)\}$ より, 任意の $v \in \mathcal{R}(A)$ に対し

$$A(u) = v$$

となる $u \in \mathcal{D}(A)$ が存在する. その上, A が単射であるため, 単射の定義の対偶より $u \in \mathcal{D}(A)$ はどんな $u \in \mathcal{R}(A)$ に対してもただ一つの元である. そのため, 作用素の定義より, 上記の $v \in \mathcal{R}(A)$ に対してただ一つの元 $u \in \mathcal{D}(A)$ を指定する規則として

$$B(v) = u$$

となる定義域 $\mathcal{D}(B) = \mathcal{R}(A)$ と値域 $\mathcal{R}(B) = \mathcal{D}(A)$ となる Y から X への作用素 B が定義できる. その上, $B(v) = u$ の $v = A(u)$ を代入すると

$$B(A(u)) = u$$

となる. 同様に, $A(u) = v$ の u に $u = B(v)$ を代入すると

$$A(B(v)) = v$$

となる. よって, 定義域 $\mathcal{D}(B) = \mathcal{R}(A)$ と値域 $\mathcal{R}(B) = \mathcal{D}(A)$ となる Y から X への作用素 B は A の逆作用素であるため, A は逆作用素を持つ. ■

定義 17 (作用素の等号)**.** 線形空間 X から線形空間 Y への作用素 A と B が等しいとは

$$\mathcal{D}(A) = \mathcal{D}(B)$$

かつ

$$Au = Bu, \quad \forall u \in \mathcal{D}(A) = \mathcal{D}(B)$$

が成立することであり,

$$A = B$$

と表記する.

定義 18 (作用素の連続)**.** ノルム空間 X からノルム空間 Y への作用素 A が $u \in \mathcal{D}(A)$ で連続であるとは

$$u_n \to u, \quad (n \to \infty)$$

となる任意 $u_n \in \mathcal{D}(A) \subset X$ に対して

$$Au_n \to Au, \quad (n \to \infty)$$

を満たすときである. さらに, A が任意の $u \in \mathcal{D}(A)$ において連続であるとき, A は連続であるという.

定義 19 (線形作用素)**.** 線形空間 X から線形空間 Y への作用素 A が, 任意の $u, v \in \mathcal{D}(A) \subset X$ と $\alpha \in \mathbb{K}$ に対し,

$$\mathcal{D}(A) \text{ が } X \text{ の線形部分空間}$$
$$A(u+v) = Au + Av$$
$$A(\alpha u) = \alpha Au$$

を満たすとき, A を作用素と呼ぶ.

定義 20 (線形作用素の加法)**.** 線形空間 X から線形空間 Y への線形作用素 A と B の和を

$$(A+B)u := Au + Bu, \quad u \in \mathcal{D}(A) \cap \mathcal{D}(B)$$

と定義する. このとき, X から Y への作用素 $A+B$ の定義域は

$$\mathcal{D}(A+B) = \mathcal{D}(A) \cap \mathcal{D}(B)$$

とする.

定義 21 (線形作用素のスカラー乗法)**.** 線形空間 X から線形空間 Y への線形作用素 A の $\alpha \in \mathbb{K}$ によるスカラー倍を

$$(\alpha A)u := \alpha(Au), \quad u \in \mathcal{D}(A)$$

と定義する. このとき, X から Y への作用素 αA の定義域は

$$\mathcal{D}(\alpha A) := \mathcal{D}(A)$$

とする.

定義 22 (合成作用素)**.** X, Y, Z を線形空間とする. A を Y から Z への線形作用素とし, B を X から Y への線形作用素とする. そのとき, A と B の合成作用素 AB は

$$(AB)u := A(Bu), \quad u \in \{v \in \mathcal{D}(B) \mid Bv \in \mathcal{D}(A)\}$$

と定義する. このとき, X から Z への合成作用素 AB の定義域は

$$\mathcal{D}(AB) := \{v \in \mathcal{D}(B) \mid Bv \in \mathcal{D}(A)\}$$

とする.

定理 4 (線形作用素に対する単射性 (1))**.** 線形空間 X から線形空間 Y への線形作用素 A において以下は同値である.

1. 線形作用素 A が単射である
2. $Au = 0, \quad u \in \mathcal{D}(A) \Rightarrow u = 0$

証明. 単射の定義の対偶は

$$Au_1 = Au_2, \quad \forall u_1, u_2 \in \mathcal{D}(A) \Rightarrow u_1 = u_2$$

となる. その上, A は線形作用素であるため

$$Au_1 = Au_2 \Leftrightarrow A(u_1 - u_2) = 0$$

となる. $u_1 - u_2 \in \mathcal{D}(A)$ を $u \in \mathcal{D}(A)$ とおきなおせば, $1 \Rightarrow 2$ は証明された. また, 証明を逆に追うことで $2 \Rightarrow 1$ も示せる. ■

定理 5 (線形作用素に対する単射性 (2))**.** ノルム空間 X からノルム空間 Y への線形作用素 A とする. 不等式

$$\|u\|_X \leq K\|Au\|_Y, \quad u \in \mathcal{D}(A)$$

を満たす定数 $K > 0$ が存在するならば, 線形作用素 A は単射である.

証明. A が線形作用素であるため, $Au = 0, \quad u \in \mathcal{D}(A) \Rightarrow u = 0$ を使って証明する. ノルムの定義より

$$Au = 0, \quad \forall u \in \mathcal{D}(A) \Leftrightarrow \|Au\|_Y = 0$$

となる. さらに, $Au = 0$ ならば,

$$\|u\|_X \leq K\|Au\|_Y = 0, \quad u \in \mathcal{D}(A)$$

より $\|u\|_X = 0$ となる. よって, 再びノルムの定義より

$$\|u\|_X = 0, \quad \forall u \in \mathcal{D}(A) \Leftrightarrow u = 0$$

より, $Au = 0$ ならば $u = 0$ となる. ■

定義 23 (有界な線形作用素)**.** ノルム空間 X から Y への線形作用素 A に対し,

$$\|Au\|_Y \leq K\|u\|_X, \quad u \in \mathcal{D}(A)$$

を満たす正の定数 K が存在するとき, 線形作用素 A を有界な作用素と呼ぶ.

定理 6 (有界な線形作用素と連続な線形作用素)**.** ノルム空間 X からノルム空間 Y への線形作用素 A に対し,

$$A \text{ が有界} \Leftrightarrow A \text{ が連続}$$

証明.「A が有界 $\Rightarrow A$ が連続」の証明

連続性の定義より, $u_n \to u$ となる任意の $u_n \in \mathcal{D}(A)$ に対して $Au_n \to Au$ となることを確かめる. $u_n \to u$ となる任意の $u_n \in \mathcal{D}(A)$ から $\|u_n - u\|_X \to 0, \quad (n \to \infty)$ を持つ. その上, A は有界であることから

$$\|Au_n - Au\|_Y \leq M\|u_n - u\|_X \to 0, \ (n \to \infty)$$

となる. よって, $u_n \to u, \ (n \to \infty)$ ならば, $Au_n \to Au$ であるため, A は連続である.

「A が連続 $\Rightarrow A$ が有界」

背理法によって証明する. すなわち, A が連続ならば, 任意の $M_2 > 0$ に対して

$$\|Au\|_Y > M_2\|u\|_X$$

を満たす $u \in \mathcal{D}(A)$ が存在すると仮定して矛盾を見つける. この仮定より自然数 n に対して,

$$\|Au_n\|_Y > n\|u_n\|_X$$

を満たす $u_n \in \mathcal{D}(A)$ が存在する．このとき, $\|u_n\|_X \neq 0$ であることに注意する．ノルム空間 X はノルム空間の定義より線形空間であるため, ゼロ元 $0 \in X$ を持つ．その上, 線形作用素の定義より $\mathcal{D}(A)$ は X の部分空間であるため, ゼロ元 $0 \in \mathcal{D}(A) \subset X$ を持つ．その上, A が連続であるため, A は $0 \in \mathcal{D}(A)$ でも連続である．$\epsilon - \delta$ 論法による A の $0 \in \mathcal{D}(A) \subset X$ における連続の定義を記述すると

$$\forall \epsilon > 0,\ \exists \delta > 0,\ \|u_n\|_X < \delta \text{ となる } \forall u_n \in X \text{ に対して } \|Au_n\|_Y < \epsilon$$

となる．その上, ϵ を $n\|u_n\|_X$ とすると, $\delta_n > 0$ が存在し, $\|u_n\|_X < \delta$ となる任意の $u_n \in \mathcal{D}(A)$ に対して,

$$\|Au_n\|_Y < n\|u_n\|_X$$

となる．有界ではないという仮定と組み合わせると

$$n\|u_n\|_X < \|Au_n\|_Y < n\|u_n\|_X$$

となるため矛盾する． ■

定義 24 (定義域が X の全体となる有界な線形作用素全体の集合 $\mathcal{L}(X,Y)$)**.** 定義域が Banach 空間 X 全体となる X から Y への有界線形作用素全体を

$$\mathcal{L}(X,Y)$$

とする．

定理 7 ($\mathcal{L}(X,Y)$ は Banach 空間)**.** X をノルム空間とし, Y を Banach 空間とする．定義域が X 全体となる X から Y への有界な線形作用素全体の集合 $\mathcal{L}(X,Y)$ のノルムを

$$\|A\|_{\mathcal{L}(X,Y)} := \sup_{u \in X \setminus \{0\}} \frac{\|Au\|_Y}{\|u\|_X}, \quad A \in \mathcal{L}(X,Y)$$

とすると, $\mathcal{L}(X,Y)$ は Banach 空間となる．

証明. 作用素の加法 (20) と作用素のスカラー乗法 (21) の定義をもとに線形空間の公理 (1) が満たされていることが導かれる．ただし, $\mathcal{L}(X,Y)$ のゼロ元は任意の $u \in X$ を $0 \in Y$ へ写す作用素であることに注意が必要である．

「ノルム空間」

$\|A\|_{\mathcal{L}(X,Y)}$ がノルムの定義を満たすことを示せばよい．ノルム空間 X と Banach 空間 Y であるため $\|\cdot\|_X \geq 0$ と $\|\cdot\|_Y \geq 0$ であることから

$$\frac{\|Au\|_Y}{\|u\|_X} \geq 0$$

となるため, $\|A\|_{\mathcal{L}(X,Y)} \geq 0$ となり, ノルムの定義 (1) はいえる．

次に, $A = 0$ ならば $\|Au\|_Y = 0$ であるため,

$$\|A\|_{\mathcal{L}(X,Y)} = \sup_{u \in X \setminus \{0\}} \frac{\|Au\|_Y}{\|u\|_X} = \sup_{u \in X \setminus \{0\}} \frac{0}{\|u\|_X} = 0$$

である. さらに, 任意の $u \in X\backslash\{0\}$ について

$$\frac{\|Au\|_Y}{\|u\|_X} = 0 \Leftrightarrow \|Au\|_Y = 0 \Leftrightarrow Au = 0$$

任意の $u \in X\backslash\{0\}$ を $0 \in Y$ へ写す作用素は $\mathcal{L}(X,Y)$ が線形空間より一意に存在し, $A = 0$ である. よって, ノルムの定義 (2) も示された.

続いて, $\alpha \in \mathbb{K}$ としたとき, Y は Banach 空間であるため $\|\cdot\|_Y$ はノルムの定義を満たすため,

$$\|\alpha A\|_{\mathcal{L}(X,Y)} = \sup_{u\in X\backslash\{0\}} \frac{\|\alpha Au\|_Y}{\|u\|_X} = |\alpha| \sup_{u\in X\backslash\{0\}} \frac{\|Au\|_Y}{\|u\|_X} = |\alpha|\|A\|_{\mathcal{L}(X,Y)}$$

となるため, ノルムの定義 (3) も示された.

最後に任意の $A, B \in \mathcal{L}(X,Y)$ について

$$\begin{aligned}
\|A+B\|_{\mathcal{L}(X,Y)} &= \sup_{u\in X\backslash\{0\}} \frac{\|(A+B)u\|_Y}{\|u\|_X} \\
&= \sup_{u\in\backslash\{0\}} \frac{\|Au+Bu\|_Y}{\|u\|_X} \\
&\leq \sup_{u\in X\backslash\{0\}} \frac{\|Au\|_Y + \|Bu\|_Y}{\|u\|_X} \\
&\leq \sup_{u\in X\backslash\{0\}} \frac{\|Au\|_Y}{\|u\|_X} + \sup_{u\in X\backslash\{0\}} \frac{\|Bu\|_Y}{\|u\|_X} \\
&= \|A\|_{\mathcal{L}(X,Y)} + \|B\|_{\mathcal{L}(X,Y)}
\end{aligned}$$

となり, ノルムの定義 (4) も示されたため, $\mathcal{L}(X,Y)$ はノルム空間である.

「Banach 空間」

Banach 空間であることを証明するには $\mathcal{L}(X,Y)$ の任意の Cauchy 列 $(A_n) \subset \mathcal{L}(X,Y)$ が極限 T を $\mathcal{L}(X,Y)$ 内に持つことを示せばよい.

まず, 極限の候補 $\tilde{A}$ が定義できるか確認する. 任意の Cauchy 列 $(A_n) \subset \mathcal{L}(X,Y)$ は Cauchy 列の定義 (4) より

$$\|A_n - A_m\|_{\mathcal{L}(X,Y)} \to 0, \quad (n,m \to 0)$$

となる. 任意の $u \in X\backslash\{0\}$ に対して, 点列 $(A_nu) \subset Y$ は

$$\begin{aligned}
\|A_nu - A_mu\|_Y &= \frac{\|(A_n - A_m)u\|_Y}{\|u\|_Y}\|u\|_X \\
&\leq \sup_{\phi\in X\backslash\{0\}} \frac{\|(A_n - A_m)\phi\|_Y}{\|\phi\|_X}\|u\|_X \\
&= \|A_n - A_m\|_{\mathcal{L}(X,Y)}\|u\|_X \to 0, \quad (n,m \to 0)
\end{aligned}$$

を持つため, 点列 $(A_nu) \subset Y$ は Cauchy 列になる. その上, Y は Banach 空間であるため, Y の任意の Cauchy 列は収束し, Y 内に極限 $\tilde{A}u$ となるような X から Y への作用素 $\tilde{A}$ が存在する. ここで, 任意の $u \in X$ に対して極限 $\tilde{A}u$ が定義されることから, $\tilde{A}$ の定義域は $\mathcal{D}(\tilde{A}) = X$ である. これにより, $\mathcal{L}(X,Y)$ の任意の Cauchy 列 (A_n) の極限の候補 $\tilde{A}$ が定義できた.

続いて, 定義した極限の候補 $\tilde{A}$ が $\mathcal{L}(X,Y)$ に属しているか確認する. $\tilde{A}$ が有界な線形作用素であり, かつ $\mathcal{D}(\tilde{A}) = X$ であることを示せばよい. $\mathcal{L}(X,Y)$ の任意の Cauchy 列 (A_n) の元 A_n は線形作用素あるため, 線形作用素の定義より任意の $\alpha, \beta \in \mathbb{K}$ と $u, v \in X$ について

$$A_n(\alpha u + \beta v) = \alpha A_n u + \beta A_n v$$

を持つ. よって $n \to \infty$ とすると

$$\tilde{A}(\alpha u + \beta v) = \alpha \tilde{A} u + \beta \tilde{A} v$$

となり, 極限の候補 $\tilde{A}$ は線形作用素である. 次に極限の候補 $\tilde{A}$ が有界作用素であることを示す. 点列 (A_n) は Cauchy 列であるため定理 (2) より有界列でもある. すなわち, どんな $n \in \mathbb{N}$ に対しても

$$\|A_n\|_{\mathcal{L}(X,Y)} \leq M$$

となる $n \in \mathbb{N}$ に依存しない定数 M が存在する. この $n \in \mathbb{N}$ に依存しない定数 M は, 任意の $n \in X$ について

$$\|A_n u\|_Y \leq M \|u\|_X$$

も満たす. $A_n u \to \tilde{A}_u$, $(n \to \infty)$ であるため, 上の不等式に対して $n \to \infty$ とすると M が n に依存しないため

$$\|\tilde{A} u\|_Y \leq M \|u\|_X$$

を得る. よって, 点列 (A_n) の極限の候補 $\tilde{A}$ は $\mathcal{L}(X,Y)$ に属する.

最後に, 点列 (A_n) の極限が $\tilde{A}$ であることを示す. 任意の $u \in X$ に対して, 点列 $(A_n u) \subset Y$ は Y 内に極限 $\tilde{A}u$ を持つこと, すなわち

$$A_n u \to \tilde{A} u, \quad (n \to \infty)$$

を持つことから

$$\|A_n u - A_m u\|_Y \to \|A_n u - \tilde{A} u\|_Y, \quad (m \to \infty)$$

となる. その上, $\mathcal{L}(X,Y)$ のノルムの定義と $\tilde{A} \in \mathcal{L}(X,Y)$ から

$$\|A_n - A_m\|_{\mathcal{L}(X,Y)} \to \|A_n - \tilde{A}\|_{\mathcal{L}(X,Y)}, \quad (m \to \infty)$$

を得る. 点列 (A_n) が Cauchy 列であるため

$$\forall \epsilon > 0,\ \exists N \in \mathbb{N},\ \forall n, m \geq N \text{に対して} \|A_n - A_m\|_{\mathcal{L}(X,Y)} < \epsilon$$

を満たす. その上, $m \to \infty$ とすると

$$\forall \epsilon > 0,\ \exists N \in \mathbb{N},\ \forall n \geq N \text{に対して} \|A_n - \tilde{A}\|_{\mathcal{L}(X,Y)} < \epsilon$$

となり, $\tilde{A} \in \mathcal{L}(X,Y)$ は Cauchy 列 (A_n) の極限である. よって, 任意の Cauchy 列は $\mathcal{L}(X,Y)$ 内に極限を持つため, ノルム空間 $\mathcal{L}(X,Y)$ は Banach 空間である. ■

定理 8 ($\mathcal{L}(X,Y)$ ノルムの性質 (1))**.** X をノルム空間とし, Y を Banach 空間とする. そのとき, 任意の $u \in X$ と任意の $A \in \mathcal{L}(X,Y)$ について以下の不等式が成り立つ.

$$\|Au\|_Y \leq \|A\|_{\mathcal{L}(X,Y)}\|u\|_X$$

証明. $u = 0$ の場合は明らかに成り立つため, $u \in X\backslash\{0\}$ について考える. $u \in X\backslash\{0\}$ について

$$\|Au\|_Y = \frac{\|Au\|_Y}{\|u\|_X}\|u\|_X \leq \sup_{\phi \in X\backslash\{0\}} \frac{\|A\phi\|_Y}{\|\phi\|_X}\|u\|_X = \|A\|_{\mathcal{L}(X,Y)}\|u\|_X$$

となるため, 題意は示された. ■

定理 9 ($\mathcal{L}(X,Y)$ のノルムの性質 (2))**.** X をノルム空間とし, Y と Z を Banach 空間とする. そのとき, 任意の $B \in \mathcal{L}(X,Y)$ と $A \in \mathcal{L}(X,Y)$ の合成作用素 AB は $\mathcal{L}(X,Y)$ に属する. その上,

$$\|AB\|_{\mathcal{L}(X,Z)} \leq \|A\|_{\mathcal{L}(Y,Z)}\|B\|_{\mathcal{L}(X,Y)}$$

証明. 合成作用素の定義 (22) から

$$\mathcal{D}(AB) = \{v \in \mathcal{D}(B) = X \mid Bv \in \mathcal{D}(A) = Y\}$$

となるが, $B \in \mathcal{L}(X,Y)$ であるため, 任意の $v \in X$ に対して Bv は Y に属する. よって,

$$\mathcal{D}(AB) = \mathcal{D}(B) = X$$

となる. その上, A も B も線形作用素であることから, 任意の $u, v \in X$ と任意の $\alpha, \beta \in \mathbb{K}$ に対して

$$AB(\alpha u + \beta v) = A(B\alpha u + B\beta v) = A(\alpha Bu + \beta Bv) = A\alpha Bu + A\beta Bv = \alpha ABu + \beta ABv$$

となるため, 合成作用素 AB は定義域が X 全体となる線形作用素である. また, $A \in \mathcal{L}(Y,Z), B \in \mathcal{L}(X,Y)$ であるため, 任意の $u \in X$ について, 定理 (8) から

$$\|ABu\|_Z \leq \|A\|_{\mathcal{L}(Y,Z)}\|Bu\|_Y \leq \|A\|_{\mathcal{L}(Y,Z)}\|B\|_{\mathcal{L}(X,Y)}\|u\|_X$$

となり, 定義域が X 全体となる線形作用素 AB は有界な線形作用素である. よって AB は $\mathcal{L}(X,Z)$ に属する. その上,

$$\begin{aligned}
\|AB\|_{\mathcal{L}(X,Z)} &= \sup_{u \in X\backslash\{0\}} \frac{\|ABu\|_Z}{\|u\|_X} \\
&\leq \sup_{u \in X\backslash\{0\}} \frac{\|A\|_{\mathcal{L}(Y,Z)}\|B\|_{\mathcal{L}(X,Y)}\|u\|_X}{\|u\|_X} \\
&= \|A\|_{\mathcal{L}(Y,Z)}\|B\|_{\mathcal{L}(X,Y)}
\end{aligned}$$

■

定義 25 (X 上の恒等作用素)**.** X を Banach 空間とする. 任意の $u \in X$ に対して

$$Iu = u$$

となる $I \in \mathcal{L}(X)$ を X 上の恒等作用素と呼ぶ.

定理 10 (Neumann 級数)**.** X を Banach 空間とする. $B \in \mathcal{L}(X)$ とし, $I \in \mathcal{L}(X)$ を X 上の恒等作用素とする. もし

$$\|I - B\|_{\mathcal{L}(X)} < 1$$

ならば, B は逆作用素をもち $B^{-1} \in \mathcal{L}(X)$ となる. そのうえ,

$$B^{-1} = I + (I - B) + (I - B)^2 + \cdots = \sum_{i=0}^{\infty} (I - B)^i$$

で, かつ

$$\|B^{-1}\|_{\mathcal{L}(X)} \leq \frac{1}{1 - \|I - B\|_{\mathcal{L}(X)}}$$

証明.

$$S_n = I + (I - B) + (I - B)^2 + \cdots + (I - B)^n$$

とすると, B と I はともに $\mathcal{L}(X)$ に属するため, 加法 $I - B$ や合成作用素 $(I - B)(I - B)$ なども $\mathcal{L}(X)$ に属する. よって S_n も $\mathcal{L}(X)$ に属する.

続いて, 点列 $(S_n) \subset \mathcal{L}(X)$ が極限 S を $\mathcal{L}(X)$ 内に持つか確認する. 定理 (9) より

$$\|(I - B)^i\|_{\mathcal{L}(X)} \leq \|I - B\|^i_{\mathcal{L}(X)}, \quad i = 0, 1, \cdots$$

となるため, $n > m > 0$ となる整数に対して

$$\|S_n - S_m\|_{\mathcal{L}(X)} = \left\| \sum_{i=m+1}^{n} (I - B)^i \right\|_{\mathcal{L}(X)} \leq \sum_{i=m+1}^{n} \|I - B\|^i_{\mathcal{L}(X)}$$

となる. 定理の仮定より $\|I - B\|_{\mathcal{L}(X)} < 1$ であるため,

$$\sum_{i=m+1}^{n} \|I - B\|^i_{\mathcal{L}(X)} \to 0, \quad (n, m \to 0)$$

となる. よって,

$$\|S_n - s_m\|_{\mathcal{L}(X)} \to 0, \quad (n, m \to 0)$$

となるため, 点列 (S_n) は Cauchy 列である. その上, $\mathcal{L}(X)$ は Banach 空間であるため, 任意の Cauchy 列は極限を $\mathcal{L}(X)$ に持つため, 点列 (S_n) は

$$\|S_n - S\|_{\mathcal{L}(X)} \to 0, \quad (n, m \to 0)$$

となる極限 $S \in \mathcal{L}(X, Y)$ を持つ.

