

# 二重指数関数型数値積分公式を用いた 行列符号関数の計算の改良と応用

2022年9月8日

日本応用数理学会2022年度年会

電気通信大学 情報理工学研究科

宮下朋也 山本有作

# はじめに

## 行列符号関数

### • 定義

- 複素数 $z$ に対して、符号関数 $\text{sign}(z)$ は次のように定義される.

$$\text{sign}(z) = \begin{cases} 1, & \text{Re } z > 0 \\ -1, & \text{Re } z < 0 \end{cases}$$

- これを行列に拡張したものが**行列符号関数**である.
- 正方行列 $A$ のJordan標準形を $A = XJX^{-1}$ とし,  $\text{diag}(J) = \text{diag}(\lambda_i)$ とすると, 行列符号関数 $\text{sign}(A)$ は次式により定義される.

$$\text{sign}(A) = X\text{sign}(J)X^{-1} = X\text{diag}(\text{sign}(\lambda_i))X^{-1}$$

### • 応用例

- シルベスター方程式の求解
- 固有値計算
- 行列平方根計算

# はじめに

- 行列符号関数の数値計算方法

- 定義に基づく計算方法

- 対称行列の場合 : 対角化の計算量大

- 非対称行列の場合: Jordan標準形の数値計算の不安定性

- Schur分解

- 高精度であるが計算量大

- Newton法

- 積分表示に基づく方法

- [1]で提案

- 数値積分の各標本点での計算が独立なので並列化に向いている

# 本発表の目的

- 二重指数関数型数値積分公式(DE公式)を用いた行列符号関数の数値積分方法について、以下の観点から評価と改良を行う
  - 積分区間の異なる2つの数値計算方法の理論的・実験的比較
    - 積分区間が半無限区間, 全無限区間の計算方法をそれぞれ比較
  - 固有値分布による収束速度の比較
  - 行列符号関数の性質を用いた最適なスケーリング
    - $\text{sign}(A) = \text{sign}(cA)$  ( $c > 0$ )を用いて収束を速める
  - 行列平方根
    - 行列符号関数と行列平方根との関係を用いて後者に対する数値計算法を導出
  - 新しい誤差上界の導出
    - 入力行列を対称行列と仮定することにより新たな上界を導出

# 目次

- はじめに
- 二重指数関数型数値積分公式(DE公式)について
- 行列符号関数について
- 2つの数値計算法の比較
- $\text{sign}(A) = \text{sign}(cA)$ を用いた最適なスケーリング
- 行列平方根への応用
- 新しい誤差上界の導出

# 二重指数関数型数値積分公式(DE公式)

## DE公式

- 台形公式は積分区間が $(-\infty, +\infty)$ で被積分関数が急減少な解析関数の場合には非常に高精度な数値積分公式である
  - 一般の積分についても、そうなるように変数変換をする
- 特に、変換後の被積分関数が二重指数関数的に減衰するように選んだものをDE公式という

台形公式による近似

$$T_h = h \sum_{k=-N_-}^{N_+} f(kh)$$

変数変換後の台形公式

$$T_h^* = h \sum_{k=-N_-}^{N_+} f(\phi(kh)) \phi'(kh)$$

# 二重指数関数型数値積分公式(DE公式)

## 台形公式の離散化誤差

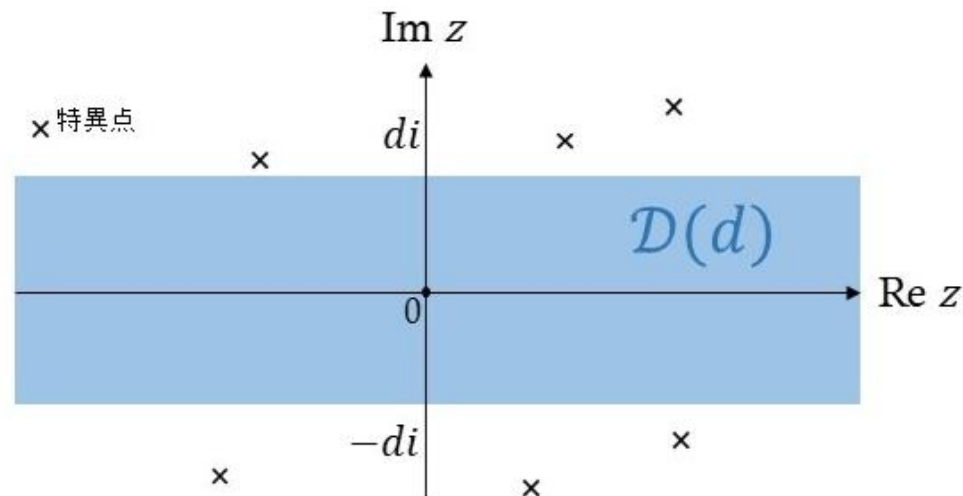
- 解析関数 $f(z)$ が次の領域 $\mathcal{D}(d)$ 上で正則とする

$$\mathcal{D}(d) = \{z \in \mathbb{C} \mid |\operatorname{Im} z| < d\}$$

このとき、積分 $I = \int_{-\infty}^{+\infty} f(x) dx$ を刻み幅 $h$ の台形公式で計算したときの離散化誤差は次のように評価される

$$|E_D| = |T_h - I| \leq \frac{\exp\left(-\frac{2\pi d}{h}\right)}{1 - \exp\left(-\frac{2\pi d}{h}\right)} \Lambda(f, d - 0)$$

$$\text{ただし, } \Lambda(f, c) \equiv \int_{-\infty}^{+\infty} |f(x) + ic| + |f(x - ic)| dx$$



