East Asian Journal on Applied Mathematics Vol. **15**, No. 1, pp. 138-162
doi: 10.4208/eajam.2023-172.281023 February 2025

A Relaxation Two-Step Newton-Based Matrix Splitting Iteration Method for Generalized Absolute Value Equations Associated with Circular Cones

Dongmei Yu[1], Yinghui Gao[1], Cairong Chen[2,*] and Deren Han[3]

[1] *Institute for Optimization and Decision Analytics & College of Science, Liaoning Technical University, Fuxin 123000, China.*
[2] *School of Mathematics and Statistics & Key Laboratory of Analytical Mathematics and Applications (Ministry of Education) & Fujian Provincial Key Laboratory of Statistics and Artificial Intelligence & Fujian Key Laboratory of Analytical Mathematics and Applications (FJKLAMA) & Center for Applied Mathematics of Fujian Province (FJNU), Fujian Normal University, Fuzhou 350117, China.*
[3] *LMIB of the Ministry of Education, School of Mathematical Sciences, Beihang University, Beijing 100191, China.*

Received 18 July 2023; Accepted (in revised version) 28 October 2023.

Abstract. A relaxation two-step Newton-based matrix splitting (RTNMS) iteration method is proposed to solve the generalized absolute value equations associated with circular cones (CCGAVE). The convergence of the RTNMS iteration method is investigated under suitable conditions. Numerical results illustrate that the RTNMS iteration method is feasible and effective for solving the CCGAVE. Moreover, some sufficient conditions for the unique solvability of the CCGAVE are given.

AMS subject classifications: 65F10, 65H10, 90C30

Key words: Generalized absolute value equations, circular cone, Newton-based matrix splitting method, convergence analysis.

1. Introduction

The circular cone (CC) is a pointed closed convex cone having hyperspherical sections orthogonal to its axis of revolution around which the cone is invariant with respect to rotation [5, p. 102]. According to the definition, one can see that [31]

$$\mathscr{L}_\theta^n = \left\{ x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|x_2\| \le x_1 \tan\theta \right\} \tag{1.1}$$

*Corresponding author. *Email addresses:* `yudongmei1113@163.com` (D. Yu), `yinghuigao1101@163.com` (Y. Gao), `cairongchen@fjnu.edu.cn` (C. Chen), `handr@buaa.edu.cn` (D. Han)

is a CC in $\mathbb{R}^n$, where $\theta \in (0, \pi/2)$ is its half-aperture angle and $\|\cdot\|$ denotes the Euclidean norm. If $\theta = \pi/4$, then $\mathscr{L}_\theta^n$ reduces to the well-known second-order cone (SOC)

$$\mathscr{K}^n = \left\{ x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|x_2\| \leq x_1 \right\}.$$

When $\theta = \pi/4$ and $n = 1$, let $\mathscr{L}_\theta^n$ be the set of nonnegative real. In general, we can consider the following Cartesian product $\mathscr{L}_\theta$ of circular cons $\mathscr{L}_\theta^{n_i}$ in $\mathbb{R}^n$:

$$\mathscr{L}_\theta = \mathscr{L}_\theta^{n_1} \times \mathscr{L}_\theta^{n_2} \times \cdots \times \mathscr{L}_\theta^{n_r},$$

where $\mathscr{L}_\theta^{n_i}$ is defined as in (1.1) and $n_1 + n_2 + \cdots + n_r = n$ with $r, n_1, n_2, \cdots, n_r \geq 1$. In addition, let us also recall — cf. [6, p. 4], that for any nonempty set $\mathscr{C} \subseteq \mathbb{R}^n$, its dual cone is defined by

$$\mathscr{C}^* := \{ y \in \mathbb{R}^n \mid \langle y, x \rangle \geq 0 \quad \text{for all} \quad x \in \mathscr{C} \}.$$

A cone $\mathscr{C}$ with $\mathscr{C} = \mathscr{C}^*$ is called a self-dual or symmetric cone. Note that unlike the SOC $\mathscr{K}^n$, the CC $\mathscr{L}_\theta^n$ is not a self-dual cone whenever $\theta \in (0, \pi/2) \backslash \{\pi/4\}$. Indeed, according to [31], we have

$$\left(\mathscr{L}_\theta^n\right)^* = \left\{ y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|y_2\| \leq y_1 \cot\theta \right\}.$$

Furthermore,

$$\mathscr{L}_\theta^* = \left(\mathscr{L}_\theta^{n_1}\right)^* \times \left(\mathscr{L}_\theta^{n_2}\right)^* \times \cdots \times \left(\mathscr{L}_\theta^{n_r}\right)^*.$$

With any CC we can associate generalized absolute value equations (CCGAVE) — viz.

