Bulletin of the Iranian Mathematical Society (2022) 48:3909–3936
https://doi.org/10.1007/s41980-022-00725-9

ORIGINAL PAPER

A New Smooth NCP Function for Solving Semidefinite Nonlinear Complementarity Problems

Mohamed Ferhaoui[1] · **Boubakeur Benahmed**[2] · **Lotfi Mouhadjer**[3]

Received: 18 December 2021 / Revised: 27 July 2022 / Accepted: 31 July 2022 /
Published online: 19 September 2022

Abstract
In this paper, we propose to solve semidefinite nonlinear complementarity problems (NCP) associated to a nonlinear matrix function , by a quasi-Newton method. For this, we reformulate this problem as a smooth nonlinear matrix equation by using a new smooth NCP matrix function, then we apply a quasi-Newton method for solving this matrix equation. We prove the local superlinear convergence of our algorithm and we give some numerical examples to illustrate the efficiency of the proposed method.

Keywords Semidefinite nonlinear complementarity problems · NCP matrix function · Matrix equation · Quasi-Newton method · Superlinear convergence

Mathematics Subject Classification 90C53 · 90C33 · 49J52 · 65F45

1 Introduction

Let $\mathcal{S}^n$ and $\mathcal{S}^n_+$ denote the space of $n \times n$ symmetric matrices, and the cone of symmetric positive semidefinite matrices respectively. We endow $\mathcal{S}^n$ with the inner product and the Frobenius norm defined by

Communicated by Behnam Hashemi.

✉ Mohamed Ferhaoui
mohamed.ferhaoui@univ-tiaret.dz

Boubakeur Benahmed
boubakeur.benahmed@enp-oran.dz

Lotfi Mouhadjer
lotfi.mouhadjer@gmail.com

[1] Department of Sciences and Technology, University of Ibn Khaldoun, Tiaret, Algeria

[2] Department of Preparatory Training in Sciences and Technology, National Polytechnic School of Oran, Maurice Audin, Oran, Algeria

[3] Higher School of Applied Sciences, Tlemcen, Algeria

$$\langle X, S\rangle = \mathrm{Tr}(XS) \quad \text{and} \quad \|X\|_F = \sqrt{\langle X, X\rangle},$$

where $X, S \in \mathcal{S}^n$ and $\mathrm{Tr}(\cdot)$ stands for the trace of a matrix.

The notation $A \succeq 0$ and $A \succ 0$ means that A is symmetric positive semidefinite and symmetric positive definite, respectively.

Let $F : \mathcal{S}^n \to \mathcal{S}^n$ be a continuously differentiable function. The Semidefinite Nonlinear Complementarity Problem associated to F (SDNCP(F)) is defined as follows:

Find $X \in \mathcal{S}^n$ such that:

$$X \in \mathcal{S}^n_+, Y = F(X) \in \mathcal{S}^n_+, \quad \text{and} \quad \langle X, Y\rangle = \mathrm{Tr}(XY) = 0. \tag{1.1}$$

If F is an affine function of the form $F(X) = L(X) + Q$ where $L : \mathcal{S}^n \to \mathcal{S}^n$ is a linear operator and $Q \in \mathcal{S}^n$, then SDNCP(F) is called a semidefinite linear complementarity problem (SDLCP(F), for short).

Note that the SDNCP(F) is a generalization of the nonnegative orthant nonlinear complementarity problem (NCP) defined by: Find $x \in \mathbb{R}^n$ such that:

$$x \geq 0, f(x) \geq 0, \quad \text{and} \quad x^{\mathrm{T}} f(x) = 0, \tag{1.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable vector function.

This problem arises naturally in the optimality conditions of semidefinite programming problems (SDP) involving inequality constraints and has wide applications in engineering, economics, management sciences and other fields (see [10, 12], for example). Some methods have been proposed to solve the SDNCP(F) problems, for example, interior-point methods and merit function methods in [11]. Chen and Tseng [13] proposed a so-called exact non-interior continuation (or smoothing-type Newton) method to solve SDNCP(F).

Another approach to solve the problem (SDNCP(F)) is to reformulate it as a matrix equation $H(X, S) = 0$ where $H : \mathcal{S}^n \times \mathcal{S}^n \to \mathcal{S}^n \times \mathcal{S}^n$ is defined by

$$H(X, S) = \begin{pmatrix} \phi(X, S) \\ F(X) - S \end{pmatrix}, \tag{1.3}$$

and $\phi : \mathcal{S}^n \times \mathcal{S}^n \to \mathcal{S}^n$ is an NCP function, that is:

$$\phi(X, S) = 0 \Leftrightarrow X \in \mathcal{S}^n_+, S \in \mathcal{S}^n_+, \quad \text{and} \quad XS = 0. \tag{1.4}$$

Most reformulations of the SDNCP(F) problem use nonsmooth NCP functions and the most popular NCP functions are Fischer–Burmeister function [6, 7, 9], defined by

$$\phi_{\mathrm{FB}}(X, S) = X + S - \sqrt{X^2 + S^2}, \tag{1.5}$$

and the natural residual function [2], defined by

$$\phi_{\mathrm{NR}}(X, S) = \min(X, S) = \frac{1}{2}\left(X + S - \sqrt{(X-S)^2}\right).$$
$$= \frac{1}{2}(X + S - |X - S|), \tag{1.6}$$

where the matrix valued functions square root, min and absolute value are defined in the following section.

It is shown in [4], that the functions ϕ_{FB} and $\phi_{\min}$ are not differentiable everywhere, but only strongly semismooth functions. To overcome the nonsmoothness, many authors [8, 13, 14] used smooth approximations of these functions. Smooth approximations of ϕ_{FB} and $\phi_{\min}$ [2, 8, 13, 15], are given for a parameter $\mu \in \mathbb{R}^*_+$, respectively by:

$$\phi^{\mu}_{\mathrm{FB}}(X, S) = X + S - \sqrt{X^2 + S^2 + 2\mu^2 I}, \tag{1.7}$$

and

$$\phi^{\mu}_{\mathrm{NR}}(X, S) = \frac{1}{2}\left(X + S - \sqrt{(X-S)^2 + 4\mu^2 I}\right). \tag{1.8}$$

In this paper, we propose a new NCP function to give a smooth reformulation of the SDNCP(F) problem. This new NCP function is continuously differentiable and is defined by $\phi_\alpha : \mathcal{S}^n \times \mathcal{S}^n \to \mathcal{S}^n$ where

$$\phi_\alpha(X, S) = XS + SX - \frac{1}{\alpha}[\min(0, X + S)]^2, \quad 0 < \alpha \leq 1. \tag{1.9}$$

The rest of the paper is organized as follows. In Sect. 2, we recall some useful preliminaries that will be needed in the sequel. We study some properties of the NCP function ϕ_α in Sect. 3. In Sect. 4, we develop our algorithm for solving the SDNCP(F) using the NCP function ϕ_α and a quasi-Newton method. For the sake of illustrating the effectiveness of our algorithm, some numerical experiments are reported in Sect. 5. We draw conclusions in Sect. 6.

2 Preliminaries

In this section, we recall the spectral definition of matrix valued function associated to a given real valued function and we present a classical result about its differentiability.

For any $X \in \mathcal{S}^n$, let $\lambda_1(X) \leq \lambda_2(X) \leq \cdots \leq \lambda_n(X)$ be eigenvalues of X, then X admits a spectral decomposition of the form $X = P D_X P^{\mathrm{T}}$ for some $P \in \mathcal{O}^n$, where $\mathcal{O}^n$ denotes the set of $P \in \mathbb{R}^{n\times n}$ that are orthogonal and $D_X = \mathrm{diag}[\lambda_1(X), \ldots, \lambda_n(X)]$ denotes the $n \times n$ diagonal matrix where $\lambda_i(X)$ are diagonal elements.

Let $f : \mathbb{R} \to \mathbb{R}$. We can define a corresponding matrix valued function $F : \mathcal{S}^n \to \mathcal{S}^n$ by

$$X \to F(X) = P\text{diag}\left(f(\lambda_1(X)), f(\lambda_2(X)), \ldots, f(\lambda_n(X))\right) P^{\mathrm{T}}, \tag{2.1}$$

where $X = P D_X P^{\mathrm{T}}$ is the spectral decomposition of X.

Example 2.1 (1) For $f(t) = \sqrt{t}$, the corresponding function $F : \mathcal{S}^n_+ \to \mathcal{S}^n_+$ is defined by

$$F(X) = \sqrt{X} = P\text{diag}\left(\sqrt{\lambda_1(X)}, \ldots, \sqrt{\lambda_n(X)}\right) P^{\mathrm{T}}.$$

is called square root function.

(2) For $f(t) = \min(0, t)$, the corresponding function $F : \mathcal{S}^n \to \mathcal{S}^n$ is defined by

$$F(X) = \min(0, X) = P\text{diag}\left(\min(0, \lambda_1(X)), \ldots, \min(0, \lambda_n(X))\right) P^{\mathrm{T}}.$$

is called the min function.

(3) For $f(t) = |t|$, the corresponding function $F : \mathcal{S}^n \to \mathcal{S}^n$ is defined by

$$F(X) = |X| = P\text{diag}\left(|\lambda_1(X)|, \ldots, |\lambda_n(X)|\right) P^{\mathrm{T}}, \tag{2.2}$$

is called the absolute value function.

