数理解析研究所講究録
第 1629 巻 2009 年 59-66

2次錐相補性問題に対するFischer-Burmeister関数を用いた平滑化ニュートン法について

東京理科大学 数理情報科学科 成島 康史
愛知大学 経営総合科学研究所 相良 信子
東京理科大学 数理情報科学科 小笠原 英穂

1 はじめに

本稿では2次錐相補性問題 (Second-Order Cone Complementarity Problem, SOCCP) に対する数値解法を提案する. SOCCP とは, 非線形相補性問題 (Nonlinear Complementarity Problem, NCP) や2次錐計画問題を拡張した次のような問題である:

$$\text{Find } (x,y,p)\in \boldsymbol{R}^n\times\boldsymbol{R}^n\times\boldsymbol{R}^{\ell} \text{ such that}$$
$$x\in\mathcal{K},\quad y\in\mathcal{K},\quad \langle x,y\rangle=0,\quad F(x,y,p)=0.$$

ここで, $\langle\cdot,\cdot\rangle$ はEuclid空間における通常の内積を表し, $F:\boldsymbol{R}^{2n+\ell}\to\boldsymbol{R}^{n+\ell}$ は連続微分可能な関数とする. また, $\mathcal{K}$ は直積 $\mathcal{K}=\mathcal{K}^{n_1}\times\mathcal{K}^{n_2}\times\cdots\times\mathcal{K}^{n_m}$ $(n_1+\cdots+n_m=n)$ により定義された凸錐とする. 各 $\mathcal{K}^{n_i}$ は $\mathcal{K}^{n_i}:=\left\{(z_1,z_2)\in\boldsymbol{R}\times\boldsymbol{R}^{n_i-1}\mid z_1\geq\|z_2\|\right\}$ で定義される自己双対な閉凸錐で, n_i 次の2次錐 (SOC) と呼ばれる. ただし $\|\cdot\|$ は2-ノルムを表す. 特に, $\mathcal{K}^1=\boldsymbol{R}_+=\{z\mid z\geq 0\}$ であり, $n_1=n_2=\cdots=n_m=1$ の場合, $\mathcal{K}=\boldsymbol{R}^n_+$, すなわち非負象限となる. さらに $\ell=0$ で, $F(x,y,p)=f(x)-y$ $(f:\boldsymbol{R}^n\to\boldsymbol{R}^n)$ のとき, SOCCP はNCPに帰着する.

SOCCP はNCPと同様に, 等価な非線形方程式系として再定式化されることが知られており, それを用いた数値解法がいくつか提案されている [1, 3]. 今回我々は, Fukushima, Luo and Tseng [2] によって提案されたSOCCPに対する平滑化Fischer-Burmeister関数を用い, それに基づく数値解法を提案する.

以降では簡単のために, $\mathcal{K}=\mathcal{K}^n$ で説明するが, 一般の場合も同様に扱うことができる.

2 SOC関数による再定式化

はじめに, 以下での説明で必要となるSOCに関連するジョルダン代数を簡単に述べておく. 任意の $x=(x_1,x_2)$, $y=(y_1,y_2)\in\boldsymbol{R}\times\boldsymbol{R}^{n-1}$ に対して, ジョルダン積は $x\cdot y=(x^Ty,\, y_1x_2+x_1y_2)$ で定義される. $x\cdot x=x^2$ と書き, $x\in\mathcal{K}^n$ に対して, その平方根 $x^{1/2}$ を

$$x^{1/2}=\begin{cases}\left(\varsigma,\dfrac{x_2}{2\varsigma}\right),\ \varsigma=\sqrt{\dfrac{1}{2}\left(x_1+\sqrt{x_1^2-\|x_2\|^2}\right)} & (x\neq 0\text{ のとき})\\ 0 & (x=0\text{ のとき})\end{cases}$$

によって定義する. ここで, $(x^{1/2})^2=x^{1/2}\cdot x^{1/2}=x$ であることを注意しておく. さらに, $x=(x_1,x_2)\in\boldsymbol{R}\times\boldsymbol{R}^{n-1}$ に対して, 次のように定義される n 次対称行列 L_x をArrow行