$$Ax - B|x| = b, \tag{1.2}$$

where $A, B \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $x = (x_1^\intercal, \cdots, x_r^\intercal)^\intercal \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_r}$, and $|x|$ denotes the absolute value of x corresponding to $\mathscr{L}_\theta$. To the best of our knowledge, CCGAVE (1.2) is first introduced in [19] and it reduces to the generalized absolute value equations associated with SOC (SOCGAVE) [9] when $\theta = \pi/4$. More specifically, if $\theta = \pi/4$ and $r = n$, CCGAVE (1.2) boils down to the generalized absolute value equations (GAVE) in $\mathbb{R}^n$ [23]. The GAVE is relevant to many scientific problems, such as linear complementarity problems, interval linear equations, quadratic programs and others — cf. [15,17,21,22] and references therein. Any SOCGAVE is equivalent to a linear complementarity problem associated with SOC, which has numerous applications in control, finance and robust optimization [1,4]. We are mainly interested in CCGAVE (1.2) because it is not only present the extensions of the GAVE and SOCGAVE but also gives an equivalent reformulation of linear complementarity problems associated with CC (CCLCP). The later plays a crucial role in the optimization community [2,13].

Over the past two decades, a lot of efforts have been made in developing numerical methods and analyzing theoretical properties of the GAVE [3, 8, 14–17, 21, 25–30] and SOCGAVE [9,10,18,20]. However, the study on CCGAVE (1.2) is still in its infancy, and to our knowledge, there is no work except for the results in [19]. This motivates us to do the job here.

This paper presents a relaxation two-step Newton-based matrix splitting (RTNMS) iteration method for solving CCGAVE (1.2). As special cases, the RTNMS iteration method can generate some well-known methods, such as the two-step Newton-based matrix splitting (TNMS) iteration method, the relaxation Newton-based matrix splitting (RNMS) iteration method, the Newton-based matrix splitting (NMS) iteration method, the relaxation modified Newton-type (RMN) iteration method, the modified Newton-type (MN) iteration method, the Picard iteration method, the Douglas-Rachford splitting (DRs) iteration method and the shift splitting (SS) iteration method. Moreover, the convergence analysis of the proposed RTNMS iteration method is established. Numerical experiments are given to demonstrate the efficiency of our method.

The rest of this paper is organized as follows. In Section 2, we review some basic concepts and important properties associated with CC. Besides, the unique solvability of CCGAVE (1.2) is studied. In Section 3, we develop the RTNMS iteration method for solving CCGAVE (1.2). In Section 4, the convergence analysis of the RTNMS iteration method is established. In Section 5, numerical results are reported to show the efficiency of the proposed RTNMS iteration method. Finally, some conclusions are given in Section 6.

Notation. The set of all $n \times n$ real matrices is denoted by $\mathbb{R}^{n\times n}$ and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. We use I to denote the identity matrix with suitable dimension. The transposition of a matrix or a vector a is denoted by $a^{\intercal}$. For any $x, y \in \mathbb{R}^n$, the Euclidean inner product is defined as $\langle x, y\rangle = x^{\intercal}y$, while the Euclidean norm $\|x\| = \sqrt{\langle x, x\rangle}$. The spectral norm of $A \in \mathbb{R}^{n\times n}$ is denoted by $\|A\|$ and is defined by the formula $\|A\| = \max\{\|Ax\| : x \in \mathbb{R}^n, \|x\| = 1\}$. We use $\mathrm{tridiag}(a, b, c)$ to denote a tridiagonal matrix, which has a, b, c as the subdiagonal, main diagonal and superdiagonal entries, respectively, and the same goes to the block tridiagonal matrix $\mathrm{blktridiag}(A, B, C)$. The set of all eigenvalues of matrix A is defined by $\mathrm{sp}(A)$, and the Kronecker product is defined as $\otimes$.

2. Preliminaries

In this section, we collect a few important results on CC which lay the foundation of our theoretical analysis. In addition, a sufficient condition for the unique solvability of CCGAVE (1.2) is provided. For the sake of simplicity, we concentrate on the case of $r = 1$, since the results can be extended to the case of $r > 1$ according to the property of the Cartesian product. Meanwhile, the theoretical analysis in this paper hold for any given $\theta \in (0, \pi/2)$.