次に S が B^{-1} になることを示す. 合成作用素の定義 (22) にしたがって合成作用素 BS_n を考える. X は Banach 空間であり, $S, B_n \in \mathcal{L}(X)$ であるため, 定理 (9) より合成作用素 BS_n は $\mathcal{L}(X)$ に属する. その上, 点列 $(BS_n) \subset \mathcal{L}(X)$ は

$$\|BS_n - BS\|_{\mathcal{L}(X)} \leq \|B\|_{\mathcal{L}(X)} \|S_n - S\|_{\mathcal{L}(X)} \to 0, \quad (n \to \infty)$$

となるため, 極限 BS を $\mathcal{L}(X)$ 内にもつ. 一方で,

$$\begin{aligned} BS_n &= (I-(I-B))S_n \\ &= S_n - (I-B)S_n \\ &= \sum_{i=0}^{n}(I-B)^i - \sim_{i=1}^{n+1}(I-B)^i \\ &= I-(I-B)^{n+1} \end{aligned}$$

となり, 定理の仮定より $\|I-B\|_{\mathcal{L}(X)} < 1$ を持つため

$$\|BS_n - I\|_{\mathcal{L}(X)} = \|(I-B)^{n+1}\|_{\mathcal{L}(X)} \leq \|I-B\|_{\mathcal{L}(X)}^{n+1} \to 0,\ (n \to 0)$$

となるため, 点列 (BS_n) は極限 I も $\mathcal{L}(X)$ 内に持つ. よって, 極限の一意性より

$$BS = I$$

を得る. $\mathcal{R}(I) = X$ であるため, $\mathcal{R}(BS) = X$ である. その上, $X = \mathcal{R}(BS) \subset \mathcal{R}(B)$ と $\mathcal{R}(B) \subset X$ となるため,

$$\mathcal{D}(S) = \mathcal{R}(B) = X$$

となる.

同様の議論を $S_nB \in \mathcal{L}(X)$ について行うと

$$SB = I$$

と

$$\mathcal{D}(B) = \mathcal{R}(S) = X$$

が得られる. そのため, B は逆作用素を持ち, 逆作用素 $B^{-1} = S \in \mathcal{L}(X)$ である.

また,

$$S_n = I + (I-B) + (I-B)^2 + \cdots + (I-B)^n \to B^{-1},\ (n \to \infty)$$

より

$$B^{-1} = I + (I-B) + (I-B)^2 + \cdots = \sum_{i=0}^{\infty}(I-B)^i$$

となる.

最後に

$$\|B^{-1}\|_{\mathcal{L}(X)} = \left\|\sum_{i=0}^{\infty}(I-B)^i\right\|_{\mathcal{L}(X)} \leq \sum_{i=0}^{\infty}\|I-B\|^i$$

となり, 初項 1, 公比 $\|I-B\|_{\mathcal{L}(X)} < 1$ の総和より

$$\|B^{-1}\|_{\mathcal{L}(X)} \leq \frac{1}{1-\|I-B\|_{\mathcal{L}(X)}}$$

■

定理 11. X と Y を Banach 空間とする. $A \in \mathcal{L}(X,Y), R \in \mathcal{L}(Y,X)$ とする. もし RA が全単射ならば, A は単射であり, R は全射である.

証明.「A が単射」の証明

定理 (4) (線形作用素に対する単射性 (1)) の (2) を用いて証明する. $u \in X$ とし, RA が単射であることに注意すると

$$Au = 0 \Rightarrow RAu = 0 \Rightarrow u = 0$$

よって, A は単射である.

「R が全射」の証明

RA が全射であるため任意の $g \in X$ に対して,

$$RAu = g$$

となる $u \in X$ が存在する. その上, $v = Au$ とすると任意の $g \in X$ に対して

$$Rv = g$$

となる $v \in Y$ が存在するため, R は全射である. ■

定義 26 (Fréchet 微分)**.** 作用素 $F : X \to Y$ が $x_0 \in X$ で Fréchet 微分可能であるとは, ある有界線形作用素 $E : X \to Y$ が存在して

$$\lim_{\|h\|_X \to 0} \frac{\|F(x_0+h) - F(x_0) - Eh\|_Y}{\|h\|_X} = 0$$

が成り立つことをいう. このとき E は作用素 F の x_0 における Fréchet 微分といい, $E = DF(x_0)$ とも書く. もしも作用素 $F : X \to Y$ がすべての $x \in X$ に対して Fréchet 微分可能ならば, F は X において C^1-Fréchet 微分可能という.

2.3 Banach の不動点定理

定義 27 (不動点)**.** X を係数体が $\mathbb{K}$ の Banach 空間とする. M はからでない閉集合で $M \subset X$ を満たすとする. A を M から M への写像とする. $x \in M$ が A の不動点であるとは, x が

$$x = Ax$$

を満たすことである.

定義 28 (距離空間)**.** X をノルム空間とし, $x, y \in X$ に対して実数値を対応させる関数 $d(\cdot,\cdot) : X \times X \to \mathbb{R}$ が定義され,

1. $d(x,y) \geq 0$ かつ $d(x,y) = 0 \Leftrightarrow x = y, \quad x, y \in X$
2. $d(x,y) = d(y,x), \quad x, y \in X$
3. $d(x,y) \leq d(x,z) + d(z,y), \quad x, y, z \in X$

を満たすとき, d を距離空間という. 距離の備わった集合を距離空間という.

定義 29 (縮小写像)**.** X を係数体が $\mathbb{K}$ の Banach 空間とする. M はからでない閉集合で $M \subset X$ を満たすとする. 写像 $A : M \to M$ が k 次の縮小写像であるとは, $0 \leq k < 1$ を満たす定数 k が存在し, $\forall x, y \in M$ について

$$\|Ax - Ay\| \leq k\|x - y\|$$

を満たすことである.

定理 12 (Banach の不動点定理)**.** X を係数体が $\mathbb{K}$ の Banach 空間とする. M はからでない閉集合で $M \subset X$ を満たすとする. A は M から M への k 次の縮小写像とする. そのとき, 問題

$$Find \quad u \in M, \quad u = Au \tag{2.1}$$

は真の解 u^* を M 内にただ一つ持つ. 即ち, 写像 A は M 上にただ一つ不動点 u^* を持つ.

証明. u_0 を閉集合 M の元として与えられていると仮定する. 点列 (u_n) は反復法

$$u_{n+1} = Au_n, \quad n = 0, 1 \cdots \tag{2.2}$$

によって得られる. そのとき, 証明のプロセスは次のように考える.

1. (u_n) が Cauchy 列になること, さらに Banach 空間の完備性を使うことで, $u_n \to u,\ n \to \infty$ となる u が X 内に存在することを示す.
2. u が (2.1) を満たす真の解 u^* と一致することを示す (解の存在性).
3. 真の解 u^* が M 内で一意であることを示す.

1

(2.2) より

$$\|u_n - u_{n+1}\| = \|Au_{n-1} - Au_n\|$$

となる. 仮定より A は k 次の縮小写像であるため,

$$\|Au_{n-1} - Au_n\| \leq k\|u_{n-1} - u_n\|$$

となる定数 k が存在する. 同様に $\|u_{n-1} - u_n\|$ に (2.2) と A の縮小写像の性質を使うと最終的に

$$\|u_n - u_{n+1}\| \leq k^n\|u_0 - u_1\| \tag{2.3}$$

を得る.

次に三角不等式より, $n = 0, 1, 2 \cdots, \quad m > n$ について

$$\begin{aligned} \|u_n - u_m\| &= \|(u_n - u_{n+1}) + (u_{n+1} - u_{n+2}) + \cdots + (u_{m-1} - u_m)\| && (2.4)\\ &\leq \|u_n - u_{m+1}\| + \|u_{n+1} - u_{n+2}\| + \cdots + \|u_{m-1} - u_m\| && (2.5) \end{aligned}$$

となる. 上の式に (2.3) を適用すると

$$\begin{aligned} \|u_n - u_m\| &\leq \|u_n - u_{m+1}\| + \|u_{n+1} - u_{n+2}\| + \cdots + \|u_{m-1} - u_m\| && (2.6)\\ &\leq k^n\|u_0 - u_1\| + k^{n+1}\|u_0 - u_1\| + \cdots + k^{m-1}\|u_0 - u_1\| && (2.7)\\ &= k^n(1 + k + \cdots + k^{m-n-1})\|u_0 - u_1\| && (2.8) \end{aligned}$$

となる. ここで, k は $0 \leq k < 1$ であるため, $1 + k + \cdots + k^{m-n-1} \leq 1 + k + \cdots + k^{m-1}$ となる. さらに, 等比級数から

$$1 + k + \cdots + k^{m-1} = \frac{1 - k^m}{1 - k} \tag{2.9}$$

であるため, (2.6) は

$$\|u_n - u_m\| \leq \frac{k^n(1 - k^m)}{1 - k}\|u_0 - u_1\| \tag{2.10}$$

となる. よって, k は $0 \leq k < 1$ から $k^n \to 0,\ n \to \infty$ と $k^m \to 0,\ m \to \infty$ となる. 即ち,

$$\|u_n - u_m\| \to 0, \quad (n, m \to \infty)$$

となるため, 点列 (u_n) は Cauchy 列である. さらに X は Banach 空間であるため, X は完備である. 即ち, 任意の Cauchy 列が X の中で極限を持つ. よって, 点列 (u_n) は

$$u_n \to u, \quad n \to \infty$$

となる $u \in X$ が存在する.

2

t_0 を M の元とする. 仮定より A は M から M への写像であるため, $A(M) \subseteq M$ となる. 即ち, $u_1 = Au_0$ が成立する $u_1 \in M$ が存在する. 同様に, $\forall n \in \mathbb{N}$ について $u_n \in M$ が存在する. さらに, M は閉集合であるため, I で存在を証明した u は M に属する. よって A_u も M に属する. そのうえで仮定より A は k 次の縮小写像であるため,

$$\|Au_n - Au\| \leq k\|u_n - u\|$$

を得る. 1 より点列 (u_n) は極限を持つため, $\|u_n - u\| \to 0,\ n \to \infty$ となる. 即ち

$$\|Au_n - Au\| \to 0, \quad n \to \infty$$

となるため, Au は Au_n の極限である. よって, (2.2) について $n \to \infty$ とすると

$$u = Au$$

が成立する. よって u は (2.1) を満たす真の解 u^* となる.

3

$u^*, v^* \in M$ を去れぞれ $u^* = Au^*$ と $v^* = Av^*$ を満たすとする. その時, A は k 次の縮小写像であるため,

$$\|u^* - v^*\| \leq \|Au^* - Av^*\| \leq k\|u^* - v^*\|$$

を得る. ここで $0 \leq k < 1$ であるため, 不等式を満たすものは $u^* = v^*$ の場合のみである. 即ち, (2.1) を満たす真の解は一意である. ■

2.4 簡易ニュートン写像

本節では, X, Y を Banach 空間とし, 写像 $F : X \to Y$ に対して

$$F(x) = 0 \quad \in Y$$

という（非線形）作用素方程式を考える.

このとき, 写像 $T : X \to X$ を

$$T(x) := x - AF(x)$$

で定義する. これを簡易ニュートン写像という. ここで $A : Y \to X$ はある全単射な線形作用素である. いま $\bar{x}$ を $F(x) \approx 0$ となる近似解とし, $\bar{x}$ の近傍を

$$B(\bar{x}, r) := \{x \in X : \|x - \bar{x}\| < r\} \text{ (開球)}$$
$$\overline{B(\bar{x}, r)} := \{x \in X : \|x - \bar{x}\| \leq r\} \text{ (閉球)}$$

で定義する. このときもし, $B(\bar{x}, r)$ 上で写像 T が縮小写像となれば, Banach の不動点定理から $F(\tilde{x}) = 0$ をみたす解 $\tilde{x} \in B(\bar{x}, r)$ がただ一つ存在することになる. このように解の存在を仮定せずに近似解近傍での収束をいう定理を Newton-Kantorovich の定理という.

2.5 フーリエ級数

2.5.1 フーリエ級数の導出

ある関数 $f(x)$ が有限の閉空間 $[a, b]$ で定義されている場合に, 次の性質をもつとき区分的に連続であるという.

1. $f(x)$ は有限個の不連続点を除いて連続である.
2. $f(x)$ の不連続点 c では $f(c + 0)$ と $f(c - 0)$ が存在する.

また, $f(x)$ が無限区間で定義されている場合には, $f(x)$ が任意の有限区間で区分的に連続であるときに, $f(x)$ は区分的に連続であるという. 以降は, 特にことわらない限り区分的に連続な関数だけを考えることにする.

関数 $f(x)(x \in [-\pi, \pi])$ を周期 2π の周期関数とする. このような $f(x)$ が以下の三角級数で表されるとして, 係数 $a_0, a_1, a_2, \cdots, b_0, b_1, b_2, \cdots$ を求めることを考える.

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx) \qquad (2.11) \\ &= \frac{a_0}{2} + (a_1 \cos x + b_1 \sin x) + (a_2 \cos 2x + b_2 \sin 2x) + \cdots \end{aligned}$$

そのために, 次の公式を用意する.

$$\int_{-\pi}^{\pi} \cos nx \cos mx dx = 0 (n \neq m)$$
$$\int_{-\pi}^{\pi} \cos nx^2 dx = \pi$$
$$\int_{-\pi}^{\pi} \sin nx \sin mx dx = 0 (n \neq m)$$
$$\int_{-\pi}^{\pi} \cos nx^2 dx = \pi$$
$$\int_{-\pi}^{\pi} \sin nx \cos mx dx = 0 \tag{2.12}$$

ここで, $m, n \in \mathbb{N}$ である. また, 明らかに次の公式も成り立つ.

$$\int_{-\pi}^{\pi} \sin nx dx = \int_{-\pi}^{\pi} \cos nx dx = 0 \tag{2.13}$$

ここで, 次のような形式的計算が許されると仮定する. まず, (2.11) の両辺を $-\pi$ から π まで積分して, (2.13) を利用すれば,

$$\begin{aligned}\int_{-\pi}^{\pi} f(x) dx &= \frac{a_0}{2} \int_{-\pi}^{\pi} dx + \sum_{n=1}^{\infty} (a_n \int_{-\pi}^{\pi} \cos nx dx + b_n \int_{-\pi}^{\pi} \sin nx dx) \\ &= \pi a_0\end{aligned}$$
$$\therefore a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) dx \tag{2.14}$$

次に, (2.11) の両辺に $\cos mx$ を掛けて, $-\pi$ から π まで積分すれば, 公式 (2.12),(2.13) によって

$$\begin{aligned}\int_{-\pi}^{\pi} f(x) \cos mx dx &= \frac{a_0}{2} \int_{-\pi}^{\pi} \cos mx dx \\ &\quad + \sum_{n=1}^{\infty} (a_n \int_{-\pi}^{\pi} \cos nx \cos mx dx + b_n \int_{-\pi}^{\pi} \sin nx \cos mx dx) \\ &= \pi a_m\end{aligned}$$
$$\therefore a_m = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos mx dx \tag{2.15}$$

同様に, (2.11) の両辺に $\sin mx$ を掛けて, $-\pi$ から π まで積分することによって次の式が得られる.

$$b_m = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin mx dx \tag{2.16}$$

(2.13), (2.15), (2.16) をまとめて次のようになる.

定義 30 (フーリエ級数)**.**

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx$$

ただし

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos nx dx, (n \geq 0)$$
$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin nx dx, (n \geq 1)$$

この無限級数を f のフーリエ級数といい, a_n, b_n をフーリエ係数という.

定義 31 (複素フーリエ級数)**.** また, $\cos nx = \frac{e^{inx}+e^{(-inx)}}{2}, \sin nx = \frac{e^{inx}-e^{(-inx)}}{2i}(i = \sqrt{-1}$ は虚数単位) という関係を用いて

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}$$
$$c_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}dx$$

と複素数を用いた形式も考えられる. これを f の複素フーリエ級数, c_k を複素フーリエ係数という.

これらには関係式

$$c_k = \begin{cases} a_k/2 & k = 0 \\ \frac{a_k - ib_k}{2} & k > 0 \\ \frac{a_{-k}+ib_{-k}}{2} & k < 0 \end{cases}$$

があり, 変換可能である.

2.5.2 フーリエ級数の性質

対称性

周期関数 $f(x)$ が偶関数, すなわち $f(x) = f(-x)$ を満たすならば, サイン関数の係数 b_n が $b_n = 0$ になる. 偶関数のフーリエ級数は

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos nx$$

と表すことができる. このようにコサイン関数のみで表されるフーリエ級数のことをフーリエ・コサイン級数という. このとき複素フーリエ係数は $c_{-k} = -c_k$(コサイン対称性) が成り立つ. 一方で, $f(x)$ が奇関数の性質 $f(x) = -f(x)$ を満たすとすると, コサインの係数 a_n が $a_n = 0$ になるので, この関数のフーリエ級数は,

$$f(x) = \sum_{n=1}^{\infty} b_n \sin nx$$

と表すことができる. このようにサイン関数のみで表されるフーリエ級数のことをフーリエ・サイン級数という. このとき $c_{-k} = -c_k$ も成り立つ.

実数値関数

$f(x)$ が実数値関数 $f(x) \in \mathbb{R}$ であるとき, 各フーリエ係数 a_n, b_n は

$$a_n, b_n \in \mathbb{R}$$

となる. このとき複素フーリエ係数 c_k は

$$c_{-k} = \overline{c_k}$$

を満たす. これは, $f(x) = \overline{f(x)}$ という実数値関数の性質を使うことで確認できる.

係数の収束

ある周期関数 $f(x)$ のフーリエ係数を a_n として, $n \to \infty$ での収束のオーダーを考えると

$$a_n = \begin{cases} \mathcal{O}(n^{-k}) & k\text{次オーダーの収束} \\ \mathcal{O}(e^{-qn^r}) & q : \text{定数}, r > 0, \text{指数オーダーの収束} \\ \mathcal{O}(e^{-qb\log(n)}) & \text{超幾何収束} \end{cases}$$

などのパターンがあり, それぞれ $f(x)$ の滑らかさによって決まる. 例えば, $f(x)$ が k 次オーダーの収束をする場合は, 関数 f は C^k-級 (k 階連続微分可能) の関数である. 指数オーダーの収束の場合は実解析関数 (極や分岐点を持つ一般的な有限区間/無限区間上の関数), 超幾何収束の場合は複素平面上で ∞ 以外で特異点を持たない関数 (整関数, entire function) とそれぞれ知られている.

ランダウの記号 $\mathcal{O}$ は, 例えば $a_n = \mathcal{O}(n^{-2})$ $(n \to \infty)$ は, 左辺の絶対値が右辺の絶対値の定数倍以下であることを表す. すなわち, ある定数 C があって, n が十分大きいすべての $n \geq 0$ に対して, $|a_n| \leq \frac{C}{n^2}$ が成立することを意味する.

微分・シフト

フーリエ級数の便利な性質として微分およびシフト (平行移動) がある.

いま $f(x)$ を複素フーリエ級数とすると微分操作は

$$\frac{d}{dx} f(x) \sim \sum_{k=-\infty}^{\infty} (ik) c_k \exp(ikx)$$

と各係数に ik を乗じることで計算できる.

同様に, シフト操作も

$$f(x-d) \sim \sum_{k=-\infty}^{\infty} c_k \exp(ik(x-d)) = \sum_{k=-\infty}^{\infty} (\exp(-ikd) c_k) \exp(ikx)$$

とすることでシフト d 分だけ x 方向に平行移動できる. このときは各係数に $\exp(-ikd)$ を乗じる. 以後, 簡単のために特に混合がなければ複素フーリエ級数も単にフーリエ級数と呼ぶ.

2.5.3 フーリエ級数の数値計算

与えられた関数のフーリエ係数を計算をする

周期関数 $f(x)$ のフーリエ係数 c_k を数値計算で求めることを考える. 無限級数を計算機で表現

することは難しいため, 係数を打ち切る操作が必要である. この時のフーリエ係数の添字のサイズ N を $|k| < N$ となるように定める ($N-1$ を最大端数という).
いま $0 = x_0 \leq x_1 \leq \cdots \leq x_{2N-1} = 2\pi$ と区間 $[0, 2\pi]$ を等間隔に分割した点 $x_j = jh$ ($j = 0, \cdots, 2N-1, h = 2\pi/(2N-1)$) を標本点といい, 標本点上での関数値を用いてフーリエ係数の近似 $\overline{c_k}$ を得る.

$$c_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)\exp(-kx)dx \approx \frac{1}{2N-1}\sum_{j=0}^{2N-2} f(x_j)\exp(-2\pi i\frac{kj}{2N-1}) = \overline{c_k},\ (|k| < N)$$

この $\overline{c_k}$ の式は, 離散フーリエ変換 $a_k = \mathcal{F}_k(b) = \sum_{j=0}^{2M-2} b_j \exp(-2\pi i\frac{jk}{2M-1})$ そのものである. したがって高速フーリエ変換 (FFT) を用いて近似フーリエ係数 $\overline{c_k}$ を計算できる. そして $\overline{c_k}$ を使って, 元の関数 $f(x)$ の近似 $f^{(N)}(x)$ が

$$f(x) \approx f^{(N)}(x) = \sum_{|k|<N} \overline{c_k}\exp(ikx)$$

と得られる.

フーリエ係数から元の関数の概形を求める

近似関数 $f^{(N)}(x)$ の係数 c_k から関数の概形をプロットする.
いま, 標本点上での関数値は

$$f^{(N)}(x_j) = \sum_{|k|<N} \overline{c_k}\exp(ikx_j) = \sum_{|k|<N} \overline{c_k}\exp(2\pi i\frac{kj}{2N-1})$$

これは逆離散フーリエ変換に相当する. そこで逆高速フーリエ変換 (IFFT) を用いて元の関数の概形を求める. しかし, このまま IFFT を用いると, 標本点と同じ数の関数値しか得られず, グラフに描画するとギザギザ下粗い概形になってしまう. そこでフーリエ係数 $\overline{c_k}$ に 0 を余分に貼り合わせて, 滑らかなグラフのプロットを得る.

周期が 2π 以外の場合の取り扱い方

周期が 2π 以外の場合も, ほぼ同様な考え方でフーリエ級数を考えられる.
いま $f(t)$ を周期 L の周期関数とする. このとき変数 $t : a \to b(L = b-a)$ に対して, 変数 x を $x = \omega(t-a)(\omega = 2\pi/L)$ と定めると, $x : 0 \to 2\pi$ となり, 関数 $g(x) = f(a + \omega^{-1}x)$ は周期 2π の周期関数である. そこで $g(x)$ がフーリエ級数

$$g(x) = \sum_{k\in\mathbb{Z}} c_k \exp(ikx)$$

で表されたとすると, $f(t)$ のフーリエ級数は

$$f(t) = g(\omega(t-a)) = \sum_{k\in\mathbb{Z}} c_k \exp(ik\omega(t-a))$$

が成り立つ. フーリエ係数は $dx = \omega dt$ より

$$\begin{aligned} c_k &= \frac{1}{2\pi}\int_0^{2\pi} g(x)\exp(-ikx)dx \\ &= \frac{1}{2\pi}\int_a^b g(\omega(t-a))\exp(-ik\omega(t-a))\omega dt \\ &= \frac{1}{L}\int_a^b f(t)\exp(-ik\omega(t-a))dt \quad (k\in\mathbb{Z}) \end{aligned}$$

と f の離散フーリエ変換で求められる. したがって, サイズ N のフーリエ係数は標本点を $a = t_0 \le t_1 \le \cdots \le t_{2N-1} = b$ と区間 $[a,b]$ を等間隔に分割した点 $t_j = a + jh$ $(j = 0,\cdots,2N-1,\ h = L/(2N-1))$ でとり, この標本点上での関数値を用いてフーリエ係数の近似 $\overline{c_k}$ を得る.

$$c_k = \frac{1}{L}\int_a^b f(t)\exp(-ik\omega(t-a))dt \approx \frac{1}{2N-1}\sum_{j=0}^{2N-2} f(t_j)\exp(-2\pi i\frac{kj}{2N-1}) = \overline{c_k},\ (|k|<N)$$

ここで, $t_j = a + \frac{jL}{2N-1}$ $(j = 0,1,\cdots,2N-2)$ から, 周期が 2π の周期関数のフーリエ係数の近似が同じ式になるため, フーリエ係数の計算方法は変わらない. 標本点がリスケール $(x_j = \omega(t_j - a))$ されたものを使っている.

