# 射影–再スケーリング法を用いた対称錐計画問題に対する後処理アルゴリズム

加納　伸一

対称錐計画問題は，対称錐と呼ばれる閉凸集合と線形制約からなる実行可能領域上で線形の目的関数を最小化（または最大化）する問題であり，主双対内点法という効率的な解法がすでに知られている．しかし，対称錐計画（特に半正定値計画）問題に対して，数値的に安定した高精度の解を手に入れることは未だに難しい．この現状に対し，筆者の博士論文では，対称錐計画の特殊型に対する新たな多項式時間アルゴリズムとして提案された射影–再スケーリング法に注目し研究を行った．本稿では筆者の博士論文を基に，射影–再スケーリング法を用いた後処理アルゴリズムとその数値実験の結果を紹介する．

キーワード：対称錐計画問題，射影–再スケーリング法，Chubanov のアルゴリズム，後処理アルゴリズム

## 1. はじめに

対称錐計画問題とは，$d$ 次元の実ベクトル空間 $\mathbb{R}^d$ に対し，対称錐 $\mathcal{K} \subseteq \mathbb{R}^d$ とアフィン空間からなる実行可能領域上で線形の目的関数を最小化または最大化する問題である．

$$\begin{aligned} &(\mathrm{P}) \quad \inf \ \langle c, x\rangle \quad \text{s.t. } \mathcal{A}x = b,\ x \in \mathcal{K}, \\ &(\mathrm{D}) \quad \sup \ b^\top y \quad \text{s.t. } c - \mathcal{A}^* y = z,\ z \in \mathcal{K}^*. \end{aligned} \tag{1}$$

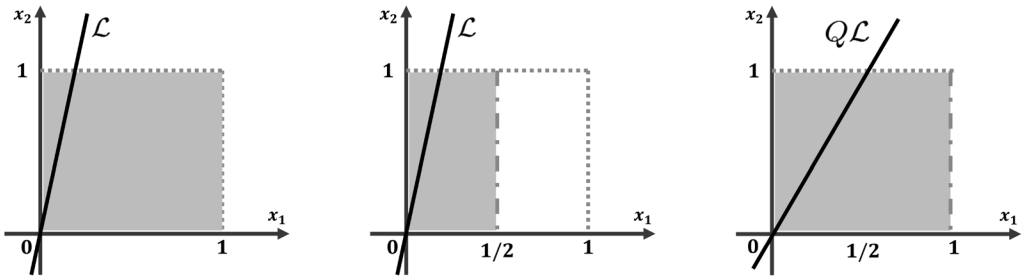
ここで，$\langle , \rangle$ は内積，$\mathcal{A}$ は $\mathbb{R}^d$ から $\mathbb{R}^m$ への線形写像，$\mathcal{A}^*$ は $\mathcal{A}$ の随伴作用素，$b \in \mathbb{R}^m$，$c \in \mathbb{R}^d$ である．また，$\mathcal{K}^*$ は $\mathcal{K}$ の双対錐であり，$\mathcal{K}^* = \{s \in \mathbb{R}^d : \langle x, s\rangle \geq 0, \forall x \in \mathcal{K}\}$ である．対称錐は自己双対性が成り立つため，$\mathcal{K} = \mathcal{K}^*$ であることに注意する．対称錐計画は線形計画，2 次錐計画，半正定値計画を含み，主双対内点法で効率よく求解できることが知られている．内点法は，現在点が最適解に十分近づくまで更新方向の計算と現在点の更新を繰り返し，最適解に収束する（実行可能）内点列を生成する手法であるが，現在点が最適解（つまり，錐の境界）に近づくと更新方向の計算が数値的に不安定になるため，高精度の解を求めることが難しいという実装面での課題がある．対称錐計画（特に半正定値計画）問題に対して高精度の解を求めることを目的とした研究には，文献 [1, 2] などがある．文献 [1] では，多倍長精度演算を用いた内点法の開発が行われており，文献 [2] では，（浮動小数点演算ではなく）厳密に正確な演算 (exact arithmetic) で実装可能な，(D) の実行可能解を求めるアルゴリズムが開発されいている．このように，文献 [1, 2] では浮動小数点演算で起こる誤差に直接的に対処した研究が行われている．一方，筆者は Kanoh and Yoshise [3] において，対称錐計画問題の特殊型に対するアルゴリズムである射影–再スケーリング法に注目し，(P) と (D) の高精度な解を求める後処理アルゴリズムの開発という研究を行った．筆者が射影–再スケーリング法に注目した理由は，Kanoh and Yoshise [4] の計算機実験において，射影–再スケーリング法が商用ソルバー [5] よりも精度の良い解を安定して返したことを確認したためである．

本稿では，筆者の博士論文，および，文献 [3, 4] を基に，対称錐上の射影–再スケーリング法と，(P) と (D) に対する射影–再スケーリング法を用いた後処理アルゴリズムを解説する．また，後処理アルゴリズムの性能を確かめた文献 [3] の計算機実験の結果を一部紹介する．