特異点が存在しない領域 $\mathcal{D}(d)$ が広いほど、つまり $d$ が大きく取れるほど収束は速くなる

# 二重指数関数型数値積分公式(DE公式)

積分区間ごとの最適な変換関数

$$I = \int_{-\infty}^{\infty} f(x) dx \Rightarrow x = \sinh\left(\frac{\pi}{2} \sinh t\right) \equiv \phi_{\sinh}(t)$$

$$I = \int_0^{\infty} f(x) dx \Rightarrow x = \exp\left(\frac{\pi}{2} \sinh t\right) \equiv \phi_{\exp}(t)$$

行列符号関数をDE公式を用いて計算するとき上記のいずれかを用いる



# 行列符号関数

## 行列符号関数の積分表示形式

$$\begin{aligned}\text{sign}(A) &= \frac{2}{\pi} A \int_0^{\infty} (t^2 I + A^2)^{-1} dt \\ &= \frac{1}{\pi} A \int_{-\infty}^{\infty} (t^2 I + A^2)^{-1} dt\end{aligned}$$

行列 $A$ の固有値を $\lambda_k$ とすると被積分関数の特異点は $\pm i\lambda_k$ となる

- DE公式を適用すると、以下の2つの計算式が得られる

$$\text{sign}(A) = \frac{2}{\pi} A \sum_{k=-N_-}^{N_+} h(\phi_{\text{exp}}^2(kh)I + A^2)^{-1} \phi'_{\text{exp}}(kh)$$

数値計算法1  
[1]で採用

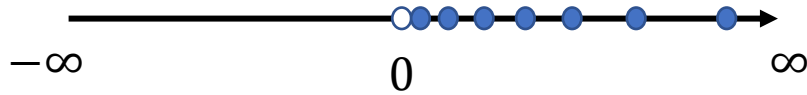
$$= \frac{A}{\pi} \sum_{k=-N_-}^{N_+} h(\phi_{\text{sinh}}^2(kh)I + A^2)^{-1} \phi'_{\text{sinh}}(kh)$$

数値計算法2

# 行列符号関数

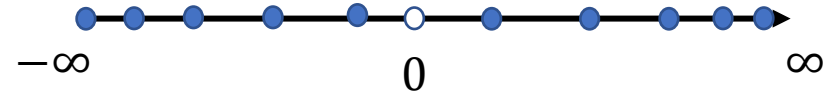
## 2つの計算法を比較する動機

DE公式での半無限区間と全無限区間での標本点分布



### 半無限区間

積分区間が原点の近くで  
標本点が非常に密になる



### 全無限区間

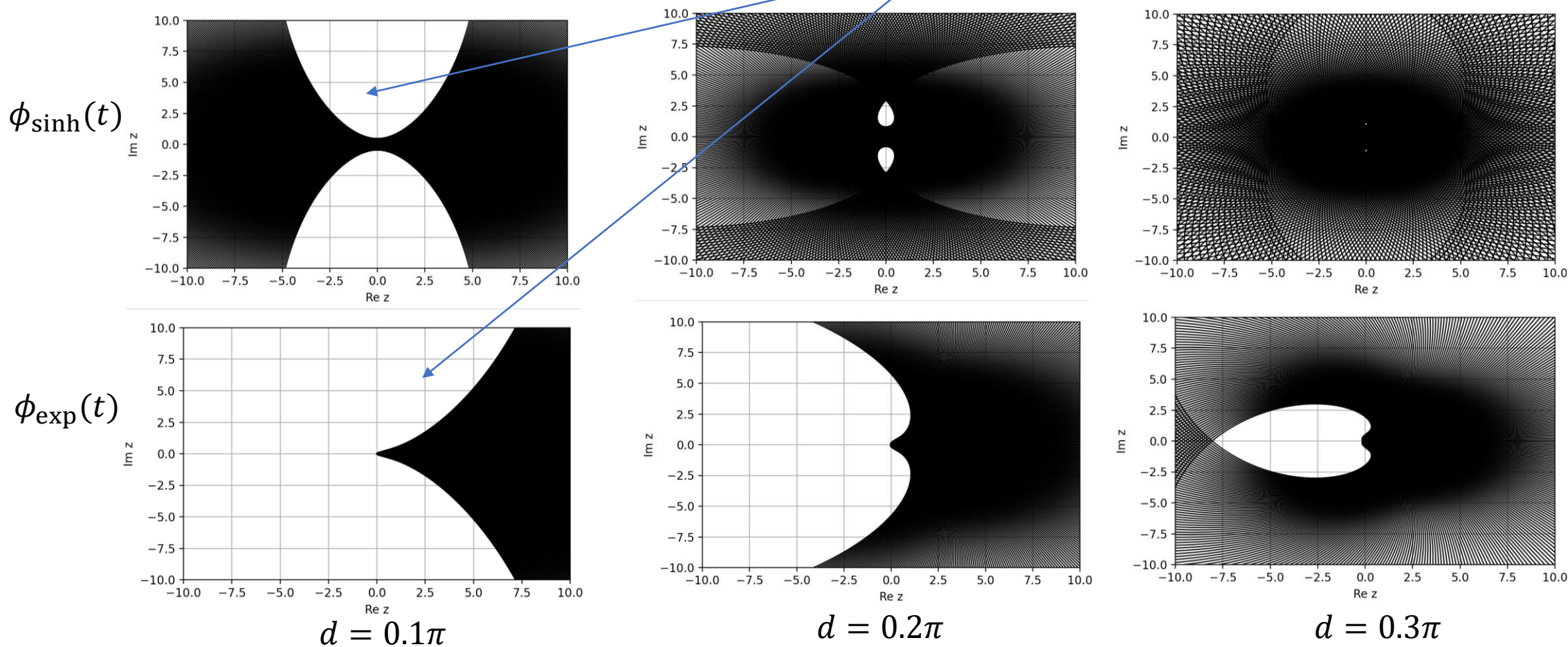
対称性を利用して積分区間を  
 $[-\infty, \infty]$ にすると、0点付近の端  
点がなくなる