2.6 チェビシェフ級数

2.6.1 チェビシェフ級数の導出

定義 32 (チェビシェフ級数)**.** チェビシェフ級数とはフーリエ・コサイン級数に対し変換 $x = \cos\theta$ を与えたものである.

$$T_n(x) := \cos n\theta, \quad \theta = \arccos x$$

を n 次の第一種チェビシェフ多項式という.

$$f(x) = \sum_{n=0}^{\infty} a_n T_n(x), \quad x\in[-1,1]$$

チェビシェフ多項式を基底とした級数展開をチェビシェフ級数という.

これは変換 $x = \cos\theta$ と $T_n(x)$ の定義から

$$f(\cos\theta) = \sum_{n=0}^{\infty} a_n \cos n\theta, \quad \theta\in[0,2\pi]$$

となるため, $g(\theta) = f(\cos\theta)$ という周期関数のフーリエ・コサイン級数である. したがって係数の収束もフーリエ級数と似ている. チェビシェフ係数は

$$a_n = \begin{cases} \frac{1}{\pi}\int_{-1}^{1}\frac{f(x)T_0(x)}{\sqrt{1-x^2}}dx & (n=0) \\ \frac{2}{\pi}\int_{-1}^{1}\frac{f(x)T_n(x)}{\sqrt{1-x^2}}dx & (n\ge 1) \end{cases}$$

で与えられる. この係数 $a_n (n \ge 0)$ を Two-sided チェビシェフ級数と呼ぶ.

$T_n(x) = \cos n\theta$ なので, $\cos 0 = 1,\ \cos 1\cdot\theta = \cos\theta,\ \cos 2\theta = 2\cos^2\theta - 1,\ \cos 3\theta = 4\cos^3\theta - 3\cos\theta,\ \cos 4\theta = 8\cos^4\theta - 8\cos^2\theta + 1, \cdots$ となる. これから

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ &\vdots \end{aligned}$$

となり, 一般的には

$$\cos n\theta + \cos(n-2)\theta = 2\cos\theta\cos(n-1)\theta$$

という三角関数の公式を使って,

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)\ (n = 1, 2, \cdots), \quad T_0(x) = 1, \quad T_1(x) = x$$

という漸化式により, 多項式が決定される.

チェビシェフ多項式のもう一つの表現方法として, $|z| = 1$ をみたす複素平面の単位円上で定義される複素変数 z を用いる方法がある. いま

$$F(z) = f(x), \quad x = \frac{z + z^{-1}}{2}, \quad z \in \{z \in \mathbb{C} : |z| = 1\}$$

をみたすような複素関数 F を考える. 変換 $x = (z + z^{-1})/2$ は x の 1 つの値に対して z の 2 つの値 (z, z^{-1}) を対応させる. すなわち F は $F(z) = F(z^{-1})$ の対称性が成り立つ. 関数 F のローラン展開 (Laurent expansion) を考えると

$$F(z) = F(z^{-1}) = \frac{1}{2}\sum_{n=0}^{\infty} a_n(z^n + z^{-n}), \quad |z| = 1$$

よって n 番目のチェビシェフ多項式 T_n は, 単位円上の変数 z^n の実部で定義できて

$$T_n(x) = \frac{1}{2}(z^n + z^{-n}), \quad n \geq 0$$

と表すことができる. この表現方法はチェビシェフ多項式がみたす漸化式を簡潔に導く. すなわち任意の $n \geq 1$ について

$$\frac{1}{2}(z + z^{-1})(z^n + z^{-n}) = \frac{1}{2}(z^{n+1} + z^{-n-1}) + \frac{1}{2}(z^{n-1} + z^{-n+1})$$

が成り立つので, (2.6.1) の漸化式を得る.

一般の定義域 $x \in [a, b]$ 上でのチェビシェフ多項式は

$$\begin{aligned} &[a, b] \to [-1, 1], \quad x \mapsto \xi = 2\frac{x - a}{b - a} - 1 \\ &[-1, 1] \to [a, b], \quad \xi \mapsto x = \frac{1 - \xi}{2}a + \frac{1 + \xi}{2}b \end{aligned}$$

という変数変換を使う.

2.6.2 チェビシェフ級数の性質

対称性

チェビシェフ多項式 $T_n(x)$ は n が偶数番目のとき偶関数, 奇数番目のとき機関数となる. したがって次のような対称性が成り立つ.

- 周期関数 $f(x)$ が偶関数, すなわち $f(-x) = f(x)$ をみたす ($x = 0$ で対称) ならば, a_0 の偶数番目 $(\cos 2n\theta, n = 1, 2, \cdots)$ だけ必要. 奇数番目は 0 となる.
- 周期関数 $f(x)$ が関数, すなわち $f(-x) = -f(x)$ をみたすならば, a_0 の奇数番目 $(\cos(2n-1)\theta, n = 1, 2, \cdots)$ だけ必要. 偶数番目は 0 となる.

チェビシェフ (-Lobatto) 点

区間 $[-1, 1]$ の点

$$x_j = \cos\frac{j\pi}{n}, \quad 0 \le j \le n$$

という点を n 次チェビシェフ多項式のチェビシェフ点 (-Lobatto) という. この点は複素平面上の単位円の円周を当分割した点 $z_j = \exp(i\theta_j)$ $(\theta_j = \frac{j\pi}{n})$ の実数部分 $x_j = \mathbb{R}[z_j]$ である. チェビシェフ点において, $T_n(x_j) = \pm 1$ となる.

フーリエ級数との関係

チェビシェフ級数はフーリエ級数への変換が容易である.

$$\begin{aligned} f(x) &= c_0 + 2\sum_{n=1}^{\infty} c_n T_n(x) \\ &= c_0 + 2\sum_{n=1}^{\infty} c_n \cos n\theta, \quad (x = \cos\theta) \\ &= \sum_{k=-\infty}^{\infty} c_{|k|} \exp(ik\theta) \end{aligned}$$

ここで

$$c_k = \frac{1}{2\pi}\int_0^{2\pi} f(\cos\theta) e^{-ik\theta} d\theta \quad (k \in \mathbb{Z}) \quad (c_{-k} = c_k)$$

が成り立つため, フーリエ係数 $c_{|k|}$ は計算できる. この $(c_n)_{n\ge 0}$ を (One-sided) チェビシェフ係数と呼ぶ. さらに, $|k| < M$ の範囲で打ち切ったチェビシェフ係数 $\overline{c_k}$ は離散フーリエ変換を用いて

$$c_k \approx \frac{1}{2M-2}\sum_{j=0}^{2M-3} f(\cos\theta_j) e^{-\pi i \frac{kj}{M-1}} = \overline{c_k} \quad (\theta_j = \frac{\pi j}{M-1},\ h = \frac{\pi}{M-1},\ |k| < M)$$

と近似できる. ここで $f(\cos\theta_j)$ はチェビシェフ点 $x_j = \cos\theta_j$ における関数 f の値を表す.

2.7 高速フーリエ変換

定義 33 (離散フーリエ変換)**.** $b=(b_0,\cdots,b_{2M-2})\in\mathbb{C}^{2M-1}$ に対して, $a=\mathcal{F}(b)\in\mathbb{C}^{2M-1}$ を

$$a_k=\mathcal{F}_k(b):=\sum_{j=0}^{2M-2}b_j\exp\left(2\pi i\frac{jk}{2M-1}\right),\quad k=-M+1,\cdots,M-1$$

とし, 離散フーリエ変換 (DFT) と呼ぶ.

定義 34 (逆離散フーリエ変換)**.** $a=(a_k)_{|k|<M}=(a_{-M+1},\cdots,a_{M-1})\in\mathbb{C}^{2M-1}$ に対して, $b=\mathcal{F}^{-1}(a)\in\mathbb{C}^{2M-1}$ を

$$b_j=\mathcal{F}_j^{-1}(a):=\sum_{k=-M+1}^{M-1}a_k\exp\left(2\pi i\frac{jk}{2M-1}\right),\quad j=0,\cdots,2M-2$$

とし, 逆離散フーリエ変換 (IDFT) と呼ぶ.

2.8 畳み込みの FFT アルゴリズム

2.8.1 畳み込み定理

u_1,u_2: 周期 L の周期関数とする.

$$u_1(t)=\sum_{k\in\mathbb{Z}}a_k^{(1)}\exp(ik\omega t),\quad a^{(1)}=(a_k^{(1)})_{k\in\mathbb{Z}}$$
$$u_2(t)=\sum_{k\in\mathbb{Z}}a_k^{(2)}\exp(ik\omega t),\quad a^{(2)}=(a_k^{(2)})_{k\in\mathbb{Z}}$$

とする. 但し, $\omega=2\pi/L$(周波数) としたとする. このとき関数の積 u_1u_2 は次のようにフーリエ級数で表せる.

$$u_1(t)u_2(t)=\sum_{k\in\mathbb{Z}}\left(a^{(1)}*a^{(2)}\right)_k\exp(ik\omega t)$$

ここで $a=(a_k)_{k\in\mathbb{Z}}$, $b=(b_k)_{k\in\mathbb{Z}}$ に対して

$$(a*b)_k=\sum_{\substack{k_1+k_2=k\\k_1,k_2\in\mathbb{Z}}}a_{k_1}b_{k_2},\quad k\in\mathbb{Z}$$

を離散畳み込み (discrete convolution) という. スペクトル法の非線形項の計算, i.e., $\mathrm{u}_1,\mathrm{u}_2$ のような (有限モードのフーリエ級数で表される) 周期関数が p 個 ($p\in\mathbb{N}$) あったとき,

$$u_l(t)=\sum_{|k|<M}a_k^l\exp(ik\omega t),\quad l=1,\cdots,p$$

とすると, 離散畳み込みはこれらの周期関数の積

$$\begin{aligned} u_1(t)u_2(t)\cdots u_p(t) &= \sum_{k\in\mathbb{Z}}(a^{(1)} * a^{(2)} * \cdots * a^{(p)}_k)_k \exp(ik\omega t) \\ &= \sum_{|k|\le p(M-1)} (a^{(1)} * a^{(2)} * \cdots * a^{(p)})_k \exp(ik\omega t) \end{aligned}$$

を表すことになる. ここで

$$(a^{(1)} * \cdots * a^{(p)})_k = \sum_{\substack{k_1+\cdots+k_p=k \\ |k|\le p(M-1) \\ |k_1|,\cdots,|k_p|<M}} a^{(1)}_{k_1}\cdots a^{(p)}_{k_p}$$

と表される.

畳み込み定理

畳み込みを離散フーリエ変換したものは, それぞれの離散フーリエ変換の積になる.

$$\begin{aligned} \mathcal{F}(a^{(1)} * a^{(2)} * \cdots * a^{(p)}) &= \mathcal{F}(a^{(1)})\mathcal{F}(a^{(2)})\cdots\mathcal{F}(a^{(p)}) \\ &= b^{(1)}b^{(2)}\cdots b^{(p)} \end{aligned}$$

但し, ベクトルの積は要素ごとの積を表す.

2.8.2 畳み込みの FFT アルゴリズム

p 個の有限点列 $\{a^{(l)}_k\}_{|k|<M}$ に対して, FFT を使って

$$(a^{(1)} * a^{(2)} * \cdots * a^{(p)})_k = \sum_{\substack{k_1+k_2+\cdots+k_p=k \\ |k_l|<M}} a^{(1)}_{k_1}a^{(2)}_{k_2}\cdots a^{(p)}_{k_p}, \quad |k| \le p(M-1)$$

を計算する.

入力:

$a^{(l)} = (a^{(l)}_k)_{|k|<M} \in \mathbb{C}^{2M-1} \quad (l = 1, \cdots, p)$

Step1:

エイリアシングエラーを防ぐために, 入力された値 $a^{(l)}$ の両脇に $(p-1)M$ 個の 0 を付け加える. これを $\tilde{a}^{(l)}$ と書く.

$$\tilde{a}^{(l)}_j = \begin{cases} a^{(l)}_j, & |j| < M \\ 0, & M \le |j| \le pM-1 \end{cases}$$

というルールで $\tilde{a} \in \mathbb{C}^{2pM-1}$ を作る.

$$\tilde{a}^{(l)}_j = (\underbrace{0,\cdots,0}_{(p-1)M}, \underbrace{a^{(l)}_{-M+1},\cdots,a^{(l)}_{M-1}}_{2M-1}, \underbrace{0,\cdots,0}_{(p-1)M}) \in \mathbb{C}^{2pM-1}$$

Step2:

$\tilde{b}^{(l)} = (\tilde{b}^{(l)}_j)_{j=0,\cdots,2pM-2} \in \mathbb{C}^{2pM-1}$ を $\tilde{a}^{(l)}$ の逆離散フーリエ変換を行い

$$\tilde{b}^{(l)}_j := \mathcal{F}^{-1}_j(\tilde{a}^{(l)}) = \sum_{k=-pM+1}^{pM-1} \tilde{a}^{(l)}_k \exp\left(2\pi i \frac{jk}{2pM-1}\right)$$

で定める.

Step3:

ベクトルの要素毎の積を計算する.

$$(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)})_j := \tilde{b}_j^{(1)}\cdots\tilde{b}_j^{(p)}, \quad j=0,\cdots,2pM-2$$

Step4:

Step3 で計算した $(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)})$ に対して, 離散フーリエ変換を行うと, $k=-pM+1,\cdots,pM-1$ で

$$\begin{aligned}\mathcal{F}_k(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)}) &= \sum_{j=0}^{2pM-2}\tilde{b}_j^{(1)}\hat{*}\cdots\hat{*}\tilde{b}_j^{(p)}\exp\left(-2\pi i\frac{jk}{2pM-1}\right)\\ &= \sum_{j=0}^{2pM-2}\underbrace{\prod_{l=1}^{p}\left(\sum_{k_l=-pM+1}^{pM-1}\tilde{a}_{k_l}^{(l)}\exp\left(2\pi i\frac{jk_l}{2pM-1}\right)\right)\exp\left(-2\pi i\frac{jk}{2pM-1}\right)}_{=:S_k(j)}\end{aligned}$$

いま

$$\begin{aligned}S_k(j) &:= \prod_{l=1}^{p}\left(\sum_{k_l=-pM+1}^{pM-1}\tilde{a}_{k_l}^{(l)}\exp\left(2\pi i\frac{jk_l}{2pM-1}\right)\right)\exp\left(-2\pi i\frac{jk}{2pM-1}\right)\\ &= \underbrace{\sum_{\substack{k_1+k_2+\cdots+k_p=k\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)} + \sum_{m=1}^{p}\left(\sum_{\substack{k_1+k_2+\cdots+k_p=k\pm m(2pM-1)\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}\right)}_{k_1+k_2+\cdots+k_P-k\equiv 0 \mod 2pM-1,\ \exp(2\pi ij)\equiv 1}\\ &\quad + \underbrace{\sum_{\substack{k_1+k_2+\cdots+k_p\notin\{k\pm m(2pM-1):m=0,\cdots,p\}\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}\exp\left(2\pi i\frac{k_1+k_2+\cdots+k_p-k}{2pM-1}j\right)}_{k_1+k_2+\cdots+k_p-k\not\equiv 0 \mod 2pM-1}\end{aligned}$$

として

$$\begin{aligned}\mathcal{F}(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)}) &= \sum_{j=0}^{2pM-2}S_k(j)\\ &= (2pM-1)\sum_{\substack{k_1+k_2+\cdots+k_p=k\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}\\ &\quad +(2pM-1)\sum_{m=1}^{p}\left(\sum_{\substack{k_1+k_2+\cdots+k_p=k\pm m(2pM-1)\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}\right)\\ &\quad +\sum_{\substack{k_1+k_2+\cdots+k_p\\ \notin\{k\pm m(2pM-1):m=0\cdots,p\}\\ |k_l|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}\left(\sum_{j=0}^{2pM-2}\exp\left(2\pi i\frac{k_1+k_2+\cdots+k_p-k}{2pM-1}j\right)\right)\end{aligned}$$

このときオイラーの公式から $k_1+k_2+\cdots+k_p-k\equiv 0 \mod 2pM-1$ のとき

$$\sum_{j=0}^{2pM-2}\exp\left(2\pi i\frac{k_1+k_[2]+\cdots+k_p-k}{2pM-1}j\right)=0$$

そして $|k_1|,\cdots,|k_p|<M$ と $|k|\le p(M-1)$(最高次) から

$$k_1+k_2+\cdots+k_p-k\in\{-2p(M-1),\cdots,2p(M-1)\}$$

となり, $k_1+k_2+\cdots+k_p=k\pm m(2pM-1)$ $(m=1,\cdots,p)$ から

$$\begin{aligned}m(2pM-1)&=k_1+k_2+\cdots+k_p-k\\&\in\{-2p(M-1),\cdots,2p(M-1)\}\end{aligned}$$

となる. しかしこれはあり得ない. なぜなら

$$2pM-1>2p(M-1)$$

なので $m=1$ としてもはみ出してしまう. すなわちは 0 になる. 最終的に

$$\sum_{\substack{k_1+k_2+\cdots+k_p=k\\|k_m|<M}}a_{k_1}^{(1)}\cdots a_{k_p}^{(p)}=\frac{1}{2pM-1}\mathcal{F}_k(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)}),\quad |k|\le p(M-1)$$

を得る.

まとめると FFT アルゴリズムは

入力： $a^{(m)}=(a^{(m)_k})_{|k|<M}(m=1,\cdots,p$ フーリエ・チェビシェフ級数)

1. $\tilde{a}_j^{(m)}=(\underbrace{0,\cdots,0}_{(p-1)M},\underbrace{a_{-M+1}^{(m)},\cdots,a_{M-1}^{(m)}}_{2M-1},\underbrace{0,\cdots,0}_{(p-1)M})\in\mathbb{C}^{2pM-1}$
2. $\tilde{b}^{(m)}=\mathcal{F}^{-1}(\tilde{a}^{(m)})\in\mathbb{C}^{2pM-1}$(IFFT を使う)
3. $(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)})_j=\tilde{b}_j^{(1)}\cdots\tilde{b}_j^{(p)}$ $(j=0,\cdots,2pM-2)$
4. $c_k=\frac{1}{2pM-1}\mathcal{F}_k(\tilde{b}^{(1)}\hat{*}\cdots\hat{*}\tilde{b}^{(p)})$ $(|k|\le p(M-1))$

出力： $c=(c_k)_{|k|\le p(M-1)}\in\mathbb{C}^{2p(M-1)+1}$

3 Newton Kantorovich の定理を用いた精度保証付き数値計算

X, Y を Banach 空間とし, 非線形方程式

$$F(u) = 0,\ u \in X$$

を求める問題を考える. この問題に対して近似解の品質保証を行う定理を紹介する.

3.1 radii polynomial approach [6]

定理 13 (Newton-Kantorovich type argument)**.** 有界線形作用素 $A^\dagger \in \mathcal{L}(X,Y), A \in \mathcal{L}(Y,X)$ を考え, 作用素 $F : X \to Y$ が C^1-Fréchet 微分可能とする. また A が単射で $AF : X \to X$ とする. いま $\bar{x} \in X$ に対して

$$\begin{aligned}
\|AF(\bar{x})\|_X &\le Y_0 \\
\|I - AA^\dagger\|_{\mathcal{L}(X)} &\le Z_0 \\
\|A(DF(\bar{x}) - A^\dagger)\|_{\mathcal{L}(X)} &\le Z_1 \\
\|A(DF(b) - DF(\bar{x}))\|_{\mathcal{L}(X)} &\le Z_2(r)r, \quad \forall b \in \overline{B(\bar{x}, r)}
\end{aligned}$$

が成り立つとする. このとき

$$p(r) := Z_2(r)r^2 - (1 - Z_1 - Z_0)r + Y_0$$

を radii polynomial といい, もしも $p(r_0) < 0$ となる $r_0 > 0$ が存在すれば, $F(\tilde{x}) = 0$ をみたす解 $\tilde{x}$ が $\overline{B(\bar{x}, r_0)}$ 内に一意存在する.

証明. 写像 T が閉球 $\overline{B(\bar{x}, r_0)}$ において縮小写像となることを示す. まず

$$DT(x) = I - ADF(x), \quad x \in X$$

から, 任意の $x \in \overline{B(\bar{x}, r_0)}$ に対して, 仮定より

$$\begin{aligned}
\|DT(x)\|_{\mathcal{L}(X)} &= \|I - ADF(x)\|_{\mathcal{L}(X)} \\
&\le \|I - AA^\dagger\|_{\mathcal{L}(X)} + \|A(A^\dagger - DF(\bar{x}))\|_{\mathcal{L}(X)} + \|A(DF(\bar{x}) - DF(x))\|_{\mathcal{L}(X)} \\
&\le Z_0 + Z_1 + Z_2(r_0)r_0
\end{aligned}$$

そして, 任意の $x \in \overline{B(\bar{x}, r_0)}$ に対して, 平均値の定理より

$$\begin{aligned}
\|T(x) - \bar{x}\|_X &\le \|T(x) - T(\bar{x})\|_X + \|T(\bar{x}) - \bar{x}\|_X \\
&\le \sup_{b \in B(\bar{x}, r_0)} \|DT(b)\|_{\mathcal{L}(X)} \|x - \bar{x}\|_X + \|AF(\bar{x})\|_X \\
&\le (Z_0 + Z_1 + Z_2(r_0)r_0)r_0 + Y_0 \\
&= p(r_0) + r_0
\end{aligned}$$

よって $p(r_0) < 0$ ならば, $\|T(x) - \bar{x}\|_X < r_0$. よって $T(x) \in \overline{B(\bar{x}, r_0)}$.

次に X の距離として $d(x,y) := \|x-y\|_X, \quad x,y \in X$ と定義すると

$$\begin{aligned} d(T(x),T(y)) &\leq \sup_{b\in B(\bar{x},r_0)} \|DT(b)\|_{\mathcal{L}(X)} d(x,y) \\ &\leq (Z_0 + Z_1 + Z_2(r_0)r_0)d(x,y), \quad x,y \in \overline{B(\bar{x},r_0)} \end{aligned}$$

このとき $p(r_0) < 0$ より

$$Z_0 + Z_1 + Z_2(r_0)r_0 + \frac{Y_0}{r_0} < 1$$

よって $k := Z_0 + Z_1 + Z_2(r_0)r_0 < 1$ となるため, T は $\overline{B(\bar{x},r_0)}$ 上で縮小写像となる.

したがって, Banach の不動点定理より, ただ一つの不動点 $\tilde{x}$ が $\overline{B(\bar{x},r_0)}$ に存在し, A の単射性より, この不動点は写像 F の零点となる, i.e., $\mathrm{F}(\tilde{\mathrm{x}}) = 0$. ■

3.2 Newton-Kantorovich の定理の亜種 [7]

定理 14 (Newton-Kantorovich の定理の亜種)**.** X と Y を Banach 空間とする. $F : X \to Y$ を与えられた作用素とし, $\bar{x} \in X$ を与えられているとする. $A \in \mathcal{L}(Y,X)$ とする. F は $\bar{x}$ で Fréchet 微分可能とし $DF(\bar{x})$ と表記する. $DF\bar{x}$ は全射であるとする. η と δ を不等式

$$\|AF(\bar{x})\|_X \leq \eta$$

と

$$\|I - ADF(\bar{x})\|_{\mathcal{L}(X)} \leq \delta < 1$$

を満たす定数とする.

$\overline{B(0,\frac{2\eta}{1-\delta})} = \{z \in X \mid \|z\|_X \leq \frac{2\eta}{1-\delta}\}$ とし, 定数 K を不等式

$$\|A(DF(\bar{x}) - DF(\bar{x}+z))\|_{\mathcal{L}(X)} \leq K,\ z \in \overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$$

を満たす定数とする. もし, $2K + \delta \leq 1$ ならば,

$$\|\tilde{x} - \bar{x}\|_X \leq \frac{\eta}{1-(K+\delta)} =: \rho$$

に対し, 真の解 $\tilde{x}$ は $\overline{B}(\bar{x},\rho)$ 内に存在する. その上, $\overline{B\left(\bar{x}, \frac{2\eta}{1-\delta}\right)}$ 内で一意である.