## 2. 射影–再スケーリング法

Chubanov [6] は，同次線形計画問題の内点解を求める多項式時間アルゴリズムを提案した．このアルゴリズムは，線形部分空間への射影とスケーリングで構成されており，文献 [4, 7, 8] により対称錐計画へ拡張された．そのアルゴリズムの構造から，本稿では，これらのアルゴリズムを“射影–再スケーリング法”と呼ぶこととする．ここで，二つの集合 $C_1, C_2 \subseteq \mathbb{R}^d$ に対し，実行可能性問題 $\mathrm{FP}(C_1, C_2)$ を式 (2) のように定義すると，文献 [4, 7, 8] の射影–再スケーリング法が扱う問題は $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ となる．ただし，$\mathcal{L} \subseteq \mathbb{R}^d$ は

かのう　しんいち
（株）日立製作所
s2130104@u.tsukuba.ac.jp




**(a)** 標準化した問題の作成　　**(b)** カットの生成　　**(c)** スケーリングによる変換

**図 1**　$\mathcal{K} = \mathbb{R}^2_+$ の場合の Kanoh and Yoshise [4] の手法の流れ

線形部分空間であり，int$\mathcal{K}$ は対称錐 $\mathcal{K}$ の内部を表す．

$$\mathrm{FP}(C_1, C_2): \quad \text{find} \quad x \in C_1 \cap C_2. \tag{2}$$

文献 [4, 7, 8] では，アルゴリズムの収束の解析に用いる評価指標の定め方がそれぞれで異なるため，三種類の異なる射影–再スケーリング法が提案されているが，アルゴリズムの大まかな流れはどれも同じである．

### 2.1　射影–再スケーリング法の流れ

射影–再スケーリング法は，ステップ 1. $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の実行可能性を判断する，ステップ 2. $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ と等価な問題をスケーリングにより作成する，という二つのステップを繰り返す．ステップ 1 では，$\mathcal{L}$ と $\mathcal{K}$ を入力とする内部関数を使用する．この内部関数は $\mathcal{L}$ への射影を用いて，

(a) $\mathcal{L} \cap \mathrm{int}\mathcal{K}$ の点，

(b) $\mathcal{L} \cap \mathrm{int}\mathcal{K} = \emptyset$ を示す点（つまり，$\mathcal{L}^\perp \cap \mathcal{K}$ を満たす非ゼロの点．ただし，$\mathcal{L}^\perp$ は $\mathcal{L}$ の直交補空間）．または，

(c) $\mathcal{L} \cap \mathrm{int}\mathcal{K}$ の存在範囲に関する情報

のいずれかを返す．(a) または (b) の出力が得られた場合，アルゴリズムは終了する．(c) の出力が得られた場合は，ステップ 2 でその情報を基に一次変換 $Q : \mathbb{R}^d \to \mathbb{R}^d$ を構築する．具体的には，スケーリング後の線形部分空間 $Q\mathcal{L}$ が $\mathcal{L}$ よりも錐の中心に近づくような $Q$ を作成する．ただし，$Q$ は $Q\mathcal{K} = \mathcal{K}$ を満たすように構築されることに注意する．その後，$\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ と等価な問題として $\mathrm{FP}(Q\mathcal{L}, \mathrm{int}\mathcal{K})$ を作成し，再びステップ 1 にて $Q\mathcal{L}$ と $\mathcal{K}$ を入力として内部関数を実行，という操作を繰り返す．

文献 [4, 7, 8] の手法は，どれも上述の流れでアルゴリズムが進むが，内部関数を用いて $\mathcal{L} \cap \mathrm{int}\mathcal{K}$ の存在範囲に関してどのような情報を手に入れるのかという部分は異なる．たとえば Kanoh and Yoshise [4] では，式 (3) のように $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の実行可能領域を標準化した問題 $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ に対するカットを見つけることで $Q$ を構築している．ただし $\|x\|_\infty \le 1$ は，$x$ のすべての固有値に対し，その絶対値が 1 以下であることを意味する．（ここでの固有値は，ユークリッド的ジョルダン代数における固有値である．ユークリッド的ジョルダン代数に関しては，文献 [9] が詳しい．）

$$\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K}): \quad \begin{array}{ll} \text{find} & x \in \mathcal{L} \cap \mathrm{int}\mathcal{K} \\ \text{s.t.} & \|x\|_\infty \le 1 \end{array}. \tag{3}$$

$\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ に対するカットから目的の一次変換 $Q$ がどのように作られるかを本稿では詳しく解説しないが，作成方法の基本的な考え方は図 1 より理解することができる．図 1(a) は，錐が 2 次の非負象限 $\mathcal{K} = \mathbb{R}^2_+$ の場合の領域 $F := \{x \in \mathbb{R}^2_+ : \|x\|_\infty \le 1\}$ を表している．仮に，図 1(b) のように $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の任意の実行可能解 $x = (x_1, x_2)^\top$ に対して，$x_1 \le 1/2$ というカットが得られたとする．このとき，限定された解の探索範囲 $\{x \in \mathbb{R}^2_+ : \|x\|_\infty \le 1, x_1 \le 1/2\}$ が領域 $F := \{x \in \mathbb{R}^2_+ : \|x\|_\infty \le 1\}$ にまんべんなく広がるようなスケーリングを行うために，一次変換 $Q$ を $Q = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ のように構築し，問題のスケーリングを行う．すると，図 1(c) のようにスケーリング後の線形部分空間 $Q\mathcal{L}$ がより錐の中心に近づくことがわかる．

