TWO CSCS-BASED ITERATION METHODS FOR SOLVING ABSOLUTE VALUE EQUATIONS*

Xianming Gu[1], Tingzhu Huang[1,†], Houbiao Li[1],

Shengfeng Wang[1] and Liang Li[1]

Abstract Recently, two families of HSS-based iteration methods are constructed for solving the system of absolute value equations (AVEs), which is a class of non-differentiable NP-hard problems. In this study, we establish the Picard-CSCS iteration method and the nonlinear CSCS-like iteration method for AVEs involving the Toeplitz matrix. Then, we analyze the convergence of the Picard-CSCS iteration method for solving AVEs. By using the theory about nonsmooth analysis, we particularly prove the convergence of the nonlinear CSCS-like iteration solver for AVEs. The advantage of these methods is that they do not require the storage of coefficient matrices at all, and the sub-system of linear equations can be solved efficiently via the fast Fourier transforms (FFTs). Therefore, computational cost and storage can be saved in practical implementations. Numerical examples including numerical solutions of nonlinear fractional diffusion equations are reported to show the effectiveness of the proposed methods in comparison with some existing methods.

Keywords Absolute value equation, CSCS-based iteration, toeplitz matrix, nonsmooth analysis, fast Fourier transform.

MSC(2010) 65F12, 65L05, 65N22.

1. Introduction

In the present paper, we are interested in the efficient solutions of absolute value equations (AVEs), i.e.,

$$A\boldsymbol{x} - |\boldsymbol{x}| = \boldsymbol{b}, \quad A \in \mathbb{C}^{n \times n}, \quad \boldsymbol{x}, \ \boldsymbol{b} \in \mathbb{C}^n, \tag{1.1}$$

where A is a non-Hermitian Toeplitz matrix and $|\boldsymbol{x}| = (|x_1|, |x_2|, \ldots, |x_n|)^H$ denotes the component-wise absolute value of the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$. Here the transpose and the conjugate transpose of a matrix A are represented by A^T and A^H, respectively. At present, both theoretical and numerical investigations of such

problems have been extensively studied in recent literature [15, 16, 18, 23, 27, 33, 36]. Additionally, a slightly more extended form of AVEs,

$$A\boldsymbol{x} - B|\boldsymbol{x}| = \boldsymbol{b}, \quad A,\ B \in \mathbb{C}^{m \times n},\ \boldsymbol{x} \in \mathbb{C}^n,\ \boldsymbol{b} \in \mathbb{C}^m, \tag{1.2}$$

was also discussed in [32] and investigated in a more general context [13, 18]. On the other side, the system of AVEs (1.1), which is generally equivalent to the linear complementarity problem (LCP) [7, 23], arises from linear programming, quadratic programming, bimatrix games and other engineering problems (see e.g. [6] for fractional diffusion equations). This means that the system of AVEs is NP-hard in its general form [15, 18, 23]. If $B = 0$, then extended AVEs (1.2) reduce to a linear system $A\boldsymbol{x} = \boldsymbol{b}$, which have many applications in the field of scientific computations [23].

The recent researches concerning AVEs contents can be summarized as the following aspects, one is the theoretical analysis, which focuses on the theorem of alternatives, various equivalent reformulations, and the existence and nonexistence of solutions; refer, e.g., to [7,13,27,32] for details, and the other is how to solve AVEs numerically. In the last decade, based on the fact that the LCP can be reduced to AVEs, which enjoys a special and simple structure, a lot of numerical methods for solving AVEs (1.1) can be found in the recent literature; see e.g. [17, 19, 23, 33, 39] and references therein. For example, a finite computational algorithm that is solved by a finite succession of linear programs (SLP) in [15], and a semi-smooth Newton method and its inexact variants are introduced in [1, 20] respectively, which largely shorten the computation time than the SLP method. Furthermore, a smoothing Newton algorithm was also presented in [7], which was proved to be globally convergent and the convergence rate was quadratic under the condition that the singular values of A exceed 1. This proposed condition was weaker than the one established in [20].

In recent years, the Picard-HSS iteration method and the nonlinear HSS-like iteration method are established to solve the AVEs in [34, 42], respectively. The sufficient condition is given to guarantee the convergence of the Picard-HSS iteration method, and numerical experiments are employed to illustrate the effectiveness of the Picard-HSS and nonlinear HSS-like iteration methods. However, the number of the inner HSS iteration steps is often problem-dependent and difficult to be determined in actual computations. Moreover, the iterates can not be updated timely. It has shown that the nonlinear HSS-like iteration method is more efficient than the Picard-HSS iteration method in aspects of the defect mentioned above, which is designed originally for solving weakly nonlinear systems in [5]. In order to improve the nonlinear HSS-like iteration method, Zhang [40] had extended the preconditioned HSS (PHSS) method [4] to solve AVEs and also used the relaxation technique to accelerate his proposed methods. Meanwhile, he successfully achieved the proof of convergence of the nonlinear PHSS-like iteration method, which was not addressed in the previous work. Numerical results also show the effectiveness of his proposed method in [40]. We consider the special case of A with non-Hermitian Toeplitz structure in this paper, and a Toeplitz matrix A has the so-called circulant and skew-circulant splitting (CSCS) [25]. Inspired by the similar strategies of [34, 42], two kinds of CSCS-based iteration methods are established to solve AVEs (1.1) efficiently. In the first, convergence conditions of the Picard-CSCS iteration method will be investigated. Then we follow Zhang's analytical techniques [40] to prove the convergence condition of the nonlinear CSCS-like iteration method.

The rest of this paper is organized as follows. In Section 2, we first introduce several preliminary results about the nonsmooth analysis. Then we briefly review the CSCS iteration method. In Section 3, we devote to introducing two CSCS-based iteration methods for solving the AVEs (1.1) and investigate their convergence properties, respectively. Numerical experiments are reported in Section 4 to demonstrate the feasibility and effectiveness of our proposed CSCS-based iteration methods. Finally, the paper closes with some conclusions in Section 5.

2. Preliminaries

In this section, similarly to [40], we review some notations and properties related to the nonsmooth analysis, which are useful for discussing the convergence of the proposed methods. Then we briefly recall the knowledge about the CSCS iteration method for solving the non-Hermitian Toeplitz system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$.

2.1. Preliminary results

Let $\Psi : \mathbb{R}^n \to \mathbb{R}^n$ be a specified function, and let $\boldsymbol{x}$ be a given point in $\mathbb{R}^n$. The function Ψ is supposed to be locally Lipschitzian near $\boldsymbol{x}$ if there exist a scalar $\kappa \in \mathbb{R}$ and $\delta > 0$ such that, for all $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n, \|\boldsymbol{y} - \boldsymbol{x}\| < \delta, \|\boldsymbol{z} - \boldsymbol{x}\| < \delta$, the following inequality holds:

$$\|\Psi(\boldsymbol{y}) - \Psi(\boldsymbol{z})\| < \kappa\|\boldsymbol{y} - \boldsymbol{z}\|.$$

Let $\Psi : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitzian function. From Rademacher's theorem [12, pp. 18–23], it notes that Ψ is differentiable almost everywhere. Denote the set of points at which Ψ is differentiable by D_Ψ. We write $\Psi'(\boldsymbol{x})$ for the usual $n \times n$ Jacobian matrix of partial derivatives whenever $\boldsymbol{x}$ is a point at which the necessary partial derivatives exist. Then, the Bouligand subdifferential of Ψ at $\boldsymbol{x} \in \mathbb{R}^n$, denoted by $\partial_B \Psi(\boldsymbol{x})$, is as follows:

$$\partial_B \Psi(\boldsymbol{x}) := \left\{ \lim_{k \to \infty} \Psi'(\boldsymbol{x})(\boldsymbol{x}^{(k)}) : \boldsymbol{x}^{(k)} \in D_\Psi, \boldsymbol{x}^{(k)} \to \boldsymbol{x} \right\}. \tag{2.1}$$

Clarke's generalized Jacobian [8, pp. 69-75] of Ψ at $\boldsymbol{x}$ is the convex hull of $\partial_B \Psi(\boldsymbol{x})$, i.e., $\partial \Psi(\boldsymbol{x}) = conv\{\partial_B \Psi(\boldsymbol{x})\}$. Since Ψ is a locally Lipschitzian function, so the set $\partial_B \Psi(\boldsymbol{x})$ and $\partial \Psi(\boldsymbol{x})$ are bounded. By the definition, $\partial_B \Psi(\boldsymbol{x})$ is also closed. Therefore, $\partial_B \Psi(\boldsymbol{x})$ and $\partial \Psi(\boldsymbol{x})$ are compact.

Definition 2.1 ([29]). Ψ is called semismooth at $\boldsymbol{x}$, if Ψ is locally Lipschitzian and for all $\boldsymbol{h} \in \mathbb{R}^n$ with $\boldsymbol{h} \neq \boldsymbol{0}$,

$$\lim_{\boldsymbol{h}' \to \boldsymbol{h}, t \downarrow 0} \{E\boldsymbol{h}' : E \in \partial\Psi(\boldsymbol{x} + t\boldsymbol{h}')\} \tag{2.2}$$

exists. If Ψ is semismooth at all points in a given set, we can state that Ψ is semismooth in this set.

If Ψ is semismooth at $\boldsymbol{x}$, then Ψ must be directionally differentiable at $\boldsymbol{x}$.