列と呼ぶ:

$$L_x = \begin{bmatrix} x_1 & x_2^T \\ x_2 & x_1 I \end{bmatrix}.$$

特に, $x \in \operatorname{int} \mathcal{K}^n$ と L_x が正則であることとは等価である. 任意の $x = (x_1, x_2) \in \boldsymbol{R} \times \boldsymbol{R}^{n-1}$ は, $x = \lambda_1 u^{(1)} + \lambda_2 u^{(2)}$ と分解される. ただし, λ_1, λ_2 と $u^{(1)}$, $u^{(2)}$ はそれぞれ x の固有値, 固有ベクトルと呼ばれ,

$$\begin{aligned} \lambda_i &= x_1 + (-1)^i \|x_2\|, \\ u^{(i)} &= \begin{cases} \dfrac{1}{2}\left(1,\ (-1)^i \dfrac{x_2}{\|x_2\|}\right) & (x_2 \neq 0 \text{ のとき}) \\ \dfrac{1}{2}\left(1,\ (-1)^i \bar{u}_2\right) & (x_2 = 0 \text{ のとき}) \end{cases} \end{aligned}$$

$(i = 1, 2)$ で与えられる. ここで $\bar{u}_2 \in \boldsymbol{R}^{n-1}$ は $\|\bar{u}_2\| = 1$ であるような任意のベクトルである.

次に, SOC 関数を用いた SOCCP の再定式化を説明する. 一般に, 関数 $\widehat{\phi} : \boldsymbol{R}^{2n} \to \boldsymbol{R}^n$ が次の性質をもつとき, $\widehat{\phi}$ は SOC 関数であるという:

$$\widehat{\phi}(x, y) = 0 \iff x \in \mathcal{K}^n, \quad y \in \mathcal{K}^n, \quad \langle x,\ y \rangle = 0.$$

したがって, SOC 関数を含む関数 $\widehat{H} : \boldsymbol{R}^{2n+\ell} \to \boldsymbol{R}^{2n+\ell}$ を

$$\widehat{H}(x, y, p) := \begin{pmatrix} \widehat{\phi}(x, y) \\ F(x, y, p) \end{pmatrix}$$

と定義することにより, SOCCP は等価な非線形方程式系 $\widehat{H}(x, y, p) = 0$ に再定式化される. Fukushima et al. [2] は以下で与えられる Natural Residual (NR) 関数と Fischer-Burmeister (FB) 関数を定義し, どちらも SOC 関数になることを示した:

$$\begin{aligned} \phi_{\text{NR}}(x, y) &:= x - [x - y]_+, \\ \phi_{\text{FB}}(x, y) &:= x + y - (x^2 + y^2)^{1/2}. \end{aligned}$$

ここで, $[z]_+$ は $z \in \boldsymbol{R}^n$ の $\mathcal{K}^n$ への射影を表す. これらは NCP の再定式化に用いられる関数の SOCCP への自然な拡張になっている. これらの関数を用いて SOCCP は非線形方程式系 $\widehat{H}(x, y, p) = 0$ に再定式化される.

3 平滑化 Fischer-Burmeister 関数

SOC 関数は一般に微分不可能であるため, 平滑化関数がしばしば使用される. Fukushima et al. [2] は NR 関数に対する平滑化関数を提案しており, Hayashi et al. [3] 及び Chen et al. [1] は, それに基づいた数値解法を提案している. 前者は, 平滑化のために導入したパラメータをそのままパラメータとして調整しているのに対し, 後者はパラメータを変数として取込んでいる.

Fukushima et al. はパラメータ t を含む以下の平滑化 FB 関数 $\phi_t : \boldsymbol{R}^{2n} \to \boldsymbol{R}^n$ も提案している:

$$\phi_t(x,y) := x + y - (2t^2e + x^2 + y^2)^{1/2}.$$

ただし, $e = (1,0,\ldots,0)^T \in \boldsymbol{R}^n$ である.

今回我々は, t をパラメータとしてではなく, Chen et al. の考え方に従い, 変数とみなした平滑化 FB 関数 $\phi : \boldsymbol{R}^{1+2n} \to \boldsymbol{R}^n$,

$$\phi(t,x,y) := x + y - (2t^2e + x^2 + y^2)^{1/2}$$

を扱う. パラメータをそのままの形で調整する解法も考えられるが, 変数として扱った方がパラメータを調整するよりも容易に, 自動的に調整できるという利点がある. 上での議論と同様に,

$$H(t,x,y,p) := \begin{pmatrix} t \\ \phi(t,x,y) \\ F(x,y,p) \end{pmatrix}$$

で定義される関数 $H : \boldsymbol{R}^{1+2n+\ell} \to \boldsymbol{R}^{1+2n+\ell}$ を考えることにより, SOCCP は非線形方程式系 $H(t,x,y,p) = 0$ に再定式化される. さらに, メリット関数 $\Psi : \boldsymbol{R}^{1+2n+\ell} \to \boldsymbol{R}$ を

$$\Psi(t,x,y,p) := \|H(t,x,y,p)\|^2 = t^2 + \|\phi(t,x,y)\|^2 + \|F(x,y,p)\|^2$$

により定義する.