このようにして，Kanoh and Yoshise [4] ではカットから一次変換 $Q$ を作成するので，アルゴリズムが進むにつれて $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の任意の実行可能解の最小固有値の上界値を狭めていくことができる．（たとえば図 1 において，$\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の任意の実行可能解の最小固有値は，図 1(a) の時点では 1 以下であることしかわからないが，カットが得られた時点では 1/2 以下に狭められることがわかる．）すなわち，彼らの手法は最小固有値が $\varepsilon$ より大きい $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の実行可能解が存在するかどうかを判断することがで

きる．ただし，$\varepsilon$ はユーザー指定の正の実数である．Kanoh and Yoshise [4] では，最小固有値が $\varepsilon$ 以上の $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の実行可能解を $\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の“$\varepsilon$ 許容解”と呼び，対象となる問題に $\varepsilon$ 許容解が存在しないと判明した場合にアルゴリズムを終了するという終了条件を設けることで，多項式時間アルゴリズムとしての射影–再スケーリング法を提案している．

このようなアルゴリズムの特徴は Lourenço et al. [7] でも見られる．Lourenço et al. [7] では，式 (3) とは異なる形で $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ を標準化した問題を作成し，その標準化問題に対するカットを内部関数で求め，得られたカットを基に一次変換 $Q$ を作成しスケーリングを行うというアルゴリズムである．そのため Lourenço et al. [7] の手法も，対象となる標準化問題の任意の実行可能解の最小固有値の上界値を用いた終了条件を設けることで多項式時間アルゴリズムとして提案されている．

一方，Peña and Soheili [8] では $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ を標準化した問題は考えず，内部関数を用いて $\mathcal{L} \cap \mathcal{K}$ の条件数を改善するような一次変換 $Q$ を求め，スケーリングを行うアルゴリズムとなっている．彼らの手法は，$\mathcal{L} \cap \mathrm{int}\mathcal{K} = \emptyset$ の場合にアルゴリズムの終了を保証できず，多項式時間アルゴリズムではないことに注意する．

### 2.2 射影–再スケーリング法に注目した理由

Kanoh and Yoshise [4] では，彼らの射影–再スケーリング法が文献 [7, 8] のものより計算コストが改善していることを理論的に証明し，その性能を確かめる数値実験が行われている．その実験では，式 (3) の実行可能解が錐の境界付近にしか存在しないような $\mathcal{L}$ を作成し，そのような問題に対し射影–再スケーリング法を用いると，商用ソルバー [5] よりも実行時間はかかるものの，線形制約の残差が小さい解が安定して得られやすかったという結果が報告されている．この結果は，対称錐計画問題の高精度な解を求めることにおいて，射影-再スケーリング法が役立つ可能性を示すものである．

## 3. Kanoh and Yoshise [3] の後処理アルゴリズム

Kanoh and Yoshise [4] の数値実験結果より，射影–再スケーリング法を用いることでソルバーよりも高精度の解の入手が期待できるが，射影–再スケーリング法は対称錐計画の特殊型に対するアルゴリズムである．そこで筆者は文献 [3] において，射影–再スケーリング法を用いた式 (1) を解くためのアルゴリズムを提案し，提案手法をソルバー（内点法）の後処理として使用した場合の数値的性能を計算機実験によって確認した．提案したアルゴリズムは，それ自体でも問題を解くことができるが，実行時間の観点，および，近似最適（内点）解を用いることで効率的に問題を解くことができるという特徴をもつことから，筆者は内点法の後処理として用いることを想定してアルゴリズムを構築した．

3.1 節と 3.2 節では，射影–再スケーリング法を用いて (P) と (D) の解を求めるための基盤となる理論を説明する．3.3 節では，Kanoh and Yoshise [3] のアルゴリズムの基本的な流れを説明する．以降では，(P) と (D) の最適値をそれぞれ $\theta_p \in \mathbb{R} \cup \{\pm\infty\}$，$\theta_d \in \mathbb{R} \cup \{\pm\infty\}$ とする．もし (P) が実行不可能ならば $\theta_p$ は $+\infty$ を取り，(D) が実行不可能ならば $\theta_d$ は $-\infty$ を取るとする．

### 3.1 基盤となる理論 1

実数 $\theta \in \mathbb{R}$ に対し，線形写像 $\mathcal{A}(\theta)$ を式 (4) のように定義する．$\mathcal{A}(\theta)$ の零空間を $\ker\mathcal{A}(\theta)$ とする．$\ker\mathcal{A}(\theta)$ の直交補空間は，$\mathcal{A}(\theta)$ の随伴作用素 $\mathcal{A}(\theta)^*$ の像空間 $\mathrm{range}\mathcal{A}(\theta)^*$ で与えられる．また，対称錐 $\bar{\mathcal{K}}$ を $\bar{\mathcal{K}} = \mathcal{K} \times \mathbb{R}^2_+$ と定義する．まずは，問題 $\mathrm{FP}(\ker\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ に対し，その実行可能解から目的関数値が $\theta$ 未満となる (P) の内点解が得られることを確認する．

$$\mathcal{A}(\theta) = \begin{pmatrix} \mathcal{A} & -b & \mathbf{0} \\ c^\top & -\theta & 1 \end{pmatrix}. \tag{4}$$

**命題 3.1**（文献 [3] (Proposition 3.1)）**.** FP ($\ker\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}}$) が実行可能であるとし，$(x, \tau, \rho)$ をその実行可能解とする．このとき，$\frac{1}{\tau}x$ は (P) の内点解であり $\langle c, \frac{1}{\tau}x\rangle < \theta$ を満たす．