Proposition 2.1 ([28,29]). *Suppose that Ψ is semismooth at $\boldsymbol{x}$. Then the classic directional derivative*

$$\Psi'(\boldsymbol{x}; \boldsymbol{h}) = \lim_{t \downarrow 0} \frac{\Psi(\boldsymbol{x} + t\boldsymbol{h}) - \Psi(\boldsymbol{x})}{t}$$

exists and is equal to the limit in (2.2).

Semismoothness was originally presented by Mifflin [21] for functionals, and then Qi and Sun [29] generalized the concept to vector valued functions. It was proved in [29, Corollary 2.4] that Ψ is semismooth at $\boldsymbol{x}$ if and only if all its component functions are the same. The class of semismooth functionals is very broad; it includes the smooth functions, all convex functions, and the piecewise-smooth functions. Moreover, the sums, differences, products, and composites of semismooth functions are still semismooth; refer, e.g., to [21, 28, 30] for details.

2.2. The CSCS iteration method

Here let $A \in \mathbb{C}^{n \times n}$ be a non-Hermitian Toeplitz matrix of the following form

$$
A = \begin{bmatrix}
a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\
a_1 & a_0 & a_{-1} & \cdots & a_{2-n} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{n-2} & \cdots & a_1 & a_0 & a_{-1} \\
a_{n-1} & a_{n-2} & \cdots & a_1 & a_0
\end{bmatrix},
$$

i.e., A is constant along its diagonals; refer to [24, 25], and $B \in \mathbb{C}^{n \times n}$ be a zero matrix, the general AVEs (1.2) reduced to the system of linear equations

$$
A\boldsymbol{x} = \boldsymbol{b}. \tag{2.3}
$$

It is well-known that a Toeplitz matrix A enjoys a circulant and skew-circulant splitting [25], i.e., $A = C + S$, where

$$
C = \frac{1}{2} \begin{bmatrix}
a_0 & a_{-1} + a_{n-1} & \cdots & a_{2-n} + a_2 & a_{1-n} + a_1 \\
a_1 + a_{1-n} & a_0 & \cdots & \cdots & a_{2-n} + a_2 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{n-2} + a_{-2} & \cdots & \cdots & a_0 & a_{-1} + a_{n-1} \\
a_{n-1} + a_{-1} & a_{n-2} + a_{-2} & \cdots & a_1 + a_{1-n} & a_0
\end{bmatrix}, \tag{2.4}
$$

and

$$
S = \frac{1}{2} \begin{bmatrix}
a_0 & a_{-1} - a_{n-1} & \cdots & a_{2-n} - a_2 & a_{1-n} - a_1 \\
a_1 - a_{1-n} & a_0 & \cdots & \cdots & a_{2-n} - a_2 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{n-2} - a_{-2} & \cdots & \cdots & a_0 & a_{-1} - a_{n-1} \\
a_{n-1} - a_{-1} & a_{n-2} - a_{-2} & \cdots & a_1 - a_{1-n} & a_0
\end{bmatrix}. \tag{2.5}
$$

As we know, C is a circulant matrix, which can be diagonalized by the discrete Fourier transform matrix F; and S is a skew-circulant matrix, which can be diagonalized by a discrete Fourier transform matrix with diagonal scaling, i.e., $\hat{F} = F\Omega$,

where $\Omega = \mathrm{diag}(1, e^{-\frac{\pi\iota}{n}}, \ldots, e^{\frac{-(n-1)\pi\iota}{n}})$ and $\iota = \sqrt{-1}$ is the imaginary unit. That is to say, it holds that

$$FCF^H = \Lambda_C, \qquad \hat{F}S\hat{F}^H = \Lambda_S, \tag{2.6}$$

where

$$F = (F)_{j,k} = \frac{1}{\sqrt{n}}e^{\frac{2\pi\iota}{n}jk}, \quad 0 \le j, k \le n-1$$

and Λ_C, Λ_S are two diagonal matrices formed by the eigenvalues of C and S, respectively, which can be obtained in $\mathcal{O}(n\log n)$ operations by using the FFTs [24, pp. 37-39]. Furthermore, Ng [25] had established the following CSCS iteration scheme to solve the non-Hermitian Toeplitz linear system (2.3).

Algorithm 1 The CSCS iteration method.
Given an initial guess $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ and compute $\boldsymbol{x}^{(k)}$ for $k = 0, 1, 2, \ldots$, using the following iterative scheme until $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ converges,

$$\begin{cases} (\sigma I + C)\boldsymbol{x}^{(k+\frac{1}{2})} = (\sigma I - S)\boldsymbol{x}^{(k)} + \boldsymbol{b}, \\ (\sigma I + S)\boldsymbol{x}^{(k+1)} = (\sigma I - C)\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{b}, \end{cases}$$

where σ is a positive constant and I is the identity matrix of order n.

In the matrix-vector form, the CSCS iteration can be equivalently rewritten as

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \mathcal{M}(\sigma)\boldsymbol{x}^{(k)} + \mathcal{G}(\sigma)\boldsymbol{b} \\ &= (\mathcal{M}(\sigma))^{k+1}\boldsymbol{x}^{(0)} + \sum_{j=0}^{k}(\mathcal{M}(\sigma))^j\mathcal{G}(\sigma)\boldsymbol{b}, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{2.7}$$

where

$$\mathcal{M}(\sigma) = (\sigma I + S)^{-1}(\sigma I - C)(\sigma I + C)^{-1}(\sigma I - S) \text{ and } \mathcal{G}(\sigma) = 2\sigma(\sigma I + S)^{-1}(\sigma I + C)^{-1}.$$

It is worth mentioning that the CSCS iteration is a stationary iterative method obtained from the matrix splitting

$$A = \mathcal{B}(\sigma) - \mathcal{C}(\sigma),$$

where

$$\mathcal{B}(\sigma) = \frac{1}{2\sigma}(\sigma I + C)(\sigma I + S) \quad \text{and} \quad \mathcal{C}(\sigma) = \frac{1}{2\sigma}(\sigma I - C)(\sigma I - S).$$

On the other hand, we have

$$\mathcal{M}(\sigma) = (\mathcal{B}(\sigma))^{-1}\mathcal{C}(\sigma) \quad \text{and} \quad \mathcal{G}(\sigma) = (B(\sigma))^{-1}.$$

Here, $\mathcal{M}(\sigma)$ is the iterative matrix of the CSCS iteration method. We mention that the CSCS iteration method is greatly similar to the HSS iteration method [2] and its variants, see e.g. [3] and references therein.

When the circulant part C and the skew-circulant part S of $A \in \mathbb{C}^{n \times n}$ are both positive definite*, Ng has proved that the spectral radius $\rho(\mathcal{M}(\sigma))$ of $\mathcal{M}(\sigma)$ is less than 1 for any parameters $\sigma > 0$, i.e., the CSCS iteration method unconditionally converges to the exact solution of $A\boldsymbol{x} = \boldsymbol{b}$ for any initial guess $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$; refer to [25, Theorem 1] for details.

---

*It means that the real parts of all their eigenvalues are positive.

3. Two CSCS-based iteration methods for AVEs

Motivated by the pioneer work of [34,42], we extend the conventional CSCS iteration method to two types of CSCS-based iteration methods for solving AVEs (1.1). These methods fully exploit the Toeplitz structure to accelerate the computation speed and save storage. Next, we will devote to establishing these two new methods, i.e., the Picard-CSCS iteration method and the nonlinear CSCS-like iteration method.

3.1. The Picard-CSCS iteration method

Recalling that the Picard iteration method is a fixed-point iterative method and the linear term $A\boldsymbol{x}$ and the nonlinear term $|\boldsymbol{x}| + \boldsymbol{b}$ are separated [34,42], the AVEs (1.1) can be solved by using the Picard iteration method

$$A\boldsymbol{x}^{(k+1)} = |\boldsymbol{x}^{(k)}| + \boldsymbol{b}, \qquad k = 0, 1, 2, \ldots. \tag{3.1}$$

We assume that the non-Hermitian Toeplitz matrix A is positive definite. In this case, the next iterate of $\boldsymbol{x}^{(k+1)}$ can be approximately computed by the CSCS iteration method with using $A = \mathcal{B}(\sigma) - \mathcal{C}(\sigma)$ as the following scheme (see [41])

$$\mathcal{B}(\sigma)\boldsymbol{x}^{(k,\ell+1)} = \mathcal{C}(\sigma)\boldsymbol{x}^{(k,\ell)} + |\boldsymbol{x}^{(k)}| + \boldsymbol{b}, \quad \ell = 0, 1, \ldots, l_k - 1, \quad k = 0, 1, 2, \ldots, \tag{3.2}$$

where $\mathcal{B}(\sigma)$ and $\mathcal{C}(\sigma)$ are the matrices defined in the previous section, σ is a positive constant, $\{l_k\}_{k=0}^{\infty}$ is a prescribed sequence of positive integers, and $\boldsymbol{x}^{(k,0)} = \boldsymbol{x}^{(k)}$ is the starting point of the inner CSCS iteration at k-th outer Picard iteration. This leads to the inexact Picard iteration method, called Picard-CSCS iteration method, for solving AVEs (1.1) which can be summarized as follows, refer to [41].