証明. まず, A と $DF\bar{x}$ が全単射であることを示す. $DF(\bar{x})$ は F の $\bar{x}$ における Frćhet 微分であることから, $DF(\bar{x})$ は $\mathcal{L}(X,Y)$ に属する. さらに, Neumann 級数の定理 (10) と仮定 $\|I - ADF(\bar{x})\|_{\mathcal{L}(X)} \leq \delta < 1$ より, $ADF(\bar{x})$ は全単射である. その上, 定理 (11) より, $DF(\bar{x})$ は単射であり, A は全射である. また, 定理の仮定より $DF(\bar{x})$ は全単射となるため, 逆作用素 $DF(\bar{x})^{-1}$ が存在し, $\mathcal{L}(Y,X)$ に属する. また, A の単射性については, 定理 (4) 逆作用素 $DF(\bar{x})^{-1}$ を用いて

$$A\phi = 0 \Rightarrow ADF(\bar{x})DF(\bar{x})^{-1}\phi = 0 \Rightarrow \phi = 0$$

となるため, A も全単射である.

続いて, Banach の不動点定理 (12) を用いて解の存在を示す. まず, 作用素方程式 $F(x) = 0$ を不動点方程式に変形する. $w := \tilde{x} - \bar{x}$ とする. A が単射であるため,

$$\begin{aligned}
&F(\tilde{x}) = 0 \\
&\Leftrightarrow w = w - AF(\bar{x} + w) \\
&\Leftrightarrow w = -AF(\bar{x}) + w - A(F(\bar{x} + w) - F(\bar{x}))
\end{aligned}$$

$\mathcal{T} : V \to V$ を

$$\mathcal{T}(w) := -AF(\bar{x}) + w - A(F(\bar{x} + w) - F(\bar{x}))$$

となる非線形作用素とし, 不動点方程式 $w = \mathcal{T}(w)$ の解の存在を Banach の不動点定理 (12) を用いて示す.

Banach の不動点定理 (12) では M を決めて, $\mathcal{T}$ が M から M への縮小写像になることを確認しなければならない. 特に, ポイントの一つは $\mathcal{T}$ の定義域を M としたときに, 値域が M に含まれることを確かめなければならない. すなわち, $\mathcal{T}(M) \subset M$ となるように M を選ぶことが重要である. この定理では, $M = \overline{B(0, \rho)},\ \rho = \frac{\eta}{1-(K+\delta)}$ と選ぶ. まず, M として選んだ閉球 $\overline{B(0, \rho)}$ と定理で出てくる, もう一つの閉球 $\overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$ の関係性を確認する. 定理の仮定より $1 - \delta > 0$ と $1 - (\delta + 2K) \geq 0$ を持つため,

$$\begin{aligned}
\rho - \frac{2\eta}{1-\delta} &= \frac{\eta(1-\delta)}{(1-(\delta+K))(1-\delta)} - \frac{2\eta(1-(\delta+K))}{(1-(\delta+K))(1-\delta)} \\
&= \frac{\eta((1-\delta) - 2(1-\delta) + 2K)}{(1-(\delta+2K))(1-\delta)} \\
&= \frac{-\eta(1-\delta-2K)}{(1-(\delta+K))(1-\delta)} \\
&\leq 0
\end{aligned}$$

となる. よって, $\rho \leq \frac{2\eta}{1-\delta}$ から $\overline{B(\bar{x}, \rho)} \subset \overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$ となる.

次に, $\mathcal{T}(\overline{B(0, \rho)}) \subset \overline{B(0, \rho)}$ を示す. 任意の $w \in \overline{B(0, \rho)}$ に対し, Fréchet 微分と Bochner 積分に対する微分積分学の基本定理を用いることで

$$\begin{aligned}
\|\mathcal{T}(w)\|_X &\leq \|AF(\bar{x})\|_X + \|w - A(F(\bar{x} + w) - F(\bar{x}))\|_X \\
&\leq \eta + \|w - R\int_0^1 DF((1-t)\bar{x} + t(\bar{x} + w))w dt\|_X \\
&\leq \eta + \int_0^1 \|w - ADF(\bar{x} + tw)w\|_X dt \\
&\leq \eta + \int_0^1 \|I - ADF(\bar{x} + tw)\|_{\mathcal{L}(V,V)}\|w\|_X dt \\
&\leq \eta + \int_0^1 (\|A(DF(\bar{x}) - DF(\bar{x} + tw))\|_{\mathcal{L}(X)} + \|I - ADF(\bar{x})\|_{\mathcal{L}(X)})\|w\|_X dt \\
&\leq \eta + \int_0^1 (\|A(DF(\bar{x}) - DF(\bar{x} + tw))\|_{\mathcal{L}(X)} + m)\|w\|_X dt
\end{aligned}$$

を得る. さらに $t \in [0, 1]$ に対し $tw \in \overline{B(0, \rho)} \subset \overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$ となるため, 定理の仮定 $\|R(DF(\bar{x}) -$

$DF(\bar{x}+tw))\|_{\mathcal{L}(X)} \leq K$ から

$$\begin{aligned}\|\mathcal{T}(w)\|_X &\leq \eta + (K+\delta)\|w\|_X \\ &\leq \eta + (K+\delta)\rho \\ &= \eta + \frac{\eta(K+\delta)}{1-(\delta+K)} \\ &= \frac{\eta-\eta(\delta+K)}{1-(\delta+K)} + \frac{\eta(K+\delta)}{1-(\delta+K)} \\ &= \rho\end{aligned}$$

となる. よって, 任意の $w \in \overline{B(0,\rho)}$ に対して $\|\mathcal{T}(w)\|_X \leq \rho$ となることから, $\mathcal{T}(\overline{B(0,\rho)}) \subset \overline{B(0,\rho)}$ となる.

次に $\mathcal{T} : \overline{B(0,\rho)} \to \overline{B(0,\rho)}$ が縮小写像になることを確認する. すなわち

$$\|\mathcal{T}(w_1) - \mathcal{T}(w_2)\|_X \leq k\|w_1 - w_2\|_X, \quad \forall w_1, w_2 \in \overline{B(0,\rho)}$$

となる定数 k が 1 未満になることを確認しなければならない. この定理では, 一意性の範囲を広げるために, $\overline{B(0,\rho)} \subset \overline{B\left(\frac{2\eta}{1-m}\right)}$ であることを用いて

$$\|\mathcal{T}(w_1) - \mathcal{T}(w_2)\|_X \leq k\|w_1 - w_2\|_X, \quad \forall w_1, w_2 \in \overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$$

となる定数 k が 1 未満になることを確かめる. その上, 任意の $w_1, w_2 \in \overline{B\left(0, \frac{\eta}{1-\delta}\right)}$ に対し,

$$\begin{aligned}\|\mathcal{T}(w_1) - \mathcal{T}(w_2)\|_X &= \|w_1 - w_2 - A(D(\bar{x}+w_1) - F(\bar{x}+w_2))\|_X \\ &= \|(w_1 - w_2) - A\int_0^1 DF(\bar{x} + (1-t)w_2 + tw_1)\|_{\mathcal{L}(X)} dt \|w_1 - w_2\|_X \\ &\leq \int_0^1 \|I - ADF(\bar{u} + (1-t)w_2 + tw_1)\|_{\mathcal{L}(X)} dt \|w_1 - w_2\|_X \\ &\leq \int_0^1 (\|A(DF(\bar{x}) - DF(\bar{x} + (1-t)w_2 + tw_1))\|_{\mathcal{L}(X)} + \delta) dt \|w_1 - w_2\|_X\end{aligned}$$

となる. 任意の $w_1, w_2 \in \overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$ と $0 \leq t \leq 1$ に対し, $\|(1-t)w_2 + tw_1\|_X \leq (1-t)\|w_2\|_X + t\|w_1\|_X \leq \frac{2\eta}{1-\delta}$ となるため,

$$\|\mathcal{T}(w_1) - \mathcal{T}(w_2)\|_X \leq (K+\delta)\|w_1 - w_2\|_X$$

となる. その上, 仮定 $2K + \delta \leq 1$ かつ $\delta < 1$ より $K + \delta < 1$ も満たすため, $\mathcal{T}$ は $\overline{B(0,\rho)}$ から $\overline{B(0,\rho)}$ への縮小写像となる. よって, Banach の不動点定理 (12) より不動点方程式 $w = \mathcal{T}(w)$ を $F(u) = 0$ を満たす解が $\overline{B(0,\rho)}$ 内に一意に存在する. $w = \tilde{x} - \bar{x}$ であったことから, 作用素方程式 $F(x) = 0$ を満たす解が $\overline{B(\bar{x},\rho)}$ 内に一意に存在する.

最後に $\overline{B\left(\bar{x}, \frac{2\eta}{1-\delta}\right)}$ 内で一意に存在することを示す. $\overline{B\left(0, \frac{2\eta}{1-\delta}\right)}$ 内に不動点方程式 $w = \mathcal{T}(w)$ の解が 2 つあったとする. 即ち 2 つの解 $\tilde{z_1}, \tilde{w_2} \in \overline{B\left(\bar{x}, \frac{2\eta}{1-\delta}\right)}$ とし, $\tilde{x_1} = \mathcal{T}(\tilde{w_1})$ と $\tilde{w_2} = \mathcal{T}(\tilde{w_2})$ を満たすとする. そのとき,

$$\|\tilde{w_1} - \tilde{w_2}\|_X = \|\mathcal{T}(\tilde{w_1}) - \mathcal{T}(\tilde{w_2})\|_X \leq (K+m)\|\tilde{w_1} - \tilde{w_2}\|_X$$

となる. ここで, $2K+m<1$ であるため, 不等式を満たすものは $\tilde{w}_1=\tilde{w}_2$ の場合のみである. すなわち, 不動点方程式 $w=\mathcal{T}(w)$ を満たす解は $\overline{B(0,\rho)}$ 内に一意である. よって, 作用素方程式 $F(x)=0$ を満たす解が $\overline{B\left(\bar{x},\frac{2\eta}{1-\delta}\right)}$ 内に一意に存在する. ■

4　既存の van der Pol 方程式の精度保証付き数値計算 [8]

4.1　van der Pol 方程式

van der Pol 方程式とは, ある発振現象をもつ電気回路の方程式であり, 以下のように表す.

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$$

未知関数は $x(t)$ で, $\mu > 0$ は非線形の減衰の強さを表すパラメータである.

4.2　フーリエ・スペクトル法

フーリエ・スペクトル法の計算に必要な参照軌道をまず得るために, van der Pol 方程式の解の挙動を数値計算する. まず, van der Pol 方程式を次の連立常微分方程式系にして ODE ソルバーで数値計算する.

$$\begin{cases} \dot{x} = y \\ \dot{y} = \mu(1-x^2)y - x \end{cases}$$

初期値 $x(0) = 0,\ y(0) = 2$ とし, $\mu = 1$ の時の数値計算を実行する. 周期解をフーリエ級数で表し, その係数と周期を求めるため, van der Pol 方程式の周期解の周期を大まかに求める. 得た近似周期軌道と近似周期を使って, 起動のフーリエ補完を計算する. van der Pol 方程式は, $\dot{x} = \frac{dx}{dt}$ とおくと, 以下のように表すことができる.

$$\ddot{x} - \mu(1-x^2)\dot{x} + x = 0$$

後の計算のために, 式を整理すると

$$\ddot{x} - \mu\dot{x} + \frac{\mu}{3}(\dot{x^3}) + x = 0$$

周期解 $x(t)$ を周期 L の周期関数とし, $\omega = \frac{2\pi}{L}$ とおくと, $x(t)$ とその微分やべき乗はフーリエ級数を使って,

$$\begin{aligned} x(t) &= \sum_{k\in\mathbb{Z}} a_k e^{ik\omega t} \\ \frac{dx(t)}{dt} &= \sum_{k\in\mathbb{Z}} (ik\omega) a_k e^{ik\omega t} \\ \frac{d^2x(t)}{dt^2} &= \sum_{k\in\mathbb{Z}} (-k^2\omega^2) a_k e^{ik\omega t} \\ x(t)^3 &= \sum_{k\in\mathbb{Z}} (a*a*a)_k e^{ik\omega t} \end{aligned}$$

と書くことができる. ここで

$$(a*a*a)_k := \sum_{\substack{k_1+k_2+k_3=k \\ k_i\in\mathbb{Z}}} a_{k_1}a_{k_2}a_{k_3}, \quad k\in\mathbb{Z}$$

は 3 次の離散畳み込みである.

そしてフーリエ係数に関する式を立てる. 係数 $a = (a_k)_{k\in\mathbb{Z}}$ に対して, van der Pol 方程式に求めたフーリエ級数を代入すると,

$$f_k(a) := -k^2\omega^2 a_k - \mu ik\omega a_k + \frac{\mu}{3}(ik\omega)(a * a * a)_k + a_k$$

となる点列 $(f_k(a))_{k\in\mathbb{Z}}$ が得られる. そして, 各 $k \in \mathbb{Z}$ について

$$f_k(a) = 0$$

となる点列 a が得られれば, van der Pol 方程式の解のフーリエ係数になる. 未知数は周期数 ω と点列 a であり, これらを並べて $x = (\omega, a)$ と書くことにする. 未知数 x に対して, $f_k(a) = 0$ という方程式だけでは不定な方程式になるため, 解の形を一つに定めることができない. そこで, 位相条件

$$\eta(a) := \sum_{|k|<N} a_k - \eta_0 = 0, \quad \eta_0 \in \mathbb{R}$$

を加える. この条件は, $x(t)$ の切片 $x(0) = \eta_0$ を表している. 最終的に van der Pol 方程式の周期解の求解は次の代数方程式を解くことに帰着される.

$$F(x) := \begin{bmatrix} \eta(a) \\ (f_k(a)_{k\in\mathbb{Z}}) \end{bmatrix} = 0$$

以下, この零点探索問題 $F(x) = 0$ について Newton 法で解を得ることを考える. まず N をフーリエ係数の打ち切り番号 (最大波数:$N-1$) とし, 周期解の近似を次のように構成する.

$$x(t) = \sum_{|k|<N} \bar{a}_k e^{ik\omega t}$$

このとき, フーリエ係数と (近似) 周期をならべた

$$\bar{x} = (\bar{\omega}, \bar{a}_{-N+1}, \cdots, \bar{a}_{N-1}) \in \mathbb{C}^{2N}$$

を近似解と呼ぶ. 近似解 $\bar{x}$ の項数は $2N$ 個. そして $f_k(a) = 0$ を $|k| < N$ の範囲で打ち切る方程式

$$F^{(N)}(x^{(N)}) = \begin{bmatrix} \eta(a^{(N)}) \\ (f_k(a^{(N)}))_{k<|N|} \end{bmatrix} = 0$$

を考える. ここで $a^{(N)} = (a_k)_{|k|<N}$, $x^{(N)} = (\omega, a^{(N)})$ をそれぞれ表し, $F^{(N)} : \mathbb{C}^{2N} \to \mathbb{C}^{2N}$ である. したがって $F^{(N)}(x^{(N)}) = 0$ という有限次元の非線形方程式を解くことで, 近似解 $\bar{x}$ が得られる.

実際に Newton 法を用いて, 周期解の数値計算を行っていく. Newton 法では, ある適当な初期値 x_0 を最初に定め, 以下の反復計算によって計算できる.

$$x_{n+1} = x_n - DF^{(N)}(x_n)^{-1} F^{(N)}(x_n), \quad n = 0, 1, \cdots$$

このことから $DF^{(N)}(x_n)^{-1}$ と $F^{(N)}(x_n)$ を計算することができれば, 近似解を得ることができる. はじめに離散畳み込みの関数を用意する.

$F^{(x)}(X^{(N)})$ のヤコビ行列は

$$DF^{(N)}(x^{(N)}) = \left[\begin{array}{c|ccc} 0 & 1 & \cdots & 1 \\ \vdots & & \vdots & \\ \partial_\omega f_k & \cdots & \partial_{a_j} f_k & \cdots \\ \vdots & & \vdots & \end{array}\right] \in \mathbb{C}^{2N\times 2N}(|k|,|j|<N)$$

ここで

$$\begin{cases} \partial_\omega f_k = (-2k^2\omega - \mu ik)a_k + \frac{\mu ik}{3}(a*a*a)_k & (|k|<N) \\ \partial_{a_j} f_k = (-k^2\omega^2 - \mu ik\omega + 1)\delta_{k_j} + \mu ik\omega(a*a)_{k-j} & (|k|,|j|<N) \end{cases}$$

$$\delta_{k_j} = \begin{cases} 1 & (k=j) \\ 0 & (k\neq j) \end{cases}$$

である. ヤコビ行列の各要素との対応は

$$(DF^{(N)}(x^{(N)}))_{\ell,m}$$

$$= \begin{cases} 0(\ell = m = 1) \\ 1(\ell = 1, m = 2, \cdots, 2N) \\ \partial_\omega f_k(\ell = 2, \cdots, 2N, m = 1, \text{ i.e.}, & \ell = k+N+1 \text{ for } |\mathrm{k}| < \mathrm{N}) \\ \partial_{a_j} f_k(\ell, m = 2, \cdots, 2N), \text{ i.e.} & \ell = k+N+1 \text{ for } |\mathrm{k}| < \mathrm{N}, \\ & m = j+N+1 \text{ for } |\mathrm{j}| < \mathrm{N} \end{cases}$$

4.3 重み付き空間と作用素の決定

定義 35 (許容重み)**.** 点列 $\omega = (\omega_k)_{k\in\mathbb{Z}}$ について,

$$\omega_k > 0, \quad \forall k \in \mathbb{Z}$$
$$\omega_{n+k} \leq \omega_n\omega_k, \quad \forall n,k \in \mathbb{Z}$$

が成立するとき許容重み (admissible wights) であるという.

定理 15 (重み付き ℓ^1 空間)**.**

$$\ell^1_\omega := \left\{ a = (a_k)_{k\in\mathbb{Z}} : a_k \in \mathbb{C},\ \|a\|_w := \sum_{k\in\mathbb{Z}} |a_k|\omega_k < \infty \right\}$$

点列 w を許容重みとする. $(\ell^1_w, *)$ は可換な Banach 環となる. すなわち, 点列 $a, b \in \ell^1_\omega$ として, 離散畳み込み " $*$ " に対して

$$\|a*b\|_\omega \leq \|a\|_\omega\|b\|_\omega$$

が成立する.

証明. $a, b \in \ell^1_\omega$ として,

$$\begin{aligned}
\|a * b\|_\omega &= \sum_{k\in\mathbb{Z}} |(a*b)_k|\omega_k \\
&= \sum_{k\in\mathbb{Z}} \left\| \sum_{\substack{k_1+k_2=k \\ k_1,k_2\in\mathbb{Z}}} a_{k_1} b_{k_2} \right\| \omega_k \\
&\leq \sum_{k\in\mathbb{Z}} \left(\sum_{\substack{k_1+k_2=k \\ k_1,k_2\in\mathbb{Z}}} |a_{k_1}||b_{k_2}| w_k \right) \\
&\leq \sum_{k\in\mathbb{Z}} \left(\sum_{\substack{k_1+k_2=k \\ k_1,k_2\in\mathbb{Z}}} |a_{k_1}|\omega_{k_1}|b_{k_2}|\omega_{k_2} \right) \\
&\leq \left(\sum_{k_1\in\mathbb{Z}} |a_{k_1}|\omega_{k_1} \right) \left(\sum_{k_2\in\mathbb{Z}} |b_{k_2}|\omega_{k_2} \right) \\
&= \|a\|_\omega \|b\|_\omega
\end{aligned}$$

■

定義 36 (Banach 空間 X)**.** Banach 空間 X を次のように定める. はじめに重み付き ℓ^1 空間を重み $\omega_k = \nu^{|k|} (\nu = 1.05)$ として次のように定める.

$$\ell^1_\nu := \left\{ a = (a_k)_{k\in\mathbb{Z}} : a_k \in \mathbb{C},\ \|a\|_\omega := \sum_{k\in\mathbb{Z}} |a_k|\nu^{|k|} < \infty \right\}$$

そして, 検証に用いる関数空間 X は

$$X := \mathbb{C} \times \ell^1_\nu,\ x = (x, a), \quad \omega \in \mathbb{C},\ a \in \ell^1_\nu$$

と定め, そのノルムを

$$\|x\|_X := \max|\omega|,\ \|a\|_\omega$$

として, 定義する. このとき, X は Banach 空間となる.

定義 37 (ヤコビ行列 $F^{(N)}(x^{(N)})$)**.** $F^{(N)}(x^{(N)})$ のヤコビ行列は, 次のような形をしている.

$$DF^{(N)}(x^{(N)}) = \left[\begin{array}{c|ccc} 0 & 1 & \cdots & 1 \\ \hline \vdots & & \vdots & \\ \partial_\omega f_k & \cdots & \partial_{a_j} f_k & \cdots \\ \vdots & & \vdots & \end{array}\right] \in \mathbb{C}^{2N\times 2N} \quad (|k|,\ |j| < N)$$

ここで

$$\begin{cases} \partial_\omega f_k = (-2k^2\omega - \mu i k)a_k + \frac{\mu i k}{3}(a*a*a)_k & (|k| < N) \\ \partial_{a_j} = (-k^2\omega^2 - \mu i k\omega + 1)\delta_{kj} + \mu i k\omega (a*a)_{k_j} & (|k|\ |j| < N) \end{cases}$$

$$\delta_{k_j} = \begin{cases} 1 & (k = j) \\ 0 & (k \neq j) \end{cases}$$

である. ヤコビ行列の各要素の伏字とその対応は

$$(DF^N(x^N))_{\ell,m} = \begin{cases} 0 & (\ell = m = 1) \\ 1 & (\ell = 1,\ m = 2\cdots 2N) \\ \partial_\omega f_k & (\ell = 2\cdots 2N,\ m = 1,\ \text{i.e.,}\ \ell = \text{k} + \text{N} + 1 \text{ for } |\text{k}| < \text{N}) \\ \partial_{a_j} f_k & (\ell, m = 2\cdots 2N,\ \text{i.e.,}\ \ell = \text{k} + \text{N} + 1 \\ & for\ |k| < N,\ m = j + N + 1 \text{ for } |\text{j}| < \text{N}) \end{cases}$$

定義 38 (作用素 $A^\dagger, A$ の定義)**.** まず, Banach 空間 $X = \mathbb{C} \times \ell^1_\nu$ から $Y = \mathbb{C} \times \ell^1_{\nu'} (\nu' < \nu)$ と設定し, $A^\dagger$ を $A^\dagger \in (X, Y)$ として, $b = (b_0, b_1) \in \mathbb{C} \times \ell^1_\nu = X$ に対して, $A^\dagger b = ((A^\dagger b)_0, (A^\dagger b)_1)$ と作用するように定義する. ここで, $A^\dagger$ を形式的に見ると,

$$A^\dagger = \left[\begin{array}{c|ccccccc} 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots \\ \hline \vdots & & \vdots & & & & \\ \partial_\omega f_k & \cdots & \partial_{a_j} f_k & \cdots & & 0 & \\ \vdots & & \vdots & & & & \\ \vdots & & & & \lambda_N & & 0 \\ 0 & & 0 & & & \lambda_{N+1} & \\ \vdots & & & & 0 & & \ddots \end{array}\right] = \left[\begin{array}{c|c} 0 & A^\dagger_{a,0} \\ \hline A^\dagger_{\omega,1} & A^\dagger_{a,1} \end{array}\right]$$

このことから, $A^\dagger b$ は,

$$A^\dagger b = \begin{bmatrix} 0 & A^{\dagger_{a,0}} \\ A^\dagger_{\omega,1} & A^\dagger_{a,1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} A^\dagger_{a,0} b_1 \\ A^\dagger_{\omega,1} b_0 + A^\dagger_{a,1} b_1 \end{bmatrix} =: \begin{bmatrix} (A^\dagger b)_0 \\ (A^\dagger b)_1 \end{bmatrix}$$

と表すことができ,

$$\begin{aligned} (A^\dagger b)_0 &= \sum_{|k|<N} (b_1)_k \\ ((A^\dagger b)_1)_k &:= \begin{cases} \partial_\omega f_k b_0 + \sum_{|j|<N} \partial_{a_j} f_k (b_1)_j & , |k| < N \\ \lambda_k (b_1)_k & , |k| \geq N,\ \lambda_k := -k^1\omega^2 - \mu i k \omega + 1 \end{cases} \end{aligned}$$

と書ける. またこのとき, $(A^\dagger b)_0$ と $(A^\dagger b)_1$ はそれぞれ

$$\begin{aligned} (A^\dagger b)_0 &= A^\dagger_{a,0} b_1 = \sum_{|k|<N} (b_1)_k \in \mathbb{C} \\ (A^\dagger b)_1 &= A^\dagger_{\omega,1} b_0 + A^\dagger_{a,1} b_1 \in \ell^1_{\nu'} \end{aligned}$$

次に作用素 A について考える.

$$A^N = \begin{bmatrix} A^{(N)}_{\omega,0} & A^{(N)}_{a,0} \\ A^{(N)}_{\omega,1} & A^{(N)}_{a,1} \end{bmatrix} \approx DF^{(N)}(\bar{x}^{-1}) \in \mathbb{C}^{2N \times 2N}$$

をヤコビ行列の近似逆行列とする. $A \in \mathcal{L}(Y, X)$ として, $b = (b_0, b_1) \in X$ に対して, $Ab = ((Ab)_0, (Ab)_1)$ と作用するように定義する. ここで,

$$\begin{aligned} (Ab)_0 &= A^{(N)}_{\omega,0} b_0 + A^{(N)}_{a,0} b^{(N)}_1 \\ (Ab)_1 &= A^{(N)}_{\omega,1} b_0 + A_{a,1} b_1 \end{aligned}$$

ただし, 無限次元の $A_{a,1}b_1$ は以下のようになる.

$$(A_{a,1}b_1)_k = \begin{cases} (A_{a,1}^{(N)}, b_1^{(N)})_k & (|k| < N) \\ \frac{(b_1)_k}{\lambda_k} & (|k| \geq N) \end{cases}$$

この定義を形式的に見ると

$$A = \left[\begin{array}{c|ccccc} A_{\omega,0}^{(N)} & A_{a,0}^{(N)} & 0 & \cdots & 0 \\ \hline A_{\omega,1}^{(N)} & A_{a,1}^{(N)} & & 0 & \\ 0 & & \frac{1}{\lambda_N} & & \\ \vdots & & & \frac{1}{\lambda_{N+1}} & \\ 0 & & 0 & & \ddots \end{array}\right] = \left[\begin{array}{c|c} 0 & A_{a,0} \\ \hline A_{\omega,1} & A_{a,1} \end{array}\right]$$

と表記できる

4.4 Y_0, Z_0, Z_1, Z_2 の評価

Y_0

$$\|AF(\bar{x})\|_x \leq Y_0$$

Z_0

$$\|I - AA^\dagger\|_{\mathcal{L}(X)} \leq Z_0$$

Z_1

$$\|A(DF(\bar{x}) - A^\dagger)c\|_X \leq Z_1,\ c \in \overline{B(0,1)}$$

また, ここでいう $\overline{B(0,1)}$ とは $\|c\|_X = 1$ ということである.