次は，$\mathrm{FP}(\ker\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ が実行不可能の場合を考える．$\ker\mathcal{A}(\theta) \cap \mathrm{int}\bar{\mathcal{K}} = \emptyset$ は，$(z, \omega, \kappa) \in \mathrm{range}\mathcal{A}(\theta)^* \cap \bar{\mathcal{K}}$ を満たす非ゼロのベクトル $(z, \omega, \kappa)$ が存在することと必要十分の関係となる．この $(z, \omega, \kappa)$ から，(P) の実行可能性状態を判別するベクトル，または，目的関数値が $\theta$ 以上の (D) の実行可能解が得られることが，命題 3.2 より確認できる．

**命題 3.2**（文献 [3] (Proposition 3.4)）**.** $(z, \omega, \kappa) \in \mathrm{range}\mathcal{A}(\theta)^* \cap \bar{\mathcal{K}}$ を満たす非ゼロのベクトル $(z, \omega, \kappa)$ が存在するとき，$z = \mathcal{A}^*y + \kappa c, \omega = -b^\top y - \kappa\theta$ を満たす $y \in \mathbb{R}^m$ が存在し，$(y, \kappa) \in \mathbb{R}^{m+1}$ に対して以下のいずれかが成り立つ．

1. $\kappa > 0$．このとき，$-\frac{1}{\kappa}y$ は $b^\top(-\frac{1}{\kappa}y) \geq \theta$ を満たす (D) の実行可能解となる．

2. $\kappa = 0$ かつ $\omega > 0$. このとき，$-y$ は (P) に対する実行不可能証明となる．($-\mathcal{A}^*(-y) \in \mathcal{K}$ と $b^\top(-y) > 0$ が成り立つ．)
3. $\kappa = \omega = 0$. このとき，$-y$ は (P) に対する内点解の非存在証明となる．($-\mathcal{A}^*(-y) \in \mathcal{K} \setminus \{0\}$ と $b^\top(-y) = 0$ が成り立つ．)

最後に，FP($\ker\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}}$) の実行可能性に関して，命題 3.3 を導入する．命題 3.3 は，$\ker\mathcal{A}(\theta) \cap \mathrm{int}\bar{\mathcal{K}} \neq \emptyset$ となるための必要十分条件を，$\mathrm{range}\mathcal{A}(\theta)^* \cap \bar{\mathcal{K}} \setminus \{0\} = \emptyset$ とは異なる形でまとめたものである．

**命題 3.3**（文献 [3] (Proposition 3.2)）. $\ker\mathcal{A}(\theta) \cap \mathrm{int}\bar{\mathcal{K}} \neq \emptyset$ となるための必要十分条件は，(P) に内点解が存在し，かつ，$\theta_p < \theta$ となることである．

命題 3.1, 3.2, 3.3 を用いると，(P) を解くアルゴリズムを以下のように構築できる．ただし，$\theta_{acc}$ はユーザー指定の正の実数である．

Step 1: **初期化：** $k \leftarrow 0$, $LB \leftarrow -\infty$, $UB \leftarrow \infty$
Step 2: **入力値の決定：** $\theta^k \in (LB, UB)$ を定める．
Step 3: **求解：** FP($\ker\mathcal{A}(\theta^k), \mathrm{int}\bar{\mathcal{K}}$) を解く．
Step 4: **出力に応じた処理 1:** $(x, \tau, \rho) \in \ker\mathcal{A}(\theta^k) \cap \mathrm{int}\bar{\mathcal{K}}$ が得られたとき，
- $x_{tmp} \leftarrow \frac{1}{\tau}x$, $UB \leftarrow \theta^k$

Step 5: **出力に応じた処理 2:** $(z, \omega, \kappa) \in \mathrm{range}\mathcal{A}(\theta^k)^* \cap \bar{\mathcal{K}} \setminus \{0\}$ が得られたとき，
- $z = \mathcal{A}^*y + \kappa c$, $\omega = -b^\top y - \kappa\theta^k$ を満たす $y \in \mathbb{R}^m$ を計算
- $\kappa > 0$ ならば，$LB \leftarrow \theta^k$
- $\kappa = 0$ ならば，$-y$ を返して終了．

Step 6: **終了条件の確認：** $UB - LB > \theta_{acc}$ ならば $k \leftarrow k+1$ として Step 2 へ，そうでない場合は $x_{tmp}$ を返して終了．

$\theta_{acc}$ が十分小さい値で，かつ，(P) が内点解をもつ限り，返される $x_{tmp}$ は (P) の近似最適解となることが命題 3.1, 3.3 よりわかる．(P) に内点解が存在しない場合は，内点解の非存在性を示す，または，実行不可能性を示すベクトルが Step 5 から得られることが，命題 3.2, 3.3 より保証できる．また，アルゴリズム実行前に (P) の近似最適値（近似最適解）がわかっていれば，入力値 $\theta$ の決定を効率的に行うことができる．

### 3.2 基盤となる理論 2

FP($\mathrm{range}\mathcal{A}(\theta)^*, \mathrm{int}\bar{\mathcal{K}}$) に対して命題 3.4, 3.5, 3.6 が成り立つことを，それぞれ 3.1 節の命題 3.1, 3.2, 3.3 と同様にして示すことができる．