Algorithm 2 The Picard-CSCS iteration method
Let $A = C + S \in \mathbb{C}^{n \times n}$ be a non-Hermitian Toeplitz matrix; C and S are the circulant and skew-circulant parts of A given in (2.4) and (2.5) and they are both positive definite. Given an initial guess $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ and a sequence $\{l_k\}_{k=0}^{\infty}$ of positive integers, compute $\boldsymbol{x}^{(k+1)}$ for $k = 0, 1, \ldots,$ using the following iterative scheme until $\{\boldsymbol{x}^{(k)}\}$ satisfies the stopping criterion:

(a) Set $\boldsymbol{x}^{(k,0)} = \boldsymbol{x}^{(k)}$;

(b) For $\ell = 0, 1, \ldots, l_k - 1$, solve the following linear systems to obtain $\boldsymbol{x}^{(k,\ell+1)}$:

$$\begin{cases} (\sigma I + C)\boldsymbol{x}^{(k,\ell+\frac{1}{2})} = (\sigma I - S)\boldsymbol{x}^{(k,\ell)} + |\boldsymbol{x}^{(k)}| + \boldsymbol{b}, \\ (\sigma I + S)\boldsymbol{x}^{(k,\ell+1)} = (\sigma I - C)\boldsymbol{x}^{(k,\ell+\frac{1}{2})} + |\boldsymbol{x}^{(k)}| + \boldsymbol{b}, \end{cases}$$

where σ is a given positive constant.

(c) Set $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k,l_k)}$.

Numerical advantages of the Picard-CSCS iteration method are obvious. First, the two linear subsystems in all inner CSCS iteration steps have the same shifted circulant coefficient matrix $\sigma I + C$ and shifted skew-circulant coefficient matrix $\sigma I + S$, which are constant with respect to the iteration index k. Second, the exact solutions can be efficiently obtained via FFTs in $\mathcal{O}(n \log n)$ operations [9,25]. Hence, the computation cost of the Picard-CSCS iteration method could be much cheaper than that of the Picard-HSS iteration method.

The next theorem suggests sufficient conditions for the convergence of the Picard-CSCS iteration method for solving the AVEs (1.1).

Theorem 3.1. *Let $A = C + S \in \mathbb{C}^{n \times n}$ be a non-Hermitian Toeplitz matrix; C and S are the circulant and skew-circulant parts of A given in (2.4)-(2.5) and they are both positive definite. Let also $\eta = \|A^{-1}\|_2 < 1$. Then the AVE (1.1) has a unique solution $\boldsymbol{x}^*$, and for any initial guess $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ and any sequence of positive integers $\{\ell_k\}, k = 0, 1, 2, \ldots$, the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ produced by the Picard-CSCS iteration method converges to $\boldsymbol{x}^*$ provided that $l = \liminf_{k \to \infty} l_k \geq N$, where N is a natural number satisfying*

$$\left\| (\mathcal{M}(\sigma))^s \right\|_2 < \frac{1-\eta}{1+\eta}, \quad \forall s \geq N.$$

Proof.   Due to $\eta < 1$ and the conclusion of [18, Proposition 4], the system of AVEs (1.1) has a unique solution $\boldsymbol{x}^* \in \mathbb{C}^n$. As seen from Eq. (2.7), it found that the $(k+1)$-th iterate of the Picard-CSCS iteration can be written as

$$\boldsymbol{x}^{(k+1)} = (\mathcal{M}(\sigma))^{l_k} \boldsymbol{x}^{(k)} + \sum_{j=0}^{l_k-1} (\mathcal{M}(\sigma))^j \mathcal{G}(\sigma)(|\boldsymbol{x}^{(k)}| + \boldsymbol{b}), \quad k = 0, 1, 2, \ldots. \quad (3.3)$$

On the other side, since $\boldsymbol{x}^*$ is the solution of AVEs (1.1), it follows

$$\boldsymbol{x}^* = (\mathcal{M}(\sigma))^{l_k} \boldsymbol{x}^* + \sum_{j=0}^{l_k-1} (\mathcal{M}(\sigma))^j \mathcal{G}(\sigma)(|\boldsymbol{x}^*| + \boldsymbol{b}), \quad k = 0, 1, 2, \ldots. \quad (3.4)$$

To subtract (3.4) from (3.3) yields

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^* = (\mathcal{M}(\sigma))^{l_k} (\boldsymbol{x}^{(k)} - \boldsymbol{x}^*) + \sum_{j=0}^{l_k-1} (\mathcal{M}(\sigma))^j \mathcal{G}(\sigma)(|\boldsymbol{x}^{(k)}| - |\boldsymbol{x}^*|). \quad (3.5)$$

Furthermore, since $\rho(\mathcal{M}(\sigma)) < 1$, we obtain

$$\sum_{j=0}^{l_k-1} (\mathcal{M}(\sigma))^j \mathcal{G}(\sigma) = (I - (\mathcal{M}(\sigma))^{l_k})(I - \mathcal{M}(\sigma))^{-1} \mathcal{G}(\sigma)$$

$$= (I - (\mathcal{M}(\sigma))^{l_k})(I - (\mathcal{B}(\sigma))^{-1}\mathcal{C}(\sigma))^{-1}(\mathcal{B}(\sigma))^{-1}$$

$$= (I - (\mathcal{M}(\sigma))^{l_k})A^{-1}.$$

Substituting the above identity in Eq. (3.5) yields

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^* = (\mathcal{M}(\sigma))^{l_k} (\boldsymbol{x}^{(k)} - \boldsymbol{x}^*) + (I - (\mathcal{M}(\sigma))^{l_k})A^{-1}(|\boldsymbol{x}^{(k)}| - |\boldsymbol{x}^*|)$$

$$= (\mathcal{M}(\sigma))^{l_k} \left[(\boldsymbol{x}^{(k)} - \boldsymbol{x}^*) - A^{-1}(|\boldsymbol{x}^{(k)}| - |\boldsymbol{x}^*|) \right] + A^{-1}(|\boldsymbol{x}^{(k)}| - |\boldsymbol{x}^*|).$$

Now, we can obtain

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_2 \leq \left(\|(\mathcal{M}(\sigma))^{l_k}\|_2 (1 + \eta) + \eta \right) \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_2.$$

Here, the above inequality is true due to the fact that for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$, it follows $\||\boldsymbol{x}| - |\boldsymbol{y}|\|_2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2$. Since $\rho(\mathcal{M}(\sigma)) < 1$, then $\lim_{s \to \infty} (\mathcal{M}(\sigma))^s = 0$. Thus, there exists a natural number N such that

$$\|(\mathcal{M}(\sigma))^s\|_2 < \frac{1-\eta}{1+\eta}, \quad \forall s \geq N.$$

At the stage, if we suppose that $l = \liminf\limits_{k \to \infty} l_k \geq N$, then the targeted result is immediately completed. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

3.2. The nonlinear CSCS-like iteration method

In the Picard-CSCS iteration method, the numbers $l_k, k = 0, 1, 2, \ldots$ of the inner CSCS iteration steps are often problem-dependent and difficult to be determined in actual computations [34, 41, 42]. Moreover, the iterative vector can not be updated timely. Thus, to avoid the defection and still preserve the advantages of the Picard-CSCS iteration method, based on the nonlinear fixed-point equations

$$(\sigma I + C)\boldsymbol{x} = (\sigma I - S)\boldsymbol{x} + |\boldsymbol{x}| + \boldsymbol{b}, \quad \text{and} \quad (\sigma I + S)\boldsymbol{x} = (\sigma I - C)\boldsymbol{x} + |\boldsymbol{x}| + \boldsymbol{b},$$

we propose the following nonlinear CSCS-like iteration method.

Algorithm 3 The nonlinear CSCS-like iteration method
Let $A = C + S \in \mathbb{C}^{n \times n}$ be a non-Hermitian Toeplitz matrix; C and S are the circulant and skew-circulant parts of A given in (2.4) and (2.5) and they are both positive definite. Choose an initial guess $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ and compute $\boldsymbol{x}^{(k+1)}$ for $k = 0, 1, 2, \ldots$, using the following iteration scheme until $\{\boldsymbol{x}^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\sigma I + C)\boldsymbol{x}^{(k+\frac{1}{2})} = (\sigma I - S)\boldsymbol{x}^{(k)} + |\boldsymbol{x}^{(k)}| + \boldsymbol{b}, \\ (\sigma I + S)\boldsymbol{x}^{(k+1)} = (\sigma I - C)\boldsymbol{x}^{(k+\frac{1}{2})} + |\boldsymbol{x}^{(k+\frac{1}{2})}| + \boldsymbol{b}, \end{cases} \tag{3.6}$$

where σ is a given positive constant.