Z_2

$b \in \overline{B(\bar{x}, r)},\ h \in \overline{B(0,1)}$ として,

$$\|A(DF(b) - DF(\bar{x}))h\|_X \leq Z_2(r)r$$

4.4.1 Y_0 を計算する

$$F(\bar{x}) = (\delta_0, \delta_1) \in \mathbb{C} \times \ell^1_{\nu'}$$

とすると, A の定義より,

$$\|AF(\bar{x})\|_X \leq \max\left\{ |A_{\omega,0}^{(N)}\delta_0 + A_{a,0}^{(N)}\delta_1^{(N)}|,\ \|A_{\omega,1}^{(N)}\delta_0 + A_{a,1}\delta_1^{(N)}\|_\omega + \sum_{|k|>N} \left\| \frac{(\delta_1^\infty)_k}{\lambda_k} \right\| \nu^{|k|} \right\}$$

ここで, $\delta_1 = (\delta_1^{(N)}, \delta_1^{(\infty)}) \in \mathbb{C}^{2\times 3(N-1)+1}$ であり,

$$(\delta_1)_k = \begin{cases} \delta_1^{(N)} & (k < |N|) \\ \delta_1^{(\infty)} & (k \leq |N|) \end{cases}$$

と表す.

4.4.2 Z_0 を計算する

$$B := I - AA^\dagger = \begin{bmatrix} B_{\omega,0} & B_{a,0} \\ B_{\omega,1} & B_{a,1} \end{bmatrix}$$

この B を $c \in \overline{B(0,1)},\ \|c\|_X \leq 1$ である $c = (c_0, c_1)$ に作用させると,

$$\begin{aligned} (Bc)_0 &= B_{\omega,0}c_0 + B_{a,0}c_1 \\ (Bc)_1 &= B_{\omega,1}c_0 + B_{a,1}c_1 \end{aligned}$$

Bc_0 はスカラ値なので,

$$\begin{aligned} |B_{a,0}c_1| &\leq \sum_{k\in\mathbb{Z}} |(B_{a,0})_k||(c_1)_k| \\ &= \sum_{k\in\mathbb{Z}} \frac{|(B_{a,0})_k|}{\omega_k} |(c_1)_k|\omega_k \\ &\leq \max_{k<|N|} \frac{|(B_{a,0})_k|}{\omega_k} \sum_{k\in\mathbb{Z}} |(c_1)_k|\omega_k \\ &\leq \max_{k<|N|} \frac{|(B_{a,0})_k|}{\omega_k},\ (\sum_{k\in\mathbb{Z}} |(c_1)_k|\omega_k = \|c_1\|_\omega \leq 1) \end{aligned}$$

以上より,

$$|(Bc)_0| \leq |B_{\omega,0}| + \max_{|k|<N} \frac{|(B_{a,0})_k|}{\omega_k} = Z_0^{(0)}$$

またここで, 作用素 $M : \ell^1_{nu} \Rightarrow \ell^1_{nu}$ の作用素ノルムについて以下の補題を準備する.

補題 1. 行列 $M^{(N)}$ を $M^{(N)} \in \mathbb{C}^{(2N-1)\times(2N-1)}$, 双方向の複素無限点列を $(\delta_k)_{|k|\geq N}$ と定義する. ここで, $\delta_N > 0$ であり,

$$|\delta_k| \leq \delta_N \quad \forall |k| \geq N$$

を満たすとする. そして, $a = (a_k)_{k\in\mathbb{Z}} \in \ell^1_\nu$ に対して $a^{(N)} = (a_{-N+1}, \cdots, a_{N-1}) \in \mathbb{C}^{2N-1}$ と表し, 作用素 $M : \ell^1_\nu \Rightarrow \ell^1_\nu$ を以下のように定義する.

$$[Ma]_k := \begin{cases} [M^{(N)}a^{(N)}]_k, & |k| < N \\ \delta_k a_k, & |k| \geq N \end{cases}$$

このとき, M は有界線形作用素であり,

$$\|M\|_{\mathcal{L}(\ell^1_\nu)} \leq \max(K, \delta_N), \quad K := \max_{|n|<N} \frac{1}{\nu^{|n|}} \sum_{|k|<N} |M_{k,n}|\nu^{|k|}$$

と評価される.

上の補題を利用すると,

$$\|(Bc)_1\|_\omega \le \|B_{\omega,1}\|_\omega + \|B_{a,1}\|_{\mathcal{L}(\ell^1_\nu)} = Z_0^{(1)}$$

が評価可能となり, 結論としては, 求めたい Z_0 は $Z_0 := \{Z_0^{(0)}, Z_0^{(1)}\}$ となる.

4.4.3 Z_1 を計算する

$$\|A(DF(\bar{x}) - A^\dagger)c\|_X \le Z_1, \quad c = (c_0, c_1) \in \overline{B(0,1)} \Leftrightarrow \|c\|_X \le 1$$

点列 z を下記のように定義する.

$$z := (DF(\bar{x}) - A^\dagger)c = \begin{bmatrix} z_0 \\ z_1 \end{bmatrix}$$

ここで, $DF(\bar{x})$ と $A^\dagger$ は

$$DF(\bar{x}) = \left[\begin{array}{c|ccc} 0 & \cdots & 0 & \cdots \\ \hline \partial_\omega f_k^{(N)} & \cdots & \partial_{a_j} f_k & \cdots \\ \vdots & & \vdots & \end{array}\right],$$

$$A^\dagger = \left[\begin{array}{c|ccccccc} 0 & \cdots & \partial_a \eta & \cdots & 0 & \cdots & 0 \\ \hline \vdots & & \vdots & & & & \\ \partial_\omega f_k^{(N)} & \cdots & \partial_{a_j} f_k^{(N)} & \cdots & & & \\ \vdots & & \vdots & & 0 & & \\ 0 & & & \lambda_N & & & \\ \vdots & & & & \lambda_{N+1} & & \\ 0 & & 0 & & & \ddots & \end{array}\right]$$

$$\lambda_k := -k^2\omega^2 - \mu i k \omega + 1$$

と表される. すると z_0 は,

$$z_0 = \sum_{|k|\ge N} (c_1)_k,$$
$$|z_0| \le \frac{1}{\omega_N} \sum_{|k|_\ge N} |(c_1)_k|\omega_k \le \frac{1}{\omega_N}$$

次に, z_1 について考える. $DF(\bar{x})c$ 部分は

$$\begin{aligned}((DF(\bar{x})c)_1)_k &= \partial_\omega f_k c_0 + \partial_a f_k c_1 \\ &= \frac{\partial \lambda_k}{\partial \omega} c_0 \bar{a}_k + \frac{\mu i k}{3}(\bar{a} * \bar{a} * \bar{a})_k c_0 + \lambda_k (c_1)_k + \mu i k \omega (\bar{a} * \bar{a} * c_1)_k, \ k \in \mathbb{Z}\end{aligned}$$

と書け, $|k| \ge N$ で $\bar{a}_k = 0$ より, $c_1 = c_1^{(N)} + c_1^{(\infty)}$ として,

$$(z_1)_k = \begin{cases} \mu i k \omega (\bar{a} * \bar{a} * c_1^{(\infty)})_k, & |k| < N \\ \frac{\mu i k}{3}(\bar{a} * \bar{a} * \bar{a})_k c_0 + \mu i k \omega (\bar{a} * \bar{a} * c_1)_k, & |k| \ge N \end{cases}$$

と表せる．ここから, z_1 の絶対値を取ると, $|k| < N$ で

$$|(z_1)_k| \le |\mu i k \omega| \max\left\{ \max_{k-N+1\le j\le -N} \frac{|(\bar{a}*\bar{a})_{k-j}|}{\omega_j}, \max_{N\le j\le k+N-1} \frac{|(\bar{a}*\bar{a})_{k-j}|}{\omega_j} \right\} =: \zeta,$$
$$\zeta = (\zeta_k)_{|k|<N} \in \mathbb{R}^{2N-1}$$

最後に, Z_1 を評価していく．$Z_1^{(0)}$ の評価は

$$\begin{aligned} |(A(DF(\bar{x}) - A^\dagger)c)_0| &= |((Az)_0)| \\ &\le |A_{\omega,0}^{(N)}||Z_0| + |A_{a,0}^{(N)}||z_1^{(N)}| \\ &\le \frac{|A_{\omega,0}^{(N)}|}{\omega_N} + |A_{a,0}^{(N)}|\zeta \\ &=: z_1^{(0)} \end{aligned}$$

$Z_1^{(1)}$ の評価は

$$\begin{aligned} \|(A(DF(\bar{x}) - A^\dagger)c)_1\|_\omega &= \|(Az)_1\|_\omega \\ &= \|A_{\omega,1}^{(N)} z_0 + A_{a,1} z_1\|_\omega \\ &\le \frac{\|A_{\omega,1}^{(N)}\|_\omega}{\omega_N} + \sum_{|k|<N} (|A_{a,1}^{(N)}|\zeta)_k \omega_k + \sum_{N\le|k|\le 3(N-1)} \frac{|\mu i k(\bar{a}*\bar{a}*\bar{a})_k|}{3|\lambda_k|}\omega_k \\ &\quad + \sum_{|k|\ge N} \frac{|\mu i k \omega(\bar{a}*\bar{a}*c_1)_k|}{|\lambda_k|}\omega_k \\ &\le \frac{\|A_{\omega,1}^{(N)}\|_\omega}{\omega_N} + \||A_{a,1}^{(N)}|\zeta\|_\omega + \sum_{N\le|k|\le 3(N-1)} \frac{|\mu i k(\bar{a}*\bar{a}*\bar{a})_k|}{3|\lambda_k|}\omega_k + \frac{1}{N}\frac{\mu\omega\|\bar{a}\|_\omega^2}{\omega^2 - \frac{1}{N^2}} \\ &=: Z_1^{(1)} \end{aligned}$$

よって,

$$Z_1 := \max\{Z_1^{(0)}, Z_1^{(1)}\}$$

4.4.4 Z_2 を計算する

$b \in \overline{B(\bar{x}, r)},\ c(c_0, c_1) \in \overline{Z(0,1)}$ について

$$\|A(DF(b) - DF(\bar{x}))c\|_X \le Z_2(r) r$$

を考える．まず z を,

$$z := (DF(b) - DF(\bar{x}))c = \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ z_1 \end{bmatrix}$$

と定義する．$z_0 = 0$ となるので, z_1 だけを考えればよく

$$(z_1)_k := (\partial_\omega f_k(b) - \partial_\omega f_k(\bar{x}))c_0 + [\partial_a f(b) - \partial_a f(\bar{x})c_1]_k,\ k \in \mathbb{Z}$$

と書ける. $b=(\omega,(a_k)_{k\in\mathbb{Z}}),\bar{x}=(\bar{\omega},(\bar{a}_k)_{|k|<N})$ として, 第 1 項は

$$\begin{aligned}(\partial_\omega f_k(b)-\partial_\omega f_k(\bar{x}))c_0 &= [((-2k^2\omega-\mu ik)a_k+\frac{\mu ik}{3}(a*a*a)_k)-((-2k^2\bar{\omega}-\mu ik)\bar{a}_k+\frac{\mu ik}{3}(\bar{a}*\bar{a}*\bar{a})_k)]c_0\\ &= [-2k^2\omega(a_k-\bar{a}_k)-2k^2(\omega-\bar{\omega})\bar{a}_k-\mu ik(a_k-\bar{a}_k)\\ &\quad+\frac{\mu ik}{3}((a*a*a)_k-(\bar{a}*\bar{a}*\bar{a})_k)]c_0\end{aligned}$$

と書ける. そして, 第 2 項は,

$$\begin{aligned}[(\partial_a f(b)-\partial_a f(\bar{x}))c_1]_k &= (-k^2\omega^2-\mu ik\omega+1)(c_1)_k+\mu ik\omega(a*a*c_1)_k\\ &\quad-(-k^2\bar{\omega}^2-\mu ik\bar{\omega}+1)(c_1)_k+\mu ik\bar{\omega}(\bar{a}*\bar{a}*c_1)_k\\ &= [-k^2(\omega+\bar{\omega})(\omega-\bar{\omega})-\mu ik(\omega-\bar{\omega})](c_1)_k+\mu ik\omega((a+\bar{a})*(a-\bar{a})*c_1)_k\\ &\quad+\mu ik(\omega-\bar{\omega})(\bar{a}*\bar{a}*c_1)_k\end{aligned}$$

と書ける. $(Az)_0,(Az)_1$ は,

$$\begin{aligned}(Az)_0 &= A_{a,0}^{(N)}z_1^{(N)}\\ (Az)_1 &= A_{a,1}z_1\end{aligned}$$

より,

$$\|Az\|_X=\max\{|A_{a,0}^{(N)}z_1^{(N)}|,\|A_{a,1}z_1\|\}$$

となる.

$|A_{a,0}^{(N)}z^{(N)_1}|$ を上から評価する. はじめに $\tilde{A}_{a,0},\tilde{B}_{a,0}$ を以下のように定義する.

$$\begin{aligned}\tilde{A}_{a,0} &:= (|k|(A_{a,0}^{(N)})_k)_{|k|<N}\\ \tilde{B}_{a,0} &:= (k^2(A_{a,0}^{(N)})_k)_{|k|<N}\end{aligned}$$

すると,

$$\begin{aligned}|A_{a,0}^{(N)}z_1^{(N)}| &\le 2(\tilde{\omega}+r)\|\tilde{B}_{a,0}\|_\omega r+2\|\tilde{B}_{a,0}\|_\omega\|\tilde{a}\|_\omega r+\mu\|\tilde{A}_{a,0}\|_\omega r+\frac{\mu}{3}\|\tilde{A}_{a,0}\|_\omega(r^2+3\|\bar{a}\|_\omega r;3\|\bar{a}\|_\omega^2)r\\ &\quad+\|\tilde{B}_{a,0}\|_\omega(2\bar{\omega}+r)r+\mu\|\tilde{A}_{a,0}\|_\omega r+\mu(\bar{\omega}+r)\|\tilde{A}_{a,0}\|_\omega(2\|\bar{a}\|_\omega+r)r+\mu\|\tilde{A}_{a,0}\|_\omega\|\bar{a}\|_\omega^2 r\\ &= Z_2^{(4,0)}r^3+Z_2^{(3,0)}r^2+Z_2^{(2,0)}r\end{aligned}$$

となる. 同様に $\|A_{a,1}z_1\|_\omega$ を上から評価する. $\tilde{A}_{a,1},\tilde{B}_{a,1}$ を以下のように定義する.

$$\begin{aligned}\tilde{A}_{a,1} &:= (|j|(A_{a,1})_{k,j})_{k,j\in\mathbb{Z}}\\ \tilde{B}_{a,1} &:= (j^2(A_{a,1})_{k,j})_{k,j\in\mathbb{Z}}\end{aligned}$$

すると,

$$\begin{aligned}\|A_{a,1}z_1\|_\omega &\le 2(\bar{\omega}+r)\|\tilde{B}_{a,1}\|_{\mathcal{L}(\ell^1_{nu})}r+2\|\tilde{B}_{a,1}\|_{\mathcal{L}(\ell^1_{nu})}\|\bar{a}\|_\omega r+\mu\|\tilde{A}_{a,1}\|_{\mathcal{L}(\ell^1_\nu)}r\\ &\quad+\frac{\mu}{3}\|\tilde{A}_{a,1}\|_{\mathcal{L}(\ell^1_\nu)}(r^2+3\|\bar{a}\|_\omega r+3\|\bar{a}\|_\omega^2)r\\ &\quad+\|\tilde{B}\|_{\mathcal{L}(\ell^1_\nu)}(2\bar{\omega}+r)r+\mu\|\tilde{A}_{a,1}\|_{\mathcal{L}(\ell^1_\nu)}r+\mu(\bar{\omega}+r)\|\tilde{A}_{a,1}\|_{\mathcal{L}(\ell^1_\nu)}(2\|\bar{a}\|_\omega+r)r+\mu\|\tilde{A}_{a,1}\|_{\mathcal{L}(\ell^1_\nu)}\|\bar{a}\|_\omega^2 r\\ &= Z_2^{(4,1)}r^3+Z_2^{(3,1)}r^2+Z_2^{(2,1)}r\end{aligned}$$

と書ける． $Z_2^{(4,1)}, Z_2^{(3,1)}, Z_2^{(2,1)}$ は, 先ほどの $\tilde{A}_{a,0}, \tilde{B}_{a,0}$ を $\tilde{A}_{a,1}, \tilde{B}_{a,1}$ に置き換えたものになる．$j = 2, 3, 4$ で

$$Z_2^{(j)} := \max\{Z_2^{(j,0)}, Z_2^{(j,1)}\}$$

とすれば

$$Z_2(r) := Z_2^{(4)} r^2 + Z_2^{(3)} r + Z_2^{(2)}$$

となる．

4.5 radii polynomial の零点探索の精度保証

以上で, $Y_0, \cdots, Z_2$ の評価を区間演算で求めた．これらの評価を用いて $p(r_0) < 0$ となる r_0 を求める．精度保証の方法は, Newton 法を反復させることで, $p(r_0) = 0$ となる r_0 の近似解を求め, これを Krawczyk 法で検証する．

4.6 Krawzczyk(クラフチック) 法

定理 16 (Krawczyk 法 [2])**.** $X \subset \mathbb{R}^n$ を区間ベクトル, $c = \mathrm{mid(X)}$, $\mathrm{R} \simeq \mathrm{Df(c)}^{-1} = \mathrm{J(c)}^{-1}$, E を単位行列とし,

$$K(X) = c - Rf(c) + (E - RDf(X))(X - c)$$

としたとき, $K(X) \subset \mathrm{int(X)}$(int(X) : Xの内部)ならばXにf(x) = 0 の解が唯一存在する．

5　提案手法

radii polynomial approach(3.1) において, A は $A^\dagger$ の近似逆作用素である. これを真の逆作用素とすることで,

$$AA^\dagger = I$$

となる. 次に Z_0 に注目すると

$$\|I - AA^\dagger\|_{\mathcal{L}(X)} = \|I - I\|_{\mathcal{L}(X)} = 0 = Z_0$$

となる. また, Z_1 に注目すると

$$\|A(DF(\bar{x}) - A^\dagger)\|_{\mathcal{L}(X)} = \|ADF(\bar{x}) - I\|_{\mathcal{L}(X)} \leq Z_1$$

となる. ここで Newton-Kantorovich の定理の亜種 (3.2) と照らし合わせると

$$\|ADF(\bar{x}) - I\|_{\mathcal{L}(X)} = \|I - ADF(\bar{x})\|_{\mathcal{L}(X)}$$

同様にして, $b \in \overline{B(x,r)},\ z \in \overline{B(0, \frac{2\eta}{1-\delta})}$ から

$$\|A(DF(b) - DF(\bar{x}))\|_{\mathcal{L}(X)} = \|A(DF(\bar{x}) - DF(\bar{x}+z))\|_{\mathcal{L}(X)}$$

より, Y_0 は η, Z_0 は δ, $Z_2(r)r$ は K を表すことになる.

ゆえに, radii polynomial の零点探索における r_0 を $r_0 = \frac{2\eta}{1-\delta}$ と表すことができることが (3.2) によって証明されていることとなる.

6 実験結果

6.1 実験環境

本研究では, van der Pol 方程式の精度保証付き数値計算において 6.2 章に示した 3 つの手法の解の保証範囲, 実行時間を比較する. (6.1) に本実験の環境を (6.1) に使用パッケージを示す.

表 1　実験環境

CPU	12th Gen Intel(R) Core(TM) i7-12700
OS	Linux (x86_64-linyc-gnu)
Julia	Version 1.8.5

表 2　使用パッケージ

IntervalArithmetic	v“0.20.9”
ForwardDiff	v“0.10.36”
DifferentialEquations	v“7.10.0”
Plots	v“1.39.0”
FFTW	v“1.7.1”

6.2 実験手法

既存手法 (Exsisting method)

4 章に示した radii polynomial approach を指す.

提案手法 1(Proposed method1)

radii polynomial approach において A を A^{dagger} の真の逆作用とし, 評価関数

$$\begin{aligned} \|AF(\bar{x})\|_X &\leq Y_0 \\ \|ADF(\bar{x}) - I\|_{\mathcal{L}(X)} &\leq Z_1 \\ \|A(DF(b) - DF(\bar{x}))\|_{\mathcal{L}(X)} &\leq Z_2(r)r,\ \forall b \in \overline{B\left(\bar{x}, \frac{2Y_0}{1-Z_1}\right)} \end{aligned}$$

が成り立つとする. もし, $2Z_2(r)r + Z_1 \leq 1$ ならば,

$$\|\tilde{x} - \bar{x}\|_X \leq \frac{Y_0}{1-(Z_2(r)r+Z_1)} =: \rho$$

に対し, 真の解 $\tilde{x}$ は $\overline{B(\bar{x}, \rho)}$ 内に存在する. その上, $\overline{B\left(\bar{x}, \frac{2Y_0}{1-Z_1}\right)}$ 内で一意である.