Remark 2.2 Note that the definition of absolute value function given by (2.2) is not applicable to the vectors, since here the absolute value is applied to the eigenvalues and not the entries of a matrix. Furthermore, we have

$$|X| = \sqrt{X^2} \quad \text{and} \quad |X|^2 = X^2.$$

It is proved in [5, 7, 14] that the matrix valued function F inherited all topological properties of the function f, in particular we have:

Proposition 2.3 (See [7, 14] for the proof) *Let $f : \mathbb{R} \to \mathbb{R}$ and $F : \mathcal{S}^n \to \mathcal{S}^n$ be the corresponding matrix valued function. Let $X = P D_X P^{\mathrm{T}}$ be the spectral decomposition of X. Then*

F is (continuously) differentiable at an $X \in \mathcal{S}^n$ with eigenvalues $\lambda_1(X), \ldots, \lambda_n(X)$ if and only if f is (continuously) differentiable at $\lambda_1(X), \ldots, \lambda_n(X)$. Moreover, $F'(X)$ is an operator given by

$$F'(X)(H) = P\left[f^{[1]}(D_X) \circ \left(P H P^{\mathrm{T}}\right)\right] P^{\mathrm{T}}, \tag{2.3}$$

where $\circ$ is the Hadamard product and $f^{[1]}(D_X)$ is the matrix whose elements are:

$$\left[f^{[1]}(D_X)\right]_{ij} = \begin{cases} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j}, & \text{if } i \neq j \in \{1, 2, \ldots, n\} \\ f'(\lambda_i) & \text{if } i = j \in \{1, 2, \ldots, n\} \end{cases}. \tag{2.4}$$

3 Properties of the Function ϕ_α

In this section, we state some properties of the function ϕ_α given by (1.9). In particular, we show that this function is an NCP function, that is continuously differentiable and we calculate its derivative.
First, we remark that our new function ϕ_α can be expressed equivalently as follows:

$$\phi_\alpha(X, S) = XS + SX - \frac{1}{4\alpha}(X + S - |X + S|)^2, \quad 0 < \alpha \leq 1, \tag{3.1}$$

or by the form:

$$\phi_\alpha(X, S) = XS + SX - \frac{1}{\alpha} P \text{diag}(f(\lambda_1), \ldots f(\lambda_n)) P^{\mathrm{T}}, \quad 0 < \alpha \leq 1, \tag{3.2}$$

with $f(\cdot) = \min^2(0, \cdot)$ and $X + S = PDP^{\mathrm{T}}$ is the spectral decomposition of the symmetric matrix $X + S$.

Proposition 3.1 *The function ϕ_α defined in (1.9) is an NCP function for all $0 < \alpha \leq 1$, i.e.:*

$$X \in \mathcal{S}_+^n, S \in \mathcal{S}_+^n, \quad \textit{and} \quad Tr(XS) = 0 \Leftrightarrow \phi_\alpha(X, S) = 0.$$

Proof Let $0 < \alpha \leq 1$.
(1) Suppose that $X \in \mathcal{S}_+^n$, $S \in \mathcal{S}_+^n$, and $\text{Tr}(XS) = 0$, then

$$XS = SX = 0 \Rightarrow XS + SX = 0,$$

and

$$X + S \in \mathcal{S}_+^n \Rightarrow \min(0, X + S) = 0.$$

Obviously, this implies that $\phi_\alpha(X, S) = 0$
(2) Conversely, suppose that $\phi_\alpha(X, S) = 0$. Using the form (3.1) of the function ϕ_α and the fact that $\phi_\alpha(X, S) = 0$, it follows that

$$XS + SX - \frac{1}{4\alpha}(X + S - |X + S|)^2 = 0.$$

So,

$$4\alpha(XS + SX) = (X + S - |X + S|)^2, \tag{3.3}$$

then $XS + SX \in \mathcal{S}_+^n$.
Next, by expanding the right-hand side of (3.3) we obtain

$$4\alpha(XS + SX) = 2(X + S)^2 - (X + S)|X + S| - |X + S|(X + S),$$

since $|X+S|^2 = (X+S)^2$.
So, for all $0 < \alpha \leq 1$, we get

$$\begin{aligned}
&(X+S)\,|X+S| + |X+S|\,(X+S)\\
&= 2\,(X+S)^2 - 4\alpha\,(XS+SX)\\
&= 2X^2 + 2S^2 + (2-4\alpha)\,(XS+SX)\\
&= 2X^2 + 2S^2 - 2\,(XS+SX) + (4-4\alpha)\,(XS+SX)\\
&= 2\,(X-S)^2 + (4-4\alpha)\,(XS+SX) \in \mathcal{S}_+^n
\end{aligned}$$

and then

$$(X+S)\,|X+S| + |X+S|\,(X+S) \in \mathcal{S}_+^n.$$

Let $X+S = PDP^{\mathrm{T}}$ be the spectral decomposition of $X+S$, where $D = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$, then

$$\begin{aligned}
&(X+S)\,|X+S| + |X+S|\,(X+S) = PDP^{\mathrm{T}}\left|PDP^{\mathrm{T}}\right| + \left|PDP^{\mathrm{T}}\right| PDP^{\mathrm{T}}\\
&= PD\,|D|\,P^{\mathrm{T}} + PD\,|D|\,P^{\mathrm{T}} (\text{since } \left|PDP^{\mathrm{T}}\right| = P\,|D|\,P^{\mathrm{T}})\\
&= 2P\mathrm{diag}\,(\lambda_i\,|\lambda_i|)\,P^{\mathrm{T}} \in \mathcal{S}_+^n,
\end{aligned}$$

then for all $i = 1, 2, \ldots, n$, we have

$$\lambda_i\,|\lambda_i| \geq 0 \Rightarrow \lambda_i \geq 0, \quad \forall i = 1, 2, \ldots, n,$$

therefore, $X+S \in \mathcal{S}_+^n$ which implies that $\min(0, X+S) = 0$, *i.e*: the eigenvalues of $X+S$ are all nonnegative
Next, since $\phi_\alpha(X, S) = 0$ we get

$$XS + SX = 0.$$

So, we have shown that

$$XS + SX = 0 \quad \text{and} \quad X+S \in \mathcal{S}_+^n.$$

Now, it remains to be shown that

$$X \in \mathcal{S}_+^n, S \in \mathcal{S}_+^n, \quad \text{and} \quad \langle X, S\rangle = \mathrm{Tr}(XS) = 0. \tag{3.4}$$

The proof of (3.4) can be given by two methods:
Method 1:
Since $XS + SX = 0$ and $X+S \in \mathcal{S}_+^n$, then:

$$XS + SX = 0 \Rightarrow (X+S)^2 = X^2 + S^2,$$

so

$$X + S = \sqrt{X^2 + S^2}, \quad (|X + S| = X + S, \text{ because } X + S \in \mathcal{S}_+^n),$$

hence

$$X + S - \sqrt{X^2 + S^2} = 0 \Leftrightarrow \phi_{\mathrm{FB}}(X, S) = 0.$$

It is known that ϕ_{FB} is an NCP function [2], then

$$X \in \mathcal{S}_+^n, S \in \mathcal{S}_+^n \quad \text{and} \quad \langle X, S \rangle = \mathrm{Tr}(XS) = 0.$$

Method 2:
(1) Let $X = LD_X L^{\mathrm{T}}$ be the spectral decomposition of X, where L is an orthogonal matrix and $D_X = \mathrm{diag}(\lambda_1(X), \ldots \lambda_n(X))$.
Since $XS + SX = 0$ and $X + S \in \mathcal{S}_+^n$, we get

$$LD_X L^{\mathrm{T}} S + SLD_X L^{\mathrm{T}} = 0 \quad \text{and} \quad LD_X L^{\mathrm{T}} + S \in \mathcal{S}_+^n, \tag{3.5}$$

next, since $L\,L^{\mathrm{T}} = L^{\mathrm{T}}\,L = I_n$, it follows from relations (3.5) that

$$D_X L^{\mathrm{T}} SL + L^{\mathrm{T}} SLD_X = 0 \quad \text{and} \quad D_X + L^{\mathrm{T}} SL \in \mathcal{S}_+^n.$$

Set $B = (b_{ij}) = L^{\mathrm{T}} SL$. Since $D_X B + BD_X = 0$, it follows that

$$\left(\lambda_i(X) + \lambda_j(X)\right) b_{ij} = 0 \quad \text{for all } i, j = 1, 2, \ldots, n,$$

in particular if we take $i = j$, we obtain

$$2\lambda_i(X)\, b_{ii} = 0, \quad \text{for all } i = 1, 2, \ldots, n.$$

Next, since $D_X + B \in \mathcal{S}_+^n$ then it follows that

$$\lambda_i(X) + b_{ii} \geq 0, \quad \text{for all } i = 1, 2, \ldots, n.$$

so, for all $i = 1, 2, \ldots, n$, we have

$$2\lambda_i(X)\, b_{ii} = 0 \quad \text{and} \quad \lambda_i(X) + b_{ii} \geq 0.$$

Obviously, this implies

$$\lambda_i(X) \geq 0, \quad \text{for all } i = 1, 2, \ldots, n,$$

then X is positive semidefinite matrix.
(2) In the same way, we prove that S is positive semidefinite matrix.
(3) We have $\mathrm{Tr}(XS) = \frac{1}{2}\mathrm{Tr}(XS + SX)$, then $\mathrm{Tr}(XS) = 0$.

Therefore, we obtain

$$\phi_\alpha(X,S)=0 \Rightarrow X\in\mathcal{S}_+^n,\ S\in\mathcal{S}_+^n,\ \text{et}\ \langle X,S\rangle=\mathrm{Tr}(XS)=0.$$

□

Proposition 3.2 (Differentiability of the function ϕ_α) *The function ϕ_α defined by (1.9) is continuously differentiable everywhere and its derivative is given by*:

$$\begin{aligned}\phi'_\alpha(X,S)(U,V)&=US+SU+XV+VX\\&\quad-\frac{1}{\alpha}P\left[f^{[1]}(D)\circ\left(P(U+V)P^{\mathrm{T}}\right)\right]P^{\mathrm{T}}.\end{aligned}\tag{3.6}$$

for all $U,V\in\mathcal{S}^n$, where $X+S=PDP^{\mathrm{T}}$ is the spectral decomposition of $X+S$ and:

$$\left[f^{[1]}(D)\right]_{ij}=\begin{cases}\frac{\min^2(0,\lambda_i)-\min^2(0,\lambda_j)}{\lambda_i-\lambda_j} & \text{if } i\neq j\in\{1,\dots,n\}\\ 2\min(0,\lambda_i) & \text{if } i=j\in\{1,\dots,n\}\end{cases}.\tag{3.7}$$

Proof From formula (1.9), we have

$$\phi_\alpha(X,S)=\psi_1(X,S)-\frac{1}{\alpha}\psi_2(X,S),$$

where $\psi_1(X,S)=XS+SX$ and

$$\psi_2(X,S)=\frac{1}{\alpha}P\mathrm{diag}(f(\lambda_1),\dots f(\lambda_n))P^{\mathrm{T}},$$

with $f(\cdot)=\min^2(0,\cdot)$.