Define

$$\begin{cases} \mathcal{U}(\boldsymbol{x}) = (\sigma I + C)^{-1}[(\sigma I - S)\boldsymbol{x} + |\boldsymbol{x}| + \boldsymbol{b}], \\ \mathcal{V}(\boldsymbol{x}) = (\sigma I + S)^{-1}[(\sigma I - C)\boldsymbol{x} + |\boldsymbol{x}| + \boldsymbol{b}], \end{cases} \tag{3.7}$$

and

$$\Theta(\boldsymbol{x}) = \mathcal{V} \circ \mathcal{U}(\boldsymbol{x}) := \mathcal{V}(\mathcal{U}(\boldsymbol{x})). \tag{3.8}$$

Then the nonlinear CSCS-like iterative scheme can be equivalently expressed as

$$\boldsymbol{x}^{(k+1)} = \Theta(\boldsymbol{x}^{(k)}), \quad k = 0, 1, 2, \ldots. \tag{3.9}$$

The Ostrowski theorem, i.e., Theorem 10.1.3 in [26, pp. 300–301], provides a local convergence theory about a one-step stationary nonlinear iteration. Based on this item, Zhu and Zhang established the local convergence theory for the nonlinear CSCS-like iteration method in [41]. However, these convergence theory has a strict requirement that $f(\boldsymbol{x}) = |\boldsymbol{x}| + \boldsymbol{b}$ is $\mathcal{F}$-differentiable at a point $\boldsymbol{x}^* \in \mathbb{D}$ (where we define $f : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$) such that $A\boldsymbol{x}^* - |\boldsymbol{x}^*| = \boldsymbol{b}$. Obviously, the absolute value function $|\boldsymbol{x}|$ is non-differentiable. In order to remedy the difficulty, Zhu, Zhang and Liang [42] attempt to introduce a smoothing approximation function [38]

$$\varphi(\boldsymbol{x}) = \frac{1}{\mu} \ln\left(\exp\left(\frac{\boldsymbol{x}}{\mu}\right) + \exp\left(\frac{-\boldsymbol{x}}{\mu}\right)\right), \quad \boldsymbol{x} \in \mathbb{C}^n \text{ and } \mu > 0$$

for $|\boldsymbol{x}|$, then they present the convergence of the nonlinear HSS-like iteration method based on the convergence of the iteration scheme

$$\begin{cases} (\sigma I + C)\boldsymbol{x}^{(k+\frac{1}{2})} = (\sigma I - S)\boldsymbol{x}^{(k)} + \varphi(\boldsymbol{x}^{(k)}) + \boldsymbol{b}, \\ (\sigma I + S)\boldsymbol{x}^{(k+1)} = (\sigma I - C)\boldsymbol{x}^{(k+\frac{1}{2})} + \varphi(\boldsymbol{x}^{(k+\frac{1}{2})}) + \boldsymbol{b}, \end{cases}$$

and their convergence result is deeply dependent on the smoothing approximate function $\varphi(\boldsymbol{x})$ of $|\boldsymbol{x}|$, nor $|\boldsymbol{x}|$ itself. Recently, Zhang [40] exploit the theory of nonsmooth analysis to introduce a framework to prove the convergence of his proposed relaxed nonlinear HSS-like iteration method completely. Inspired by Zhang's framework, we will similarly analyze the (local) convergence of the nonlinear CSCS-like iteration method in the next context. Firstly, the following definition in [26, pp. 299–300] needs to be cited here.

Definition 3.1. Let $\Theta : \mathbb{D} \subset \mathbb{R}^n \to \mathbb{R}^n$. Then $\boldsymbol{x}^*$ is a point of attraction of the iteration (3.9), if there is an open neighborhood S of the point $\boldsymbol{x}^*$ such that $S \subset \mathbb{D}$ and, for any $\boldsymbol{x}^{(0)} \in S$, the iterates $\{\boldsymbol{x}^{(k)}\}$ all lie in $\mathbb{D}$ and converge to $\boldsymbol{x}^*$.

Based on the above definition, we can obtain the following proposition, which is useful for studying the convergence of the nonlinear CSCS-like iteration method.

Proposition 3.1 ([40]). *Suppose that $\Theta : \mathbb{R}^n \to \mathbb{R}^n$ has a fixed-point $\boldsymbol{x}^* \in \mathbb{R}^n$ and is semismooth at $\boldsymbol{x}^*$. If for all $E \in \partial_B \Theta(\boldsymbol{x}^*)$, we have $\rho(E) < 1$, where $\rho(E)$ denotes the spectral radius of E. Then $\boldsymbol{x}^*$ is a point of attraction of the iteration scheme (3.9).*

From statements in [40], let $\boldsymbol{x}^*$ satisfy $A\boldsymbol{x}^* - |\boldsymbol{x}^*| = \boldsymbol{b}$. We compute the Bouligand subdifferential of $\Theta(\boldsymbol{x})$ defined by (3.8)-(3.9) at $\boldsymbol{x}^*$. Due to the special form of $\mathcal{V}$ and $\mathcal{U}$, it is easy to verify that, $\boldsymbol{x}^* = \mathcal{U}(\boldsymbol{x}^*)$, $\boldsymbol{x}^* = \mathcal{V}(\boldsymbol{x}^*)$, and $\boldsymbol{x}^* = \Theta(\boldsymbol{x}^*)$. Observe the special form of Θ, we have that

$$
\begin{aligned}
\partial_B \Theta(\boldsymbol{x}^*) &= \big\{ \lim_{k \to \infty} \Theta'(\boldsymbol{x}^{(k)}) : \boldsymbol{x}^{(k)} \in D_\Theta,\ \boldsymbol{x}^{(k)} \to \boldsymbol{x}^* \big\} \\
&= \Big\{ \lim_{k \to \infty} \mathcal{V}'(\boldsymbol{y}^{(k)}) \mathcal{U}'(\boldsymbol{x}^{(k)}) : \boldsymbol{x}^{(k)} \in D_\mathcal{U},\ \boldsymbol{y}^{(k)} = \mathcal{U}(\boldsymbol{x}^{(k)}) \in D_\mathcal{V},\ \boldsymbol{x}^{(k)} \to \boldsymbol{x}^* \Big\} \\
&\subset \Big\{ \lim_{\boldsymbol{y}^{(k)} \to \boldsymbol{x}^*} \mathcal{V}'(\boldsymbol{y}^{(k)}) : \boldsymbol{y}^{(k)} \in D_\mathcal{V} \Big\} \Big\{ \lim_{\boldsymbol{x}^{(k)} k \to \boldsymbol{x}^*} \mathcal{U}'(\boldsymbol{x}^{(k)}) : \boldsymbol{x}^{(k)} \in D_\mathcal{U} \Big\} \\
&\subset \partial_B \mathcal{V}(\boldsymbol{x}^*) \partial_B \mathcal{U}(\boldsymbol{x}^*),
\end{aligned}
$$

where

$$
\partial_B \mathcal{V}(\boldsymbol{x}^*) \partial_B \mathcal{U}(\boldsymbol{x}^*) := \{ W : W = EF_1,\ E \in \partial_B \mathcal{V}(\boldsymbol{x}^*),\ F_1 \in \partial_B \mathcal{U}(\boldsymbol{x}^*) \}.
$$

Using the above discussion and Proposition 3.1, it immediately obtains the following conclusion about the convergence of the nonlinear CSCS-like iteration solver.

Theorem 3.2. *Let the point $\boldsymbol{x}^*$ satisfy $A\boldsymbol{x}^* = |\boldsymbol{x}^*| + \boldsymbol{b}$. Suppose that C and S are the circulant and skew-circulant parts of the Toeplitz matrix $A = C + S$ given in (2.4) and (2.5). Moreover, C and S are both positive definite matrices, and $F_1, \tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$. Denote by*

$$
\mathcal{M}(\sigma; F_1, \tilde{F}_1) = \mathcal{T}_1(\sigma; F_1) \mathcal{T}_2(\sigma; \tilde{F}_1),
$$

where

$$
\mathcal{T}_1(\sigma; F_1) = (\sigma I + S)^{-1}[(\sigma I - C) + F_1],
$$
$$
\mathcal{T}_2(\sigma; \tilde{F}_1) = (\sigma I + C)^{-1}[(\sigma I - S) + \tilde{F}_1].
$$

If for all $F_1, \tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$, $\rho(\mathcal{M}(\sigma; F_1, \tilde{F}_1)) < 1$, then $\boldsymbol{x}^*$ is a point of attraction of the nonlinear CSCS-like iteration method.

Proof. It is clear that $\mathcal{U}$ and $\mathcal{V}$ are semismooth, so Θ is semismooth. Let $E \in \partial_B |\boldsymbol{x}^*|$, then it is not hard to find that E is a diagonal matrix. Assume

$$E = \text{diag}(E_{11}, E_{22}, \ldots, E_{nn}),$$

we have $E_{ii} = 1$, if $x_i^* > 0$; $E_{ii} = -1$, if $x_i^* < 0$, and $E_{ii} \in \{1, -1\}$, if $x_i^* = 0$. If $W \in \partial_B \mathcal{V}(\boldsymbol{x}^*)$, then $W = (\sigma I + S)^{-1}[(\sigma I - C) + F_1]$, where $F_1 \in \partial_B |\boldsymbol{x}^*|$. If $\tilde{W} \in \partial_B \mathcal{U}(\boldsymbol{x}^*)$, then $\tilde{W} = (\sigma I + C)^{-1}[(\sigma I - S) + \tilde{F}_1]$, where $\tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$. Since $\partial_B \Theta(\boldsymbol{x}^*) \subset \partial_B \mathcal{V}(\boldsymbol{x}^*) \partial_B \mathcal{U}(\boldsymbol{x}^*)$, if for all $F_1, \tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$, $\rho(\mathcal{M}(\sigma; F_1, \tilde{F}_1)) < 1$, then for all $W \in \partial_B \Theta(\boldsymbol{x}^*)$, we have $\rho(W) < 1$. This can complete the desired proof. $\square$

Corollary 3.1. *Let the point $\boldsymbol{x}^*$ satisfies $A\boldsymbol{x}^* = |\boldsymbol{x}^*| + \boldsymbol{b}$. Suppose that C and S are the circulant and skew-circulant parts of the Toeplitz matrix $A = C + S$ given in (2.4) and (2.5). Moreover, C and S both are positive definite matrices, and $F_1, \tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$. Denote by*

$$t_1(\sigma) = \|(\sigma I + S)^{-1}(\sigma I - C)\|,$$

$$t_2(\sigma) = \|(\sigma I + C)^{-1}(\sigma I - S)\|,$$

and

$$\delta = \max\{\|(\sigma I + C)^{-1}\tilde{F}_1\|, \|(\sigma I + S)^{-1}F_1\|\}.$$

If $t_1(\sigma)t_2(\sigma) < 1$ and for all $F_1, \tilde{F}_1 \in \partial_B |\boldsymbol{x}^*|$,

$$\delta < \frac{2 - 2t_1(\sigma)t_2(\sigma)}{\sqrt{(t_1(\sigma) - t_2(\sigma))^2 + 4} + (t_1(\sigma) + t_2(\sigma))}, \tag{3.10}$$

then $\boldsymbol{x}^*$ is a point of attraction of the nonlinear CSCS-like iteration method.