**命題 3.4**（文献 [3] (Proposition 3.6)）. FP ($\mathrm{range}\mathcal{A}(\theta)^*, \mathrm{int}\bar{\mathcal{K}}$) が実行可能であり，$(z, \omega, \kappa)$ をその実行可能解とする．このとき，$z = \mathcal{A}^*y + \kappa c$, $\omega = -b^\top y - \kappa\theta$ を満たす $y \in \mathbb{R}^m$ が存在し，$-\frac{1}{\kappa}y$ は $b^\top(\frac{-1}{\kappa}y) > \theta$ を満たす (D) の内点解となる．

**命題 3.5**（文献 [3] (Proposition 3.9)）. $(x, \tau, \rho) \in \ker\mathcal{A}(\theta) \cap \bar{\mathcal{K}}$ を満たす非ゼロのベクトル $(x, \tau, \rho)$ が存在するとき，以下のいずれかが成り立つ．

1. $\tau > 0$. このとき，$\frac{1}{\tau}x$ は $\langle c, \frac{1}{\tau}x\rangle \leq \theta$ を満たす (P) の実行可能解となる．
2. $\tau = 0$ かつ $\rho > 0$. このとき，$x$ は (D) に対する実行不可能証明となる．($x \in \mathcal{K}$, $\mathcal{A}x = 0$, $\langle c, x\rangle < 0$ が成り立つ．)
3. $\tau = \rho = 0$. このとき，$x$ が (D) に対する内点解の非存在証明となる．($x \in \mathcal{K} \setminus \{0\}$, $\mathcal{A}x = 0$, $\langle c, x\rangle = 0$ が成り立つ．)

**命題 3.6**（文献 [3] (Proposition 3.7)）. $\mathrm{range}\mathcal{A}(\theta)^* \cap \mathrm{int}\bar{\mathcal{K}} \neq \emptyset$ となるための必要十分条件は，(D) に内点解が存在し，かつ，$\theta_d > \theta$ となることである．

命題 3.4, 3.5, 3.6 を用いると，(D) を解くアルゴリズムを前節と同様に構築できる．ただし，$\theta_{acc}$ はユーザー指定の正の実数である．

Step 1: **初期化：** $k \leftarrow 0$, $LB \leftarrow -\infty$, $UB \leftarrow \infty$
Step 2: **入力値の決定：** $\theta^k \in (LB, UB)$ を定める．
Step 3: **求解：** FP($\mathrm{range}\mathcal{A}(\theta^k)^*, \mathrm{int}\bar{\mathcal{K}}$) を解く．
Step 4: **出力に応じた処理 1:** $(z, \omega, \kappa) \in \mathrm{range}\mathcal{A}(\theta^k)^* \cap \mathrm{int}\bar{\mathcal{K}}$ が得られたとき，
- $z = \mathcal{A}^*y + \kappa c$, $\omega = -b^\top y - \kappa\theta^k$ を満たす $y \in \mathbb{R}^m$ を計算
- $y_{tmp} \leftarrow \frac{-1}{\kappa}y$, $LB \leftarrow \theta^k$

Step 5: **出力に応じた処理 2:** $(x, \tau, \rho) \in \ker\mathcal{A}(\theta^k) \cap \bar{\mathcal{K}} \setminus \{0\}$ が得られたとき，
- $\tau > 0$ ならば，$UB \leftarrow \theta^k$
- $\tau = 0$ ならば，$x$ を返して終了．

Step 6: **終了条件の確認：** $UB - LB > \theta_{acc}$ ならば $k \leftarrow k+1$ として Step 2 へ，そうでない場合は $(y_{tmp}, c - \mathcal{A}^*y_{tmp})$ を返して終了．

### 3.3 Kanoh and Yoshise [3] の手法の流れ

Kanoh and Yoshise [3] では，3.1 節と 3.2 節のアル

ゴリズムの Step 3 において Kanoh and Yoshise [4] の射影–再スケーリング法を用いることで，射影–再スケーリング法を用いた (P) と (D) を解くアルゴリズムを実装している．ここで，Kanoh and Yoshise [4] の射影–再スケーリング法は入力された問題 $\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ に対して，$\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の解・$\mathrm{FP}(\mathcal{L}, \mathrm{int}\mathcal{K})$ の実行不可能証明という出力以外に，$\mathrm{FP}_{S_\infty}(\mathcal{L}, \mathrm{int}\mathcal{K})$ に $\varepsilon$ 許容解が存在しない証明という出力を返す場合があることに注意する．つまり，3.1 節と 3.2 節のアルゴリズムを射影–再スケーリング法で実装する場合，Step 4 や Step 5 に該当する出力が毎反復で必ず得られるとは限らないということである．そのため Kanoh and Yoshise [3] では，3.1 節のアルゴリズムに対し，“出力に応じた処理 3”として

- $\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta^k), \mathrm{int}\bar{\mathcal{K}})$ に $\varepsilon$ 許容解が存在しないと判断されたとき，$LB \leftarrow \theta^k$

という操作を加えている．このように $LB$ を更新するのは，$\theta < \theta^k$ を満たす任意の $\theta$ を用いて作られる問題 $\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ に対して，射影–再スケーリング法は同様の終了条件で，つまり入力された問題に $\varepsilon$ 許容解が存在しない証明を返して終了する可能性が非常に高く，そのような $\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ の実行可能解を射影–再スケーリング法で求めることは難しいと考えられるためである．このことは，3.1 節のアルゴリズムの実行前に (P) の近似最適値が得られているという前提の下，以下の補題より確認することができる．