It is clear that ψ_1 is continuously differentiable everywhere. Now, by using the Proposition 2.3 and the fact that f is continuously differentiable everywhere, then the function ψ_2 is continuously differentiable everywhere. Hence ϕ_α is continuously differentiable since it is the sum of two continuously differentiable functions. Now, we calculate the derivative of ϕ_α

(a) It is clear that

$$\psi'_1(X,S)(U,V)=US+SU+XV+VX,$$

for all $U,V\in\mathcal{S}^n$.

(b) We compute the derivative of ψ_2 in the following way

$$\begin{aligned}\psi_2(X,S)&=[\min(0,X+S)]^2\\&=P\mathrm{diag}(f(\lambda_1),\dots f(\lambda_n))P^{\mathrm{T}}.\end{aligned}$$

We put

$$\psi_2(X, S) = (\theta \circ \Sigma)(X, S),$$

where

$$\begin{aligned} \Sigma : \mathcal{S}^n \times \mathcal{S}^n &\to \mathcal{S}^n \\ (X, S) &\to \Sigma(X, S) = X + S, \end{aligned}$$

and

$$\begin{aligned} \theta : \mathcal{S}^n &\to \mathcal{S}^n \\ Z &\to \theta(Z) = [\min(0, Z)]^2, \end{aligned}$$

then:

$$\psi_2'(X, S)(\cdot, \cdot) = \theta'(\Sigma(X, S)) \circ \left(\Sigma'(X, S)(\cdot, \cdot)\right).$$

It is clear that the derivative of Σ at (X, S) is

$$\begin{aligned} \Sigma'(X, S) : \mathcal{S}^n \times \mathcal{S}^n &\to \mathcal{S}^n \\ (U, V) &\to \Sigma'(X, S)(U, V) = U + V, \end{aligned}$$

for the derivative of $\theta(Z)$:

$$\begin{aligned} \theta'(Z) : \mathcal{S}^n &\to \mathcal{S}^n \\ K &\to \theta'(Z)(K), \end{aligned}$$

we have

$$\begin{aligned} \theta(Z) = [\min(0, Z)]^2 &= P\operatorname{diag}(f(\lambda_1(Z)), \ldots, f(\lambda_n(Z))) \\ &= P\operatorname{diag}\left(\min(0, \lambda_1(Z))^2, \ldots, \min(0, \lambda_n(Z))^2\right) P^{\mathrm{T}}, \end{aligned}$$

where $Z = P D_Z P^{\mathrm{T}}$ and $D_Z = \operatorname{diag}(\lambda_1(Z), \ldots, \lambda_n(Z))$, $i = 1, \ldots, n$. According to Proposition 2.3, the derivate of θ is given by

$$\theta'(Z)(K) = P\left[f^{[1]}(D_Z) \circ \left(P K P^{\mathrm{T}}\right)\right] P^{\mathrm{T}}.$$

Next, using the chain rule formula, we obtain

$$\psi_2'(X, S)(\cdot, \cdot) = \theta'(\Sigma(X, S)) \circ \left(\Sigma'(X, S)(\cdot, \cdot)\right),$$

that is

$$\psi_2'(X,S)(U,V) = \theta'(\overbrace{X+S}^{Z},\overbrace{U+V}^{K}).$$

Hence, the derivative of ψ_2 is given by

$$\psi_2'(X,S)(U,V) = P\left[f^{[1]}(D) \circ \left(P(U+V)P^{\mathrm{T}}\right)\right]P^{\mathrm{T}},$$

where $X+S = PDP$ is the spectral decomposition of $X+S$ and

$$\left[f^{[1]}(D)\right]_{ij} = \begin{cases} \frac{\min^2(0,\lambda_i)-\min^2(0,\lambda_j)}{\lambda_i-\lambda_j}, & \text{if } i \neq j \in \{1,\ldots,n\} \\ 2\min(0,\lambda_i) & \text{if } i = j \in \{1,\ldots,n\}. \end{cases}$$

Finally, the derivate of the function ϕ_α is given by

$$\begin{aligned}\phi_\alpha'(X,S)(U,V) = {} & US + SU + XV + VX \\ & -\frac{1}{\alpha}P\left[f^{[1]}(D) \circ \left(P(U+V)P^{\mathrm{T}}\right)\right]P^{\mathrm{T}}\end{aligned}$$

for all $U, V \in \mathcal{S}^n$. □

Remark 3.3 We can give the matrix function $f^{[1]}(D)$ by the form

$$\left[f^{[1]}(D)\right]_{ij} = \begin{cases} 0 & \text{if } \lambda_i \geq 0 \text{ and } \lambda_j \geq 0 \\ \frac{-\lambda_j^2}{\lambda_i-\lambda_j} & \text{if } \lambda_i > 0 \text{ and } \lambda_j < 0 \\ \frac{\lambda_i^2}{\lambda_i-\lambda_j} & \text{if } \lambda_i < 0 \text{ and } \lambda_j > 0 \\ \lambda_i+\lambda_j & \text{if } \lambda_i < 0 \text{ and } \lambda_j < 0, \end{cases} \quad \forall i,j \in \{1,2,\ldots,n\},$$

then we obtain

$$\left[f^{[1]}(D)\right]_{ij} = \begin{cases} = 0 & \text{if } X+S \succeq 0 (\lambda_i \geq 0 \text{ and } \lambda_j \geq 0) \\ < 0 & \text{if not } \left(\text{i.e. } \lambda_i < 0 \text{ or } \lambda_j < 0\right) \end{cases}.$$

4 On the Applicability of Newton's Method for Solving the SDNCP

Solving the SDNCP(F) problem comes back to solving the smooth equation $H_\alpha(X,S) = 0$, where

$$H_\alpha(X,S) = \begin{pmatrix} \phi_\alpha(X,S) \\ F(X) - S \end{pmatrix}, \tag{4.1}$$

and $\phi_\alpha(X,S)$ is defined by (1.9).

Since the functions ϕ_α and F are continuously differentiable then H_α is a continuously differentiable function. Moreover, based on Proposition 3.2, the derivative (or Jacobian) operator $H'_\alpha(X, S)$ of H_α at (X, S) is given by:

$$H'_\alpha(X,S)(U,V) = \begin{pmatrix} \begin{pmatrix} US + SU + XV + VX \\ -\frac{1}{\alpha} P\left[f^{[1]}(D) \circ \left(P(U+V)P^{\mathrm{T}}\right)\right] P^{\mathrm{T}} \end{pmatrix} \\ \left(F'(X)U - V\right) \end{pmatrix}, \quad (4.2)$$

for all $U, V \in \mathcal{S}^n$, where $X + S = PDP^{\mathrm{T}}$ is the spectral decomposition of $X + S$.

To solve the equation $H_\alpha(X, S) = 0$ by Newton's method, we must study the invertibility property of $H'_\alpha(X, S)$ at all (X, S) in $\mathcal{S}^n \times \mathcal{S}^n$ near the solution. We need the following lemma:

Lemma 4.1 (i) *The operator of Lyapunov $L_A : \mathcal{S}^n \to \mathcal{S}^n$, defined by*

$$L_A(X) = XA + AX,$$

is monotone (resp. strongly monotone) operator if $A \succeq 0$ (resp. $A \succ 0$).

(ii) *If $X, S \succ 0$, then L_X, L_S and $L_S \circ L_X$ are strongly monotone and L_X and L_S are self-adjoint.*

(iii) *Let $B : \mathcal{S}^n \to \mathcal{S}^n$ be the operator defined by*

$$B(U) = -\frac{1}{\alpha} P\left[f^{[1]}(D) \circ \left(P(U)P^{\mathrm{T}}\right)\right] P^{\mathrm{T}},$$

where $X + S = PDP^{\mathrm{T}}$ is the spectral decomposition of $X + S$. Then $B = 0$ if $X + S \succeq 0$, and B is strongly monotone operator if not.

Proof For the proof of (i) and (ii) see Lemma 4.2 in [2].
(iii) According to Remark 3.3, it follows that

(1) If $X + S \succeq 0$, then

$$\left[f^{[1]}(D)\right]_{ij} = 0,$$

for all $i, j = 1, 2, \ldots, n$. So,

$$B = 0.$$

(2) If $X + S \not\succeq 0$, then

$$\left[f^{[1]}(D)\right]_{ij} < 0,$$

for all $i, j = 1, 2, \ldots, n$.

We have

$$B \text{ is strongly monotone } \Leftrightarrow \langle U, B(U)\rangle > 0,$$

so, for all $U \in \mathcal{S}^n$ $(U \neq 0)$, we have

$$\begin{aligned}\langle U, B(U)\rangle &= \left\langle U, -\frac{1}{\alpha} P\left[f^{[1]}(D) \circ \left(PUP^{\mathrm{T}}\right)\right] P^{\mathrm{T}}\right\rangle \\ &= \mathrm{Tr}\left(U\left(-\frac{1}{\alpha} P\left[f^{[1]}(D) \circ \left(PUP^{\mathrm{T}}\right)\right] P^{\mathrm{T}}\right)\right) \\ &= -\frac{1}{\alpha}\mathrm{Tr}\left(UP\left[f^{[1]}(D) \circ \left(PUP^{\mathrm{T}}\right)\right] P^{\mathrm{T}}\right).\end{aligned}$$

Now, since

$$\mathrm{Tr}(X) = \mathrm{Tr}(P^{\mathrm{T}} X P), \quad \forall P \in \mathcal{O}^n,$$

then

$$\begin{aligned}\langle U, B(U)\rangle &= -\frac{1}{\alpha}\mathrm{Tr}\left(P^{\mathrm{T}}\left(UP\left[f^{[1]}(D) \circ \left(PUP^{\mathrm{T}}\right)\right] P^{\mathrm{T}}\right) P\right) \\ &= -\frac{1}{\alpha}\mathrm{Tr}\left(\left(P^{\mathrm{T}}UP\right) f^{[1]}(D) \circ \left(PUP^{\mathrm{T}}\right)\right).\end{aligned}$$

From the commutativity of the Hadamard product, we get

$$\begin{aligned}\langle U, B(U)\rangle &= -\frac{1}{\alpha}\mathrm{Tr}\left(\left(P^{\mathrm{T}}UP\right)^2 \circ f^{[1]}(D)\right) \\ &= -\frac{1}{\alpha}\sum_{i,j=1}^{n}\left(\left(P^{\mathrm{T}}UP\right)^2_{ij}\left(f^{[1]}(D)\right)_{ij}\right) > 0.\end{aligned}$$

Then, the operator B is strongly monotone. □

Proposition 4.2 (Invertibility of the derivative operator $H'_\alpha(X, S)$ for the monotone case) *Suppose that F is a monotone function.*

Then for all $S \succ 0$ and $X \succ 0$, the derivative operator $H'_\alpha(X, S)$ defined by formula (4.2) is strongly monotone (so, invertible).