アルゴリズムを構築する上で重要となる H の性質を以下に与えておく.

命題 1 $w = (w_1, w_2) := 2t^2e + x^2 + y^2 \in \boldsymbol{R} \times \boldsymbol{R}^{n-1}$ とし, λ_1, λ_2 を w の固有値とする. さらに, $u := w^{1/2}$ とおく. このとき, 関数 H は $\boldsymbol{R}_{++} \times \boldsymbol{R}^{2n+\ell}$ 上で連続微分可能であり, (転置) ヤコビ行列 ∇H は次式で与えられる:

$$\nabla H(t,x,y,p) = \begin{bmatrix} 1 & -2te^TL_u^{-1} & 0 \\ 0 & I - L_xL_u^{-1} & \nabla_x F(x,y,p) \\ 0 & I - L_yL_u^{-1} & \nabla_x F(x,y,p) \\ 0 & 0 & \nabla_p F(x,y,p) \end{bmatrix}, \tag{1}$$

ただし,

$$L_u^{-1} = \begin{cases} \begin{bmatrix} b & -\dfrac{cw_2^T}{\|w_2\|} \\ -\dfrac{cw_2}{\|w_2\|} & aI + (b-a)\dfrac{w_2w_2^T}{\|w_2\|^2} \end{bmatrix} & (w_2 \neq 0 \text{ のとき}) \\ \dfrac{1}{\sqrt{w_1}} I & (w_2 = 0 \text{ のとき}) \end{cases}$$

とし, $a = \dfrac{2}{\sqrt{\lambda_1} + \sqrt{\lambda_2}}$, $b = \dfrac{1}{2}\left(\dfrac{1}{\sqrt{\lambda_1}} + \dfrac{1}{\sqrt{\lambda_2}}\right)$, $c = \dfrac{1}{2}\left(\dfrac{1}{\sqrt{\lambda_1}} - \dfrac{1}{\sqrt{\lambda_2}}\right)$ とする.

命題 2 点 $(x,y,p) \in \boldsymbol{R}^{2n+\ell}$ は以下の仮定を満たすとする:

1. rank $\nabla_p F(x,y,p) = \ell$,

2. 任意の $(\xi,\eta,\varphi) \in \boldsymbol{R}^{2n+\ell}$ に対して，もし $\nabla F(x,y,p)^T(\xi,\eta,\varphi) = 0$ ならば $\xi^T\eta \geq 0$ が成立する．

このとき，$t > 0$ ならば，(1) によって与えられる $\nabla H(t,x,y,p)$ は正則である．

命題 3 関数 H は $\boldsymbol{R}^{1+2n+\ell}$ 上で semismooth である．さらに，もし ∇F が $\boldsymbol{R}^{2n+\ell}$ 上で局所 Lipschitz 連続ならば，関数 H は $\boldsymbol{R}^{1+2n+\ell}$ 上で strongly semismooth である．

4 アルゴリズムと収束性

本節では前節で与えた平滑化 FB 関数に基づくアルゴリズムを提案し，その収束性を議論する．簡単のために $s = (t,x,y,p)$ などと表記する．$\gamma\|\bar{s}\| < 1$ となるような $\bar{s} = (\bar{t},\bar{x},\bar{y},\bar{p}) \in \boldsymbol{R}_{++} \times \boldsymbol{R}^{2n+\ell}$ と $\gamma \in (0,1)$ を選ぶ．さらに，関数 $\beta : \boldsymbol{R}^{1+2n+\ell} \to \boldsymbol{R}_+$ と $\boldsymbol{R}^{1+2n+\ell}$ の部分集合 Ω を次のように定義する:

$$\beta(s) := \gamma \min\{1, \Psi(s)\},$$
$$\Omega := \{\, s = (t,x,y,p) \mid t \geq \beta(s)\bar{t} \,\}.$$