Proof. By simple calculations we obtain

$$\begin{aligned}
\mathcal{M}(\sigma; F_1, \tilde{F}_1) &= (\sigma I + S)^{-1}[(\sigma I - C) + F_1](\sigma I + C)^{-1}[(\sigma I - S) + \tilde{F}_1] \\
&= (\sigma I + S)^{-1}(\sigma I - C)(\sigma I + C)^{-1}(\sigma I - S) + (\sigma I + S)^{-1}(\sigma I - C)(\sigma I \\
&\quad + C)^{-1}\tilde{F}_1 + (\sigma I + S)^{-1}F_1(\sigma I + C)^{-1}(\sigma I - S) \\
&\quad + (\sigma I + S)^{-1}F_1(\sigma I + C)^{-1}\tilde{F}_1.
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\mathcal{M}(\sigma; F_1, \tilde{F}_1)\| &\leq \|(\sigma I + S)^{-1}(\sigma I - C)\| \|(\sigma I + C)^{-1}(\sigma I - S)\| + \|(\sigma I + S)^{-1}(\sigma I \\
&\quad - C)\| \cdot \|(\sigma I + C)^{-1}\tilde{F}_1\| + \|(\sigma I + S)^{-1}F_1\| \|(\sigma I + C)^{-1}(\sigma I - \\
&\quad S)\| + \|(\sigma I + S)^{-1}F_1\| \|(\sigma I + C)^{-1}\tilde{F}_1\| \\
&\leq t_1(\sigma)t_2(\sigma) + \delta(t_1(\sigma) + t_2(\sigma)) + \delta^2.
\end{aligned}$$

With the help of the condition (3.10), we obtain

$$t_1(\sigma)t_2(\sigma) + \delta(t_1(\sigma) + t_2(\sigma)) + \delta^2 < 1.$$

Therefore, we have

$$\rho(\mathcal{M}(\sigma; F_1, \tilde{F}_1)) \leq \|\mathcal{M}(\sigma; F_1, \tilde{F}_1)\| < 1,$$

which follows the desired result by using Theorem 3.2. □

Remark. An attractive feature of the nonlinear CSCS-like iteration method is that it avoids the use of the differentiable in actual iterative scheme. Although we present our convergence analysis of the nonlinear CSCS-like iteration method under real matrices and vectors, the condition is not necessary in the actual implementation corresponding to numerical experiments of the next section.

4. Numerical results

In this section, numerical performances of the Picard-CSCS and the nonlinear CSCS-like iterative solvers are investigated and compared experimentally by a suit of test problems. All the tests are performed in MATLAB R2014a (64bit) on Intel(R) Core(TM) i5-3470 CPU @ 3.2 GHz and 8.00 GB of RAM, with machine precision 10^{-16}, and terminated when the current residual satisfies

$$\frac{\|A\boldsymbol{x}^{(k)} - |\boldsymbol{x}^{(k)}| - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2} < 10^{-7},$$

where $\boldsymbol{x}^{(k)}$ is the computed solution by each of the methods at iteration step k, and a maximum number of the iterations 200 is used.

Morover, the stopping criterion for the inner iterations of the Picard-CSCS iterative method is

$$\frac{\|\boldsymbol{b}^{(k)} - A\boldsymbol{x}^{(k,l_k)}\|_2}{\|\boldsymbol{b}^{(k)}\|_2} \leq \eta_k,$$

where l_k is the number of the inner iteration steps and η_k is the prescribed tolerance for controlling the accuracy of the inner iterations at the k-th outer iteration step. If η_k is fixed for all k, then it is simply denoted by $\tilde{\eta}$.

In our numerical experiments, we use the zero vector as the initial guess, the accuracy of the inner iterations η_k for both Picard-CSCS and Picard-HSS iterative methods is fixed and set to be $\tilde{\eta} = 0.01$, a maximum number of iterations 15 ($l_k = 15, k = 0, 1, 2, \ldots,$) for inner iterations, and the right-hand side vector $\boldsymbol{b}$ of the AVEs (1.1) is taken in such a way that the vector $\boldsymbol{x}^* = (x_1, x_2, \ldots, x_n)^H$ with

$$x_k = (-1)^k \iota, \qquad k = 1, 2, \ldots, n \tag{4.1}$$

is the exact solution. The two sub-systems of linear equations involved are solved in the way if $A\boldsymbol{x}^* = \boldsymbol{b}$, then $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$. Moreover, if the two sub-systems of linear equations involved in the Picard-CSCS and the nonlinear CSCS-like iteration methods are solved by exploiting the method introduced in [9] and using parallel computing, numerical performances of the Picard-CSCS and the nonlinear CSCS-like iteration methods should become better.

On the other hand, Mangasarian modified the classical Newton iteration method for solving AVEs by introducing the auxiliary diagonal matrix $\hat{D}(\boldsymbol{x}) = \partial|\boldsymbol{x}| = \text{diag}(\text{sign}(\boldsymbol{x}))$, refer to [20] for details; then he established the generalized Newton iterative scheme with the initial guess $\boldsymbol{x}^{(0)}$,

$$\boldsymbol{x}^{(k+1)} = (A - \hat{D}(\boldsymbol{x}^{(k)}))^{-1}\boldsymbol{b}, \tag{4.2}$$

so it notes that we need to solve a system of linear equations with the coefficient matrix $J^{(k)} = A - \hat{D}(\boldsymbol{x}^{(k)})$, i.e., Eq. (4.2). If the matrix $J^{(k)} = A - \hat{D}(\boldsymbol{x}^{(k)})$ is

very sparse, then Eq. (4.2) can be solved by using MATLAB's function "\". If the matrix $J^{(k)} = A - \hat{D}(\boldsymbol{x}^{(k)})$ is large-scale (even dense), the Eq. (4.2) can be solved by using Krylov subspace methods, such as GMRES [35] and TFQMR [10]. This consideration just follows the recent method named the inexact semi-smooth Newton method, which has been introduced in [1]. In our numerical experiments, we also give the compared results between the proposed method and the above generalized Newton iterative scheme.

In practical implementations, the optimal parameter $\sigma_{\mathrm{HSS}} = \sqrt{\lambda_{\max}\lambda_{\min}}$ recommended in [2] is employed for the Picard-HSS and nonlinear HSS-like iteration methods, where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and the maximum eigenvalues of the Hermitian part H of the matrix A. Similarly, we adopt the optimal parameter σ_{CSCS} given in [25, 31] for the Picard-CSCS iteration method and the nonlinear CSCS-like iteration method. More precisely, in our calculations, σ_{CSCS} is chosen according to the following formula

$$\sigma_{\mathrm{CSCS}} = \begin{cases} \sqrt{\gamma_{\min}\gamma_{\max} - \zeta_{\max}^2}, & \text{for } \zeta_{\max} < \sqrt{\gamma_{\min}\gamma_{\max}}, \\ \sqrt{\gamma_{\min}^2 + \zeta_{\max}^2}, & \text{for } \zeta_{\max} \geq \sqrt{\gamma_{\min}\gamma_{\max}}, \end{cases}$$

where γ_{min} and γ_{max} are the lower and the upper bounds of the real part of the eigenvalues of the matrices C and S, and ζ_{max} is the upper bound of the absolute values of the imaginary part of the eigenvalues of the matrices C and S. Meanwhile, it should mention that two optimal parameters σ_{HSS} and σ_{CSCS} only minimize the bounds of the convergence factors (not the spectral radiuses selves) of the HSS and CSCS iteration matrices, respectively [41]. Admittedly, the optimal parameters are crucial for guaranteeing fast convergence of these parameter-dependent iteration methods, but they are generally difficult to be determined, see e.g. [2, 5, 34, 41] for a discussion of these issues.

To show that the proposed iteration methods can also be efficiently applied to deal with the complex system of AVEs (1.1), we first construct and test the following example, which is a system of AVEs with complex Toeplitz matrix.