For any vector $x \in \mathbb{R}^n$, here and in the sequel, let x_+ denote the projection[†] of x onto $\mathscr{L}_\theta^n$ and x_- be the projection of $-x$ onto $(\mathscr{L}_\theta^n)^*$. According to the definition of the projection, it is obvious that $x_+ \in \mathscr{L}_\theta^n$ and $x_- \in (\mathscr{L}_\theta^n)^*$. Concretely, we have [31]

$$x_+ = \begin{cases} x, & \text{if} \quad x \in \mathscr{L}_\theta^n, \\ 0, & \text{if} \quad x \in -(\mathscr{L}_\theta^n)^*, \\ s, & \text{otherwise}, \end{cases}$$

[†]The projection mapping from $\mathbb{R}^n$ onto Ω, denoted by P_Ω, is defined as $P_\Omega[x] = \arg\min\{\|x - z\| | z \in \Omega\}$.

where

$$s=\begin{bmatrix}\dfrac{x_1+\|x_2\|\tan\theta}{1+\tan^2\theta}\\ \left(\dfrac{x_1+\|x_2\|\tan\theta}{1+\tan^2\theta}\tan\theta\right)\dfrac{x_2}{\|x_2\|}\end{bmatrix},$$

$$x_-=\begin{cases}0, & \text{if} \quad x\in\mathcal{L}_\theta^n,\\ -x, & \text{if} \quad x\in-(\mathcal{L}_\theta^n)^*,\\ e, & \text{otherwise},\end{cases}$$

and

$$e=\begin{bmatrix}-\dfrac{x_1-\|x_2\|\cot\theta}{1+\cot^2\theta}\\ \left(\dfrac{x_1-\|x_2\|\cot\theta}{1+\cot^2\theta}\cot\theta\right)\dfrac{x_2}{\|x_2\|}\end{bmatrix}.$$

For any vector $x\in\mathbb{R}^n$, it can be confirmed that

$$x=x_+-x_- \tag{2.1}$$

and $\langle x_+,x_-\rangle=0$. In addition, the absolute value of $x\in\mathbb{R}^n$ with respect to the CC $\mathcal{L}_\theta^n$ is defined as

$$|x|=x_++x_-. \tag{2.2}$$

According to (2.1) and (2.2), we have the following lemma.

Lemma 2.1. *For any two vectors $x,y\in\mathbb{R}^n$, it holds that*

$$\|\,|x|-|y|\,\|\le\|x-y\|.$$

Proof. It follows from (2.1) and (2.2) that

$$\begin{aligned}&\|x-y\|^2-\|\,|x|-|y|\,\|^2\\ =&\langle x-y,x-y\rangle-\langle|x|-|y|,|x|-|y|\rangle\\ =&x^\intercal x-x^\intercal y-y^\intercal x+y^\intercal y-|x|^\intercal|x|+|x|^\intercal|y|+|y|^\intercal|x|-|y|^\intercal|y|\\ =&2\big(\langle|x|,|y|\rangle-\langle x,y\rangle\big)\\ =&2\big(\langle x_++x_-,y_++y_-\rangle-\langle x_+-x_-,y_+-y_-\rangle\big)\\ =&4\big(\langle x_+,y_-\rangle+\langle x_-,y_+\rangle\big)\ge0,\end{aligned}$$

where the last inequality use the definition of the dual cone and the fact that $x_+,y_+\in\mathcal{L}_\theta^n$ and $x_-,y_-\in(\mathcal{L}_\theta^n)^*$. Then the proof is completed by the fact that $\|x-y\|\ge0$ and $\|\,|x|-|y|\,\|\ge0$. ☐

The proof of Lemma 2.1 is inspired by the proof of [18, Lemma 4.1], which seems to be simpler than that of [12, Proposition 2.2].

In the following, we will explore the expression of $|x|$ associate with CC $\mathcal{L}_\theta^n$. To this end, the following spectral decomposition theorem is included.

Theorem 2.1 (cf. Zhou & Chen [31, Theorem 3.1]). *If $x=(x_1,x_2)\in\mathbb{R}\times\mathbb{R}^{n-1}$, then x can be represented in the form*

$$x=\lambda_1^{(x)}\mu_1^{(x)}+\lambda_2^{(x)}\mu_2^{(x)},$$

where $\lambda_1^{(x)}$, $\lambda_2^{(x)}$ and $\mu_1^{(x)}$, $\mu_2^{(x)}$ are the spectral values and associated spectral vectors of x, viz.

$$\begin{aligned}
&\lambda_1^{(x)}=x_1-\|x_2\|\cot\theta, && \lambda_2^{(x)}=x_1+\|x_2\|\tan\theta,\\
&\mu_1^{(x)}=\frac{1}{1+\cot^2\theta}\begin{pmatrix}1\\ -(\cot\theta)\tilde{\mu}\end{pmatrix}, && \mu_2^{(x)}=\frac{1}{1+\tan^2\theta}\begin{pmatrix}1\\ (\tan\theta)\tilde{\mu}\end{pmatrix}
\end{aligned}$$

with $\tilde{\mu}=x_2/\|x_2\|$ if $x_2\neq 0$, and any vector in $\mathbb{R}^{n-1}$ satisfying $\|\tilde{\mu}\|=1$ if $x_2=0$.