このとき, Qi, Sun and Zhou [4] によって以下の命題が示されている．

命題 4 $\quad H(s) = 0 \iff \beta(s) = 0 \iff H(s) = \beta(s)\bar{s}$

この命題は, $H(s) = 0$ を解くことと $H(s) = \beta(s)\bar{s}$ を解くことは等価であることを示している．よって我々は, $H(s) = \beta(s)\bar{s}$ を解くためのアルゴリズムとして以下を提案する．

アルゴリズム 1

Step 0. 定数 $\rho \in (0,1)$, $\sigma \in (0,1/2)$ を選び, 初期点 $s^{(0)} := (t^{(0)},x^{(0)},y^{(0)},p^{(0)}) \in \Omega$ を与える．$k := 0$ として *Step 1* へ.

Step 1. 収束判定条件を満たしていれば終了する．

Step 2. 次の方程式系を $d^{(k)}$ について解く．

$$H(s^{(k)}) + \nabla H(s^{(k)})^T d^{(k)} = \beta(s^{(k)})\bar{s}.$$

Step 3. 次の不等式を満たす最小の非負整数 m を求め, $m_k := m$ とする．

$$\Psi(s^{(k)} + \rho^m d^{(k)}) \leq [1 - 2\sigma(1-\gamma\|\bar{s}\|)\rho^m]\,\Psi(s^{(k)}). \tag{2}$$

Step 4. $s^{(k+1)} := s^{(k)} + \rho^{m_k} d^{(k)}$ により点列を更新し，$k := k+1$ として *Step 1* へ.

次に，提案したアルゴリズムの収束性を考える．まず, 実行可能性について次の命題が成立する．

命題 5 任意に固定された $k \geq 0$ に対して $t^{(k)} > 0,\ s^{(k)} \in \Omega$ であるとし，さらに $\nabla H(s^{(k)})$ が正則であるとする．このとき，(2) を満たす非負整数 m が存在し，$s^{(k+1)}$ が定義可能である．さらに，$t^{(k+1)} > 0$ かつ $s^{(k+1)} \in \Omega$ が成立する．

以下ではアルゴリズム 1 によって生成される点列を $\{s^{(k)}\}$ とする．さらに，収束性を議論するために以下の仮定をする．

仮定 1

(A1) 各 k に対して，$\nabla H(s^{(k)})$ は正則である．

(A2) 点列 $\{s^{(k)}\}$ は有界である．

(A3) $\{s^{(k)}\}$ の各集積点 $s^* = (t^*, x^*, y^*, p^*)$ に対して，もし $t^* > 0$ かつ $s^* \in \Omega$ ならば，$\nabla H(s^*)$ は正則である．

上の仮定の下で，大域的収束性および局所的収束性に関して以下の定理が成立する．

定理 1 (大域的収束性) 仮定 A1–A3 を満たしているとする．このとき，$\{s^{(k)}\}$ の各集積点 $s^* = (t^*, x^*, y^*, p^*)$ は方程式系 $H(s) = 0$ の解である．したがって，(x^*, y^*, p^*) はSOCCP の解である．

定理 2 (局所的収束性) 仮定 A1–A3 を満たしているとする．さらに，$s^* = (0, x^*, y^*, p^*)$ を $\{s^{(k)}\}$ の集積点とし，すべての $V \in \partial H(s^*)$ は正則であるとする．このとき，

$$\|s^{(k+1)} - s^*\| = o(\|s^{(k)} - s^*\|), \qquad t^{(k+1)} = o(t^{(k)})$$

が成立する．さらに，もし ∇F が局所 Lipschitz 連続ならば，

$$\|s^{(k+1)} - s^*\| = O(\|s^{(k)} - s^*\|^2), \qquad t^{(k+1)} = O((t^{(k)})^2)$$

が成立する．ここで，$\partial H(s^*)$ は H の s^* における Clarke 劣勾配を表す．

5 数値実験

本節では提案した数値解法が理論どおり収束するかを確認するために数値実験結果を報告する．すべての数値実験は Compaq nx 9030 において，Matlab 7 を使用し実行された．

今回，我々は $F(x, y, p) = f(x) - y$ の場合の SOCCP，つまり，

$$x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad \langle x, y \rangle = 0, \quad y = f(x)$$