提案手法 2(Proposed method2)

radii polynomial approach における評価関数

$$\begin{aligned}\|AF(\bar{x})\|_X &\le Y_0\\ \|I - AA^\dagger\|_{\mathcal{L}(X)} &\le Z_0\\ \|A(DF(\bar{x}) - A^\dagger)\|_{\mathcal{L}(X)} &\le Z_1\\ \|A(DF(b) - DF(\bar{x}))\|_{\mathcal{L}(X)} &\le Z_2(r)r,\ \forall b \in \overline{B\left(\bar{x}, \frac{2Y_0}{1-(Z_0+Z_1)}\right)}\end{aligned}$$

が成り立つとする. もし, $2Z_2(r)r + (Z_0 + Z_1) \le 1$ ならば,

$$\|\tilde{x} - \bar{x}\|_X \le \frac{Y_0}{1-(Z_2(r)r+(Z_0+Z_1))} =: \rho$$

に対し, 真の解 $\tilde{x}$ は $\overline{B(\bar{x}, \rho)}$ 内に存在する. その上, $\overline{B\left(\bar{x}, \frac{2Y_0}{1-(Z_0+Z_1)}\right)}$ 内で一意である.

6.3 実験結果

6.3.1 近似解の保証範囲

まず, 3 つの手法に対して周期解を求める際のフーリエ級数の次数を変え, 計算した評価定数の値 $Y_0, Z_0, Z_1, Z_2^{(2,1)}, Z_2^{(3,1)}, Z_2^{(4,1)}$ を表 3, 表 4, 表 5, 表 6, 表 7, 表 8 にまとめる.

表 3　評価定数 Y_0

次数	既存手法	提案手法 1	提案手法 2
25	$2.189956031972\times10^{-7}$	$2.189956031983\times10^{-7}$	$2.189956031972\times10^{-7}$
50	$2.470208168889\times10^{-11}$	$2.470208378229\times10^{-11}$	$2.470208168889\times10^{-11}$
75	$6.633346365723\times10^{-10}$	$6.633346394726\times10^{-10}$	$6.633346365723\times10^{-10}$
100	$2.399308427746\times10^{-8}$	$2.399308428175\times10^{-8}$	$2.399308427746\times10^{-8}$
125	$8.096510415868\times10^{-7}$	$8.096510415910\times10^{-7}$	$8.096510415868\times10^{-7}$
150	$2.736836105769\times10^{-5}$	$2.736836105770\times10^{-5}$	$2.736836105769\times10^{-5}$
175	$8.973956292883\times10^{-4}$	$8.973956292883\times10^{-4}$	$8.973956292883\times10^{-4}$
200	$4.046551016784\times10^{-2}$	$4.046551016784\times10^{-2}$	$4.046551016784\times10^{-2}$

表 4　評価定数 Z_0

次数	既存手法	提案手法 1	提案手法 2
25	$1.836467033545\times10^{-12}$	0	$1.8364670335451\times10^{-12}$
50	$4.001051392601\times10^{-12}$	0	$4.0010513926017\times10^{-12}$
75	$7.651418180561\times10^{-12}$	0	$7.6514181805613\times10^{-12}$
100	$1.705957387609\times10^{-11}$	0	$1.7059573876094\times10^{-11}$
125	$3.499159693630\times10^{-11}$	0	$3.4991596936308\times10^{-11}$
150	$1.089712413407\times10^{-10}$	0	$1.0897124134072\times10^{-10}$
175	$3.925276157847\times10^{-10}$	0	$3.9252761578479\times10^{-10}$
200	$1.029153445722\times10^{-9}$	0	$1.0291534457222\times10^{-9}$

表 5　評価定数 Z_1

次数	既存手法	提案手法 1	提案手法 2
25	1.017826203905	1.017826376314	1.017826203905
50	0.374621316999	0.374621382386	0.374621316999
75	0.193438233453	0.193438257620	0.193438233453
100	0.125685732055	0.125685739822	0.125685732055
125	0.094240924212	0.094240926472	0.094240924212
150	0.076505001721	0.076505002584	0.076505001721
175	0.064907843167	0.064907843445	0.064907843167
200	0.056562245284	0.056562245366	0.056562245284

表 6　評価定数 $Z_2^{(2,1)}$

次数	既存手法	提案手法 1	提案手法 2
25	64.070811472180	64.070858383310	64.070811472180
50	64.070904136351	64.070971561361	64.070904136351
75	64.070904136400	64.070991753888	64.070904136400
100	64.070904136401	64.071001097115	64.070904136401
125	64.070904136402	64.071000671209	64.070904136402
150	64.070904136402	64.071030038766	64.071030038766
175	64.070904136404	64.071042264246	64.070904136404
200	64.070904136407	64.071042084744	64.070904136407

表 7　評価定数 $Z_2^{(3,1)}$

次数	既存手法	提案手法 1	提案手法 2
25	25.283272305673	25.283290232754	25.283272305673
50	25.283307550480	25.283333316401	25.283307550480
75	25.283307550499	25.283341030975	25.283307550499
100	25.283307550499	25.283344603009	25.283307550499
125	25.283307550500	25.283344440127	25.283307550500
150	25.283307550500	25.283355660176	25.283307550500
175	25.283307550500	25.283360332882	25.283307550500
200	25.283307550501	25.283360261149	25.283307550501

表 8　評価定数 $Z_2^{(4,1)}$

次数	既存手法	提案手法 1	提案手法 2
25	2.319296043152	2.319297322038	2.319299126108
50	2.319299126108	2.319300963858	2.319299126108
75	2.319299126110	2.319301512913	2.319299126110
100	2.319299126110	2.319301768718	2.319299126110
125	2.319299126110	2.319301757020	2.319299126110
150	2.319299126110	2.319302555658	2.319299126110
175	2.319299126110	2.319302889478	2.319299126110
200	2.319299126110	2.319302882336	2.319299126110

6.3.2　解の保証範囲

3 つの手法において近似解の一意性が保証される範囲 $(r_0, \frac{2\eta}{1-\delta})$ を表 9, 存在が保証される範囲 (r_0, ρ) を表 10 にまとめる. 値は近似解の精度が保証された場合のときだけを抽出した. また図 1 は表 9 を, 図 2 は表 10 を常用対数グラフで示したものである.

表 9　フーリエ級数の次数における解の一意性の保証範囲 $r_0, \frac{2\eta}{1-\delta}$

次数	既存手法	提案手法 1	提案手法 2
50	$3.949940108631\times10^{-11}$	$7.899881155698\times10^{-11}$	$7.899879660287\times10^{-11}$
75	$8.224226679561\times10^{-10}$	$1.644845284914\times10^{-9}$	$1.644845228453\times10^{-9}$
100	$2.744222919362\times10^{-8}$	$5.488434851077\times10^{-8}$	$5.488434801444\times10^{-8}$
125	$8.939488051415\times10^{-7}$	$1.787784556079\times10^{-6}$	$1.787784551679\times10^{-6}$
150	$2.969682129426\times10^{-5}$	$5.927127084457\times10^{-5}$	$5.927127079615\times10^{-5}$
175	$1.032803976652\times10^{-3}$	$1.919373663864\times10^{-3}$	$1.919373664098\times10^{-3}$

表 10　フーリエ級数の次数における解の存在保証範囲 r_0, ρ

次数	既存手法	提案手法 1	提案手法 2
50	$3.949940108631\times10^{-11}$	$3.949940577849\times10^{-11}$	$3.949939830143\times10^{-11}$
75	$8.224226679561\times10^{-10}$	$8.224226424573\times10^{-10}$	$8.224226142265\times10^{-10}$
100	$2.744222919362\times10^{-8}$	$2.744217425539\times10^{-8}$	$2.744217400722\times10^{-8}$
125	$8.939488051415\times10^{-7}$	$8.938922782418\times10^{-7}$	$8.938922760420\times10^{-7}$
150	$2.969682129426\times10^{-5}$	$2.963564264566\times10^{-5}$	$2.963564262143\times10^{-5}$
175	$1.032803976652\times10^{-3}$	$0.959929322342\times10^{-3}$	$0.959929321937\times10^{-3}$

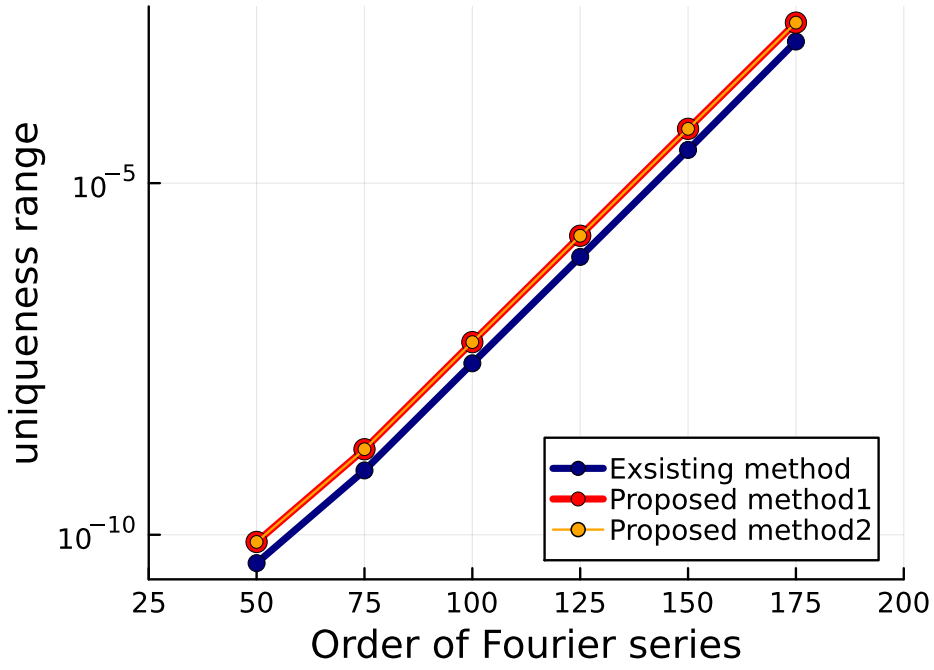

図 1　フーリエ級数の次数における一意性の保証範囲 $r_0, \frac{2\eta}{1-\delta}$

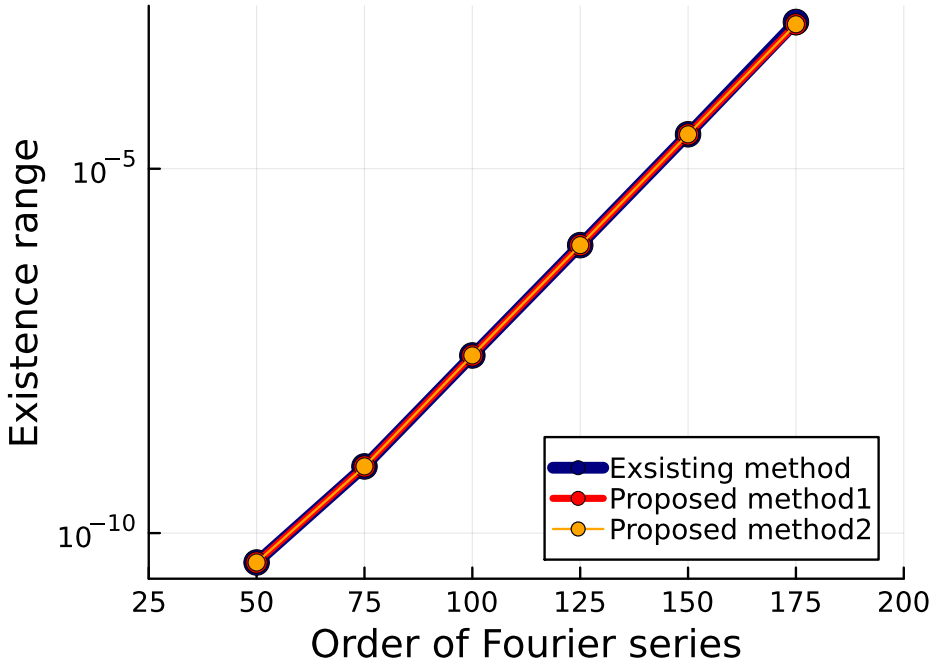

図 2　フーリエ級数の次数における存在保証範囲 r_0, ρ

全ての手法においてフーリエ級数の次数を増加していくと, 指数関数的に保証範囲が広がっていることがわかる. まず解の一意性が保証される範囲は, 提案手法 1,2 の方が既存手法に比べて明らかに広く, これは提案手法の方が精度が高いことを意味する. しかし, 本実験では既存手法において radii polynomial の零点探索にクラフチック法を用いたが, その性質から保証範囲にこのような差が出る. そのため, 一概に提案手法の方が精度が高くなるとは現時点ではいえない.

次に解の存在が保証される範囲をみてみる. この範囲は, 狭いほど精度が高いが全ての手法に大きな差異はない.

6.3.3　実行速度

3 つの手法に対して周期解を求める際のフーリエ級数の次数を変え, 数値計算を行ったときの実行時間をまとめる. 値は近似解の精度が保証された場合のときだけを抽出した. 表 11 は各手法をそれぞれ 10 回実行し, その平均値をまとめ表である. また図 3 は表 11 をグラフで示したものである.

表 11　フーリエ級数の次数におけるプログラム実行時間 [ms]

次数	既存手法	提案手法 1	提案手法 2
50	27436.2	25670.1	25804.1
75	27391.6	26152.5	26309.4
100	29041.0	27245.8	27327.4
125	29882.0	29084.2	28850.1
150	32609.5	31437.5	31086.2
175	34883.3	34755.9	33725.4

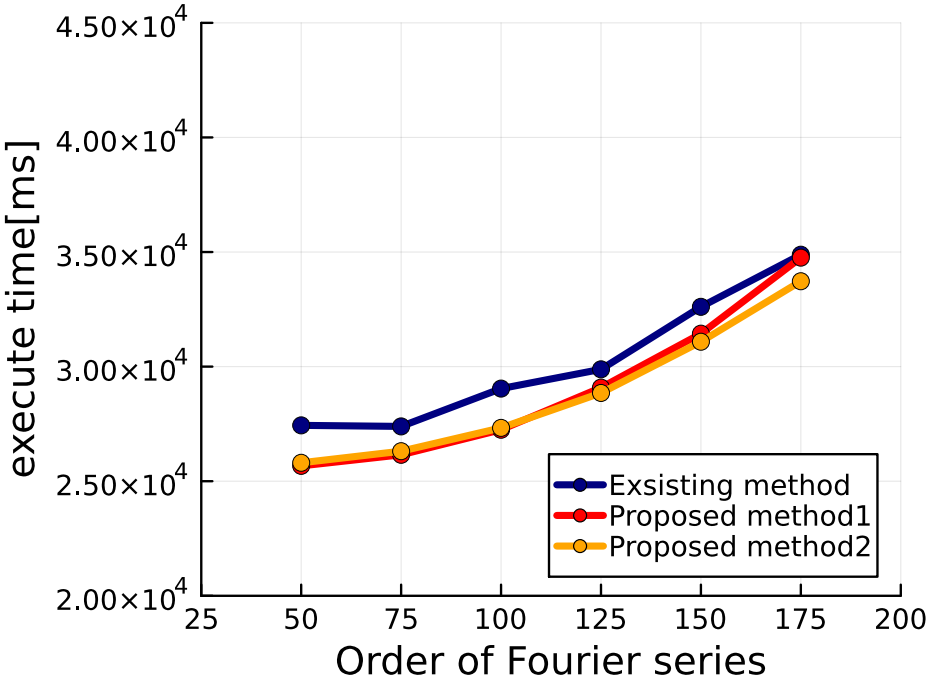

図 3　フーリエ級数の次数におけるプログラム実行時間 [ms]

実行時間は, 提案手法 1,2 の方が既存手法よりも明らかに短いことがわかる. これは, radii polynomial の零点探索手順が省略されたからである. また, 提案手法 1 では $A^\dagger$ の真の逆作用素を求めているため, フーリエ級数の次数が増加することによる実行時間が長くなる影響は大きい.

7　おわりに

本研究は radii polynomial approach の問題であった零点探索手順を省略することを目的としていた. そのため, 別の Newton-Kantorovich の定理を用いた精度保証法を参考に実現した. また, 常微分方程式である van der Pol 方程式の近似周期解に対して, それぞれの精度保証付き数値計算の精度および実行速度の比較を行った. すると解の一意性が保証される範囲は拡大し精度は向上し, 実行時間は短縮することができた.

今後の課題としては, 次のようなものが考えられる. 本研究では,

$$F(\bar{x}) = (\delta_0, \delta_1) \in \mathbb{C} \times \ell^1_{\nu'}$$

と定義して Y_0 の評価を行った. ここで, 無限次元ガウスの消去法を適用することで評価値の精度を高めることが可能であると考えられるため検討していく.

謝辞

本研究を進めるに際して, 千葉工業大学の関根晃太准教授には多くのご指導, ご助言を頂きましたこと深く感謝いたします。そして最後に、本研究に対しご助力頂いたすべての皆様に感謝し、お礼申し上げることで謝辞とさせて頂きます。

参考文献

[1] 中尾充宏, 山本野人, 精度保証付き数値計算 チュートリアル: 応用数理の最前線, 日本評論社, (1998/6/15)

[2] 大石進一, 精度保証付き数値計算, コロナ社, (2000/1/5)

[3] 石岡圭一, スペクトル法による数値計算入門, 東京大学出版会, (2004/11/22)

[4] 菊池文雄, 数値計算の誤差と精度, 丸善出版, (2022/10/25)

[5] 大石進一, 精度保証付き数値計算の基礎, コロナ社, (2018)

[6] 高安亮紀, Radii-polynomial 勉強会 講義資料, (2021/12/12)

[7] 関根晃太, 中尾充宏, 大石進一, Numerical verification methods for a system of elliptic PDEs, and their software library, (2021/1/1)

[8] 舩越康太, 井藤佳奈子, 大谷俊輔, 近藤慎佑, 高橋和暉, 瀬戸翔太二平泰知, 高安亮紀,Julia 言語を使った精度保証付き数値計算のチュートリアル, https://github.com/tak-lab/rigorous_numerics_tutorial_julia, (2023/4/13)

付録

既存手法で使用したプログラム [1]

ソースコード 1　Exsisting method.jl

```
#van der Pol方程式
 include("FourierChebyshev.jl")

 function vanderpol(du, u , μ ,t)
     x,y = u
     du[1] = y
     du[2] = μ*(1- x ^2)*y - x
 end

 function F_fourier(x, μ , η_0)
     N = length(x)/2
     ω = x[1]
     a = x[2:end]
     (a³,~) = powerconvfourier(a,3)
     eta = sum(a) - η_0

     k = -(N-1):(N-1)
     f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3

     return [eta;f]
 end

 function DF_fourier(x, μ)
     N = Int((length(x))/2)
     ω = x[1]
     a = x[2:end]
     k = (-N+1):(N-1)
     (a³,~) = powerconvfourier(a,3)

     DF = zeros(ComplexF64,2N,2N)

     DF[1,2:end] .= 1
     DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a³/3

     (~,a2) = powerconvfourier(a,2)

     M = zeros(ComplexF64,2*N-1, 2*N-1)

```

```
    for j=(-N+1):(N-1)
        M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
    end

    L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)

    DF[2:end,2:end] = L + M
    return DF
end


using DifferentialEquations
u_0 = [0.0; 2.0]
tspan = (0.0, 300)
μ = 1.0
prob = ODEProblem(vanderpol, u_0, tspan, μ)
sol = solve(prob,Tsit5(),reltol=1e-8,abstol=1e-8)
u = hcat(sol.u...)
ind = floor(Int, length(sol.t)/2)


#おおよその周期
# a = 30
# b = 36.55
a = 30
app_period = 6.55
timestep = 0.1

f_tmp = sol(a+app_period/2:timestep:a+3*app_period/2)
find_period = abs.(f_tmp .- sol(a))
(~,ind) = findmin(find_period[1,:])
b = a+app_period/2 + timestep*(ind-1)
#calc fouriercoeffs
N = 50 # size of Fourier
println("size of Fourier = $N")
a_0 = odefouriercoeffs(sol,N,[a,b])


using LinearAlgebra
# Initial value of Newton method
η_0 = 0.0
x = [2*pi/(b-a); a_0]

# Newton iteration
```

```
tol = 5e-12
F = F_fourier(x, μ, η_0)
println("Before step #1, ||F||_1 = $(norm(F,1))")
num_itr = 0

while num_itr ≤ 100
    global x = x - DF_fourier(x, μ)\F;
    global num_itr += 1
    global F = F_fourier(x, μ, η_0)
    # println("After step #$(num_itr), ||F||_1 = $(norm(F,1))")
    if norm(F,1) < tol
        break
    end
end


include("IntervalFunctions.jl")
setformat(:full)

ix = map(Interval,x)
iω = map(Interval,real(x[1]))
iā = map(Interval,x[2:end])
ν = 1.05

function DF_fourier(x::Vector{Complex{Interval{T}}}, μ) where T
    N = Int((length(x))/2)
    ω = x[1]
    a = x[2:end]
    k = (-N+1):(N-1)
    (a³,~) = powerconvfourier(a,3)

    DF = zeros(Complex{Interval{T}},2N,2N)

    DF[1,2:end] .= 1
    DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a³/3

    (~,a2) = powerconvfourier(a,2)

    M = zeros(Complex{Interval{T}},2*N-1, 2*N-1)

    for j=(-N+1):(N-1)
        M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
    end