According to the spectral decomposition of x, we have [31]

$$x_+=\left(\lambda_1^{(x)}\right)_+\mu_1^{(x)}+\left(\lambda_2^{(x)}\right)_+\mu_2^{(x)}$$

and [19]

$$x_-=\left(\lambda_1^{(x)}\right)_-\mu_1^{(x)}+\left(\lambda_2^{(x)}\right)_-\mu_2^{(x)},$$

where $\lambda_+=\max\{0,\lambda\}$ and $\lambda_-=\max\{0,-\lambda\}$ for $\lambda\in\mathbb{R}$. Then

$$\begin{aligned}
|x|&=\left[\left(\lambda_1^{(x)}\right)_++\left(\lambda_1^{(x)}\right)_-\right]\mu_1^{(x)}+\left[\left(\lambda_2^{(x)}\right)_++\left(\lambda_2^{(x)}\right)_-\right]\mu_2^{(x)}\\
&=\left|\lambda_1^{(x)}\right|\mu_1^{(x)}+\left|\lambda_2^{(x)}\right|\mu_2^{(x)}.
\end{aligned}$$

More exactly,

$$|x|=\begin{cases}\begin{bmatrix}|x_1|\\ 0\end{bmatrix}, & \text{if}\quad x_2=0,\\[2ex] \begin{bmatrix}\dfrac{|x_1-\|x_2\|\cot\theta|}{1+\cot^2\theta}+\dfrac{|x_1+\|x_2\|\tan\theta|}{1+\tan^2\theta}\\ \left(\dfrac{|x_1+\|x_2\|\tan\theta|}{1+\tan^2\theta}\tan\theta-\dfrac{|x_1-\|x_2\|\cot\theta|}{1+\cot^2\theta}\cot\theta\right)\dfrac{x_2}{\|x_2\|}\end{bmatrix}, & \text{if}\quad x_2\neq 0.\end{cases}\tag{2.3}$$

By some calculation, it can be shown that $|\cdot|$ defined as in (2.2) is equal to that defined as in (2.3).

Before ending this section, we will give a sufficient condition for the unique solvability of CCGAVE (1.2). In [19], the authors pointed out that CCGAVE (1.2) has at least one solution for any $b\in\mathbb{R}^n$ provided that all singular values of $A\in\mathbb{R}^{n\times n}$ exceed the maximal singular value of $B\in\mathbb{R}^{n\times n}$ (or equivalently, $\sigma_{\max}(B)<\sigma_{\min}(A)$). Note the same result is obtained for SOCGAVE in [9, Corollary 3.2]. In the following, as a direct extension of the result in [11, Proposition 2.3], we will prove that CCGAVE (1.2) has a unique solution for any $b\in\mathbb{R}^n$ if $\sigma_{\max}(B)<\sigma_{\min}(A)$.

Theorem 2.2. *If* $\sigma_{\max}(B) < \sigma_{\min}(A)$*, then CCGAVE* (1.2) *has a unique solution for any* $b \in \mathbb{R}^n$.

Proof. Since $\sigma_{\max}(B) < \sigma_{\min}(A)$, the matrix A is nonsingular and we can define

$$\mathscr{H}(x) = x - \varepsilon A^{-1} c(x),$$

where $c(x) = Ax - B|x| - b$ and $\varepsilon \in (0,1)$. Then for any $b \in \mathbb{R}^n$, $x^* \in \mathbb{R}^n$ is a solution of CCGAVE (1.2) if and only if $x^* = \mathscr{H}(x^*)$. Hence, the result holds if the mapping $\mathscr{H}$ has a unique fixed-point in $\mathbb{R}^n$ for any $b \in \mathbb{R}^n$. According to the Banach fixed-point theorem [6, p. 144], for any $b \in \mathbb{R}^n$, one only has to show that the mapping $\mathscr{H}$ is contractive in $\mathbb{R}^n$. Using the triangle inequality and Lemma 2.1 gives

$$\begin{aligned}
&\|\mathscr{H}(x) - \mathscr{H}(y)\| \\
&= \left\| x - \varepsilon A^{-1}(Ax - B|x| - b) - y + \varepsilon A^{-1}(Ay - B|y| - b) \right\| \\
&= \left\| (1-\varepsilon)(x-y) + \varepsilon A^{-1} B(|x| - |y|) \right\| \\
&\le (1-\varepsilon)\|x-y\| + \varepsilon \|A^{-1}B\| \|x-y\| \\
&= \left(1 - \varepsilon + \varepsilon \|A^{-1}B\|\right) \|x-y\|
\end{aligned}$$

for all $x, y \in \mathbb{R}^n$. Thus the mapping $\mathscr{H}$ is contractive in $\mathbb{R}^n$ if $\|A^{-1}B\| < 1$, which is true whenever $\sigma_{\max}(B) < \sigma_{\min}(A)$. □