に対して実験を行った．ただし，$f : \boldsymbol{R}^n \to \boldsymbol{R}^n$ は連続微分可能とする．この問題においては，変数 p が存在しないため，$s = (t, x, y)$ となる．以降では，2 次錐の直積 $\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$ $(m, n_1, \ldots, n_m \geq 1, n = n_1 + \cdots + n_m)$ を簡単に $\mathcal{K} = [n_1, n_2, \ldots, n_m]$ と表す．アルゴリズム 1 で用いられるパラメータの設定値は $\sigma = 0.4$, $\rho = 0.5$, $\bar{t} = 2.0$, $\bar{s} = (\bar{t}, 0, 0)$, $\gamma = 0.4$ とし，初期点の選択については，ξ を $[0, 5]$ の中から，$(a, b) \in \boldsymbol{R}^n \times \boldsymbol{R}^n$

の各要素を $[0,1]$ の中からランダムに選択し，$(x^{(0)}, y^{(0)}) := \xi(a,b)/\|(a,b)\|$ とした．また，収束判定条件を $\|H_{\mathrm{FB}}(x^{(k)}, y^{(k)})\| < 10^{-8}$ とした．ただし，$H_{\mathrm{FB}}(x,y) = \begin{pmatrix} \phi_{\mathrm{FB}}(x,y) \\ f(x) - y \end{pmatrix}$ である．数値実験結果は表 1–6 に示したとおりである．ただし, 表中の (Iter) と (CPU) はそれぞれ, 各問題に対し数値実験を 20 回実行した時の反復回数と実行時間 (秒) の平均を表している．

5.1 線形 SOCCP

まず，関数 f が線形の場合, すなわち線形 SOCCP

$$x \in \mathcal{K}, \quad y \in \mathcal{K}, \quad \langle x, y \rangle = 0, \quad y = Mx + q,$$

に対して実験を行った．ただし，$M \in \boldsymbol{R}^{n\times n}$ は (半) 正定値対称行列であるものとし，$q := 10^{\alpha}\sqrt{n}\zeta - Me$ とした．ここで，$e = (1,0,\dots,0)^T \in \mathrm{int}\,\mathcal{K}^n$ であり，α は $[-1,1]$ の範囲でランダムに選択した．さらに，$\zeta \in \mathrm{int}\,\mathcal{K}^n$ かつ $\|\zeta\| = 1$ を満たすように $\zeta := \cos\theta(1, v/\|v\|)/\sqrt{2} + \sin\theta(1, -v/\|v\|)/\sqrt{2}$ とした．ただし，$v \in \boldsymbol{R}^{n-1}$ の各要素は $[-1,1]$ の範囲でランダムに選択し，$\theta = \pi/5$ とした．

(例 1) パスカル行列

まず，行列 M にパスカル行列を用いた場合の数値実験結果を報告する．パスカル行列は正定値対称行列ではあるが, 悪条件であることが知られている．以下に, $n = 5$ の場合のパスカル行列と $n = 2, \dots, 17$ における条件数を掲げておく．

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix}$$

n	条件数
2	6.8541
5	8.51 e+003
10	4.16 e+009
13	1.29 e+013
15	2.84 e+015
17	6.34 e+017

表 1，2 はそれぞれ，$\mathcal{K}$ を $\mathcal{K} = [4, n-8, 1, 1, 1, 1]$ (6 分割) および $\mathcal{K} = [n]$ (分割なし) とした場合の実験結果である．

表 1: $\mathcal{K} = [4, n-8, 1, 1, 1, 1]$

n	Iter	CPU(s)
13	17.75	0.199
15	18.90	0.222
17	23.05	0.280

表 2: $\mathcal{K} = [n]$

n	Iter	CPU(s)
13	13.85	0.143
15	8.75	0.142
17	10.10	0.134

表 1 から，反復回数，実行時間は次元とともに，ある程度増加しているが，次元による条

件数の増加と比較すると，極端な増加は見られず，安定して解けていることがわかる．一方，表 2 からは，次元に関係なく，ほぼ一定時間で解けていることが見てとれる．

(例 2) ランダム行列 ($r = n - 2$)
次に，行列 M が非正則な半正定値対称行列である問題に対して実験を行った．行列 M のランク $\operatorname{rank} M = r$ が $r < n$ となるように，$n \times r$ 行列 B に対し，$M := nBB^T/\|BB^T\|$ とした．ただし，B は 各要素を $[-1, 1]$ の範囲からランダムに選択したランダム行列である．表 3 及び 4 は，$r = n - 2$ とし，分割をそれぞれ $\mathcal{K} = [n-4, 1, 1, 1, 1]$ (5 分割)，$\mathcal{K} = [n/2-2, n/2-2, 1, 1, 1, 1]$ (6 分割) とした場合の実験結果である．