全無限区間の方が端点がなくなるため密集が避けられ効率的ではないだろうか？  
(0は見かけ上の端点)

# 2つの数値計算法の比較

## 変換後の領域 $D(d)$ の比較

- 領域 $D(d)$ を変換したときの領域を調べる。(白い領域に特異点が存在してもよい)



# 2つの数値計算法の比較

- 行列符号関数の被積分関数の特異点は $A$ の固有値 $\lambda_k$ を用いて $\pm i\lambda_k$ となることを示した。そのため、変換後の領域 $D(d)$ 上に特異点を含まないようにするには $d$ がどれだけ大きく取れるかを調べた。

表1 変換式 $\phi_{\sinh}(t)$

特異点( $i\lambda_k$ )	$d$ の最大値 $\times \pi$
10000 $i$	0.049
1000 $i$	0.063
100 $i$	0.088
10 $i$	0.14
0.1 $i$	0.02
0.01 $i$	0.002
0.001 $i$	0.0002
0.0001 $i$	0.00002

表2 変換式 $\phi_{\exp}(t)$

特異点( $i\lambda_k$ )	$d$ の最大値 $\times \pi$
10000 $i$	0.053
1000 $i$	0.069
100 $i$	0.099
10 $i$	0.16
0.1 $i$	0.16
0.01 $i$	0.099
0.001 $i$	0.069
0.0001 $i$	0.053

$\phi_{\exp}(t)$ の方が収束が速くなることが予想される

# 2つの数値計算法の比較

## 数値実験

### • 実験条件

- 入力サイズ:  $100 \times 100$
- テスト行列:  $A = X\Lambda X^{-1}$  として生成
  - $X$ : ランダムな正則行列
  - $\Lambda$ : 対角成分に固有値を持つ対角行列
    - 最小固有値  $0.0001 \sim 0.1$ 、最大固有値  $10$
- 解析解: 行列符号関数の定義式  
 $\text{sign}(A) = X \text{diag}(\text{sign}(\lambda_k)) X^{-1}$  によって求める
- 誤差: 解析解との差の行列2ノルム