Example 4.1. We consider that $A \in \mathbb{C}^{n \times n}$ is a complex non-Hermitian, sparse and positive definite Toeplitz matrix with the following form

$$A = \begin{pmatrix} \gamma & c\iota & d\iota & & & & \\ -1-c\iota & \gamma & c\iota & d\iota & & & \\ -1-d\iota & -1-c\iota & \gamma & c\iota & d\iota & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1-d\iota & -1-c\iota & \gamma & c\iota & d\iota \\ & & & -1-d\iota & -1-c\iota & \gamma & c\iota \\ & & & & -1-d\iota & -1-c\iota & \gamma \end{pmatrix}, \quad (4.3)$$

where $\iota = \sqrt{-1}$ and $c, d, \gamma \in \mathbb{R}$ are three given parameters. It means that the matrices A in the targeted AVEs are defined as Eq. (4.3). According to the performances of HSS-based methods, see [34, 40, 42], compared with other early established methods, we compare the proposed CSCS-based methods with HSS-based methods in Example 4.1. Then we choose different parameters c and d and present the corresponding numerical results in Tables 2-3.

Table 1. The optimal parameters σ_{opt}^* for Example 4.1.

γ	$(c,\ d)$	σ_{opt}^*	$n = 128$	$n = 256$	$n = 512$	$n = 1024$	$n = 2048$	$n = 4096$
10	(2, 3)	σ_{HSS}	2.9710	2.9524	2.9477	2.9465	2.9462	2.9461
		σ_{CSCS}	1.1817	1.1818	1.1813	1.1813	1.1813	1.1813
13.5	(3, 4)	σ_{HSS}	3.6871	3.6595	3.6525	3.6507	3.6503	3.6502
		σ_{CSCS}	1.6008	1.5997	1.5989	1.5989	1.5988	1.5989

Table 2. Numerical results for Example 4.1 with order n, $\gamma = 10$, and $(c, d) = (2, 3)$.

Method		$n = 128$	$n = 256$	$n = 512$	$n = 1024$	$n = 2048$	$n = 4096$
Picard-HSS	IT_out	6	6	6	6	5	5
	IT_inn	9.8333	9.8333	9.5000	9.5000	9.2000	9.2000
	IT	59	59	57	57	46	46
	CPU	0.0158	0.0215	0.0243	0.0338	0.0511	0.1044
Picard-CSCS	IT_out	6	6	6	6	6	5
	IT_inn	6.3333	6.3333	6.0000	6.0000	6.0000	5.6000
	IT	38	38	36	36	36	28
	CPU	0.0112	0.0146	0.0189	0.0258	0.0431	0.0546
HSS-like	IT	37	36	35	34	33	31
	CPU	0.0136	0.0173	0.0218	0.0275	0.0531	0.1013
CSCS-like	IT	24	23	22	22	21	21
	CPU	0.0072	0.0097	0.0131	0.0215	0.0276	0.0514
GN	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–

Table 3. Numerical results for Example 4.1 with order n, $\gamma = 13.5$, and $(c, d) = (3, 4)$.

Method		$n = 128$	$n = 256$	$n = 512$	$n = 1024$	$n = 2048$	$n = 4096$
Picard-HSS	IT_out	5	5	5	5	5	5
	IT_inn	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
	IT	50	50	50	50	50	50
	CPU	0.0129	0.0161	0.0206	0.0298	0.0544	0.1135
Picard-CSCS	IT_out	5	5	5	5	5	5
	IT_inn	6.2000	6.0000	6.0000	6.0000	5.8000	5.8000
	IT	31	30	30	30	29	29
	CPU	0.0098	0.0123	0.0166	0.0248	0.0332	0.0581
HSS-like	IT	41	40	39	38	37	36
	CPU	0.0121	0.0174	0.0192	0.0306	0.0587	0.1131
CSCS-like	IT	24	23	23	22	21	21
	CPU	0.0075	0.0093	0.0136	0.0212	0.0289	0.0496
GN	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–

Firstly, the optimal parameters σ_{CSCS} and σ_{HSS} for Example 4.1 are listed in Table 1. It is worth mentioning that with the increase of the matrix dimension n, the optimal parameters σ_{CSCS} and σ_{HSS} are almost fixed or decreasing slightly. Moreover, in Tables 2-3, we report numerical results with respect to the Picard-HSS, the nonlinear HSS-like, the Picard-CSCS, the nonlinear CSCS-like iterations, and the generalized Newton iterations using the MATLAB's function "\" (referred to as GN). We also present the elapsed CPU time in seconds (denoted as CPU) and the number of outer, inner and total iteration steps (outer and inner iterations only for both Picard-HSS and Picard-CSCS) for the convergence performances (denoted as IT_out, IT_inn and IT, respectively).

As seen from Tables 2-3, it finds that except the GN method, the Picard-HSS, the nonlinear HSS-like, the Picard-CSCS and the nonlinear CSCS-like iterative methods can successfully achieve approximate solutions of the AVEs with all different matrix dimensions. When the dimension n is increasing, the number of outer and inner iteration steps are almost fixed for all iteration methods, and the number of total iteration steps shows the similar phenomena. But the total CPU time for all iteration methods are increasing quickly. Moreover, in terms of outer iteration steps, the Picard-HSS iteration method and the Picard-CSCS iteration method have almost the same results, but the Picard-CSCS iteration method is better than the Picard-HSS iteration method in terms of inner iteration steps. Then as a result, the Picard-CSCS iteration method is also more competitive than the Picard-HSS iteration method in aspects of the elapsed CPU time.
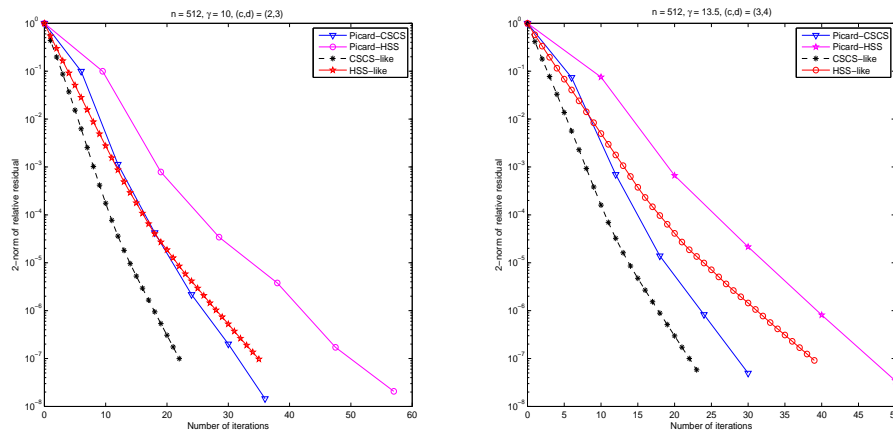
Figure 1. Convergence histories of the different iterative methods for two systems of AVEs with the size $n = 512$ in Example 4.1.

On the other hand, from Tables 2-3, we also observe that both the nonlinear CSCS-like and the Picard-CSCS iteration methods are better than the nonlinear HSS-like and the Picard-HSS iteration methods in terms of the number of iteration steps and the elapsed CPU time for solving AVEs. In particular, the nonlinear CSCS-like method often enjoys the better performance than the Picard-CSCS method in our implementations. Moreover, the convergence histories of residual 2-norms of these four different iterative algorithms are displayed in Fig. 1, and the performance profile based on CPU time for Example 4.1 with increasing the matrix size n is illustrated in Fig. 2. In conclusion, the nonlinear CSCS-like iteration method is the best choice for coping with AVEs concerning in Example 4.1. Besides, the Picard-CSCS iteration method can be regarded as an acceptable alternative.

Example 4.2. In order to evaluate the performances of the propose methods comprehensively, we consider a family of the practical problems about the AVEs arising in numerical solutions of the following one-dimensional nonlinear space fractional
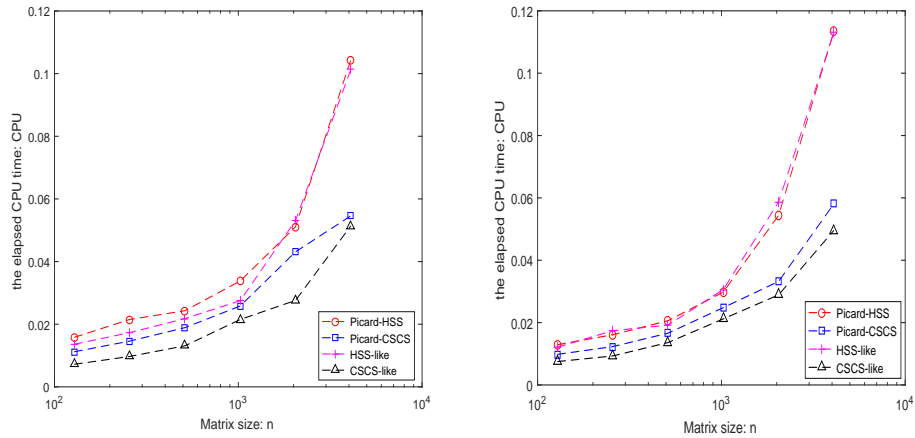
Figure 2. Performance profile based on CPU time under the matrix size n in Example 4.1.