Proof $H'_\alpha(X, S)$ is invertible operator $\Leftrightarrow \left[H'_\alpha(X, S)(U, V) = 0 \Rightarrow (U, V) = (0, 0)\right]$

$$H'_{\alpha}(X,S)(U,V)=0 \Leftrightarrow \begin{pmatrix} \begin{pmatrix} US+SU+XV+VX \\ -\frac{1}{\alpha}P\left[f^{[1]}(D)\circ\left(P(U+V)P^{\mathrm{T}}\right)\right]P^{\mathrm{T}} \end{pmatrix} \\ \left[F'(X)U-V\right] \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} L_S(U)+L_X(V)+B(U)+B(V) \\ F'(X)U-V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} L_S(U)+L_X(V)+B(U)+B(V)=0 \\ V=F'(X)U. \end{cases}$$

we replace $V=F'(X)U$ in the first equation:

$$L_S(U)+L_X\left(F'(X)U\right)+B(U)+B\left(F'(X)U\right)=0$$

$$\Leftrightarrow L_S(U)+\left(L_X\circ F'(X)\right)(U)+B(U)+\left(B\circ F'(X)\right)(U)=0$$

$$\Leftrightarrow \overbrace{\left(L_S+\left(L_X\circ F'(X)\right)+B+\left(B\circ F'(X)\right)\right)}^{A}(U)=0.$$

To show that the operator $H'_{\alpha}(X,S)$ is strongly monotone (invertible), just show that the operator A is strongly monotone.
We have $S\succ 0$ and $X\succ 0$, then $S+X\succ 0$. So, by Lemma 4.1 we have

$$B(U)=0 \quad \text{and} \quad \left(B\circ F'(X)\right)(U)=0,$$

hence,

$$A=L_S+L_X\circ F'(X),$$

since L_X is strongly monotone, we put $D=L_X^{-1}(U)$, then $U=L_X(D)$

$$\begin{aligned} A(U) &= (L_S+L_X\circ F'(X))(U) \\ &= (L_S\circ L_X+L_X\circ F'(X)\circ L_X)(D) \\ &= L_S\circ L_X(D)+L_X\circ F'(X)\circ L_X(D). \end{aligned}$$

Now, by using Lemma 4.1 (property (i) and (ii)) we have L_X, L_S and $L_S\circ L_X$ are strongly monotone and L_X and L_S are self-adjoint.
To show that A is strongly monotone we need to show that $L_X\circ F'(X)\circ L_X)$ is at least monotone. Then for all $D\in S^n$ $(D\neq 0)$, we have

$$\begin{aligned} \langle L_X\circ F'(X)\circ L_X)(D),(D)\rangle &= \langle F'(X)\circ L_X)(D),L_X(D)\rangle, \\ &\quad \text{(since } L_X \text{ is self-adjoint)} \\ &= \langle F'(X)(L_X(D)),L_X(D)\rangle. \end{aligned}$$

By the fact that $F'(X)$ is a monotone operator, we have

$$\langle F'(X)(L_X(D)), L_X(D)\rangle \geq 0,$$

then, $L_X \circ F'(X) \circ L_X$ is monotone.
So, A is strongly monotone (sum of strongly monotone and monotone operator), so it is invertible. Consequently, the operator $H'_\alpha(X, S)$ is strongly monotone, so it is invertible. □

Remark 4.3 If $S \not\succ 0$ or $X \not\succ 0$, the derivative operator $H'_\alpha(X, S)$ may be not invertible. The following counter-example shows it.
Let $F : \mathcal{S}^2 \to \mathcal{S}^2$ be the function defined by

$$F(X) = \begin{pmatrix} 2x_{11} & x_{12} \\ x_{12} & x_{22}^3 + x_{22} \end{pmatrix}.$$

By using theorem 2.5 in [3], we have

$$F'(X)U = \begin{pmatrix} 2u_{11} & u_{12} \\ u_{12} & \left(3x_{22}^2 + 1\right) u_{22} \end{pmatrix}, \quad \text{for all } U = \begin{pmatrix} u_{11} & u_{12} \\ u_{12} & u_{22} \end{pmatrix} \in \mathcal{S}^2.$$

It is clear that F is a strongly monotone function since $F'(X)$ is a strongly monotone operator that is

$\left\langle F'(X)U, U\right\rangle > 0$, for all $U \in \mathcal{S}^2$.

Let $X_0 = \begin{pmatrix} -1 & 0 \\ 0 & 2 \end{pmatrix}$ and $S_0 = \begin{pmatrix} 2 & 0 \\ 0 & -3 \end{pmatrix}$ which are indefined matrices.

$X_0 + S_0 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = PDP^{\mathrm{T}}$, where $P = I_2$ and $D = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

Let $(U, V) = \left(\begin{pmatrix} u_{11} & u_{12} \\ u_{12} & u_{22} \end{pmatrix}, \begin{pmatrix} v_{11} & v_{12} \\ v_{12} & v_{22} \end{pmatrix}\right)$, then

$$H'_\alpha(X_0, S_0)(U, V) = \begin{pmatrix} \begin{pmatrix} US + SU + XV + VX \\ -\frac{1}{\alpha} P\left[f^{[1]}(D) \circ \left(P(U+V)P^{\mathrm{T}}\right)\right] P^{\mathrm{T}} \end{pmatrix} \\ \left(F'(X)U - V\right) \end{pmatrix},$$

then for $\alpha = 1$,

$$H'_\alpha(X_0, S_0)(U, V) = \begin{pmatrix} \begin{pmatrix} 4u_{11} - 2v_{11} & 1.5v_{12} - 0.5u_{12} \\ 1.5v_{12} - 0.5u_{12} & 2v_{22} - 8u_{22} \end{pmatrix} \\ \begin{pmatrix} 2u_{11} - v_{11} & u_{12} - v_{12} \\ u_{12} - v_{12} & 13z - v_{22} \end{pmatrix} \end{pmatrix},$$

then

$$H'_\alpha(X_0, S_0)(U, V) = \overbrace{\begin{pmatrix} 4 & 0 & 0 & -2 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & -8 & 0 & 0 & 2 \\ 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 13 & 0 & 0 & -1 \end{pmatrix}}^{M} \begin{pmatrix} u_{11} \\ u_{12} \\ u_{22} \\ v_{11} \\ v_{12} \\ v_{22} \end{pmatrix}.$$

Since $\det(M) = 0$, then the operator $H'(X_0, S_0)$ is not invertible for X_0 and S_0.

Remark 4.4 Since we can not guarantee the invertibility of $H'_\alpha(X_k, S_k)$ at each iteration k, we can't apply Newton's method for solving the equation $H_\alpha(X, S) = 0$. To avoid the inversion and the computation of the Jacobian of H_α at each iteration, we propose to use a quasi-Newton method in Hilbertian space as defined in [1].

Recall that, quasi-Newton's method for solving nonlinear equation

$$G(Z) = 0,$$

where $G : \mathcal{X} \to \mathcal{Y}$ is a Frechet-differentiable function and $\mathcal{X}$,$\mathcal{Y}$ are two Hilbert spaces, is defined by:

$$Z_{k+1} = Z_k - B_k^{-1}(G(Z_k)),$$

where the operators B_k are invertibles and approximate $G'(Z_k)$ at each iteration k.

In this paper, we use the Broyden's method [1], to update B_k at each iteration, where the update formula is:

$$B_{k+1} = B_k + \frac{(Y_k - B_k(W_k)) \otimes W_k}{\langle W_k, W_k \rangle},$$

where $W_k = Z_{k+1} - Z_k$, $Y_k = G(Z_{k+1}) - G(Z_k)$ and $\otimes$ is the dyadic product operator defined by

$$(A \otimes C)(\cdot) = \langle C, (\cdot) \rangle A, \quad A \otimes C \in L(\mathcal{X}, \mathcal{Y}). \tag{4.3}$$

Remark 4.5 The notation $\otimes$ is used in the literature for the Kronecker product, but here it is used for the dyadic (or tensor) operator defined by (4.3).

Note also that for all $A, B \in \mathcal{S}^n$, the dyadic operator $A \otimes C$ is a linear operator, but the Kronecker product of matrices A and B, (Kron(A, B) in Matlab notation) is a matrix of order $n^2 \times n^2$. The relation between the dyadic operator and the Kronecker product is given in [1, Prop. 13].

In the space of symmetric matrices ($\mathcal{X} = \mathcal{Y} = \mathcal{S}^n$), the dyadic operator is given by

$$(A \otimes C)(M) = \langle C, M \rangle A = \mathrm{Tr}(CM)A, \quad \forall M \in \mathcal{S}^n$$

The following theorem shows that Broyden's method converges superlinearly to the solution of the equation of $G(Z) = 0$.