### • 計算法

1. 初めに刻み幅  $h$  を決める
2. 各積分点でのノルムを計算
3.  $\|f\|_2 < 1e-15$  となるまで積分点を増やしていく

### • 計算機環境

- AMD Ryzen 7 3700X 8-Core Processor (3.59GHz, 8コア)
- Python NumPyモジュール

# 2つの数値計算法の比較

- 固有値は正の実数
- 最大固有値を10に固定し，最小固有値を0.1から0.0001まで変えて計算

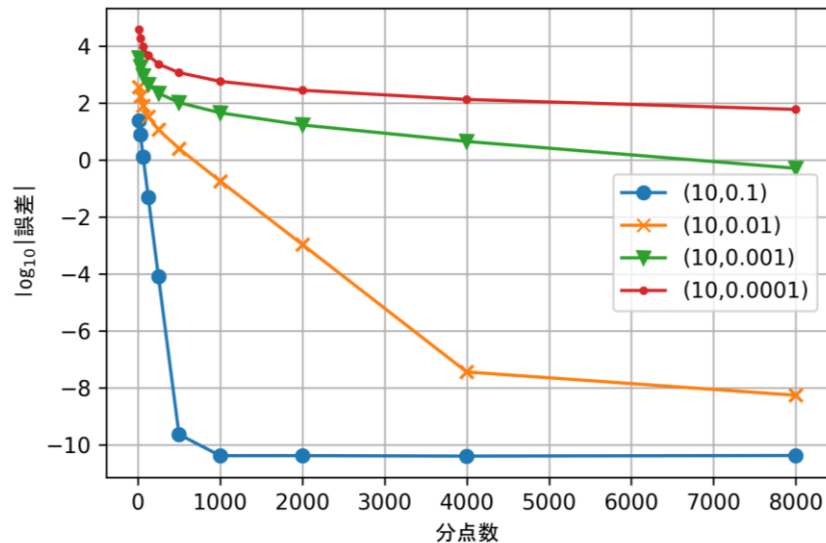


図1 変換式 $\phi_{\sinh}(t)$

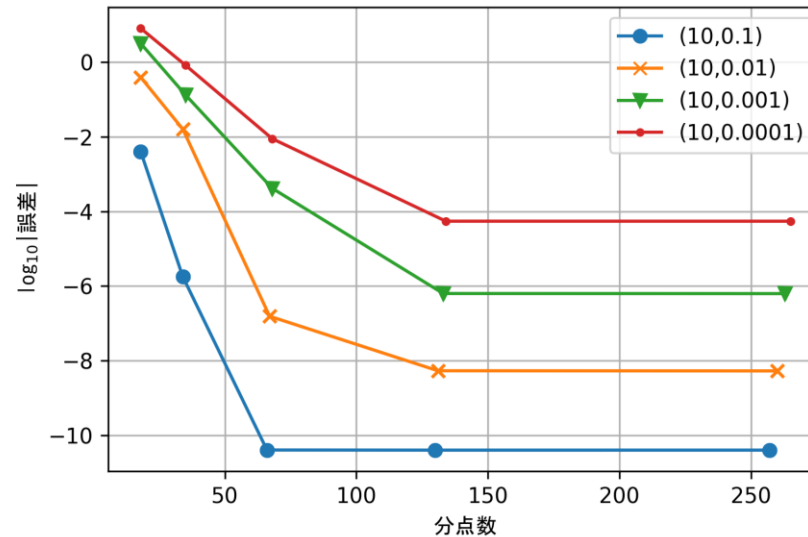


図2 変換式 $\phi_{\exp}(t)$

- 最小固有値が1より小さい場合には，変換式 $\phi_{\sinh}(t)$ での誤差の収束がかなり遅い
- 領域 $\mathcal{D}(d)$ の比較から得られる予想と一致      以降 $\phi_{\exp}(t)$ を使用

# $\text{sign}(cA) = \text{sign}(A)$ を用いた最適なスケーリング

- 行列符号関数の定義より,  $c > 0$ に対して,


$$\text{sign}(cA) = X\text{sign}(cJ)X^{-1} = X\text{diag}(\text{sign}(c\lambda_i))X^{-1} = X\text{diag}(\text{sign}(\lambda_i))X^{-1} = \text{sign}(A).$$

符号関数の定義より成り立つ

つまり,  $c > 0$ のとき  $\text{sign}(cA) = \text{sign}(A)$  が成り立つ

- これにより結果を変えずに  $A$  の固有値分布を変えることができる

- 固有値がすべて実数のとき  $|\lambda_{\min}| = 1/|\lambda_{\max}|$  となるように  $c$  を選べば, 特異点の存在してよい領域を最大限に活用でき, 効率的である



特異点( $i\lambda_k$ )	$d$ の最大値 $\times \pi$
10000 $i$	0.053
1000 $i$	0.069
100 $i$	0.099
10 $i$	0.16
0.1 $i$	0.16
0.01 $i$	0.099
0.001 $i$	0.069
0.0001 $i$	0.053

# $\text{sign}(cA) = \text{sign}(A)$ を用いた最適なスケーリング

## 固有値がわかっている場合

- 絶対値最大の固有値を $\lambda_{\max}$ , 絶対値最小の固有値を $\lambda_{\min}$ とする. このとき, 最適なスケーリングの条件は次のようになる.

$$\begin{aligned} 1/(c|\lambda_{\min}|) &= c|\lambda_{\max}| \\ \Leftrightarrow c &= \frac{1}{\sqrt{|\lambda_{\max}||\lambda_{\min}|}} \quad \cdots (*) \end{aligned}$$

## 固有値がわかっていない場合

- 固有値の絶対値の最大と最小について, 次の上界, 下界が成り立つ.

$$\begin{aligned} |\lambda_{\max}| &\leq \|A\|_{\infty} \\ |\lambda_{\min}| &\geq (\|A^{-1}\|_{\infty})^{-1} \end{aligned}$$

- 固有値最大値, 最小値を以下のように近似して(\*)に適用して $c$ を決める.

$$\lambda'_{\max} = \|A\|_{\infty}, \quad \lambda'_{\min} = (\|A^{-1}\|_{\infty})^{-1}$$