3. RTNMS Iteration Method

In this section, the RTNMS iteration method for solving CCGAVE (1.2) is suggested. To this end, inspired by [30], we split the matrix A as

$$A = M_1 - N_1 = M_2 - N_2,$$

and introduce two nonnegative relaxation parameters φ_1 and φ_2. Let

$$F(x) = Ax - B|x| - b,$$

then we have

$$F(x) = G_1(x) + T_1(x) = G_2(x) + T_2(x),$$

where

$$\begin{aligned}
G_1(x) &= \Omega x + \varphi_1 M_1 x, \\
G_2(x) &= \Omega x + \varphi_2 M_2 x
\end{aligned}$$

are differentiable,

$$\begin{aligned}
T_1(x) &= -\Omega x - N_1 x - (\varphi_1 - 1) M_1 x - B|x| - b, \\
T_2(x) &= -\Omega x - N_2 x - (\varphi_2 - 1) M_2 x - B|x| - b
\end{aligned}$$

are non-differentiable, $\Omega \in \mathbb{R}^{n\times n}$ is a given matrix such that $\Omega + \varphi_1 M_1$ and $\Omega + \varphi_2 M_2$ are nonsingular. It follows from $F(x) = 0$ that

$$(\varphi_1 M_1 + \Omega)x = \big(N_1 + (\varphi_1 - 1)M_1 + \Omega\big)x + B|x| + b,$$
$$(\varphi_2 M_2 + \Omega)x = \big(N_2 + (\varphi_2 - 1)M_2 + \Omega\big)x + B|x| + b,$$

from which the following RTNMS iteration method for solving CCGAVE (1.2) is established.

Algorithm 3.1 RTNMS Iteration Method.

1: Let $A = M_1 - N_1 = M_2 - N_2$ be two splittings of the matrix $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$.
2: Assume that $x^{(0)} \in \mathbb{R}^n$ is an arbitrary initial guess.
3: **for** $k = 0, 1, 2, \ldots$ until the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ is convergent **do**
4: Compute $x^{(k+1)} \in \mathbb{R}^n$ by

$$\begin{aligned} x^{(k+1/2)} &= (\varphi_1 M_1 + \Omega)^{-1}\big[\big(N_1 + (\varphi_1 - 1)M_1 + \Omega\big)x^{(k)} + B|x^{(k)}| + b\big], \\ x^{(k+1)} &= (\varphi_2 M_2 + \Omega)^{-1}\big[\big(N_2 + (\varphi_2 - 1)M_2 + \Omega\big)x^{(k+1/2)} + B|x^{(k+1/2)}| + b\big], \end{aligned} \tag{3.1}$$

where $\varphi_1,\ \varphi_2 \geq 0$ are nonnegative relaxation parameters, and Ω is a given matrix such that $\varphi_1 M_1 + \Omega$ and $\varphi_2 M_2 + \Omega$ are nonsingular.
5: **end for**

To the best of our knowledge, the only method proposed in the literature for solving CCGAVE (1.2) is the generalized Newton (GN) method [19]. In this sense, the RTNMS iteration method gives a new general framework for solving CCGAVE (1.2), which contains some extensions of the existing methods for solving the GAVE (see the following remarks for more details). Particularly, if $\varphi_1 = \varphi_2 = 1$, the RTNMS iteration method will be called as the TNMS iteration method. By suitably selecting the matrix splittings, it can also generate some relaxation versions of the RTNMS iteration method, see Remarks 3.1-3.4.

Remark 3.1. Let $A = D - L - U$, where D is the diagonal part of A, and $-L$ and $-U$ are the strictly lower-triangular and the strictly upper-triangular parts of A, respectively. If

$$\begin{aligned} M_1 &= \frac{1}{\alpha}(D - \beta L), \quad N_1 = \frac{1}{\alpha}\big[(1-\alpha)D + (\alpha - \beta)L + \alpha U\big], \\ M_2 &= \frac{1}{\alpha}(D - \beta U), \quad N_2 = \frac{1}{\alpha}\big[(1-\alpha)D + (\alpha - \beta)U + \alpha L\big], \end{aligned}$$

then (3.1) leads to the relaxation two-step Newton-based accelerated overrelaxation (RTNAOR) iteration scheme

$$\begin{aligned} x^{(k+1/2)} &= (\varphi_1 D - \varphi_1 \beta L + \alpha\Omega)^{-1} \\ &\quad \times \big[(\alpha\Omega + (\varphi_1 - \alpha)D - (\varphi_1\beta - \alpha)L + \alpha U)x^{(k)} + \alpha(B|x^{(k)}| + b)\big], \\ x^{(k+1)} &= (\varphi_2 D - \varphi_2 \beta U + \alpha\Omega)^{-1} \\ &\quad \times \big[(\alpha\Omega + (\varphi_2 - \alpha)D - (\varphi_2\beta - \alpha)U + \alpha L)x^{(k+1/2)} + \alpha(B|x^{(k+1/2)}| + b)\big]. \end{aligned}$$

When $\alpha = \beta$ and $\alpha = \beta = 1$, the RTNAOR iteration scheme reduces to the relaxation two-step Newton-based successive overrelaxation (RTNSOR) iteration scheme and the relaxation two-step Newton-based Gauss-Seidel (RTNGS) iteration scheme, respectively.