diffusion equation, which is specially modified from Refs. [6, 22],

$$
\begin{cases}
\dfrac{\partial u(x,t)}{\partial t} = d_+ \dfrac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} + d_- \dfrac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} + |u(x,t)|/\varsigma, & x \in (0,1), \quad t \in [0,1], \\
u(0,t) = u(1,t) = 0, & 0 \le t \le 1, \\
u(x,0) = \phi(x), & 0 \le x \le 1,
\end{cases}
$$

$$(4.4)$$

where $\alpha \in (1,2)$ is the order of the fractional derivative, $\varsigma > 0$, and diffusion coefficients $d_\pm$ are nonnegative; i.e., $d_\pm \ge 0$. Moreover, $\phi(x)$ is a known function. To solve Eq. (4.4) numerically, let N and M be positive integers, and $h = 1/(N+1)$ and $\tau = 1/M$ be the sizes of spatial grid and time step, respectively. We define a spatial and temporal partition $x_j = jh$ for $j = 0, 1, \dots, N+1$ and $t_m = m\tau$ for $m = 0, 1, \dots, M$. Let $u_j^{(m)} = u(x_j, t_m)$. In [22], Meerschaert and Tadjeran proposed the shifted Grünwald approximation as follows,

$$
\frac{\partial^\alpha u(x_j, t_m)}{\partial_+ x^\alpha} = \frac{1}{h^\alpha} \sum_{k=0}^{j+1} g_k^{(\alpha)} u_{j-k+1}^{(m)} + \mathcal{O}(h), \tag{4.5a}
$$

$$
\frac{\partial^\alpha u(x_j, t_m)}{\partial_- x^\alpha} = \frac{1}{h^\alpha} \sum_{k=0}^{N-j+2} g_k^{(\alpha)} u_{j+k-1}^{(m)} + \mathcal{O}(h), \tag{4.5b}
$$

where the coefficients $g_k^{(\alpha)}$ and corresponding properties are given in [14, 22, Proposition 1]. Combining the implicit Euler scheme with Eqs. (4.5) to discrete Eq. (4.4), then the final numerical scheme is

$$
\frac{u_j^{(m)} - u_j^{(m-1)}}{\tau} = \frac{d_+}{h^\alpha} \sum_{k=0}^{j+1} g_k^{(\alpha)} u_{j-k+1}^{(m)} + \frac{d_-}{h^\alpha} \sum_{k=0}^{N-j+2} g_k^{(\alpha)} u_{j+k-1}^{(m)} + |u_j^{(m)}|/\varsigma. \tag{4.6}
$$

By using the similar ways given in [6], it is not difficult to prove that the numerical scheme (4.6) is unconditionally stable, which we will not pursue here. Let $\boldsymbol{u}^{(m)} =$

$(u_1^{(m)}, u_2^{(m)}, \ldots, u_N^{(m)})^T$, $m = 0, 1, \ldots, M$ and I_N be the identity matrix of order N. Then the numerical scheme (4.6) at the first temporal level $m = 1$ can be written in the following matrix form

$$\left[I_N - \frac{\tau}{h^\alpha}(d_+ G_\alpha + d_- G_\alpha^T)\right] \boldsymbol{u}^{(1)} - |\boldsymbol{u}^{(1)}| = \boldsymbol{u}^{(0)}, \qquad (4.7)$$

where we take $\varsigma = \tau$ and $G_\alpha \in \mathbb{R}^{N \times N}$ is a nonsymmetric Toeplitz matrix defined in [14]. According to Eq. (4.6), it implies that we need to handle a system of nonlinear equations like the AVEs in (1.1) at each time step, i.e., there is a need for solving the AVEs with the form $A\boldsymbol{u} - |\boldsymbol{u}| = \boldsymbol{u}^{(0)}$, where $A = I_N - \frac{\tau}{h^\alpha}(d_+ G_\alpha + d_- G_\alpha^T)$ is also a nonsymmetric Toeplitz matrix. Meanwhile, for simplicity, the vector $\boldsymbol{u}^{(0)}$ is still chosen as the same as that $\boldsymbol{x}^*$ in Eq. (4.1) is the solution of AVEs in (1.1).

Next, for the first temporal level $m = 1$, we employ these two CSCS-based iteration methods to solve the above resultant AVEs, then the necessary condition for analyzing the convergence of the CSCS-based iteration method is that both the circulant part C and the skew-circulant part S of A are positive definite. In fact, we have already mentioned that both the circulant part C and the skew-circulant part S of the matrix $A = I_N - \frac{\tau}{h^\alpha}(d_+ G_\alpha + d_- G_\alpha^T)$ are positive definite (see [11] for details) via the similarly analyzed methods in [31]. It means that exploiting the CSCS-based iteration methods for solving the resulting AVEs is reasonable. *At the same time, it is worth mentioning that HSS-based iteration methods are not suitable for Example 4.2 due to the Toeplitz coefficient matrix. Otherwise, it will lead to the complicated computations for solving two sub-systems with the dense coefficient matrices $\sigma I + H$ and $\sigma I + S$. In this example, since the matrix $J^{(k)}$ is a Toeplitz-plus-diagnoal matrix, so there are no fast direct solvers for $J^{(k)} \tilde{\boldsymbol{x}} = \boldsymbol{u}^{(0)\dagger}$.* Fortunately, it should note that the matrix-vector product involving $J^{(k)}$ can be implemented via FFTs due to having the Toeplitz part A. It tells us that the Krylov subspace methods can be compatibly exploited for solving $J^{(k)} \tilde{\boldsymbol{x}} = \boldsymbol{u}^{(0)}$ at each iteration step, we denote them as the GN-TFQMR method and the GN-GMRES method. In conclusion, we will compared the proposed CSCS-based iteration method with both the GN-GMRES and GN-TFQMR methods for solving the resultant AVEs in Example 4.2. Numerical results are reported in the following tables under different values of $\alpha, d_\pm$ and $h = \tau$. The total number of (inner) iteration steps used for both GMRES and TFQMR methods, which are used to solve $J^{(k)} \tilde{\boldsymbol{x}} = \boldsymbol{u}^{(0)}$, is no more than 15 in our practical implementations.

Table 4. The optimal parameters σ_{opt}^* of the CSCS iteration method in Example 4.2.

α	$(d_+,\ d_-)$	σ_{opt}^*					
		$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$	$N = 4096$
1.2	(0.5, 0.8)	1.4499	1.5338	1.6233	1.7180	1.8175	1.9216
1.5	(0.6, 0.4)	2.7848	3.2426	3.7564	4.3094	4.8598	5.3131
1.8	(0.7, 0.3)	5.8492	6.9416	7.0941	15.3896	26.7749	46.6033

First of all, the optimal parameters σ_{CSCS} for Example 4.2 are listed in Table 4. It is remarked that with the increase of the matrix dimension n, the optimal

---

†It is mainly because the displacement rank of the matrix $J^{(k)}$ can take any value between 0 and n. Hence, fast Toeplitz direct solvers that are based on small displacement rank of matrices cannot be applied [24, p.142].

parameters σ_{CSCS} are almost fixed or increasing slightly for the cases of $\alpha = 1.2$ and $\alpha = 1.5$. Since the case of $\alpha = 1.8$ corresponding to the coefficient matrix A is very ill-conditioned, so the optimal parameters σ_{CSCS} are varied intensely. Moreover, in Tables 5-7, we report the numerical results with respect to the Picard-CSCS, nonlinear CSCS-like, GN-GMRES and GN-TFQMR iterative methods. Similar to Example 4.1, we report the elapsed CPU time in seconds and the number of outer, inner and total iteration steps (outer and inner iterations only for Picard-CSCS, GN-GMRES and GN-TFQMR) for showing the convergence performances.

Table 5. Numerical results for Example 4.2 with order N, $\alpha = 1.2$, and $(c, d) = (0.5, 0.8)$.

Method		$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$	$N = 4096$
Picard-CSCS	IT_out	6	6	6	6	6	6
	IT_inn	4.0000	4.0000	4.0000	4.1667	5.0000	5.0000
	IT	24	24	24	25	30	30
	CPU	0.0077	0.0098	0.0136	0.01998	0.0331	0.0585
CSCS-like	IT	12	13	14	15	16	18
	CPU	0.0032	0.0051	0.0078	0.0099	0.01776	0.0334
GN-GMRES	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–
GN-TFQMR	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–

Table 6. Numerical results for Example 4.2 with order N, $\alpha = 1.5$, and $(d_+, d_-) = (0.6, 0.4)$.

Method		$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$	$N = 4096$
Picard-CSCS	IT_out	6	6	6	6	6	6
	IT_inn	7.0000	8.1667	9.1667	10.6667	13.0000	14.3333
	IT	42	49	55	64	78	86
	CPU	0.0161	0.0189	0.0234	0.0335	0.0609	0.1307
CSCS-like	IT	24	29	35	43	54	69
	CPU	0.0063	0.0084	0.0119	0.0218	0.0486	0.1133
GN-GMRES	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–
GN-TFQMR	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–

Based on numerical results in Tables 5-7, it finds that these two iterative solvers, i.e., the Picard-CSCS and the nonlinear CSCS-like, can successfully obtain approximate solutions to the AVEs for all different matrix dimensions; whereas both the GN-GMRES and GN-TFQMR iterative methods fully fail to converge. It is mainly because the Newton-like iterative methods are usually sensitive to the initial guess and the accuracy of solving the inner linear system corresponding to (4.2) per iterative step. When the matrix dimension N is increasing, the number of outer iteration steps are almost fixed or increasing slightly for all iteration methods, whereas the number of inner iteration steps show the contrary phenomena for the cases with

Table 7. Numerical results for Example 4.2 with order N, $\alpha = 1.8$, and $(d_+, d_-) = (0.7, 0.3)$.