# $\text{sign}(cA) = \text{sign}(A)$ を用いた最適なスケーリング

## スケーリング前後での計算結果の比較

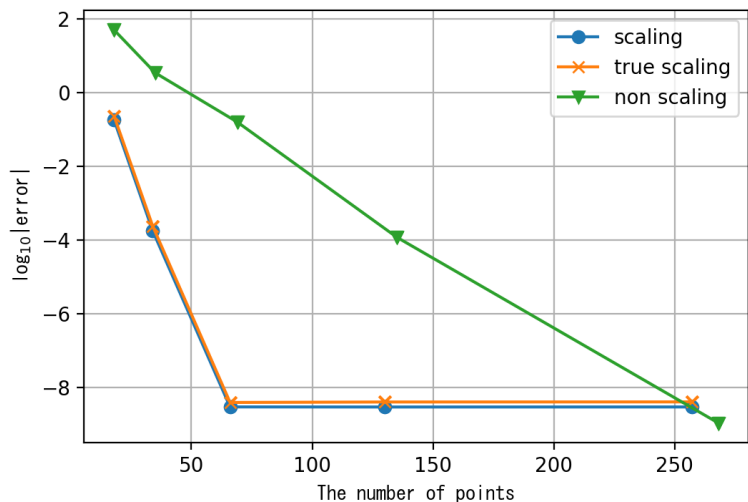


図1 最大固有値 $1e-3$ , 最小固有値 $1e-5$

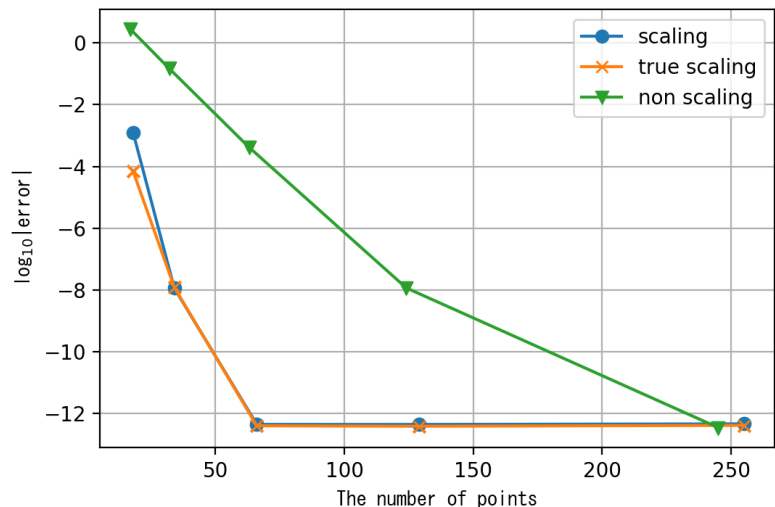
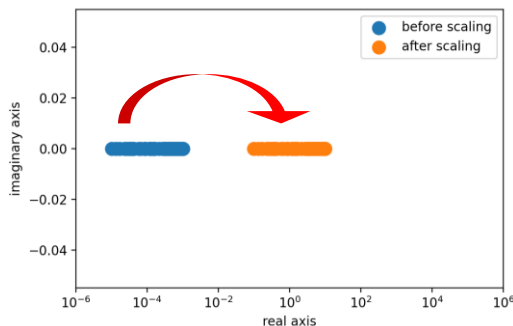
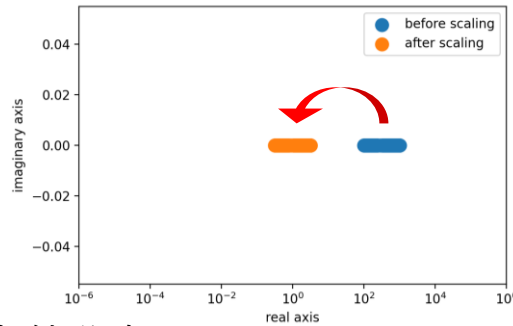


図2 最大固有値 $1e+3$ , 最小固有値 $1e+2$



凡例  
scaling: 近似固有値によるスケーリング  
true scaling: 既知の固有値によるスケーリング  
non scaling: スケーリングせずに計算

スケーリングにより収束が速くなることが確認できる

スケーリング前後の固有値分布

# $\text{sign}(cA) = \text{sign}(A)$ を用いた最適なスケーリング

## スケーリング前後での計算結果の比較 (負の固有値も含む場合)

凡例  
**scaling**: 近似固有値によるスケーリング  
**true scaling**: 既知の固有値によるスケーリング  
**non scaling**: スケーリングせずに計算