Remark 3.2. Let $A = M - N$. If $M_1 = M_2 = M$, $N_1 = N_2 = N$, $\varphi_1 = \varphi_2 = \varphi \geq 0$, then (3.1) reduces to the RNMS iteration scheme

$$x^{(k+1)} = (\varphi M + \Omega)^{-1}\big[(N + (\varphi - 1)M + \Omega)x^{(k)} + B|x^{(k)}| + b\big]. \tag{3.2}$$

The RNMS iteration scheme can be seen as an extension of the scheme proposed in [29], in which the GAVE is concerned. In addition, in the context of the GAVE, the method proposed in [29] includes the NMS iteration method [30], the Picard iteration method [24], the MN iteration method [26], and the SS iteration method [14] as special cases. These methods can also be extended to solve CCGAVE (1.2), which are special cases of the RTNMS iteration method.

Remark 3.3. Let $M_1 = M_2 = A$, $N_1 = N_2 = 0$ and $\varphi_1 = \varphi_2 = \varphi \geq 0$, then (3.1) reduces to the RMN iteration scheme

$$x^{(k+1)} = (\varphi A + \Omega)^{-1}\big[\big((\varphi - 1)A + \Omega\big)x^{(k)} + B|x^{(k)}| + b\big],$$

which is an extension of the scheme proposed in [25].

Remark 3.4. Let $M_1 = M_2 = A$, $N_1 = N_2 = 0$, $\Omega = (2/\gamma - 1)A$ with $\gamma \in (0,2)$ and $\varphi_1 = \varphi_2 = 1$, then (3.1) with $B = I$ translates into the DRs iteration scheme

$$x^{(k+1)} = \left(1 - \frac{1}{2}\gamma\right)x^{(k)} + \frac{1}{2}\gamma A^{-1}\left(|x^{(k)}| + b\right),$$

which is an extension of the scheme proposed in [3].

4. Convergence Analysis

In this section, we will establish the convergence analysis of the RTNMS iteration method for solving CCGAVE (1.2).

Theorem 4.1. *Let $A, B \in \mathbb{R}^{n\times n}$ and $A = M_1 - N_1 = M_2 - N_2$ be two splittings of the matrix A. Assume that φ_1, $\varphi_2 \geq 0$ and $\Omega \in \mathbb{R}^{n\times n}$ is such that the matrices $\varphi_1 M_1 + \Omega$, $\varphi_2 M_2 + \Omega$ and $\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A$ are nonsingular. Define*

$$\begin{aligned}
&\eta_1(\Omega) = \left\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1 + \Omega)\right\|, && \xi_1(\Omega) = \left\|(\varphi_1 M_1 + \Omega)^{-1}B\right\|,\\
&\eta_2(\Omega) = \left\|(\varphi_2 M_2 + \Omega)^{-1}(N_2 + (\varphi_2 - 1)M_2 + \Omega)\right\|, && \xi_2(\Omega) = \left\|(\varphi_2 M_2 + \Omega)^{-1}B\right\|,\\
&\delta = \big(\eta_1(\Omega) + \xi_1(\Omega)\big)\big(\eta_2(\Omega) + \xi_2(\Omega)\big), && \rho = \left\|(\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)^{-1}B\right\|.
\end{aligned}$$

If

$$\max\{\delta, \rho\} < 1, \tag{4.1}$$

then for any $b \in \mathbb{R}^n$, CCGAVE (1.2) *has a unique solution x^* and the sequence generated by Algorithm* 3.1 *converges to x^* for any initial vector $x^{(0)} \in \mathbb{R}^n$.*

Proof. To begin with, we prove that $\{x^{(k)}\}_{k=0}^{\infty}$ and $\{x^{(k+1/2)}\}_{k=0}^{\infty}$ are Cauchy sequences. It follows from (3.1) that

$$\begin{aligned}
x^{(k-1/2)} &= (\varphi_1 M_1+\Omega)^{-1}\big[(N_1+(\varphi_1-1)M_1+\Omega)x^{(k-1)}+B|x^{(k-1)}|+b\big],\\
x^{(k)} &= (\varphi_2 M_2+\Omega)^{-1}\big[(N_2+(\varphi_2-1)M_2+\Omega)x^{(k-1/2)}+B|x^{(k-1/2)}|+b\big].
\end{aligned} \tag{4.2}$$