Theorem 4.6 (See [1] and references therein) *Let $G : \mathcal{X} \to \mathcal{Y}$ a continuously differentiable function in $D \subset \mathcal{X}$, where D is a convex, open set. Assume that $Z^* \in D$ is a zero point of G that $G'(Z^*) \in L(\mathcal{X}, \mathcal{Y})$ is inversible and that $G'(\cdot)$ satisfies the Lipchitz condition*:

$$\left\| G(Z) - G'(Z^*) \right\| \leq L \left\| Z - Z^* \right\|,$$

for each $Z \in D$.
Then, if $Z_0 \in D$ and $B_0 \in L(\mathcal{X},\mathcal{Y})$ are near Z^* and $G'(Z^*)$ respectively, we have:
(1) *the sequence $\{Z_k\}$ defined by the Broyden's method is well-defined and converges superlinearly to the solution Z^*.*
(2) *Furthermore, B_k^{-1} exists for each k and the sequences $\{\|B_k\|\}$ and $\left\{ \left\| B_k^{-1} \right\| \right\}$ are bounded.*

In practical implementation, we use the Broyden's method in the form

$$\begin{cases} B_k (W_k) = -G (Z_k) \\ Z_{k+1} = Z_k + W_k. \end{cases}$$

Now, we apply the Broyden's method to solve the equation $H_\alpha (X, S) = 0$ defined by (4.1). This method is defined by

$$\begin{cases} B_k (U_k, V_k) = -H_\alpha (X_k, S_k) \\ (X_{k+1}, S_{k+1}) = (X_k, S_k) + (U_k, V_k), \end{cases}$$

where B_k is updated by the following formula :

$$B_{k+1} = B_k + \frac{(Y_k - B_k (W_k)) \otimes W_k}{\langle W_k, W_k \rangle},$$

where $W_k = (X_{k+1}, S_{k+1}) - (X_k, S_k)$ and $Y_k = H_\alpha (X_{k+1}, S_{k+1}) - H_\alpha (X_k, S_k)$.

Lemma 4.7 *The operator B_k can be define by the formula*

$$B_{k+1} (\cdot, \cdot) = B_k (\cdot, \cdot) + \beta_k (\langle U_k, \cdot \rangle + \langle V_k, \cdot \rangle) (\phi_\alpha (X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}), \tag{4.4}$$

where $\beta_k = \frac{1}{\langle U_k, U_k \rangle + \langle V_k, V_k \rangle}$.

Proof We have

$$B_{k+1} = B_k + \frac{(Y_k - B_k (W_k)) \otimes W_k}{\langle W_k, W_k \rangle},$$

and $B_k(W_k) = -H_\alpha(X_k, S_k)$, then

$$B_{k+1} = B_k + \frac{(Y_k + H_\alpha(X_k, S_k)) \otimes W_k}{\langle W_k, W_k \rangle},$$

where $Y_k = H_\alpha(X_{k+1}, S_{k+1}) - H_\alpha(X_k, S_k)$ and $W_k = (U_k, V_k)$.
Hence, we obtain

$$B_{k+1} = B_k + \frac{H_\alpha(X_{k+1}, S_{k+1}) \otimes (U_k, V_k)}{\langle U_k, U_k \rangle + \langle V_k, V_k \rangle},$$

but $H_\alpha(X_{k+1}, S_{k+1}) = (\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1})$, then

$$B_{k+1} = B_k + \frac{1}{\langle U_k, U_k \rangle + \langle V_k, V_k \rangle} (\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}) \otimes (U_k, V_k).$$

Let $(U, V) \in \mathcal{S}^n \times \mathcal{S}^n$, by the definition of the dyadic product operator we have

$$\begin{aligned}&((\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}) \otimes (U_k, V_k))(U, V)\\ &= \langle (U_k, V_k), (U, V) \rangle (\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}),\end{aligned}$$

then $B_{k+1}(U, V) = B_k(U, V) + \frac{\langle U_k, U \rangle + \langle V_k, V \rangle}{\langle U_k, U_k \rangle + \langle V_k, V_k \rangle} \big(\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}\big)$,
we put $\beta = \frac{1}{\langle U_k, U_k \rangle + \langle V_k, V_k \rangle}$, then

$B_{k+1}(U, V) = B_k(U, V) + \beta(\langle U_k, U \rangle + \langle V_k, V \rangle) \big(\phi_\alpha(X_{k+1}, S_{k+1}), F(X_{k+1}) - S_{k+1}\big)$. □

The following algorithm represents the quasi-Newton method applied to the equation $H_\alpha(X, S) = 0$.

Algorithm 1: (Quasi-Newton's method for SDNLCP(F))

Step 1: *Initialization*
Input a parameter $\alpha \in (0, 1]$ and a tolerance $\varepsilon > 0$. Set $k = 0$.
Initialize $X_0, S_0 \in \mathcal{S}^n$ and $B_0 \in L\left(\mathcal{S}^n \times \mathcal{S}^n\right)$ be invertible.
Step 2: *Stopping criteria*
If $\|H_\alpha(X_k, S_k)\|_F < \varepsilon$, stop. Otherwise go to Step 3.
Step 3: *Compute the matrices* $U_k, V_k \in \mathcal{S}^n$
Find the solution $(U_k, V_k) \in \mathcal{S}^n$ of the linear system:
$B_k(U_k, V_k) = -H_\alpha(X_k, S_k)$.
Step 4: *Updating formula.*
Compute $B_{k+1} \in L\left(\mathcal{S}^n \times \mathcal{S}^n\right)$ by using formula (4.4).
Set $\left(X_{k+1}, S_{k+1}\right) = (X_k, S_k) + \left(U^k, V^k\right)$.
Set $k = k + 1$ and go to Step 2.

Remark 4.8 Since we must choose the operator B_0 invertible (to guarantee that B_k is invertible for each iteration k), one possible choice is $B_0 = I$ (the identity operator in $L(\mathcal{S}^n \times \mathcal{S}^n)$) or $B_0 = H'_\alpha(X_0, S_0)$ for $S_0 \succ 0$ and $X_0 \succ 0$ (cf. Proposition 4.2).

Now, we will show the convergence of the Broyden's method for the equation $H_\alpha(X, S) = 0$ defined by (4.1).

Theorem 4.9 (The convergence of Broyden's method for the function H_α) *Let $H_\alpha : \mathcal{S}^n \times \mathcal{S}^n \to \mathcal{S}^n \times \mathcal{S}^n$ be the function defined by (4.1). Suppose that Eq. (4.1) admits a strict solution denoted by (X^*, S^*), i.e. $(X^* \succ 0, S^* \succ 0$ and $H_\alpha(X^*, S^*) = 0)$. If $(X_0, S_0) \in \mathcal{S}^n \times \mathcal{S}^n$ and $B_0 \in L(\mathcal{S}^n \times \mathcal{S}^n)$ are chosen near (X^*, S^*) and $H'_\alpha(X^*, S^*)$ respectively such that B_0 is invertible, then*:

(1) *the sequence $\{(X_k, S_k)\}$ defined by the Broyden's method is well-defined and converges super-linearly to the solution (X^*, S^*).*
(2) *the operator B_k^{-1} exists for all $k \geq 0$, and the sequences $\{\|B_k\|\}$ and $\left\{\left\|B_k^{-1}\right\|\right\}$ are bounded.*

Proof The proof is based on Theorem 4.6. We have

(1) $H'_\alpha(X^*, S^*) \in L(\mathcal{S}^n \times \mathcal{S}^n)$ is an invertible operator, since $H'_\alpha(X^*, S^*)$ is invertible for any $X^* \succ 0$ and $S^* \succ 0$ (Proposition 4.2).

(2) $H'_\alpha(X, S)$ is a Lipchitz operator for all $(X, S) \in \mathcal{S}^n \times \mathcal{S}^n$, since $H'_\alpha(X, S)$ is a Lipchitz operator everywhere. □

4.1 Smoothing Newton's Method for Solving SDNCP(F)

In this subsection, we develop the smoothing Newton's method for solving SDNCP(F) problem.

Solving the SDNCP(F) problem by smoothing Newton's method comes back to solving the smooth equation $H_\mu(X, S) = 0$ where $H_\mu : \mathcal{S}^n \times \mathcal{S}^n \times \mathbb{R} \to \mathcal{S}^n \times \mathcal{S}^n$ is defined by

$$H_\mu(X, S) = \begin{pmatrix} \phi^\mu_{\mathrm{FB}}(X, S) \\ F(X) - S \end{pmatrix} \tag{4.5}$$

$$= \begin{pmatrix} X + S - \sqrt{X^2 + S^2 + 2\mu^2 I} \\ F(X) - S \end{pmatrix}. \tag{4.6}$$

Smoothing Newton's method applied to the equation $H_\mu(X, S) = 0$ is defined by

$$(X_{k+1}, S_{k+1}) = (X_k, S_k) - \left(H'_{\mu_k}(X_k, S_k)\right)^{-1} \left(H_{\mu_k}(X_k, S_k)\right)$$

where $H'_{\mu_k}(X_k, S_k)$ is the derivative operator of H_{μ_k} at (X_k, S_k).

In practical implementation, we use the smoothing Newton's method in the form

$$\begin{cases} H'_{\mu_k}(X_k, S_k)(U_k, V_k) = -H_{\mu_k}(X_k, S_k) \\ (X_{k+1}, S_{k+1}) = (X_k, S_k) + (U_k, V_k). \end{cases} \tag{4.7}$$

So, at each iteration k of the smoothing Newton's method we need to solve the system of linear equations defined in (4.7), where (μ_k) is a decreasing sequence of positive numbers that tends to 0.
From [13, Lemma 2(c)], The derivative of ϕ^{μ}_{FB} defined by (1.7) is given by

$$\left(\phi^{\mu}_{\mathrm{FB}}\right)'(X,S)(U,V) = U + V - L_C^{-1}[XU + UX + SV + VS] \tag{4.8}$$

where $C = \sqrt{X^2 + S^2 + 2\mu^2 I}$ and L_C is a Lyapunov operator associated to C defined by

$$L_C(X) = CX + XC$$

Consequently the derivative of H_μ is given by

$$\begin{aligned} H'_\mu(X,S)(U,V) &= \begin{pmatrix} \left(\phi^{\mu}_{\mathrm{FB}}\right)'(X,S)(U,V) \\ F'(X)U - V \end{pmatrix} \\ &= \begin{pmatrix} U + V - L_C^{-1}[XU + UX + SV + VS]. \\ F'(X)U - V \end{pmatrix}. \end{aligned} \tag{4.9}$$