負の固有値を含む場合も  
スケーリングにより収束を  
速めることが確認できる

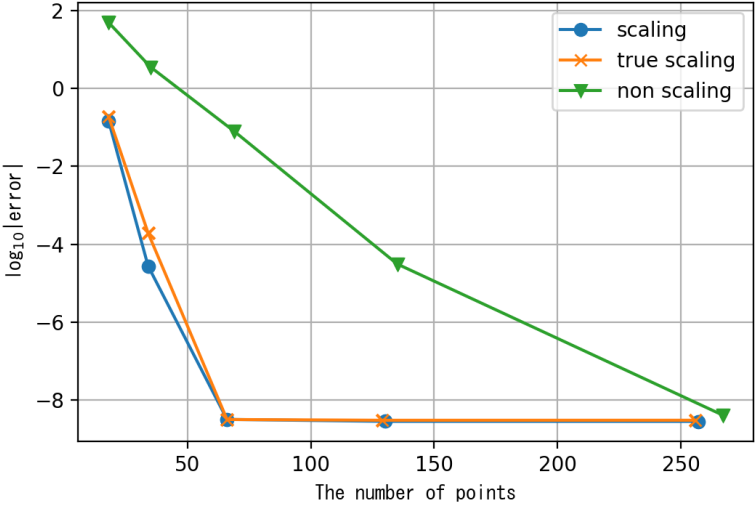


図1 絶対値最大固有値 $1e-3$ , 最小固有値 $1e-5$

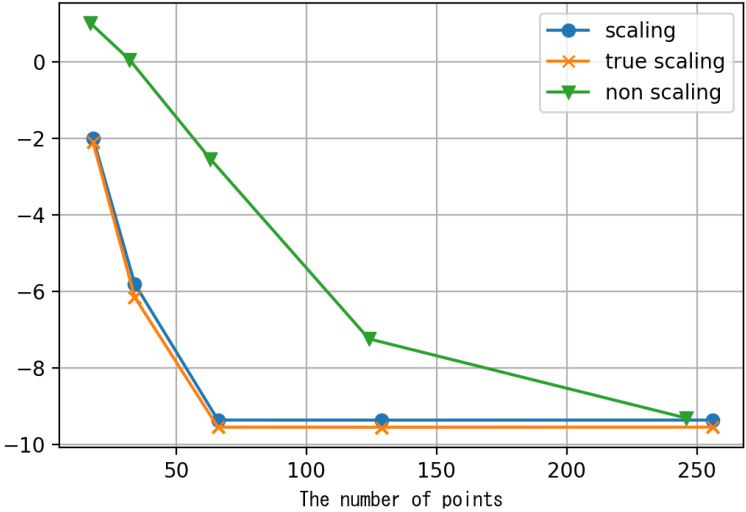
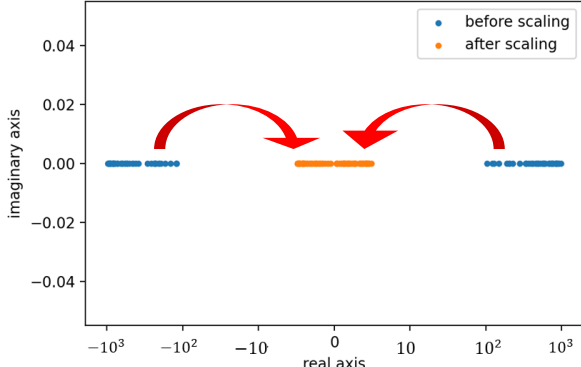
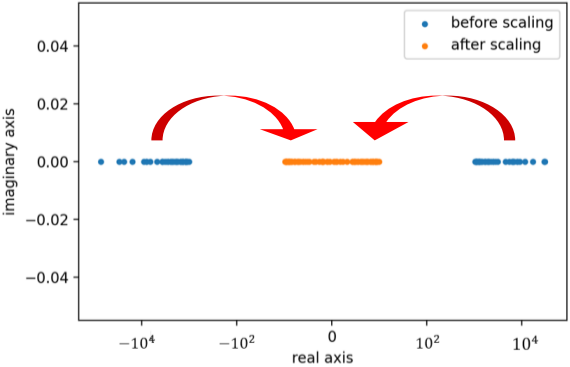


図2 絶対値最大固有値 $1e+3$ , 最小固有値 $1e+2$



スケーリング前後の固有値分布

# 行列平方根への応用

## 行列符号関数の性質

行列  $A, B \in \mathbb{C}^{n \times n}$  があるとし,  $AB$  および  $BA$  が  $\mathbb{R}^-$  上 (負の実数上) に固有値を持たないと仮定する. このとき, 以下が成り立つ.

$$\text{sign} \left( \begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & C \\ C^{-1} & 0 \end{bmatrix}$$

ここで,  $C = A(BA)^{-\frac{1}{2}}$  である.