**補題 1.** ある正の実数 $\varepsilon \in \mathbb{R}$ と，ある実数 $\bar{\theta} \in \mathbb{R}$ に対し，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\bar{\theta}), \mathrm{int}\bar{\mathcal{K}})$ は $\varepsilon$ 許容解をもたないとする．このとき，$\bar{\theta} - \theta_p \leq 1$ ならば，$\theta \leq \bar{\theta}$ を満たす任意の $\theta \in \mathbb{R}$ に対し，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ は $\varepsilon$ 許容解をもたない．

**証明.** 命題 3.3 より，(P) に内点解が存在しない場合，任意の $\theta \in \mathbb{R}$ に対し，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ は実行不可能となる．同様に命題 3.3 より，(P) に内点解が存在し，かつ，$\theta \leq \theta_p$ が成り立つとき，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ は実行不可能となる．よって，(P) に内点解が存在し，かつ，$\theta_p < \theta \leq \bar{\theta}$ を満たす任意の $\theta \in \mathbb{R}$ に対して，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ が $\varepsilon$ 許容解をもたないことを示せばよい．

ここで，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ に $\varepsilon$ 許容解 $(x, \tau, \rho)$ が存在すると仮定する．つまり，$(x, \tau, \rho)$ は

$$Ax - \tau b = 0, \tag{5}$$
$$\langle c, x\rangle - \tau\theta + \rho = 0 \tag{6}$$
$$\min\{\lambda_{\min}(x), \tau, \rho\} \geq \varepsilon, \tag{7}$$
$$\max\{\lambda_{\max}(x), \tau, \rho\} \leq 1 \tag{8}$$

を満たす．ただし，$\lambda_{\min}(x)$ と $\lambda_{\max}(x)$ は，それぞれ $x$ の最小固有値と最大固有値を表す．ここで，$\tau > 0$ と $\rho > 0$ であることに注意すると，$\frac{1}{\tau}x$ は (P) の内点解であり，$\theta_p < \langle c, \frac{1}{\tau}x\rangle < \theta < \bar{\theta}$ が成り立つことがわかる．よって，

$$\bar{\theta} - \langle c, \frac{1}{\tau}x\rangle < \bar{\theta} - \theta_p \tag{9}$$

を得る．また，$\alpha = \bar{\theta} - \theta$ として，$\theta = \bar{\theta} - \alpha$ を式 (6) に代入すると，式 (10) を得る．

$$\langle c, x\rangle - \tau\bar{\theta} + \tau\alpha + \rho = 0. \tag{10}$$

前提条件より，$\bar{\theta} - \theta_p \leq 1$ が成り立つので，

$$\begin{aligned}\tau\alpha + \rho &= \tau\bar{\theta} - \langle c, x\rangle \quad (\text{by (10)})\\ &< \tau(\bar{\theta} - \theta_p) \quad (\text{by (9)})\\ &\leq \tau \leq 1\end{aligned}$$

となり，$\tau\alpha + \rho < 1$ が成り立つ．また，式 (7) より $\rho \geq \varepsilon$ が成り立つので $\tau\alpha + \rho > \varepsilon$ となる．すると，$(x, \tau, \tau\alpha + \rho)$ は $\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\bar{\theta}), \mathrm{int}\bar{\mathcal{K}})$ の $\varepsilon$ 許容解となり，前提条件に矛盾する．よって，$\mathrm{FP}_{S_\infty}(\mathrm{ker}\mathcal{A}(\theta), \mathrm{int}\bar{\mathcal{K}})$ は $\varepsilon$ 許容解をもたない． □

同様の理由から，3.2 節のアルゴリズムに対しても，“出力に応じた処理 3”として

- $\mathrm{FP}_{S_\infty}(\mathrm{range}\mathcal{A}(\theta^k)^*, \mathrm{int}\bar{\mathcal{K}})$ に $\varepsilon$ 許容解が存在しないと判断されたとき，$UB \leftarrow \theta^k$

という操作が加えられている．

本稿では，紙面の都合上，Kanoh and Yoshise [3] の手法をこれ以上細かく解説することは難しいが，文献 [3] で実際に提案されているアルゴリズムは，入力値 $\theta$ の決定以外でも (P) や (D) の近似最適内点解を用いてアルゴリズムの効率化を図る方法や，$k$ 反復目までの情報を用いて $k+1$ 反復目に実行する射影–再スケーリング法の実行時間を短縮する方法など，いくつかの実装上の工夫が施されたものとなっている．