In [13, Lemma 7], it is proved that if F is monotone then the operator $H'_\mu(X,S)$ is nonsingular for all $(X,S) \in \mathcal{S}^n \times \mathcal{S}^n$ and $\mu \in \mathbb{R}_{++}$.
Now, for each iteration, the system of linear equations defined in (4.7) equivalent to

$$U_k + V_k - L_{C_k}^{-1}[X_kU_k + U_kX_k + S_kV_k + V_kS_k] = -\phi^{\mu_k}_{\mathrm{FB}}(X_k,S_k) \tag{4.10}$$

and

$$F'(X_k)U_k - V_k = -(F(X_k) - S_k). \tag{4.11}$$

By applying L_{C_k} to (4.10), we obtain

$$L_{C_k}(U_k + V_k) - [X_kU_k + U_kX_k + S_kV_k + V_kS_k] = -L_{C_k}\left(\phi^{\mu_k}_{\mathrm{FB}}(X_k,S_k)\right)$$

then

$$L_{C_k - X_k}(U_k) + L_{C_k - S_k}(V_k) = -L_{C_k}\left(\phi^{\mu_k}_{\mathrm{FB}}(X_k,S_k)\right) \tag{4.12}$$

and by (4.11), we have

$$V_k = F'(X_k)U_k + F(X_k) - S_k \tag{4.13}$$

substituting (4.13) in (4.12), we have

$$\begin{aligned} &L_{C_k - X_k}(U_k) + L_{C_k - S_k}\left(F'(X_k)U_k\right) \\ &\quad = -L_{C_k}\left(\phi^{\mu_k}_{\mathrm{FB}}(X_k,S_k)\right) - L_{C_k - S_k}(F(X_k) - S_k). \end{aligned} \tag{4.14}$$

Algorithm 2: (Smoothing Newton's method for SDNLCP(F))

Step 1: *Initialization*
Input a tolerance $\varepsilon > 0$. Set $k = 0$.
Initialize $X_0, S_0 \in \mathcal{S}^n$ and $\mu_0 > 0$.
Step 2: *Stopping criteria*
If $\left\| H_{\mu_k}(X_k, S_k) \right\|_F < \varepsilon$, stop. Otherwise go to Step 3.
Step 3: *Compute* (U_k, V_k) *as follows:*
Find the solution $U_k \in S^n$ of linear system (4.14)
Compute the matrix $V_k \in S^n$ by the relation (4.13)
Step 4: *Updating formula.*
Set $\left(X_{k+1}, S_{k+1}\right) = (X_k, S_k) + \left(U^k, V^k\right)$.
Set $k = k + 1$ and go to Step 2.

5 Numerical Experiments

In this section, some numerical experiments are given to show the performance of Algorithm 1 and the Smoothing Newton's method (SNM for short) for solving SDNCP(F). We used a personal computer with 8.0 GB for random memory and Intel(R) core(TM) i7-4600M CPU 2.90 GHz to perform all numerical experiments. We used Windows 8 as operating system and Matlab R2017a to write the computer codes. For all examples, the stop criterion used in Algorithm 1 and SNM is $er_1 := \|H_\alpha(X, S)\|_F < \varepsilon = 10^{-10}$ and $er_2 := \|H_\mu(X, S)\|_F < 10^{-10}$ respectively or if the number of iterations is greater than 1000.

Example 5.1 Let $F : S^4 \to S^4$ be the function defined by

$$F(X) = \begin{pmatrix} 2x_{11} - 2 & x_{12} & x_{13} & x_{14} \\ x_{12} & x_{22}^3 + x_{22} - 2 & x_{23} & x_{24} \\ x_{13} & x_{23} & x_{33}^3 - 1 & x_{34} \\ x_{14} & x_{24} & x_{34} & x_{44} - 1 \end{pmatrix}, \quad \text{for all } X = (x_{ij}) \in S^4.$$

We can verify that the exact solution of the SDNCP(F) associated to this function is $(X^*, S^*) = (I_4, 0_4)$. Furthermore, for all $X = (x_{ij})$, $U = (u_{ij}) \in S^4$, we have

$$F'(X)U = \begin{pmatrix} 2u_{11} & u_{12} & u_{13} & u_{14} \\ u_{12} & (3x_{22}^2 + 1)\, u_{22} & u_{23} & u_{24} \\ u_{13} & u_{23} & 3x_{33}^2\, u_{33} & u_{34} \\ u_{14} & u_{24} & u_{34} & u_{44} \end{pmatrix}$$

Hence, for all $U \in S^4$ such that $U \neq 0_4$, we have

$$\begin{aligned} \langle < F'(X)U, U \rangle &= \mathrm{Tr}(F'(X)U.U) \\ &= 2\sum_{j=1}^{4} u_{1j}^2 + 2(u_{23}^2 + u_{24}^2 + u_{34}^2) + (3x_{22}^2 + 1)u_{22}^2 \\ &\quad + 3x_{33}^2 u_{33}^2 + u_{44}^2 > 0. \end{aligned}$$

Table 1 Iterations, CPU time and $\|H_\alpha(X_k, S_k)\|$ for different initial guesses

X_0	S_0	k (iterations)	CPU time	$er1(k)$
$1.5\, I_4$	$0.01 * \text{ones}(4)$	15	0.1203	2.553477e−10
$4 * I_4$	$2 * I_4$	41	0.110084	5.1362e−11
$10 * \text{hilb}(4)$	$0.01 * \text{hilb}(4)$	50	0.1746	8.3595e−12

Table 2 Iterations, CPU time and $er_2(k)$ for different initial guesses

X_0	S_0	k (iterations)	CPU time	$er2(k)$
$1.5\, I_4$	$0.01 * \text{ones}(4)$	9	0.0978	8.7635e−12
$4 * I_4$	$2 * I_4$	9	0.0245	8.7633e−12
$10 * \text{hilb}(4)$	$0.01 * \text{hilb}(4)$	9	0.0695	1.4394e−11

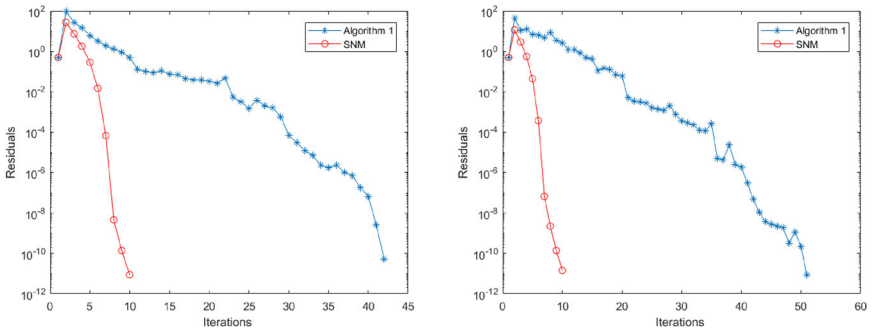
So, F is strongly monotone function since $F'(X)$ is strongly monotone operator.

We apply Algorithm 1 and SNM to solve the SDNCP(F), with $\alpha = 1$, $B_0 = H'(X_0, S_0)$, $\mu_k = (0.85)^k$, for different choices of initial guess (X_0, S_0). The number of iterations k , CPU time in seconds and the values $er1(k) := \|H_\alpha(X_k, S_k)\|$ and $er2(k) := \|H_{\mu_k}(X_k, S_k)\|$ are listed in Tables 1 and 2.

The values of approximate solutions (X_k, S_k) corresponding to different initial guesses (X_0, S_0) finded using Algorithm 1 are shown in Table 3 (Fig. 1).

Table 3 Approximate solution (X_k, S_k) for different initial guesses

X_0	S_0	k (iterations)	X_k	S_k
$1.5 * I_4$	$0.01 * \text{ones}(4)$	15	$I_4 + 10^{-12} D_1$	$10^{-12} Z_1$
$4 * I_4$	$2 * I_4$	41	$I_4 + 10^{-11} D_2$	$10^{-11} Z_2$
$10 * \text{hilb}(4)$	$0.01 * \text{hilb}(4)$	50	$I_4 + 10^{-11} D_3$	$10^{-11} Z_3$



(a) $(X_0, S_0) = (4 * I_8, 2 * I_4)$ (b) $(X_0, S_0) = (10 * hilb(4), 0.01 * hilb(4))$

Fig. 1 Convergence of Algorithms 1 and SNM for some choices of (X_0, S_0)

where

$$D_1 = \begin{pmatrix} 0.0184 & -0.1242 & 0.0267 & 0.0033 \\ -0.1242 & -0.0251 & -0.0424 & -0.1572 \\ 0.0267 & -0.0424 & 0.0386 & -0.0059 \\ 0.0033 & -0.1572 & -0.0059 & -0.0303 \end{pmatrix},$$

$$Z_1 = \begin{pmatrix} 0.0369 & -0.1242 & 0.0267 & 0.0033 \\ -0.1242 & -0.2131 & -0.0424 & -0.1572 \\ 0.0267 & -0.0424 & 0.1208 & -0.0059 \\ 0.0033 & -0.1572 & -0.0059 & -0.0303 \end{pmatrix};$$

$$D_2 = \begin{pmatrix} 0.291 & 0 & 0 & 0 \\ 0 & 0.1453 & 0 & 0 \\ 0 & 0 & -0.1158 & 0 \\ 0 & 0 & 0 & -0.0631 \end{pmatrix},$$

$$Z_2 = \begin{pmatrix} 0.0581 & 0 & 0 & 0 \\ 0 & -0.1726 & 0 & 0 \\ 0 & 0 & 0.1740 & 0 \\ 0 & 0 & 0 & -0.0631 \end{pmatrix};$$

$$D_3 = \begin{pmatrix} 0.0144 & -0.1070 & 0.0951 & 0.0106 \\ -0.1070 & -0.0256 & 0.0051 & -0.0339 \\ 0.0951 & 0.0051 & 0.0315 & 0.0316 \\ 0.0106 & -0.0339 & 0.0316 & 0.0037 \end{pmatrix},$$

$$Z_3 = \begin{pmatrix} 0.0288 & -0.1070 & 0.0951 & 0.0106 \\ -0.1070 & -0.1310 & 0.0051 & -0.0339 \\ 0.0951 & 0.0051 & 0.0714 & 0.0316 \\ 0.0106 & -0.0339 & 0.0316 & 0.0037 \end{pmatrix}.$$