この定理の特別な例として以下が成り立つ. [2]

$$\text{sign} \left( \begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & A^{\frac{1}{2}} \\ A^{-\frac{1}{2}} & 0 \end{bmatrix}$$

これより行列平方根と行列逆平方根の積分表示形式を導ける

# 行列平方根への応用

$A = \begin{bmatrix} 0 & B \\ I & 0 \end{bmatrix}$ とすると,  $A^2 = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}$ となることから次が導ける.

$$\text{sign}(A) = \frac{2}{\pi} A \int_0^\infty (t^2 I + A^2)^{-1} dt = \frac{2}{\pi} \begin{bmatrix} 0 & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \int_0^\infty (t^2 I + B)^{-1} dt & 0 \\ 0 & \int_0^\infty (t^2 I + B)^{-1} dt \end{bmatrix} \quad \cdots (1)$$

また,  $B^{\frac{1}{2}}$ を $B$ の主平方根(principal square root)とすると行列 $A$ は次のようにも表せる.

$$\text{sign}(A) = \text{sign} \left( \begin{bmatrix} 0 & B \\ I & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & B^{\frac{1}{2}} \\ B^{-\frac{1}{2}} & 0 \end{bmatrix} \quad \cdots (2)$$

# 行列平方根への応用

(1), (2)より次が成り立つ

$$\begin{aligned} \begin{bmatrix} 0 & B^{\frac{1}{2}} \\ B^{-\frac{1}{2}} & 0 \end{bmatrix} &= \frac{2}{\pi} \begin{bmatrix} 0 & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \int_0^\infty (t^2 I + B)^{-1} dt & 0 \\ 0 & \int_0^\infty (t^2 I + B)^{-1} dt \end{bmatrix} \\ &= \frac{2}{\pi} \begin{bmatrix} 0 & B \int_0^\infty (t^2 I + B)^{-1} dt \\ \int_0^\infty (t^2 I + B)^{-1} dt & 0 \end{bmatrix} \end{aligned}$$

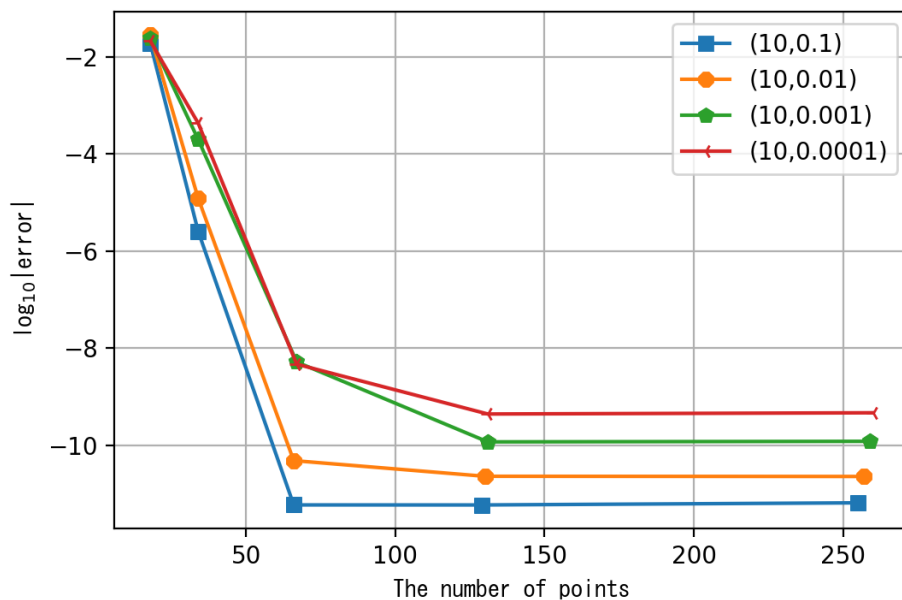
以上より, 次が成り立つことがわかる

$$B^{\frac{1}{2}} = \frac{2}{\pi} B \int_0^\infty (t^2 I + B)^{-1} dt$$

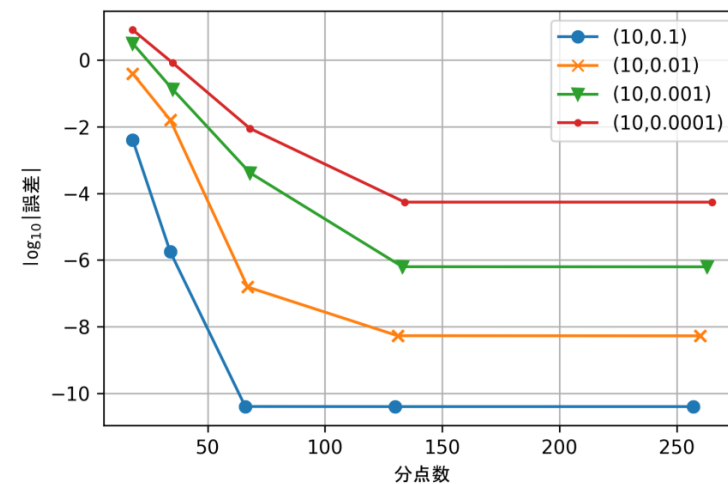
$B$ を掛けなければ, 行列逆平方根についても求められる

# 行列平方根への応用

## 計算結果



行列平方根をDE公式を用いて計算



(再掲)行列符号関数をDE公式を用いて計算

- 行列符号関数と比較したとき，被積分関数に行列の2乗がないため小さい固有値を含む場合でも誤差収束が悪くならないことがわかる

# 新しい誤差上界の導出

- 行列符号関数をDE公式で求めるときの全体の誤差上界(離散化誤差+打切り誤差)を導出する
- 導出にあたっての方針は次の通りである
  - 行列ノルムとしてフロベニウスノルムを用いる  
(2ノルムでの誤差上界も同様に導出可能)
  - 誤差行列 $E$ の各要素に対して上界を導出してから、それを用いて $\|E\|_F$ を計算するのではなく、 $\|E\|_F$ の行列ノルムとしての性質を十分に活用する
  - 行列 $A$ が $A = X\Lambda X^{-1}$ のように対角化できると仮定し、固有ベクトル行列 $X$ の条件数 $\kappa_2(X) = \|X\|_2\|X^{-1}\|_2$ を積極的に活用する