## 4. Kanoh and Yoshise [3] の数値実験

### 4.1 実験の概要

Kanoh and Yoshise [3] の実験では，以下の三つの方法で得られる解の精度を比較している．

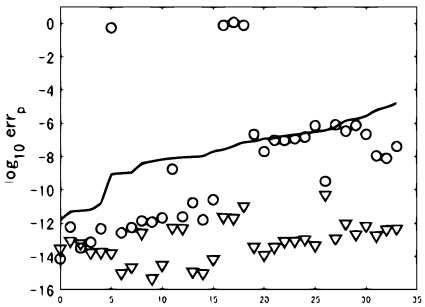



**図 2**　評価指標 1 に基づく解の精度の比較

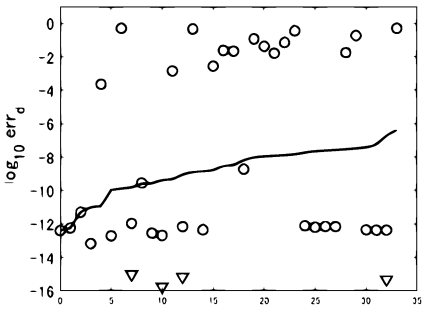



**図 3**　評価指標 2 に基づく解の精度の比較

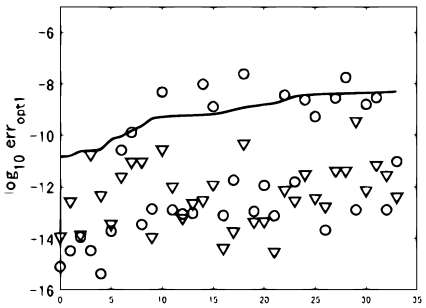



**図 4**　評価指標 3 に基づく解の精度の比較

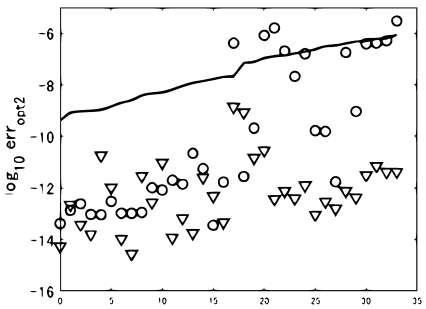



**図 5**　評価指標 4 に基づく解の精度の比較

1. $(x_{\rm def}, y_{\rm def}, z_{\rm def})$：通常設定のソルバーが返す解
2. $(x_{\rm pro}, y_{\rm pro}, z_{\rm pro})$: $(x_{\rm def}, y_{\rm def}, z_{\rm def})$ を入力として実行した Kanoh and Yoshise [3] のアルゴリズムが返す解
3. $(x_{\rm tight}, y_{\rm tight}, z_{\rm tight})$: “厳しい終了条件” に設定したソルバーが返す解

ソルバーの通常設定では，求める主双対解の線形制約の残差，および，双対ギャップに対する許容誤差値が 1e-8 程度であることが多いが，文献 [3] では許容誤差値を 1e-12 に設定した場合を “厳しい終了条件” としている．本稿では，文献 [3] の実験結果の中から，SDPLIB [10] と呼ばれるベンチマーク問題集に含まれる (P) と (D) が共に内点解をもつ問題に対して，ソルバーに Mosek を用いた場合の実験結果を載せる．計算環境は，Intel (R) Core (TM) i9-10980XE CPU @ 3.00GHz (RAM 128GB) であり，射影–再スケーリング法，および，後処理アルゴリズムの精度保証パラメータはどちらも，$\varepsilon = \theta_{acc}$ =1e-12 である．

### 4.2　実験結果

図 2–5 はそれぞれ，三種類の解に対して表 1 にある四つの評価指標の値をまとめたものである．評価指標 1 から評価指標 4 は，文献 [11] にて用いられた (P) と (D) の最適解としての精度を測る六つの指標の内の一部である．（文献 [11] では，表 1 にある四つの指標に加えて，$x$ と $z$ の錐制約の違反度合いを測定する指標も使用されている．しかし，文献 [3] の計算機実験では，錐制約を大きく違反するような近似最適解は得られなかったため，それらの評価指標については省略している．）

**表 1**　評価指標一覧表

| 指標名 | 測定項目 |
|---|---|
| 評価指標 1 | $\mathrm{err_p} := \dfrac{\|\mathcal{A}x - b\|}{1 + \max\limits_{i=1,\ldots,m} \|b_i\|}$ |
| 評価指標 2 | $\mathrm{err_d} := \dfrac{\|c - \mathcal{A}^* y - z\|}{1 + \max\limits_{i=1,\ldots,d} \|c_i\|}$ |
| 評価指標 3 | $\mathrm{err_{opt1}} := \dfrac{\|\langle c, x\rangle - b^\top y\|}{1 + \|\langle c, x\rangle\| + \|b^\top y\|}$ |
| 評価指標 4 | $\mathrm{err_{opt2}} := \dfrac{\|\langle x, z\rangle\|}{1 + \|\langle c, x\rangle\| + \|b^\top y\|}$ |

各グラフにおいて，実線は $(x_{\rm def}, y_{\rm def}, z_{\rm def})$，丸は $(x_{\rm tight}, y_{\rm tight}, z_{\rm tight})$，三角は $(x_{\rm pro}, y_{\rm pro}, z_{\rm pro})$ を示しており，対応する評価指標の値に対数を取ってプロットしている．ただし，対応する評価指標の値が 0 であった場合はプロットを省略している．また，それぞれのグラフの横軸は解いた問題を表しており，$(x_{\rm def}, y_{\rm def}, z_{\rm def})$ に対する各評価指標の値が昇順になるように並べ替え

ている．図 2–5 からは，以下の二点が観測できる．

1. ほぼすべての問題に対し，$(x_{\rm pro}, y_{\rm pro}, z_{\rm pro})$ は，ソルバーが返した解よりも，最適解としての精度が良い．
2. $(x_{\rm tight}, y_{\rm tight}, z_{\rm tight})$ の最適解としての精度は $(x_{\rm def}, y_{\rm def}, z_{\rm def})$ より良くなる場合もあれば，悪くなる場合もある．