Example 5.2 Let n be a positive integer and let $a_1, b_1, a_2, b_2, \ldots, a_n, b_n$ be a fixed real parameters such that

$$a_{2i} = 0, \quad a_{2i+1} > 0, \ b_{2i} > 0, \ \text{and } b_{2i+1} = 0.$$

We define a function $F : \mathcal{S}^n \to \mathcal{S}^n$ by

$$F(X) = \text{diag}(P_1(X), \ldots, P_n(X))$$

where for all $X = (x_{ij}) \in \mathcal{S}^n$,

$$P_i(X) = x_{ii}^2 - a_i\, x_{ii} + b_i, \quad i \in \{1, 2, \ldots, n\}.$$

By a simple verification, the exact solution of SDNCP(F) associated to function F is (X^*, S^*), where

$$X^* = \text{diag}(a_1, 0, a_3, 0, \ldots, a_n), \quad S^* = \text{diag}(0, b_2, 0, b_4, \ldots, b_n).$$

Using the definition of differentiability, we can prove that for all $X = (x_{ij})$, $U = (u_{ij}) \in \mathcal{S}^n$, we have

$$F'(X)U = \text{diag}((2x_{11} - a_1)\, u_{11}, \ldots, (2x_{nn} - a_n)\, u_{nn}).$$

Now, we take for example

$$a_{2i} = 0, \quad a_{2i+1} = \frac{2i+1}{2i+2}, \; b_{2i} = \frac{2i+1}{2i}, \text{ and } b_{2i+1} = 0.$$

We apply Algorithm 1 and SNM to solve the SDNCP(F), with $\alpha = 0.85$, $\mu_k = \frac{1}{2^{k+1}}$, $(X_0, S_0) = (I_n, I_n)$ (for both Algorithm 1 and SNM) and $B_0 = H'(X_0, S_0)$ for different values of dimension n. The number of iterations k , CPU time in seconds and the values $er_1(k) := \|H_\alpha(X_k, S_k)\|$ and $er_2(k) := \|H_\mu(X_k, S_k)\|$ are listed in Table 4. (Fig 2).
In the case $n = 8$, the approximate solution (X_{32}, S_{32}) is

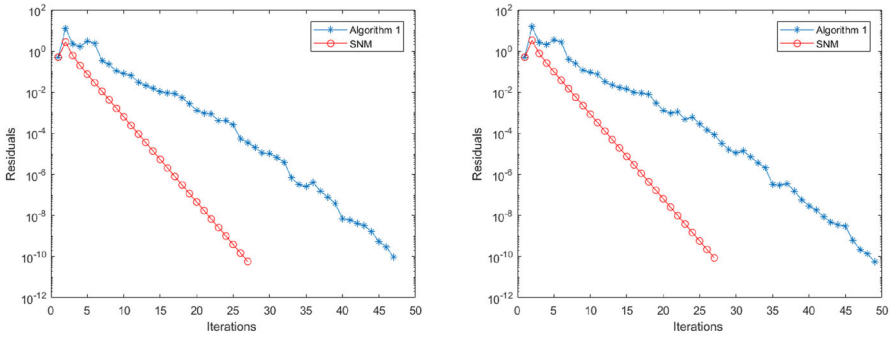
$$X_{32} = \text{diag}(5.0, -4.21 \times 10^{-14}, 0.750, 3.73 \times 10^{-13}, 0.83, \\ -7.28 \times 10^{-13}, 0.875, 6.78 \times 10^{-13});$$

and

$$S_{32} = \text{diag}(2.18 \times 10^{-13}, 1.50, -1.30 \times 10^{-12}, 1.25,$$

Table 4 Performance of Algorithm 1 and SNM for $n = 8, 20, 40, 60$

	Algorithm 1			SNM		
	k	CPU time (s)	$er_1(k)$	k	CPU time (s)	$er_2(k)$
$n = 8$	36	0.1941	5.4697e−11	23	0.0994	7.4825e−11
$n = 20$	46	2.22	7.8560e−11	25	1.46	6.2815e−11
$n = 40$	46	38.58	9.3522e−11	26	32.34	5.8101e−11
$n = 60$	48	353.16	5.5015e−11	28	289.11	8.7176e−11

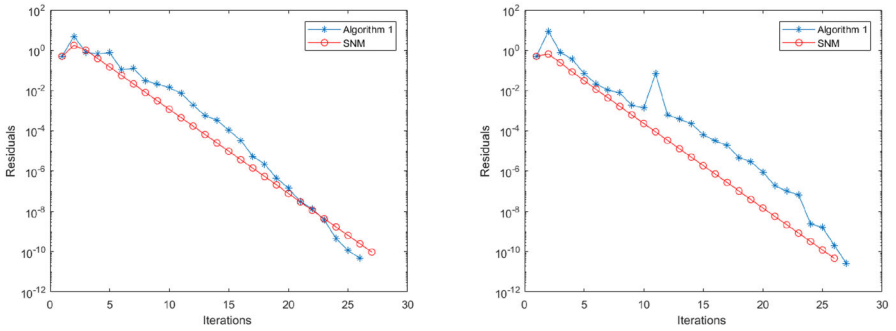


(a) $n = 20$ (b) $n = 60$

Fig. 2 Convergence of Algorithm 1 and SNM for $(X_0, S_0) = (I_n, I_n)$ where $n = 20, 60$

Table 5 Performance of Algorithm 1 and SNM for $\lambda = 0.55, 0.70, 1.25$

	Algorithm 1			SNM		
	k	CPU time (s)	$er_1(k)$	k	CPU time (s)	$er_2(k)$
$\lambda = 0.55$	36	7.95	$4.6963e-11$	27	8.31	$4.8189e-11$
$\lambda = 0.70$	25	5.60	$4.7850e-11$	26	8.05	$9.6569e-11$
$\lambda = 1.25$	26	5.78	$2.7463e-11$	25	7.56	$4.7523e-11$



(a) $\lambda = 0.70$ (b) $\lambda = 1.25$

Fig. 3 Convergence of Algorithms 1 and NM for $(X_0, S_0) = (\lambda * \text{diag}(a), I_{30})$ where $\lambda = 0.7, 1.25$

$$-2.87 \times 10^{-12}, 1.16, 5.46 \times 10^{-12}, 1.125).$$

Note that, in this case $(n = 8)$, the exact solution of SDNCP(F) is

$$X^* = \text{diag}\left(\frac{1}{2}, 0, \frac{3}{4}, 0, \frac{5}{6}, 0, \frac{7}{8}, 0\right), \quad S^* = \text{diag}\left(0, \frac{3}{2}, 0, \frac{5}{4}, 0, \frac{7}{6}, 0, \frac{9}{8}\right).$$

Now, we repeat the same experiment with fixed dimension $n = 30$ and $(X_0, S_0) = (\lambda * \text{diag}(a), I_n)$ where $\lambda \in \{0.55, 1, 1.25\}$, we get the results showing in Table 5 (Figs. 2, 3).

Example 5.3 Let $Q \in S_+^n$, $n \geq 1$. We define a function $F : \mathcal{S}^n \to \mathcal{S}^n$ by $F(X) = X^2 - Q$. It is clear that $(X^*, S^*) = (Q^{\frac{1}{2}}, 0_n)$ is the exact solution of SDNCP(F) associated to function F, where $Q^{\frac{1}{2}}$ is the square root of the matrix Q.
The function F is derivable on $\mathcal{S}^n$, and for all $(X, U) \in \mathcal{S}^n \times \mathcal{S}^n$, we have

$$F'(X)U = XU + UX.$$

According to Lemma 4.1, $F'(X)$ is strongly monotone. So F is also strongly monotone.
Now, we consider the following two choices of the matrix Q

$$Q_1 = \text{diag}(1, 2, \ldots, 10), \quad Q_2 = M^2;$$

where

$$M = \begin{pmatrix} 3&2&2&1&2&2&2&2&2&2\\ 2&2&2&1&2&2&2&2&2&2\\ 2&2&3&1&2&2&2&2&2&2\\ 1&1&1&1&1&1&1&1&1&1\\ 2&2&2&1&3&2&2&2&2&2\\ 2&2&2&1&2&3&2&2&2&2\\ 2&2&2&1&2&2&3&2&2&2\\ 2&2&2&1&2&2&2&3&2&2\\ 2&2&2&1&2&2&2&2&3&2\\ 2&2&2&1&2&2&2&2&2&3 \end{pmatrix} \in S_+^{10}.$$

We remark that when $Q = Q_1$, the exact solution of the SDNCP(F) is

$$(X^*, S^*) = (\text{diag}(1, \sqrt{2}, \sqrt{3}, \ldots, \sqrt{10}), 0_{10});$$

and when $Q = Q_2$, then the exact solution of the SDNCP(F) is $(X^*, S^*) = (M, 0_{10})$. We apply Algorithm 1 to solve the SDNCP(F), with $n = 10$, for the two choices $Q = Q_1$ and $Q = Q_2$ and different values of parameter $\alpha \in \Omega := \{0.10, 0.25, 0.50, 0.75, 1\}$.

First case : $Q = Q_1$. In the case, we take the initial guess $(X_0, S_0) = (2 * I_{10}, 0.1 * I_{10})$ and $B_0 = H'(X_0, S_0)$. We get the results listed in Table 6.

From Table 6, we remark that for all $\alpha \in \Omega$, we need 20 iterations to get desired approximate solution (X_{20}, S_{20}), where

$$\begin{aligned} X_{20} &= diag(1.0, 1.4142, 1.7321, 2.0, 2.2361, 2.4495, 2.6458, 2.8284, 3.0, 3.1623),\\ S_{20} &= 10^{-10}\, diag(-0.0533, 0.1041, -0.0285, 0.0153,\\ &\quad -0.0165, 0.0244, -0.0039, -0, 0, -0). \end{aligned}$$

However, the optimal value of α from Ω, which makes Algorithm more faster is 0.5.

Second case : $Q = Q_2$. In the case, we take the initial guess $(X_0, S_0) = (2 * M, 0.1 * I_{10})$ and $B_0 = H'(X_0, S_0)$.