# 新しい誤差上界の導出

- 行列  $A \in \mathbb{C}^{n \times n}$  の行列符号関数をDE公式で計算したときの全体の誤差上界は次のように評価される.

$$\|E\|_F \leq 4\kappa_2(X) \left\{ \frac{1}{\cos d} \left( \frac{\|A^2\|_F}{R^2} + \frac{\sqrt{n}}{3} \right) + \|A\|_F \right\} \exp \left( -\frac{2\pi d N_+}{\log(8dN_+)} \right)$$

- 一方, [1]では一般の(対角化も不可能な場合も含む)行列に対し, 次の上界が与えられる.

$$\|E\|_F \leq \frac{n\sqrt{n}C(A, d)\|A\|_F(\sqrt{n}C'(A, d) + \|A^2\|_F)^{n-1}}{\pi R^{2n}} \exp \left( -\frac{2\pi d N}{\log(8dN)} \right)$$

- ここで導出された上界は, 従来の評価と異なり  $n$  に関する指数関数的な依存性が消え,  $C(A, d)$  や  $C'(A, d)$  のような係数が消えて, 上界の式を簡略化することができた



# おわりに

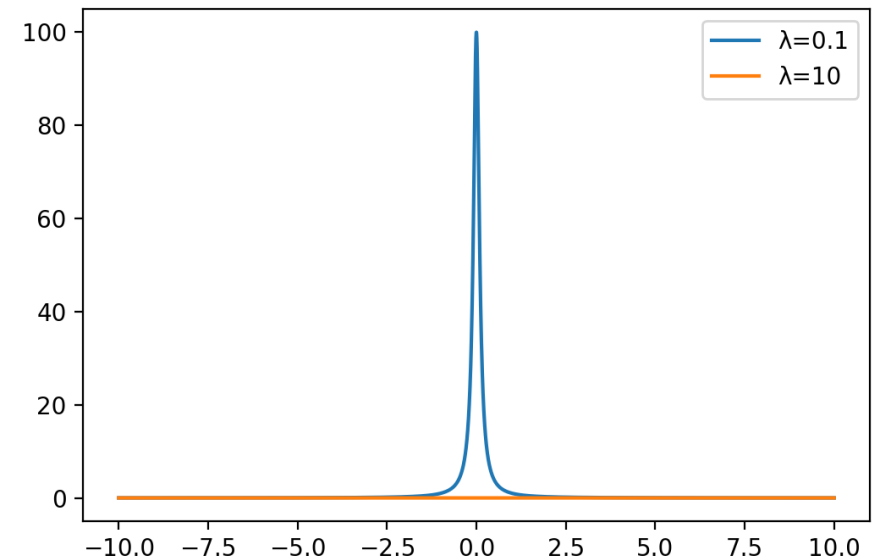
- DE公式を用いて行列符号関数を計算するとき，変換関数として $\exp\left(\frac{\pi}{2}\sinh t\right)$ を用いた方が収束が速くなることが分かった．
- 行列符号関数の性質を用いたスケーリングにより収束を速めることができた．
- 行列平方根に対する数値計算法を導出し，評価した．
- 新たな誤差上界を導出した．

# (補足) なぜ半無限区間の方が収束が速かったのか

- 行列符号関数の被積分関数がスカラーの場合は以下ようになる

$$\frac{1}{x^2 + \lambda^2}$$

- ここで、 $\lambda$ は行列のとき固有値となる定数である.
- この関数の概形は右図のようになる.
- $\lambda$ が0に近いほど0点付近での被積分関数の値は非常に大きくなる.
- つまり、0点付近での標本点分布が多くなる半無限区間積分の方が少ない分点数で近似できることが考えられる.



# (補足) 誤差上界のアプローチの違いについて

- 行列符号関数の被積分関数には逆行列が含まれる．そのため，逆行列に関する評価が必要．
  - 先行行列では一般の行列 $A$ に対して以下のような性質を用いて評価を行っている．

$$\|A^{-1}\| \leq \gamma \frac{\|A\|^{n-1}}{|\det A|}$$

- 本研究では対角化可能と仮定し， $A = X\Lambda X^{-1}$ とすることにより以下のような評価式を得る．

$$\begin{aligned}\|A^{-1}\| &= \|X\Lambda^{-1}X^{-1}\| \\ &\leq \|X\|\|X^{-1}\|\|\Lambda^{-1}\| \\ &= \kappa(X)\|\Lambda^{-1}\|\end{aligned}$$