```

```
127     L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)
128
129     DF[2:end,2:end] = L + M
130     return DF
131 end
132
133 iDF = DF_fourier(ix, μ);
134 iA = map(Interval,inv(mid.(iDF)))
135 A_a0 = iA[1,2:end]
136 A_a1 = iA[2:end,2:end]
137 A_01 = iA[2:end,1];
138
139 function F_fourier_ext(x::Vector{Complex{Interval{T}}}, μ, η_0) where T
140     N = length(x)/2
141     ω = x[1]
142     a = [zeros(Complex{Interval{T}},2*(Int(N)-1));x[2:end]; zeros(Complex{Interval{
            T}},2*(Int(N)-1))]
143     (~,a³) = powerconvfourier(x[2:end],3)
144     eta = sum(a) - η_0
145
146     k = -3*(N-1):3*(N-1)
147     f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3
148
149     return [eta;f]
150 end
151
152 function wnorm(a, ν)
153     N = (length(a)+1)/2 # length(a) = 2*N-1
154     k = (-N+1):(N-1)
155     w = ν.^abs.(k)
156     return sum(abs.(a).*w)
157 end
158
159 δ = F_fourier_ext(ix, μ, η_0)
160 δ_0 = δ[1]
161 δ_1 = δ[2:end]
162 δ_1_N = δ_1[2*(N-1)+1:end-2*(N-1)] #N-1 ,1 , N-1 = 2N-1
163 δ_1[2*(N-1)+1:end-2*(N-1)] .= 0
164 δ_1_tail = δ_1
165
166 λ_k(k,ω) = - k.^2 * ω^2 - μ* im * k * ω .+ 1
167
168 k_tail = -3*(N-1):3*(N-1)
```

```
Y0 = sup(max(abs(iA[1,1]*δ_0 + dot(A_a0,δ_1_N)), wnorm(A_01*δ_0 + A_a1*δ_1_N, ν
    ) + wnorm(δ_1_tail./(abs.(λ_k(map(Interval,Vector(k_tail)),iϖ))), ν)))

@show Y0;


# Z0 bounds
function wnorm_mat(A, ν)
    m = size(A,1) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
    return maximum(sum(w.*abs.(A),dims=1)./w')
end

function wsnorm(a, ν) # the input should be vector
    # the norm of dual space of the weighted ell^1
    m = length(a) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
    return maximum(abs.(a)./w)
end

B = I - iA*iDF #2N × 2N
Z0_0 = abs(B[1,1]) + wsnorm(B[1,2:end], ν)
Z0_1 = wnorm(B[2:end,1],ν) + wnorm_mat(B[2:end,2:end],ν)
Z0 = sup(max(Z0_0, Z0_1))
println("Z0 = $Z0")


# Z1 bounds
(~,ia²) = powerconvfourier(iā,2)
(~,ia³) = powerconvfourier(iā,3)

ζ = map(Interval,zeros(2*N-1))
for ell = -N+1:N-1
  j = ell-2*(N-1) : -N
  if isempty(j)
    ζ_1 = -1
  else
    w_j = ν.^abs.(j)
    ζ_1 = abs(μ*im*ell*iϖ) * maximum( abs.( ia²[ell.-j.+2*N.-1])./w_j)
  end
```

```
    j = N:ell+2*(N-1)
    if isempty(j)
      ζ_2 = -1
    else
      w_j = ν.^abs.(j)
      ζ_2 = abs(μ * im * ell * iϖ)* maximum( abs.(ia²[ell.-j.+2*N.-1])./w_j)
    end
    ζ[ell+N] = max(ζ_1, ζ_2)
  end

  conv = map(Interval,0)
  for k = N:2*(N-1)
      #positive
      global conv += abs(μ*im*k*ia³[k+4(N-1)+1])*ν^(k)/(3*abs(λ_k(k,iϖ)))
      #negative
      global conv += abs(-μ*im*k*ia³[-k+2*(N-1)+1]*ν^(k))/(3*abs(λ_k(-k,iϖ)))
  end

  w_n = ν^(N)
  iā_norm = wnorm(iā, ν)
  Z1_0 = abs(iA[1,1])/w_n + dot(abs.(A_a0), ζ)
  Z1_1 = wnorm(A_01, ν)/w_n + wnorm(abs.(A_a1)*ζ, ν) + conv +abs(μ*im*iϖ)*iā_norm
      ^2/(N*iϖ^2 - 1/N)
  Z1 = sup(max(Z1_0,Z1_1))
  println("Z1 = $Z1")


  #Z2 bound
  function bopnorm(A,tail_es, ν) # the operator norm of bounded operators with tail
      return max(wnorm_mat(A, ν),tail_es)
  end

  k = -N+1:N-1
  Ã = abs.(k).*abs.(A_a0)
  B = (k.^2).*abs.(A_a0)
  Ã_norm = wsnorm(Ã, ν)
  B_norm = wsnorm(B, ν)
  A_norm = wsnorm(A_a0, ν)

  Z2_20 = B_norm * (4*iϖ + 2*iā_norm) + 2*μ*Ã_norm * (1 + iϖ*iā_norm + iā_norm^2)
  Z2_30 = 3*B_norm + μ*Ã_norm*(3*iā_norm + iϖ)
  Z2_40 = 4*μ* Ã_norm/3

  tA = transpose(abs.(k)).*abs.(A_a1)
```

```
tB = transpose(k.^2).*abs.(A_a1)
tA_bopnorm = bopnorm(tA,(N+1)/abs(λ_k(N+1,iϖ)),ν)
tB_bopnorm = bopnorm(tB,1/(iϖ^2 - 1/(N^2)),ν)

Z2_21 = tB_bopnorm * (4*iϖ + 2*iā_norm) + 2*μ*tA_bopnorm * (1 + iϖ*iā_norm +
    iā_norm^2)
Z2_31 = 3*tB_bopnorm + μ*tA_bopnorm*(3*iā_norm + iϖ)
Z2_41 = 4*μ* tA_bopnorm/3

Z2_2 = sup(max(Z2_20, Z2_21))
Z2_3 = sup(max(Z2_30, Z2_31))
Z2_4 = sup(max(Z2_40, Z2_41))

@show Z2_2
@show Z2_3
@show Z2_4

Z2(r) = Z2_4*r.^3 + Z2_3*r.^2 + Z2_2*r


using ForwardDiff
#ニュートン法で近似解を計算する
function newton(F,x0)
    tol = 5e-10; count = 0;
    x = x0;
    Fx = F(x);
    while maximum(abs,Fx) ≥ tol && count ≤ 20
        DF = ForwardDiff.derivative(F,x);
        x -= DF\Fx;
        Fx = F(x);
        count += 1;
    end
    return x
end

#クラフチック写像
function krawczyk(F,X)
    iDF = ForwardDiff.derivative(F,X);
    c = mid.(X); ic = map(Interval,c);
    DF = ForwardDiff.derivative(F,c);
    R = inv(DF);
    M = I - R*iDF;
    return c - R*F(ic) + M*(X - c)
end
```

```
298
299 function verifynlss_krawczyk(F,c)
300     DF = ForwardDiff.derivative(F,c)
301     R = inv(DF)
302     r = abs.(R*F(c))
303     u = r .+ (sum(r)/length(r))
304     X = c .± u
305     K = krawczyk(F,X)
306     if all(K .⊂ X)
307         tol = 5e-10
308         count = 0
309         while maximum(radius,K) >= tol && count ≤ 100
310             K = krawczyk(F,K)
311             count += 1
312         end
313         success = 1
314         return success, K
315     end
316     println("Oh my way, verification is failed...return a improved approximate
            solution")
317     success = 0
318     return success, newton(F,c)
319 end
320
321
322 rp(r) = Z2(r).*r - (1-Z1-Z0)*r + Y0 # radii-polynomial
323 r0_mid = newton(rp,1e-10)
324 success, r0 = verifynlss_krawczyk(rp,r0_mid)
325 @show success
326 @show r0
327 rp(interval(sup(r0))) < 0
```

提案手法 1 で使用したプログラム

ソースコード 2　Proprsed method1.jl

```
1 #van der Pol方程式
2 include("FourierChebyshev.jl")
3
4 function vanderpol(du, u , μ ,t)
5     x,y = u
6     du[1] = y
7     du[2] = μ*(1- x ^2)*y - x
8 end
```

```
9
10  function F_fourier(x, μ, η_0)
11      N = length(x)/2
12      ω = x[1]
13      a = x[2:end]
14      (a³,~) = powerconvfourier(a,3)
15      eta = sum(a) - η_0
16
17      k = -(N-1):(N-1)
18      f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3
19
20      return [eta;f]
21  end
22
23  function DF_fourier(x, μ)
24      N = Int((length(x))/2)
25      ω = x[1]
26      a = x[2:end]
27      k = (-N+1):(N-1)
28      (a³,~) = powerconvfourier(a,3)
29
30      DF = zeros(ComplexF64,2N,2N)
31
32      DF[1,2:end] .= 1
33      DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a³/3
34
35      (~,a2) = powerconvfourier(a,2)
36
37      M = zeros(ComplexF64,2*N-1, 2*N-1)
38
39      for j=(-N+1):(N-1)
40          M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
41      end
42
43      L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)
44
45      DF[2:end,2:end] = L + M
46      return DF
47  end
48
49
50  #微分方程式数値計算パッケージ(DifferentialEquations)を用いた求解
51  using DifferentialEquations
52  u_0 = [0.0; 2.0]
```

```
tspan = (0.0, 300)
μ = 1.0
prob = ODEProblem(vanderpol, u_0, tspan, μ)
sol = solve(prob,Tsit5(),reltol=1e-8,abstol=1e-8)
u = hcat(sol.u...)
ind = floor(Int, length(sol.t)/2)
# plot(u[1, ind:end], u[2, ind:end], legend=false, size=(720,400))

#おおよその周期
# a = 30
# b = 36.55
a = 30
app_period = 6.55
timestep = 0.1

f_tmp = sol(a+app_period/2:timestep:a+3*app_period/2)
find_period = abs.(f_tmp .- sol(a))
(~,ind) = findmin(find_period[1,:])
b = a+app_period/2 + timestep*(ind-1)

#フーリエ係数の計算
N = 50 # size of Fourier
println("size of Fourier = $N")
a_0 = odefouriercoeffs(sol,N,[a,b])


using LinearAlgebra
#Newton法の初期値
η_0 = 0.0
x = [2*pi/(b-a); a_0]

#Newton反復
tol = 5e-12
F = F_fourier(x, μ, η_0)
# println("Before step #1, ||F||_1 = $(norm(F,1))")
num_itr = 0

while num_itr ≤ 100
    global x = x - DF_fourier(x, μ)\F;
    global num_itr += 1
    global F = F_fourier(x, μ, η_0)
# println("After step #$(num_itr), ||F||_1 = $(norm(F,1))")
    if norm(F,1) < tol
        break
```

```
97      end
98  end
99
100 # plot_solution(x,3)
101
102 include("IntervalFunctions.jl")
103
104 ix = map(Interval,x)
105 iω̄ = map(Interval,real(x[1]))
106 iā = map(Interval,x[2:end])
107 ν = 1.05
108
109 function DF_fourier(x::Vector{Complex{Interval{T}}}, μ) where T
110     N = Int((length(x))/2)
111     ω = x[1]
112     a = x[2:end]
113     k = (-N+1):(N-1)
114     (a³,~) = powerconvfourier(a,3)
115
116 DF = zeros(Complex{Interval{T}},2N,2N)
117
118     DF[1,2:end] .= 1
119     DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a³/3
120
121     (~,a2) = powerconvfourier(a,2)
122
123     M = zeros(Complex{Interval{T}},2*N-1, 2*N-1)
124
125     for j=(-N+1):(N-1)
126         M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
127     end
128
129     L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)
130
131     DF[2:end,2:end] = L + M
132     return DF
133 end
134
135 iDF = DF_fourier(ix, μ);
136 iA = inv(iDF) # map(Interval,inv(mid.(iDF)))
137 A_a0 = iA[1,2:end]
138 A_a1 = iA[2:end,2:end]
139 A_01 = iA[2:end,1];
140
```

```
function F_fourier_ext(x::Vector{Complex{Interval{T}}}, μ, η_0) where T
    N = length(x)/2
    ω = x[1]
    a = [zeros(Complex{Interval{T}},2*(Int(N)-1));x[2:end]; zeros(Complex{Interval{
        T}},2*(Int(N)-1))]
    (~,a³) = powerconvfourier(x[2:end],3)
    eta = sum(a) - η_0

    k = -3*(N-1):3*(N-1)
    f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3

    return [eta;f]
end

function wnorm(a, ν)
    N = (length(a)+1)/2 # length(a) = 2*N-1
    k = (-N+1):(N-1)
    w = ν.^abs.(k)
    return sum(abs.(a).*w)
end

δ = F_fourier_ext(ix, μ, η_0)
δ_0 = δ[1]
δ_1 = δ[2:end]
δ_1_N = δ⊠[2*(N-1)+1:end-2*(N-1)] #N-1 ,1 , N-1 = 2N-1
δ_1[2*(N-1)+1:end-2*(N-1)] .= 0
δ_1_tail = δ_1

λ_k(k,ω) = - k.^2 * ω^2 - μ* im * k * ω .+ 1


k_tail = -3*(N-1):3*(N-1)
Y0 = sup(max(abs(iA[1,1]*δ_0 + dot(A_a0,δ_1_N)), wnorm(A_01*δ_0 + A_a1*δ_1_N, ν
    ) + wnorm(δ_1_tail./(abs.(λ_k(map(Interval,Vector(k_tail)),iω̄))), ν)))

println("Y0 = $Y0")


# Z0 bounds
function wnorm_mat(A, ν)
    m = size(A,1) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
```

```
    return maximum(sum(w.*abs.(A),dims=1)./w')
end

function wsnorm(a, ν) # the input should be vector
# the norm of dual space of the weighted ell^1
    m = length(a) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
    return maximum(abs.(a)./w)
end


# Z1 bounds
(~,ia²) = powerconvfourier(iā,2)
(~,ia³) = powerconvfourier(iā,3)

ζ = map(Interval,zeros(2*N-1))
for ell = -N+1:N-1
    j = ell-2*(N-1) : -N
    if isempty(j)
        ζ_1 = -1
    else
        w_j = ν.^abs.(j)
        ζ_1 = abs(μ*im*ell*iω) * maximum( abs.( ia²[ell.-j.+2*N.-1])./w_j)
    end
    j = N:ell+2*(N-1)
    if isempty(j)
        ζ_2 = -1
    else
        w_j = ν.^abs.(j)
        ζ_2 = abs(μ * im * ell * iω)* maximum( abs.(ia²[ell.-j.+2*N.-1])./w_j)
    end
    ζ[ell+N] = max(ζ_1, ζ_2)
end

conv = map(Interval,0)
for k = N:2*(N-1)
#positive
    global conv += abs(μ*im*k*ia³[k+4(N-1)+1])*ν^(k)/(3*abs(λ_k(k,iω)))
#negative
    global conv += abs(-μ*im*k*ia³[-k+2*(N-1)+1]*ν^(k))/(3*abs(λ_k(-k,iω)))
end

```

```
w_n = ν^(N)
iā_norm = wnorm(iā, ν)

Z1_0 = abs(iA[1,1])/w_n + dot(abs.(A_a0), ζ)

Z1_1 = wnorm(A_01, ν)/w_n + wnorm(abs.(A_a1)*ζ, ν) + conv +abs(μ*im*iϖ)*iā_norm
    ^2/(N*iϖ^2 - 1/N)
Z1 = sup(max(Z1_0,Z1_1))
println("Z1 = $Z1")


#Z2 bound
function bopnorm(A,tail_es, ν) # the operator norm of bounded operators with tail
    return max(wnorm_mat(A, ν),tail_es)
end

k = -N+1:N-1
Ã = abs.(k).*abs.(A_a0)
B = (k.^2).*abs.(A_a0)
Ã_norm = wsnorm(Ã, ν)
B_norm = wsnorm(B, ν)
A_norm = wsnorm(A_a0, ν)

Z2_20 = B_norm * (4*iϖ + 2*iā_norm) + 2*μ*Ã_norm * (1 + iϖ*iā_norm + iā_norm^2)
Z2_30 = 3*B_norm + μ*Ã_norm*(3*iā_norm + iϖ)
Z2_40 = 4*μ* Ã_norm/3

tA = transpose(abs.(k)).*abs.(A_a1)
tB = transpose(k.^2).*abs.(A_a1)
tA_bopnorm = bopnorm(tA,(N+1)/abs(λ⊠(N+1,iϖ)), ν)
tB_bopnorm = bopnorm(tB,1/(iϖ^2 - 1/(N^2)), ν)

Z2_21 = tB_bopnorm * (4*iϖ + 2*iā_norm) + 2*μ*tA_bopnorm * (1 + iϖ*iā_norm +
    iā_norm^2)
Z2_31 = 3*tB_bopnorm + μ*tA_bopnorm*(3*iā_norm + iϖ)
Z2_41 = 4*μ* tA_bopnorm/3

@show Z2_2 = sup(max(Z2_20, Z2_21))
@show Z2_3 = sup(max(Z2_30, Z2_31))
@show Z2_4 = sup(max(Z2_40, Z2_41))

Z2(r) = Z2_4*r.^3 + Z2_3*r.^2 + Z2_2*r


```

```
#精度保証付き数値計算
r = 2*Y0/(1-Z1)
@show r
@show 2*Z2(r)*r+Z1
if 2*Z2(r)*r+Z1 <= 1
    p = Y0 / (1-(Z2(r)*r+Z1))
@show p
end
```

提案手法 2 で使用したプログラム

ソースコード 3　Proposed method2.jl

```
#van der Pol方程式
include("FourierChebyshev.jl")

function vanderpol(du, u , μ ,t)
    x,y = u
    du[1] = y
    du[2] = μ*(1- x ^2)*y - x
end

function F_fourier(x, μ , η_0)
    N = length(x)/2
    ω = x[1]
    a = x[2:end]
    (a³,~) = powerconvfourier(a,3)
    eta = sum(a) - η_0

    k = -(N-1):(N-1)
    f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3

    return [eta;f]
end

function DF_fourier(x, μ)
    N = Int((length(x))/2)
    ω = x[1]
    a = x[2:end]
    k = (-N+1):(N-1)
    (a³,~) = powerconvfourier(a,3)

    DF = zeros(ComplexF64,2N,2N)

```

```
32      DF[1,2:end] .= 1
33      DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a^3/3
34
35      (~,a2) = powerconvfourier(a,2)
36
37      M = zeros(ComplexF64,2*N-1, 2*N-1)
38
39      for j=(-N+1):(N-1)
40          M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
41      end
42
43      L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)
44
45      DF[2:end,2:end] = L + M
46      return DF
47  end
48
49
50  #微分方程式数値計算パッケージ(DifferentialEquations)を用いた求解
51  using DifferentialEquations
52  u_0 = [0.0; 2.0]
53  tspan = (0.0, 300)
54  μ = 1.0
55  prob = ODEProblem(vanderpol, u_0, tspan, μ)
56  sol = solve(prob,Tsit5(),reltol=1e-8,abstol=1e-8)
57  u = hcat(sol.u...)
58  ind = floor(Int, length(sol.t)/2)
59
60
61  #おおよその周期
62  # a = 30
63  # b = 36.55
64  a = 30
65  app_period = 6.55
66  timestep = 0.1
67
68  f_tmp = sol(a+app_period/2:timestep:a+3*app_period/2)
69  find_period = abs.(f_tmp .- sol(a))
70  (~,ind) = findmin(find_period[1,:])
71  b = a+app_period/2 + timestep*(ind-1)
72
73  #フーリエ係数の計算
74  N = 50 # size of Fourier
75  println("size of Fourier = $N")
```

```
a_0 = odefouriercoeffs(sol,N,[a,b])


using LinearAlgebra
#Newton法の初期値
η_0 = 0.0
x = [2*pi/(b-a); a_0]

#Newton反復
tol = 5e-12
F = F_fourier(x, μ, η_0)
# println("Before step #1, ||F||_1 = $(norm(F,1))")
num_itr = 0

while num_itr ≤ 100
    global x = x - DF_fourier(x, μ)\F;
    global num_itr += 1
    global F = F_fourier(x, μ, η_0)
    # println("After step #$(num_itr), ||F||_1 = $(norm(F,1))")
    if norm(F,1) < tol
        break
    end
end


include("IntervalFunctions.jl")

ix = map(Interval,x)
iω̄ = map(Interval,real(x[1]))
iā = map(Interval,x[2:end])
ν = 1.05

function DF_fourier(x::Vector{Complex{Interval{T}}}, μ) where T
    N = Int((length(x))/2)
    ω = x[1]
    a = x[2:end]
    k = (-N+1):(N-1)
    (a³,~) = powerconvfourier(a,3)

    DF = zeros(Complex{Interval{T}},2N,2N)

    DF[1,2:end] .= 1
    DF[2:end,1] = (- 2*ω*k.^2 - μ*im*k) .* a + μ*im*k .*a³/3

```

```
    (~,a2) = powerconvfourier(a,2)

    M = zeros(Complex{Interval{T}},2*N-1, 2*N-1)

    for j=(-N+1):(N-1)
        M[k.+N, j+N] = μ*im*k*ω.*a2[k.-j.+(2*N-1)]
    end

    L = diagm(- k.^2 * ω^2 - μ* im * k * ω .+ 1)

    DF[2:end,2:end] = L + M
    return DF
end

iDF = DF_fourier(ix, μ);
iA = map(Interval,inv(mid.(iDF)))
A_a0 = iA[1,2:end]
A_a1 = iA[2:end,2:end]
A_01 = iA[2:end,1];

function F_fourier_ext(x::Vector{Complex{Interval{T}}}, μ, η_0) where T
    N = length(x)/2
    ω = x[1]
    a = [zeros(Complex{Interval{T}},2*(Int(N)-1));x[2:end]; zeros(Complex{Interval{
        T}},2*(Int(N)-1))]
    (~,a³) = powerconvfourier(x[2:end],3)
    eta = sum(a) - η_0

    k = -3*(N-1):3*(N-1)
    f = (- k.^2 * ω^2 - μ* im * k * ω .+ 1) .* a + μ*im * k *ω .* a³ / 3

    return [eta;f]
end

function wnorm(a, ν)
    N = (length(a)+1)/2 # length(a) = 2*N-1
    k = (-N+1):(N-1)
    w = ν.^abs.(k)
    return sum(abs.(a).*w)
end

δ = F_fourier_ext(ix, μ, η_0)
δ_0 = δ[1]
δ_1 = δ[2:end]
```

```
δ_1_N = δ_1[2*(N-1)+1:end-2*(N-1)] #N-1 ,1 , N-1 = 2N-1
δ_1[2*(N-1)+1:end-2*(N-1)] .= 0
δ_1_tail = δ_1

λ_k(k,ω) = - k.^2 * ω^2 - μ* im * k * ω .+ 1


k_tail = -3*(N-1):3*(N-1)
Y0 = sup(max(abs(iA[1,1]*δ_0 + dot(A_a0,δ_1_N)), wnorm(A_01*δ_0 + A_a1*δ_1_N, ν
    ) + wnorm( δ_1_tail./(abs.(λ_k(map(Interval,Vector(k_tail)),iω))), ν )))

println("Y0 = $Y0")


# Z0 bounds
function wnorm_mat(A, ν)
    m = size(A,1) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
    return maximum(sum(w.*abs.(A),dims=1)./w')
end

function wsnorm(a, ν) # the input should be vector
    # the norm of dual space of the weighted ell^1
    m = length(a) # m = 2*N-1
    N = (m+1)/2
    k = -N+1:N-1
    w = ν.^abs.(k)
    return maximum(abs.(a)./w)
end

B = I - iA*iDF #2N × 2N
Z0_0 = abs(B[1,1]) + wsnorm(B[1,2:end], ν)
Z0_1 = wnorm(B[2:end,1],ν) + wnorm_mat(B[2:end,2:end],ν)
Z0 = sup(max(Z0_0, Z0_1))

println("Z0 = $Z0")


# Z1 bounds
(~,ia²) = powerconvfourier(iā,2)
(~,ia³) = powerconvfourier(iā,3)

```

```
ζ = map(Interval,zeros(2*N-1))
for ell = -N+1:N-1
  j = ell-2*(N-1) : -N
  if isempty(j)
    ζ_1 = -1
  else
    w_j = ν.^abs.(j)
    ζ_1 = abs(μ*im*ell*iϖ) * maximum( abs.( ia²[ell.-j.+2*N.-1])./w_j)
  end
  j = N:ell+2*(N-1)
  if isempty(j)
    ζ_2 = -1
  else
    w_j = ν.^abs.(j)
    ζ_2 = abs(μ * im * ell * iϖ)* maximum( abs.(ia²[ell.-j.+2*N.-1])./w_j)
  end
  ζ[ell+N] = max(ζ_1, ζ_2)
end

conv = map(Interval,0)
for k = N:2*(N-1)
    #positive
    global conv += abs(μ*im*k*ia³[k+4(N-1)+1])*ν^(k)/(3*abs(λ_k(k,iϖ)))
    #negative
    global conv += abs(-μ*im*k*ia³[-k+2*(N-1)+1]*ν^(k))/(3*abs(λ_k(-k,iϖ)))
end

w_n = ν^(N)
iā_norm = wnorm(iā, ν)

Z1_0 = abs(iA[1,1])/w_n + dot(abs.(A_a0), ζ)

Z1_1 = wnorm(A_01, ν)/w_n + wnorm(abs.(A_a1)*ζ, ν) + conv +abs(μ*im*iϖ)*iā_norm
    ^2/(N*iϖ^2 - 1/N)
Z1 = sup(max(Z1_0,Z1_1))
println("Z1 = $Z1")


#Z2 bound
function bopnorm(A,tail_es, ν) # the operator norm of bounded operators with tail
    return max(wnorm_mat(A, ν),tail_es)
end

k = -N+1:N-1
```

```
  Ã = abs.(k).*abs.(A_a0)
  B = (k.^2).*abs.(A_a0)
  Ã_norm = wsnorm(Ã,ν)
  B_norm = wsnorm(B,ν)
  A_norm = wsnorm(A_a0,ν)

  Z2_20 = B_norm * (4*iω + 2*iā_norm) + 2*μ*Ã_norm * (1 + iω*iā_norm + iā_norm^2)
  Z2_30 = 3*B_norm + μ*Ã_norm*(3*iā_norm + iω)
  Z2_40 = 4*μ* Ã_norm/3

  tA = transpose(abs.(k)).*abs.(A_a1)
  tB = transpose(k.^2).*abs.(A_a1)
  tA_bopnorm = bopnorm(tA,(N+1)/abs(λ_k(N+1,iω)),ν)
  tB_bopnorm = bopnorm(tB,1/(iω^2 - 1/(N^2)),ν)

  Z2_21 = tB_bopnorm * (4*iω + 2*iā_norm) + 2*μ*tA_bopnorm * (1 + iω*iā_norm +
      iā_norm^2)
  Z2_31 = 3*tB_bopnorm + μ*tA_bopnorm*(3*iā_norm + iω)
  Z2_41 = 4*μ* tA_bopnorm/3

  @show Z2_2 = sup(max(Z2_20, Z2_21))
  @show Z2_3 = sup(max(Z2_30, Z2_31))
  @show Z2_4 = sup(max(Z2_40, Z2_41))

  Z2(r) = Z2_4*r.^3 + Z2_3*r.^2 + Z2_2*r


  #精度保証付き数値計算

  r = 2*Y0/(1-(Z0+Z1))
  @show r

  if 2*Z2(r)*r+Z1 <= 1
      p = Y0 / (1-(Z2(r)*r+Z0+Z1))
      @show p
  end
```