表 3: $\mathcal{K} = [n-4, 1, 1, 1, 1]$

n	Iter	CPU(s)
100	9.3	0.275
200	9.2	1.053
500	9.5	13.677
800	9.7	49.289
1000	9.6	89.447

表 4: $\mathcal{K} = [n/2-2, n/2-2, 1, 1, 1, 1]$

n	Iter	CPU(s)
100	9.4	0.286
200	9.3	1.093
500	9.4	12.793
800	9.7	47.774
1000	10.0	91.121

表 3 と 4 から，次元に関係なく 10 回程度で収束していることがわかる．また，次元の増加に伴い実行時間が急激に増加している．反復回数がほとんど変化していないことを考慮すると，その理由は，Step 2 で線形方程式系を解くのに時間がかかっているためと考えられる．

(例 3) ランダム行列 ($n = 300,\ r = 200,\ 150$)
次に行列 M が非正則性の強い行列である場合の問題を実験するために，行列の次元を $n = 300$ に固定し，行列のランクを $r = 200,\ 150$ とした．なお，行列 M の生成法は (例 2) と同様である．表 5 は $\mathcal{K} = [1, 1, 296, 1, 1]$ (5 分割) とした場合の実験結果である．

表 5: $\mathcal{K} = [1, 1, 296, 1, 1]$

r	Iter	CPU(s)
200	10.0	3.577
150	14.5	5.224

表 5 から，$r = 150$ の場合のほうが $r = 200$ の場合よりも反復回数，実行時間ともに増加している．しかし，どちらの場合も解に収束しており，$r = 200$ の時は (例 2) (つまり非正則性の弱い場合) と同程度の反復回数で収束している．

5.2 非線形 SOCCP

次に非線形の SOCCP に対して数値実験を行った．関数 $f : \boldsymbol{R}^5 \to \boldsymbol{R}^5$ を

$$f(x) := \begin{pmatrix} 24(2x_1 - x_2)^3 + \exp(x_1 - x_3) - 4x_4 + x_5 \\ -12(2x_1 - x_3)^3 + 3(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 6x_4 - 7x_5 \\ -\exp(x_1 - x_3) + 5(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 3x_4 + 5x_5 \\ 4x_1 + 6x_2 + 3x_3 - 1 \\ -x_1 + 7x_2 - 5x_3 + 2 \end{pmatrix}$$

とし, $\mathcal{K} = [3,\ 2]$ とした. この問題は Hayashi et al. [3] の数値実験で使用されている問題で, 2 次錐計画問題の KKT 条件から派生する問題である.

表 6: $\mathcal{K} = [3,\ 2]$

Iter	CPU(s)
13.2	0.201

表 6 が示すとおり, 提案手法は非線形の問題に対しても確かに収束していることがわかる.

6 終わりに

本論文で我々は, 2 次錐相補性問題に対する Fischer-Burmeister 関数を用いた平滑化ニュートン法を提案し, その大域的収束性及び 2 次収束性を証明した. また, 提案した手法が理論どおり収束するかを検証するため, 数値実験を行った. それによれば, 解に安定して収束することが確認できた. したがって我々は, 本手法が 2 次錐相補性問題の有望な解法になり得ると考えている. しかしながら, 今回の数値実験は予備的なものにとどまっているため, より多くの数値実験を行い, 他の手法と比較してみることは今後の課題である.

参考文献

[1] X.-D. Chen, D. Sun and J. Sun, Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementaritiy problems, *Computational Optimization and Applications* **25** (2003), 39–56.

[2] M. Fukushima, Z.-Q. Luo and P. Tseng, Smoothing functions for second-order-cone complementarity problems, *SIAM Journal on Optimization* **12** (2001), 436–460.

[3] S. Hayashi, N. Yamashita and M. Fukushima, A combined smoothing and regularization method for monotone second-order cone complementarity problems, *SIAM Journal on Optimization* **15** (2005), 593–615.

[4] L. Qi, D. Sun and G. Zhou, A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities, *Mathematical Programming* **87** (2000) 1–35.