文献 [3] では，内点解が存在しない問題や他のソルバーを用いた数値実験も行い，ソルバーは許容誤差値を小さくすると，その挙動が不安定になってしまう一方で，Kanoh and Yoshise [3] の後処理アルゴリズムは $(x_{\rm def}, y_{\rm def}, z_{\rm def})$ よりも精度の良い近似最適解を安定して返すことを確認した．

## 5. おわりに

本稿では，文献 [3, 4] の内容を基に，対称錐上の射影–再スケーリング法と射影–再スケーリング法を用いた後処理アルゴリズムに関して，アルゴリズムの簡単な流れと一部の計算機実験結果を紹介した．

今後の研究の方向性としては，大きく二つある．一つ目は，後処理アルゴリズムの改良である．本稿では説明しきれなかったが，提案した後処理アルゴリズムは実行時間の面で課題が残っている．文献 [3] の計算機実験で，提案した後処理アルゴリズムは，小規模問題に対しては数秒程度で処理を完了できたが，扱う問題サイズが大きくなるにつれ，設定した最大実行時間を超えて打ち切られることが多くなった．この主な原因は，射影行列の計算にかかる時間が増加してしまうことにある．そのため，なるべく精度を落とさずに近似的な射影行列を計算する方法を組み込む，といった改良の方針が考えられる．

二つ目は，提案手法がソルバー（内点法）よりも精度の良い解を求めることができた理由について調べることである．そのためには，二つの手法の異なる特徴に注目するのが良いと考えている．たとえば，文献 [3] の手法が内点法と異なる点として，射影とスケーリングを駆使した内部関数を用いて実行可能解を求めているという違いはもちろんだが，中心パスに依存しない内点を求めているという違いもある．実際に，2 次の半正定値行列錐 $\mathbb{S}^2_+$ を用いた以下の半正定値計画 (Ex) を，同じ初期解 ($a = 2$，$b = 1.5$，$c = 1$) を入力として，Kanoh and Yoshise [3] の手法で解いた場合に得られる近似解の点列と内点法により作成される点列をまとめたものが図 6 である．

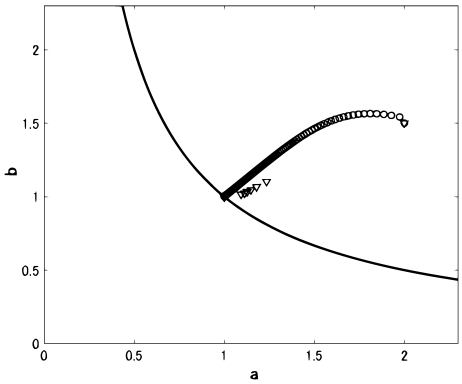



**図 6** Kanoh and Yoshise [3] の手法と内点法が生成した内点列の違い

$$\text{(Ex)} \quad \inf\ a+b \quad \text{s.t.}\ \ c=1, \quad \begin{pmatrix} a & c \\ c & b \end{pmatrix} \in \mathbb{S}^2_+. \tag{11}$$

図 6 において，丸は内点法，三角は Kanoh and Yoshise [3] の手法がそれぞれ作成した内点列となっており，三角のプロットが示す内点列は中心パスに依存していないことがわかる．このように，二つの手法の異なる特徴に注目しながら解の精度に貢献していた原因を考えることは，対称錐計画問題に対するアルゴリズムの今後の発展に大きく貢献できるはずである．

**謝辞**　本稿を執筆する機会を下さりました編集委員の先生方に感謝申し上げます．

### 参考文献

[1] K. Fujisawa, M. Fukuda, K. Kobayashi, M. Kojima, K. Nakata, M. Nakata and M. Yamashita, “Sdpa (semidefinite programming algorithm) and sdpa-gmp user’s manual—version 7.1.1.,” Research Report B-448, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2008.

[2] D. Henrion, S. Naldi and M. S. El Din, “Exact algorithms for linear matrix inequalities,” *SIAM Journal on Optimization*, **26**, pp. 2512–2539, 2016.

[3] S. Kanoh and A. Yoshise, “Post-processing with projection and rescaling algorithms for semidefinite programming,” *arXiv*, arXiv:2401.10429, 2024.

[4] S. Kanoh and A. Yoshise, “A new extension of Chubanov’s method to symmetric cones,” *Mathematical Programming*, **205**, pp. 773–812, 2024.

[5] MOSEK ApS, The Mosek optimization toolbox for MATLAB manual, https://www.mosek.com/documentation/ （2024 年 3 月 11 日閲覧）

[6] S. Chubanov, “A polynomial projection algorithm for linear feasibility problems,” *Mathematical Programming*, **153**, pp. 687–713, 2015.

[7] B. F. Lourenço, T. Kitahara, M. Muramatsu and T. Tsuchiya, “An extension of Chubanov’s algorithm to symmetric cones,” *Mathematical Programming*,

**173**, pp. 117–149, 2019.
[8] J. Peña and N. Soheili, "Solving conic systems via projection and rescaling," *Mathematical Programming*, **166**, pp. 87–111, 2017.
[9] J. Faraut and A. Korányi, *Analysis on Symmetric Cones Oxford mathematical monographs*, Clarendon Press, 1994.
[10] B. Borchers, "SDPLIB 1.2, a library of semidefinite programming test problems," *Optimization Methods and Software*, **11**, pp. 683–690, 1999.
[11] H. D. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Mathematical Programming*, **95**, pp. 407–430, 2003.