区間行列積の関数 [1]

ソースコード 4　IntervalFunctions.jl

```
  ### Interval Matrix Multiplication
function ufp(P)
    u = 2.0^(-53);
```

```
    va = 2.0^52 + 1;
    q = va * P;
    T = (1 - u)*q;
    return abs(q - T);
end

function succ(c)
    s_min = 2.0^-1074;
    u = 2.0^-53;
    va = u * (1.0 + 2.0 * u);
    if abs(c) >= (1. / 2.) * u^(-2) * s_min # 2^(-969)(Float64)
        e = va * abs(c);
        succ = c + e;
    elseif abs(c) < (1. / 2.) * u^(-1) * s_min # 2^(-1022)(Float64)
        succ = c + s_min;
    else
        C = u^(-1) * c;
        e = va * abs(C);
        succ = (C + e) * u;
    end
    return succ
end

function pred(c)
    s_min = 2.0^-1074;
    u = 2.0^-53;
    va = u * (1.0 + 2.0 * u);
    if abs(c) >= (1. / 2.) * u^(-2) * s_min # 2^(-969)(Float64)
        e = va * abs(c);
        pred = c - e;
    elseif abs(c) < (1. / 2.) * u^(-1) * s_min # 2^(-1022)(Float64)
        pred = c - s_min;
    else
        C = u^(-1) * c;
        e = va * abs(C);
        pred = (C - e) * u;
    end
    return pred
end

function mm_ufp(A_mid, B_mid) # A_mid, B_mid: Point matrix
    u = 2.0^(-53);
    realmin = 2.0^(-1022);
    n = size(A_mid,2);
```

```
48
49      if(2*(n+2)*u>=1)
50          error("mm_ufp is failed!(2(n+2)u>=1)")
51      end
52      # C_mid = A_mid * B_mid;
53      # C_rad = (n+2) * u * ufp.(abs.(A_mid)*abs.(B_mid)) .+ realmin;
54      # return C_mid, C_rad;
55      return A_mid * B_mid, (n+2) * u * ufp.(abs.(A_mid)*abs.(B_mid)) .+ realmin
56  end
57
58  function imm_ufp(A_mid, A_rad, B_mid, B_rad) # A = <A_mid, A_rad>, B = <B_mid, B_rad
        >: Interval matrix
59      u = 2.0^(-53);
60      realmin = 2.0^(-1022);
61      n = size(A_mid,2);
62
63      if(2*(n+2)*u>=1)
64          error("mm_ufp is failed!(2(n+2)u>=1)")
65      end
66  # C, R = mm_ufp(A_mid,B_mid);
67      # C_mid = A_mid * B_mid;
68      R = (n+2) * u * ufp.(abs.(A_mid)*abs.(B_mid)) .+ realmin;
69
70  # T_1, T_2 = mm_ufp(abs.(A_mid), B_rad);
71      T1 = abs.(A_mid) * B_rad;
72      T2 = (n+2)*u*ufp.(T1) .+ realmin;
73
74  # T_3 = succ.(abs.(B_mid)+B_rad);
75      T3 = succ.(abs.(B_mid)+B_rad)
76
77  # T_4, T_5 = mm_ufp(A_r, T_3);
78      T4 = A_rad * T3;
79      T5 = (n+2)*u*ufp.(T4) .+ realmin;
80
81      rad_sum = R + T1 + T2 + T4 + T5;
82
83      # C_rad = succ.(rad_sum + 4*u*ufp.(rad_sum));
84
85      # return C_mid, C_rad;
86      return A_mid * B_mid, succ.(rad_sum + 4*u*ufp.(rad_sum))
87  end
88
89  # USE IntervalArithmetic.jl
90  using IntervalArithmetic
```

```
function int_mul(A::Matrix{T}, B::Matrix{T}) where T
    Cmid, Crad = mm_ufp(A, B);
    return Cmid .± Crad
end

function int_mul(A::Matrix{Interval{T}}, B::Matrix{T}) where T
    Cmid, Crad = imm_ufp(mid.(A), radius.(A), B, zeros(size(B)));
    return Cmid .± Crad
end

function int_mul(A::Matrix{T}, B::Matrix{Interval{T}}) where T
    Cmid, Crad = imm_ufp(A, zeros(size(A)), mid.(B), radius.(B));
    return Cmid .± Crad
end

function int_mul(A::Matrix{Interval{T}}, B::Matrix{Interval{T}}) where T
    Cmid, Crad = imm_ufp(mid.(A), radius.(A), mid.(B), radius.(B));
    return Cmid .± Crad
end

function int_mul(A::Matrix{Complex{T}}, B::Matrix{T}) where T
    Ar = real.(A); Ai = imag.(A); # (Ar + im*Ai)*B = Ar*B + im*(Ai*B)
    return int_mul(Ar, B) + im * int_mul(Ar, B)
end

function int_mul(A::Matrix{T}, B::Matrix{Complex{T}}) where T
    Br = real.(B); Bi = imag.(B); # A*(Br + im*Bi) = A*Br + im*(A*Bi)
    return int_mul(A, Br) + im * int_mul(A, Bi)
end

function int_mul(A::Matrix{Complex{T}}, B::Matrix{Complex{T}}) where T
    Ar = real.(A); Ai = imag.(A); Br = real.(B); Bi = imag.(B);
    # (Ar + im*Ai)*(Br + im*Bi) = (Ar*Br - Ai*Bi) + im*(Ar*Bi + Ai*Br)
    return (int_mul(Ar,Br) - int_mul(Ai, Bi)) + im * (int_mul(Ar, Bi) + int_mul(Ai,
        Br))
end


### Interval Linear system solver



### Verify FFT using Interval Arithmetic
function verifyfft(z::Vector{T}, sign=1) where T
```

```
    n = length(z); col = 1; array1 = true
    if n==1
        Z = map(T,z)
        return Z
    else
        isrow_ = false
    end
    log2n = Int(round(log2(n))) #check dimension
    if 2^log2n ≠ n # return error if n is not the powers of 2
        error("length must be power of 2")
    end
    #bit-reversal
    f = 2^(log2n-1)
    v = [0;f]
    for k = 1:log2n-1
        f = f >> 1
        v = append!(v,f.+v)
    end
    z2 = zeros(n,col)
    if isa(real(z[1]),Interval)
        z2 = map(T,z2)
    end
    # replace z
    for j = 1: n
        z2[j,:] = z[v[j]+1,:]
    end
    #Danielson-Lanczos algorithm
    Z = complex(map(Interval,z2))
    Index = reshape([1:n*col;],n,col)

    theta = map(Interval,sign * (0:n-1)/n); # division exact because n is power of 2
    Phi = cospi.(theta) + im*sinpi.(theta) # SLOW?

    v = [1:2:n;]
    w = [2:2:n;]
    t = Z[w,:]
    Z[w,:] = Z[v,:] - t
    Z[v,:] = Z[v,:] + t
    for index in 1: (log2n-1)
        m = 2^index
        m2 = 2*m
        vw = reshape([1:n;],m2,Int(n/m2))
        v = vw[1: m, :]
        w = vw[m+1: m2, : ]
```

```
178         indexv = reshape(Index[v[:],:],m,Int(col*n/m2))
179         indexw = reshape(Index[w[:],:],m,Int(col*n/m2))
180         Phi1 = repeat(Phi[1:Int(n/m):end],outer=[1,Int(col*n/m2)])
181         t = Phi1 .* Z[indexw]
182         Z[indexw] = Z[indexv] - t
183         Z[indexv] = Z[indexv] + t
184     end
185     reverse(Z[2:end,:],dims=2)
186      if sign==-1
187         Z = Z/n
188     end
189     if isrow_
190         Z = transpose(Z) #transpose of Z
191     end
192     if array1
193         Z = Z[:,1]
194     end
195     return Z
196 end
197
198 ### Rigorous convolution algorithm via FFT
199 function powerconvfourier(a::Vector{Complex{Interval{T}}},p) where T
200     M = Int((length(a)+1)/2) # length(a) = 2M-1
201     N = (p-1)*M
202     ia = map(Interval, a)
203
204     length_ia = 2*p*M-1
205     length_ia_ext = nextpow(2,length_ia)# 2pM-2+2L
206
207     L = Int((length_ia_ext - length_ia + 1)/2)
208
209     # step.1 : padding (p-1)M + L zeros for each sides
210     ia_ext = map(Complex{Interval},zeros(length_ia_ext))
211     ia_ext[L+N+1:end-L-N+1] = ia #\tilda{a}
212
213     # step.2 : inverse fft
214     ib_ext = verifyfft(ifftshift(ia_ext), -1) #sign = -1 : ifft
215
216     # step.3 : power p elementwisely
217     ib_ext  = ib_ext.^p
218
219     # step.4 : fft with rescaling
220     ic_ext  = fftshift(verifyfft(ib_ext , 1)) * length_ia_ext^(p-1) #sign = 1 : fft
221
```

```
222 # return ic_ext ,ic_ext
223     return ic_ext [L+N+1:end-N-L+1], ic_ext [L+p:end-(L+p-2)] # return (truncated,
        full) version
224 end
```

フーリエ級数・チェビシェフ級数を扱う関数 [1]

ソースコード 5　FourierChebyshev.jl

```
1   ### Fourier functions
2  using FFTW, Plots
3  function fouriercoeffs(f, N, I=[0,2π])
4      # f: any periodic function on I
5      # N: size of Fourier series
6      a = I[1]; b = I[2]
7      h = (b-a)/(2N-1)
8      j = 0:2N-2
9      x_j = a .+ j*h
10     f_j = f.(x);
11     return fftshift(fft(f_j))/(2N-1)
12 end
13
14 function odefouriercoeffs(f, N, I, n=1)
15     a = I[1]; b = I[2];
16     h = (b-a)/(2N-1)
17     j = 0:2N-2
18     x_j = a .+ j*h
19     f_j = f(x_j)[n,:]
20     return fftshift(fft(f_j))/(2N-1)
21 end
22
23 function plot_fourier(bc, I=[0,2π])
24     # bc: Fourier coefficients
25     a = I[1]; b = I[2]
26     N = (length(bc)+1)/2 # 2N-1
27     n_pad = 500
28     bc_pad = [zeros(n_pad);bc;zeros(n_pad)]
29     N_pad = N + n_pad
30     h_pad = (b-a)/(2N_pad-1)
31     xj_pad = a .+ h_pad*(0:2N_pad-2)
32     fNj_pad = real((2N_pad-1)*ifft(ifftshift(bc_pad)))
33     plot(xj_pad, fNj_pad, legend=false, xlabel = "\$x\$", ylabel = "\$f(x)\$")
34 end
35
```

```
function plot_fourier!(bc, I=[0,2π];label="")
    # bc: Fourier coefficients
    a = I[1]; b = I[2]
    N = (length(bc)+1)/2 # 2N-1
    n_pad = 500
    bc_pad = [zeros(n_pad);bc;zeros(n_pad)]
    N_pad = N + n_pad
    h_pad = (b-a)/(2N_pad-1)
    xj_pad = a .+ h_pad*(0:2N_pad-2)
    fNj_pad = real((2N_pad-1)*ifft(ifftshift(bc_pad)))
    plot!(xj_pad, fNj_pad, label=label)
end

function plot_fouriercoeffs(bc)
    N = (length(bc)+1)/2 # 2N-1
    plot(-N+1:N-1,abs.(bc),yscale=:log10,
    legend=false,
    xlabel = "\$k\$",
    ylabel = "\$|\\bar{c}_k\\,|\$",
    title = "Absolute values of Fourier coefficients"
    )
end

function plot_solution(u, index) # u = [ω, a_{-N+1}, ..., a_0, ..., a_{N-1}],
    length(u) = 2N
    # index = 1: profile of solution
    # 2: Fourier mode
    # 3: phase profile
    ω = real(u[1])
    L = 2π / ω
    a = u[2:end]
    N = length(u)/2 # N: size of Fourier
    n_pad = 500
    a_pad = [zeros(n_pad);a;zeros(n_pad)]
    N_pad = N + n_pad
    dx = L/(2*N_pad-1)
    x = dx*(0:2*N_pad-2)
    if index == 1
    # Plot profile:
        plot(plot_fourier(a, [0,L]),
        # plot(x,real((2N_pad-1)*ifft(ifftshift(a_pad))),
            xlabel = "\$t\$",
            ylabel = "\$x\\,(t)\$",
            line = 1.6,
```

```
79              title = "Profile of solution",
80              size = (720,400),
81              legend = false,
82          )
83      elseif index == 2
84      # Plot Fourier coefficients:
85          plot(plot_fouriercoeffs(a),
86          # plot((-N+1):(N-1),abs.(a),yscale=:log10,
87              xlabel = "\$k\$",
88              ylabel = "\$|a_k\\,|\$",
89              line = 1.6,
90              title = "Absolute values of Fourier coefficients",
91              size = (720,400),
92              legend = false,
93          )
94      elseif index == 3
95      # Plot phase:
96        k = (-N_pad+1):(N_pad-1)
97        plot(real((2N_pad-1)*ifft(ifftshift(a_pad))),real((2N_pad-1)*ifft(ifftshift(
              a_pad.*(im*k*ω)))),
98              xlabel = "\$x(t)\$",
99              ylabel = "\$\\dot{x}\\,(t)\$",
100             line = 1.6,
101             title = "Phase plot of a numerical solution",
102             size = (720,400),
103             legend = false,
104         )
105     end
106 end
107
108 function plot_solution!(u)
109     L = 2π/real(u[1])
110     a = u[2:end]
111     N = length(u)/2
112     n_pad = 1000
113     a_pad = [zeros(n_pad);a;zeros(n_pad)]
114     N_pad = N+n_pad
115     k = (-N_pad+1):(N_pad-1)
116     dx = L/(2*N_pad-1)
117     x = dx*(0:2*N_pad-2)
118     plot!(real((2N_pad-1)*ifft(ifftshift(a_pad))),real((2N_pad-1)*ifft(ifftshift(
            a_pad.*(im*k)))),line=1.6,)
119 end
120
```

```
function powerconvfourier(a::Vector{Complex{T}},p) where T
    M = Int((length(a)+1)/2)
    N = (p-1)*M
    ta = [zeros(N,1);a;zeros(N,1)] # 1. Padding zeros: size(ta) = 2pM-1
    tb = ifft(ifftshift(ta)) # 2. IFFT of ta
    tb  = tb.^p # 3. tb*^tb
    c  = fftshift(fft(tb ))*(2.0*p*M-1)^(p-1)
    return c [N+1:end-N], c [p:end-(p-1)]# return (truncated, full) version
end


### Chebyshev functions
function chebpts(n, a=-1, b=1) # n: maximum order of Chebyshev polynomials
    tt = range(0, stop=π, length=n+1)
    x = cos.(tt)
    return (1.0 .- x).*a/2 + (1.0 .+ x).*b/2
end

function chebcoeffs(f,M,I=[-1,1])
    a = I[1]; b = I[2]
    n = M-1
    cpts = chebpts(n, a, b)
    fvals = f.(cpts)
    FourierCoeffs = real(fft([fvals;reverse(fvals[2:end-1])]))
    ChebCoeffs = FourierCoeffs[1:n+1]/n
    ChebCoeffs[1] = ChebCoeffs[1]/2
    ChebCoeffs[end] = ChebCoeffs[end]/2
    return ChebCoeffs # return Two-sided Chebyshev
end

function cheb(f,I=[-1;1]; tol = 5e-15,Nmax = 10000)
    a = I[1]; b = I[2]; m = 0.5*(a+b); r = 0.5*(b-a); x = rand(5)
    x1 = m .+ x*r; x2 = m .- x*r
    if f.(x1) ≈ f.(x2)
        odd_even = 1 # even function: 1
    elseif f.(x1) ≈ -f.(x2)
        odd_even = -1 # odd function: -1
    else
        odd_even = 0 # otherwise: 0
    end
    i = 3
    schbc = 0 # sampling chebyshev coefficients
    while true
        schbc = chebcoeffs(f,2^i+1,I)
```

```
        if all(abs.(schbc[end-2:end]) .< tol) || (2^i+1 > Nmax)
            break
        end
        i += 1
    end
    M = findlast(abs.(schbc) .> tol)
    cc = chebcoeffs(f,M,I)
    if odd_even == 1 # even function
        cc[2:2:end] .= 0
    elseif odd_even == -1 # odd function
        cc[1:2:end] .= 0
    end
    return cc # return Two-sided Chebyshev
end

function plot_chebcoeffs(f)
    zero_ind = findall(x->x==0, f)
    f[zero_ind] .= f[zero_ind .+ 1]
    plot(0:length(f)-1, abs.(f),
        yscale=:log10,
        title="Chebyshev coefficients",
        xlabel="Degree of Chebyshev polynomial",
        ylabel="Magnitude of coefficient",
        size = (800,400),
        legend = false,
    )
end

function clenshaw(a,x) # Clenshaw's algorithm
# a: (Two-sided) Chbyshev coefficients
# x: evaluating points in [-1,1]
    n = length(a)-1
    bk1 = 0.0
    bk2 = 0.0
    x = 2x
    for r = (n+1):-2:3
        bk2 = x.*bk1 .- bk2 .+ a[r]
        bk1 = x.*bk2 .- bk1 .+ a[r-1]
    end
    if isodd(n)
        b2 = x.*bk1 .- bk2 .+ a[2]
        bk2 = bk1 # b3
        bk1 = b2
    end
```

```
209     return -bk2 .+ 0.5x .* bk1 .+ a[1] # y = c(1) + .5*x.*bk1 - bk2;
210 end
211
212 function clenshaw_secondkind(a,x) # Clenshaw's algorithm
213 # a: (Two-sided) Chbyshev coefficients
214 # x: evaluating points in [-1,1]
215     n = length(a)-1
216     bk0 = 0
217     bk1 = 0
218     for r = (n+1):-1:1
219         tmp = 2x.*bk0 .- bk1 .+ a[r]
220         bk1 = bk0
221         bk0 = tmp
222     end
223     return bk0 #.+ bk1*(-x)
224 end
225
226 function eval_cheb(a,x; I=[-1,1])
227 # a: (Two-sided) Chbyshev coefficients
228 # x: evaluating points in domain
229     ξ = 2*(x.-I[1])/(I[2]-I[1]) .- 1
230     return clenshaw(a, ξ)
231 end
232
233 function eval_cheb_naive(ChebCoeffs_twosided,x; I=[-1,1])
234     M = length(ChebCoeffs_twosided) # M: size of chebyshev
235     a = I[1]; b = I[2]
236     k = 0:M-1
237     ξ = 2*(x.-a)/(b-a) .- 1
238     return cos.(Vector(k)' .* acos.(ξ)) * ChebCoeffs_twosided
239 end
240
241 function eval_cheb_bc(ChebCoeffs_twosided,x,n=200; I=[-1,1]) # Barycentric
        interportion formula
242     M = length(ChebCoeffs_twosided) # M: size of chebyshev
243     a = I[1]; b = I[2]
244     k = 0:M-1
245     ξ  = chebpts(n)
246     xc = (1.0 .- ξ )*a/2 + (1.0 .+ ξ )*b/2 # Chebyshev points in [a,b]
247     fxc = cos.(Vector(k)' .* acos.(ξ )) * ChebCoeffs_twosided
248     valnum = length(x)
249     ξ = 2*(x.-a)/(b-a) .- 1
250     # ξ = range(-1,stop=1,length=valnum)
251     # x = (1.0 .- ξ)*a/2 + (1.0 .+ ξ)*b/2
```

```
252     λ = [1/2; ones(n-1); 1/2] .* (-1).^(0:n)
253
254     numer = zeros(valnum)
255     denom = zeros(valnum)
256     exact = zeros(Bool,valnum)
257
258     for j = 1:n+1
259         xdiff = x .- xc[j]
260         temp = λ[j] ./ xdiff
261         numer += temp * fxc[j]
262         denom += temp
263         exact[xdiff.==0] .= true
264     end
265
266     fx = numer ./ denom
267     jj = findall(exact)
268     fx[jj] = f.(x[jj])
269     fx[jj] = cos.(Vector(k)' .* acos.(ξ[jj])) * ChebCoeffs_twosided
270     return fx
271 end
272
273 function plot_cheb(ChebCoeffs_twosided; I=[-1,1],title="",label="",legend=true) #
        Input: Two-sided Chebyshev
274     # M = length(ChebCoeffs_twosided) # M: size of chebyshev
275     # a = I[1]; b = I[2];
276     x = range(I[1],stop=I[2],length=5000)
277     # ξ = 2*(x.-a)/(b-a) .- 1
278     fx = eval_cheb(ChebCoeffs_twosided,x,I=I)
279     plot(x, fx, legend=legend, label=label, title=title, xlabel="\$x\$",ylabel="\$f(x
            )\$")
280 end
281
282 function plot_cheb!(ChebCoeffs_twosided; I=[-1,1],title="",label="",legend=true) #
        Input: Two-sided Chebyshev
283     # M = length(ChebCoeffs_twosided) # M: size of chebyshev
284     # a = I[1]; b = I[2];
285     x = range(I[1],stop=I[2],length=5000)
286     # ξ = 2*(x.-a)/(b-a) .- 1
287     fx = eval_cheb(ChebCoeffs_twosided,x,I=I)
288     plot!(x, fx, legend=legend, label=label, title=title, xlabel="\$x\$",ylabel="\$f(
            x)\$")
289 end
290
291 function chebdiff(a; I=[-1,1])# Input is Two-sided
```

```
292     M = length(a)
293     b = zeros(M+1)
294     for r = M-1:-1:1
295         b[r] = b[r+2] + 2*r*a[r+1]
296     end
297     b[1] /= 2.0
298     return b[setdiff(1:end,end)]*(2/(I[2]-I[1])) # Output is Two-sided
299 end
300
301 function chebdiff_oneside(a; I=[-1,1])# Input is One-sided
302     M = length(a)
303     b = zeros(M+1)
304     for r = M-1:-1:1
305         b[r] = b[r+2] + 2*r*a[r+1]
306     end
307     return b[setdiff(1:end,end)]*(2/(I[2]-I[1])) # Output is One-sided
308 end
309
310 function chebdiff_secondkind(a; I=[-1,1]) # Input is Two-sided
311     M = length(a)
312     b = zeros(M-1)
313     for n = 0:M-2
314         b[n+1] = (n+1)*a[n+2]
315     end
316     return b*(2/(I[2]-I[1])) # Output is second kind (Two-sided)
317 end
318
319 function chebindefint(a; I=[-1,1])# Input is Two-sided
320     M = length(a)
321     a_ext = zeros(M+2)
322     a_ext[1] = 2*a[1]
323     a_ext[2:M] = a[2:M]
324     A = zeros(M+1)
325     for n = 1:M
326         A[n+1] = (a_ext[n] - a_ext[n+2])/(2n)
327     end
328     # A[1] = sum(A[2:2:end]) - sum(A[3:2:end]) # takes the value 0 at the left
            endpoint
329     return A * (I[2]-I[1])/2
330 end
331
332 function chebint(a; I=[-1,1])# Input is Two-sided
333     M = length(a)
334     n = 0:2:M-1
```

```
335     return sum(2a[1:2:end]./(1.0 .- n.^2))*((I[2]-I[1])/2)
336 end
```