Table 6 Iterations, CPU time and $\|H_\alpha(X_k, S_k)\|$ for different values of α

alpha	k (iterations)	CPU time	$er_1(k)$
0.10	20	0.456582	$2.219155e-10$
0.25	20	0.184321	$2.219155e-10$
0.50	20	0.146197	$2.219155e-10$
0.75	20	0.163383	$2.219155e-10$
1.00	20	0.171410	$2.219155e-10$

Table 7 Iterations, CPU time and $\|H_\alpha(X_k, S_k)\|$ for different values ofα.

alpha	k (iterations)	CPU time	$\|H_\alpha(X_k, S_k)\|$
0.10	24	0.231196	$6.714579e-10$
0.25	24	0.180104	$6.714579e-10$
0.50	24	0.209106	$6.714579e-10$
0.75	24	0.196145	$6.714579e-10$
1.00	24	0.196950	$6.714579e-10$

From Table 7, we remark that Algorithm 1 has the same behavior for all $\alpha \in \Omega$. We need 24 iterations to get desired approximate solution $(X_{24}, S_{24}) = (M + 10^{-11}Z_1\ ,\ 10^{-11}Z_2)$, where

$$Z_1 = \begin{pmatrix} -0.0386 & -0.0095 & 0.0303 & 0.0235 & 0.0301 & 0.0303 & 0.0303 & 0.0301 & 0.0302 & 0.0301 \\ -0.0095 & 0.2692 & -0.0090 & -0.0327 & -0.0093 & -0.0092 & -0.0091 & -0.0088 & -0.0093 & -0.0089 \\ 0.0303 & -0.0090 & -0.0387 & 0.0235 & 0.0303 & 0.0300 & 0.0301 & 0.0302 & 0.0301 & 0.0302 \\ 0.0235 & -0.0327 & 0.0235 & -0.1474 & 0.0235 & 0.0235 & 0.0235 & 0.0234 & 0.0235 & 0.0236 \\ 0.0301 & -0.0093 & 0.0303 & 0.0235 & -0.0388 & 0.0302 & 0.0304 & 0.0302 & 0.0302 & 0.0302 \\ 0.0303 & -0.0092 & 0.0300 & 0.0235 & 0.0302 & -0.0390 & 0.0301 & 0.0303 & 0.0304 & 0.0304 \\ 0.0303 & -0.0091 & 0.0301 & 0.0235 & 0.0304 & 0.0301 & -0.0389 & 0.0301 & 0.0302 & 0.0302 \\ 0.0301 & -0.0088 & 0.0302 & 0.0234 & 0.0302 & 0.0303 & 0.0301 & -0.0387 & 0.0301 & 0.0301 \\ 0.0302 & -0.0093 & 0.0301 & 0.0235 & 0.0302 & 0.0304 & 0.0302 & 0.0301 & -0.0386 & 0.0302 \\ 0.0301 & -0.0089 & 0.0302 & 0.0236 & 0.0302 & 0.0304 & 0.0302 & 0.0301 & 0.0302 & -0.0390 \end{pmatrix}$$

and

$$Z_2 = \begin{pmatrix} 0.0974 & -0.0821 & -0.1103 & -0.0150 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 \\ -0.0821 & -0.0325 & -0.0822 & -0.0672 & -0.0821 & -0.0822 & -0.0822 & -0.0822 & -0.0822 & -0.0822 \\ -0.1103 & -0.0822 & 0.0974 & -0.0150 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 \\ -0.0150 & -0.0672 & -0.0150 & -0.3824 & -0.0150 & -0.0150 & -0.0150 & -0.0150 & -0.0150 & -0.0150 \\ -0.1103 & -0.0821 & -0.1103 & -0.0150 & 0.0974 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 \\ -0.1103 & -0.0822 & -0.1103 & -0.0150 & -0.1103 & 0.0974 & -0.1103 & -0.1103 & -0.1103 & -0.1103 \\ -0.1103 & -0.0822 & -0.1103 & -0.0150 & -0.1103 & -0.1103 & 0.0974 & -0.1103 & -0.1103 & -0.1103 \\ -0.1103 & -0.0822 & -0.1103 & -0.0150 & -0.1103 & -0.1103 & -0.1103 & 0.0974 & -0.1103 & -0.1103 \\ -0.1103 & -0.0822 & -0.1103 & -0.0150 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & 0.0974 & -0.1103 \\ -0.1103 & -0.0822 & -0.1103 & -0.0150 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & -0.1103 & 0.0974 \end{pmatrix}$$

Example 5.4 We show here an example in which Algorithm 1 converge while SNM diverge. Let n be a positive integer. Consider the problem posed in Example 4 such that $Q = (P'P)^2 \in \mathcal{S}^n_+$ where $P = (p_{ij})_{1\leq i,j\leq n}$ and

$$p_{ij} = \frac{|i-j|}{i+j}$$

It is clear that the exact solution of the SDNCP(F) is $(X^*, S^*) = (P'P, 0_n)$. We apply Algorithm 1 and SNM to solve the SDNCP(F), with initial guess $(X_0, S_0) = (1.25 * P'P, 0.1 * I_n)$ which is closed to (X^*, S^*) and $\alpha = 0.5$ and $\mu_k = \frac{1}{2^{k+1}}$. For different values of n we obtain results showing in Table 8.

Table 8 Convergence of Algorithm 1 and divergence of SNM for $n = 4, 10, 15$

	Algorithm 1			SNM		
	k	CPU time (s)	$er_1(k)$	k	CPU time (s)	$er_2(k)$
$n = 4$	28	0.1136	$4.8611e-11$	–	–	$2.2189e+154$
$n = 10$	508	4.62	$8.1889e-11$	–	–	$9.2080e+159$
$n = 15$	1001	21.10	$9.3522e-9$	–	–	$2.1375e+155$

For $n = 4$ we get using Algorithm 1 the following approximation (X_{28}, S_{28}) of exact solution $(P'P, 0_n)$:

$$X_{28} = \begin{pmatrix} 0.7211 & 0.3000 & 0.1524 & 0.1825 \\ 0.3000 & 0.2622 & 0.2143 & 0.2286 \\ 0.1524 & 0.2143 & 0.3104 & 0.3667 \\ 0.1825 & 0.2286 & 0.3667 & 0.4915 \end{pmatrix}$$

and

$$S_{28} = 10^{-9} * \begin{pmatrix} 0.0023 & -0.0117 & 0.0306 & -0.0198 \\ -0.0117 & 0.0404 & -0.1048 & 0.0653 \\ 0.0306 & -0.1048 & 0.2031 & -0.1177 \\ -0.0198 & 0.0653 & -0.1177 & 0.0649 \end{pmatrix}$$

6 Conclusion

In this paper, we proposed a new smooth NCP matrix function and studied various properties of this function. Using these properties, we reformulated the SDNCP(F) problem as a smooth equation. We proved that Newton's method can not be applied to solve this matrix equation since we can not guarantee that its Jacobian operator is invertible at each iteration. So, we applied a quasi-Newton's method and proved that the convergence is superlinear. Also we give some developments of the smoothing Newton's method for solving this problem. We concluded this paper by some numerical tests which confirm the theoretical results and demonstrate the efficiency of the proposed method, and we compared between the both methods.

References

1. Benahmed, B., Mokhtar-Kharroubi, H., de Malafosse, B., Yassine, A.: Quasi-Newton methods in infinite-dimensional spaces and application to matrix equations. J. Glob. Optim. (2011). https://doi.org/10.1007/s10898-010-9564-2
2. Kanzaw, C., Nagel, C.: Semidefinite programs: new search directions, smoothing-type methods and numerical results. SIAM J. Optim. **13**(1), 1–23 (2002)
3. Sun, D., Sun, J.: Semismooth matrix-valued functions. Math. Oper. Res. **27**(1), 150–169 (2002)
4. Sun, D., Sun, J.: Strong semismoothness of the Fischer–Burmeister SDC and SOC complementarity functions. Math. Program. **103**, 575–581 (2005). https://doi.org/10.1007/s10107-005-0577-4

5. Buchholzer, H., Kanzow, C., Knabner, P., Kräutle, S.: The semismooth Newton method for the solution of reactive transport problems including mineral precipitation-dissolution reactions. Comput. Optim. Appl. **50**, 193–221 (2011). https://doi.org/10.1007/s10589-010-9379-6
6. Han, L., Bi, S., Pan, S.: Nonsingularity of FB system and constraint nondegeneracy in semidefinite programming. Numer. Algorithms **62**, 79–113 (2013). https://doi.org/10.1007/s11075-012-9567-9
7. Zhang, L., Zhang, N., Pang, L.: Differential properties of the symmetric matrix-valued Fischer–Burmeister function. J. Optim. Theory Appl. **153**, 436–460 (2012). https://doi.org/10.1007/s10957-011-9962-8
8. Rui, S., Xu, C.: Inexact non-interior continuation method for monotone semidefinite complementarity problems. Optim. Lett. **6**, 1411–1424 (2012). https://doi.org/10.1007/s11590-011-0337-8
9. Bi, S., Pan, S., Chen, J.-S.: Nonsingularity conditions for FB system of nonlinear SDPs. SIAM J. Optim. **21**, 1392–1417 (2011)
10. Todd, M.J.: A study of search directions in primal-dual interior-point methods for semidefinite programming. Optim. Methods Softw. **11**, 1–46 (1999)
11. Tseng, P.: Merit functions for semi-definite complementarity problems. Math. Program. **83**, 159–185 (1998)
12. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming. Kluwer, Boston (2000)
13. Chen, X., Tseng, P.: Non-interior continuation methods for solving semidefinite complementarity problems. Math. Program. Ser. A **95**, 431–474 (2003). https://doi.org/10.1007/s10107-002-0306-1
14. Chen, X., Qi, H., Tseng, P.: Analysis of nonsmooth symmetric-matrix-valued functions with applications to complementarity problems. SIAM J. Optim. **13**(4), 960–985 (2003)
15. Huang, Z., Han, J.: Non-interior continuation method for solving the monotone semidefinite complementarity problem. Appl. Math. Optim. **47**, 195–211 (2003). https://doi.org/10.1007/s00245-003-0765-7