Subtracting (4.2) from (3.1), we get

$$\begin{aligned}
&x^{(k+1/2)}-x^{(k-1/2)} \qquad\qquad (4.3a)\\
&=(\varphi_1 M_1+\Omega)^{-1}\big[\big(N_1+(\varphi_1-1)M_1+\Omega\big)\big(x^{(k)}-x^{(k-1)}\big)+B\big(|x^{(k)}|-|x^{(k-1)}|\big)\big],\\
&x^{(k+1)}-x^{(k)} \qquad\qquad (4.3b)\\
&=(\varphi_2 M_2+\Omega)^{-1}\big[\big(N_2+(\varphi_2-1)M_2+\Omega\big)\big(x^{(k+1/2)}-x^{(k-1/2)}\big)+B\big(|x^{(k+1/2)}|-|x^{(k-1/2)}|\big)\big].
\end{aligned}$$

Taking the Euclidean norm on both sides of Eq. (4.3a) and using the triangle inequality and Lemma 2.1, we can conclude that

$$\begin{aligned}
&\|x^{(k+1/2)}-x^{(k-1/2)}\|\\
&=\Big\|(\varphi_1 M_1+\Omega)^{-1}\big[\big(N_1+(\varphi_1-1)M_1+\Omega\big)(x^{(k)}-x^{(k-1)})+B\left(|x^{(k)}|-|x^{(k-1)}|\right)\big]\Big\|\\
&=\Big\|(\varphi_1 M_1+\Omega)^{-1}\big(N_1+(\varphi_1-1)M_1+\Omega\big)(x^{(k)}-x^{(k-1)})\\
&\qquad+(\varphi_1 M_1+\Omega)^{-1}B\left(|x^{(k)}|-|x^{(k-1)}|\right)\Big\|\\
&\le\Big\|(\varphi_1 M_1+\Omega)^{-1}\big(N_1+(\varphi_1-1)M_1+\Omega\big)(x^{(k)}-x^{(k-1)})\Big\|\\
&\qquad+\Big\|(\varphi_1 M_1+\Omega)^{-1}B\left(|x^{(k)}|-|x^{(k-1)}|\right)\Big\|\\
&\le\Big\|(\varphi_1 M_1+\Omega)^{-1}\big(N_1+(\varphi_1-1)M_1+\Omega\big)\Big\|\cdot\|x^{(k)}-x^{(k-1)}\|\\
&\qquad+\Big\|(\varphi_1 M_1+\Omega)^{-1}B\Big\|\cdot\Big\||x^{(k)}|-|x^{(k-1)}|\Big\|\\
&\le\Big\|(\varphi_1 M_1+\Omega)^{-1}\big(N_1+(\varphi_1-1)M_1+\Omega\big)\Big\|\cdot\|x^{(k)}-x^{(k-1)}\|\\
&\qquad+\|(\varphi_1 M_1+\Omega)^{-1}B\|\cdot\|x^{(k)}-x^{(k-1)}\|\\
&=\big(\|(\varphi_1 M_1+\Omega)^{-1}\big(N_1+(\varphi_1-1)M_1+\Omega\big)\|+\|(\varphi_1 M_1+\Omega)^{-1}B\|\big)\cdot\|x^{(k)}-x^{(k-1)}\|\\
&=\big(\eta_1(\Omega)+\xi_1(\Omega)\big)\|x^{(k)}-x^{(k-1)}\|.
\end{aligned}\tag{4.4}$$

Exploiting a similar strategy, we obtain from Eq. (4.3b) that

$$\begin{aligned}
\|x^{(k+1)}-x^{(k)}\| &\le\Big(\Big\|(\varphi_2 M_2+\Omega)^{-1}\big(N_2+(\varphi_2-1)M_2+\Omega\big)\Big\|+\|(\varphi_2 M_2+\Omega)^{-1}B\|\Big)\\
&\qquad\times\|x^{(k+1/2)}-x^{(k-1/2)}\|\\
&=\big(\eta_2(\Omega)+\xi_2(\Omega)\big)\|x^{(k+1/2)}-x^{(k-1/2)}\|.
\end{aligned}\tag{4.5}$$

As a result, we get

$$\|x^{(k+1)}-x^{(k)}\|\le\delta\|x^{(k)}-x^{(k-1)}\|.$$

Then for each $p \geq 1$, we can deduce that

$$\begin{aligned}
&\|x^{(k+p)} - x^{(k)}\| \\
\leq &\sum_{i=1}^{p} \|x^{(k+i)} - x^{(k+i-1)}\| \\
\leq &(\delta^{p-1} + \cdots + 1)\|x^{(k+1)} - x^{(k)}\| \\
\leq &\frac{\delta^k(1-\delta^p)}{1-\delta}\|x^{(1)} - x^{(0)}\| \\
\leq &\frac{\delta^k}{1-\delta}\|x^{(1)} - x^{(0)}\|.
\end{aligned}$$