Method		$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$	$N = 4096$
Picard-CSCS	IT_out	5	6	10	8	9	13
	IT_inn	14.4000	15.0000	15.0000	15.0000	15.0000	15.0000
	IT	72	90	150	120	135	195
	CPU	0.0185	0.0252	0.0415	0.0505	0.0987	0.2817
CSCS-like	IT	59	86	146	117	117	118
	CPU	0.0101	0.0179	0.0386	0.0517	0.1007	0.1911
GN-GMRES	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–
GN-TFQMR	IT_out	max	max	max	max	max	max
	IT_inn	–	–	–	–	–	–
	IT	Fail	Fail	Fail	Fail	Fail	Fail
	CPU	–	–	–	–	–	–

$\alpha = 1.2$ and $\alpha = 1.5$. Meanwhile, the total CPU time and the total iteration steps for both the Picard-CSCS and the nonlinear CSCS-like iteration methods are increasing quickly except the cases of $\alpha = 1.8$ with $N = 1024$ and $N = 2048$. On the other hand, from Tables 5-7, we also observe that the nonlinear CSCS-like method is almost more competitive than the Picard-CSCS iteration method in terms of the number of iterations and the elapsed CPU time for solving the AVEs. In particular, we can find that the nonlinear CSCS-like iteration method can require slightly less number of iterations to converge than the Picard-CSCS iterative solver, but the Picard-CSCS iterative solver can save a little elapsed CPU time with compared to the nonlinear CSCS-like iteration method in our implementations. However, it still concludes that the nonlinear CSCS-like iterative method is the first choice for solving the AVEs concerning in Example 4.2. At the same time, the Picard-CSCS iteration method can be considered as a possible alternative.

5. Conclusions

In this paper, we have constructed two CSCS-based iteration methods for solving AVEs (1.1) with non-Hermitian Toeplitz matrix. Two CSCS-based iteration methods are based on separable property of the linear term $A\boldsymbol{x}$ and the nonlinear term $|\boldsymbol{x}| + \boldsymbol{b}$ as well as on the CSCS of the involved non-Hermitian positive definite Toeplitz matrix A. By leveraging the theory of nonsmooth analysis, the local convergence of nonlinear CSCS-like iteration method has been investigated. Further numerical experiments have shown that the Picard-CSCS and nonlinear CSCS-like iteration methods are feasible and efficient nonlinear solvers for the AVEs. In particular, the nonlinear CSCS-like iteration method often does better than the Picard-CSCS iteration method for solving the AVEs. Finally, it is worth mentioning that how to employ suitable acceleration techniques [3, 37, 40] for enhancing the convergence of CSCS-based iteration methods, which are affiliated with the fixed-point iteration, can remain an interesting topic of further research.

Acknowledgements

References

[1] J. Y. Bello Cruz, O. P. Ferreira and L. F. Prudente, *On the global convergence of the inexact semi-smooth Newton method for absolute value equation*, Comput. Optim. Appl., 2016, 65(1), 93–108.

[2] Z. Z. Bai, G. H. Golub and M. K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 2003, 24(3), 603–626.

[3] Z. Z. Bai, G. H. Golub and M. K. Ng, *On successive overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations*, Numer. Linear Algebra Appl., 2007, 14(4), 319–335.

[4] Z. Z. Bai, G. H. Golub and C. K. Li, *Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices*, Math. Comp., 2007, 76(257), 287–298.

[5] Z. Z. Bai and X. Yang, *On HSS-based iteration methods for weakly nonlinear systems*, Appl. Numer. Math., 2009, 59(12), 2923–2936.

[6] H. W. Choi, S. K. Chung and Y. J. Lee, *Numerical solutions for space fractional dispersion equations with nonlinear source terms*, Bull. Korean Math. Soc., 2010, 47(6), 1225–1234.

[7] L. Caccetta, B. Qu and G. Zhou, *A globally and quadratically convergent method for absolute value equations*, Comput. Optim. Appl., 2011, 48(1), 45–58.

[8] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, USA, 1990.

[9] M. Chen, *On the solution of circulant linear systems*, SIAM J. Numer. Anal., 1987, 24(3), 668–683.

[10] R. W. Freund, *A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 1993, 14(2), 470–482.

[11] X. M. Gu, T. Z. Huang, H. B. Li, L. Li and W. H. Luo, *On k-step CSCS-based polynomial preconditioners for Toeplitz linear systems with application to fractional diffusion equations*, Appl. Math. Lett., 2015, 42, 53–58.

[12] J. Heinonen, *Lectures on Lipschitz Analysis*, Technical Report, Vol. 100, Department of Mathematics and Statistics, University of Jyväskylä, Jyväskylä, Finland, 2005, 77 pages. Also available online at http://www.math.jyu.fi/research/reports/rep100.pdf.

[13] S. L. Hu and Z. H. Huang, *A note on absolute value equations*, Optim. Lett., 2010, 4(3), 417–424.

[14] S. L. Lei and H. W. Sun, *A circulant preconditioner for fractional diffusion equations*, J. Comput. Phys., 2013, 242, 715–725.

[15] O. L. Mangasarian, *Absolute value equation solution via concave minimization*, Optim. Lett., 2007, 1(1), 3–8.

[16] H. Moosaei, S. Ketabchi, M. A. Noor, J. Iqbal and V. Hooshyarbakhsh, *Some techniques for solving absolute value equations*, Appl. Math. Comput., 2015, 268(C), 696–705.

[17] O. L. Mangasarian, *Knapsack feasibility as an absolute value equation solvable by successive linear programming*, Optim. Lett., 2009, 3(2), 161–170.

[18] O. L. Mangasarian and R. R. Meyer, *Absolute value equations*, Linear Algebra Appl., 2006, 419(2–3), 359–367.

[19] O. L. Mangasarian, *Primal-dual bilinear programming solution of the absolute value equation*, Optim. Lett., 2012, 6(7), 1527–1533.

[20] O. L. Mangasarian, *A generalized Newton method for absolute value equations*, Optim. Lett., 2009, 3(1), 101–108.

[21] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 1977, 15(6), 959–972.

[22] M. M. Meerschaert and C. Tadjeran, *Finite difference approximations for two-sided space-fractional partial differential equations*, Appl. Numer. Math., 2006, 56(1), 80–90.

[23] M. A. Noor, J. Iqbal, K. I. Noor and E. Al-Said, *On an iterative method for solving absolute value equations*, Optim. Lett., 2012, 6(5), 1027–1033.

[24] M. K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, UK, 2004.

[25] M. K. Ng, *Circulant and skew-circulant splitting methods for Toeplitz systems*, J. Comput. Appl. Math., 2003, 159(1), 101–108.

[26] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, Philadelphia, USA, 2000.

[27] O. Prokopyev, *On equivalent reformulations for absolute value equations*, Comput. Optim. Appl., 2009, 44(3), 363–372.

[28] J. S. Pang and L. Qi, *Nonsmooth equations: motivation and algorithms*, SIAM J. Optim., 1993, 3(3), 443–465.

[29] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 1993, 58(1), 353–367.

[30] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 1993, 18(1), 227–244.

[31] W. Qu, S. L. Lei and S. W. Vong, *Circulant and skew-circulant splitting iteration for fractional advection-diffusion equations*, Int. J. Comput. Math., 2014, 91(10), 2232–2242.

[32] J. Rohn, *A theorem of the alternatives for the equation $A\boldsymbol{x} + B|\boldsymbol{x}| = \boldsymbol{b}$*, Linear and Multilinear Algebra, 2004, 52(6), 421–426.

[33] J. Rohn, V. Hooshyarbakhsh and R. Farhadsefat, *An iterative method for solving absolute value equations and sufficient conditions for unique solvability*, Optim. Lett., 2014, 8(1), 35–44.

[34] D. K. Salkuyeh, *The Picard-HSS iteration method for absolute value equations*, Optim. Lett., 2014, 8(8), 2191–2202.

[35] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 1986, 7(3), 856–869.

[36] S. L. Wu and P. Guo, *On the unique solvability of the absolute value equation*, J. Optim. Theory Appl., 2016, 169(2), 705–712.

[37] H. F. Walker and P. Ni, *Anderson acceleration for fixed-point iterations*, SIAM J. Numer. Anal., 2011, 49(4), 1715–1735.

[38] L. Yong, *Particle swarm optimization for absolute value equations*, J. Comput. Inf. Syst., 2010, 6(7), 2359–2366.

[39] C. Zhang and Q. J. Wei, *Global and finite convergence of a generalized Newton method for absolute value equations*, J. Optim. Theory Appl., 2009, 143(2), 391–403.

[40] J. J. Zhang, *The relaxed nonlinear PHSS-like iteration method for absolute value equations*, Appl. Math. Comput., 2015, 265, 266–274.

[41] M. Z. Zhu and G. F. Zhang, *On CSCS-based iteration methods for Toeplitz system of weakly nonlinear equations*, J. Comput. Appl. Math., 2011, 235(17), 5095–5104.

[42] M. Z. Zhu, G. F. Zhang and Z. Z. Liang, *The nonlinear HSS-like iteration method for absolute value equations*, arXiv:1403.7013v2, 1 Apr. 2015, 14 pages. Available online at http://arxiv.org/abs/1403.7013v2.