Since $\delta^k/(1-\delta) \to 0$ as $k \to \infty$ due to (4.1), we obtain that $\{x^{(k)}\}_{k=0}^{\infty}$ is a Cauchy sequence. Conversely, it follows from (4.5) that

$$\|x^{(k)} - x^{(k-1)}\| \leq \big(\eta_2(\Omega) + \xi_2(\Omega)\big)\|x^{(k-1/2)} - x^{(k-3/2)}\|.$$

Combining it with (4.4) yields

$$\|x^{(k+1/2)} - x^{(k-1/2)}\| \leq \delta\|x^{(k-1/2)} - x^{(k-3/2)}\|.$$

Similarly, for each $p \geq 1$, we get

$$\begin{aligned}
&\|x^{(k+1/2+p)} - x^{(k+1/2)}\| \\
\leq &\sum_{i=1}^{p} \|x^{(k+1/2+i)} - x^{(k-1/2+i)}\| \\
\leq &(\delta^{p-1} + \cdots + 1)\|x^{(k+3/2)} - x^{(k+1/2)}\| \\
\leq &\frac{\delta^k(1-\delta^p)}{1-\delta}\|x^{3/2} - x^{1/2}\| \\
\leq &\frac{\delta^k}{1-\delta}\|x^{3/2} - x^{1/2}\|.
\end{aligned}$$

Since $\delta^k/(1-\delta) \to 0$ as $k \to \infty$ due to (4.1), this shows that $\{x^{(k+1/2)}\}_{k=0}^{\infty}$ is also a Cauchy sequence.

Let $x^{(k)} \to x^*$ and $x^{(k+1/2)} \to \tilde{x}^*$ as $k \to \infty$. We prove $x^* = \tilde{x}^*$ by contradiction. Taking into account (4.2) with $k \to \infty$, we have

$$(\varphi_1 M_1 + \Omega)\tilde{x}^* = \big(N_1 + (\varphi_1 - 1)M_1 + \Omega\big)x^* + B|x^*| + b, \tag{4.6}$$
$$(\varphi_2 M_2 + \Omega)x^* = \big(N_2 + (\varphi_2 - 1)M_2 + \Omega\big)\tilde{x}^* + B|\tilde{x}^*| + b. \tag{4.7}$$

Subtracting (4.7) from (4.6), and combining with $A = M_1 - N_1 = M_2 - N_2$, we deduce that

$$\begin{aligned}
&B(|x^*| - |\tilde{x}^*|) \\
= &(\varphi_1 M_1 + \Omega)\tilde{x}^* - (\varphi_2 M_2 + \Omega)x^* - \big(N_1 + (\varphi_1 - 1)M_1 + \Omega\big)x^* + \big(N_2 + (\varphi_2 - 1)M_2 + \Omega\big)\tilde{x}^*
\end{aligned}$$

$$
\begin{aligned}
&= \big(\varphi_1 M_1 + \Omega + N_2 + (\varphi_2 - 1)M_2 + \Omega\big)\tilde{x}^* - \big(\varphi_2 M_2 + \Omega + N_1 + (\varphi_1 - 1)M_1 + \Omega\big)x^* \\
&= (\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)\tilde{x}^* - (\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)x^* \\
&= (\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)(\tilde{x}^* - x^*).
\end{aligned}
$$

Then based on the assumption, we have

$$
\tilde{x}^* - x^* = (\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)^{-1} B(|x^*| - |\tilde{x}^*|). \tag{4.8}
$$

Taking the Euclidean norm on both sides of (4.8), by the condition (4.1) and Lemma 2.1, it follows that

$$
\begin{aligned}
\|\tilde{x}^* - x^*\| &= \|(\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)^{-1} B(|x^*| - |\tilde{x}^*|)\| \\
&\le \|(\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)^{-1} B\| \cdot \|\,|\tilde{x}^*| - |x^*|\,\| \\
&\le \|(\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A)^{-1} B\| \cdot \|\tilde{x}^* - x^*\| \\
&< \|\tilde{x}^* - x^*\|,
\end{aligned}
$$

which is a contradiction if $\tilde{x}^* \ne x^*$. Therefore, the sequence generated by Algorithm 3.1 converges to x^* and it follows from (4.6) and (4.7) that

$$
Ax^* - B|x^*| - b = 0,
$$

that is, x^* is a solution of CCGAVE (1.2).

Next, we prove the uniqueness of the solution for CCGAVE (1.2). If y^* is another solution of CCGAVE (1.2) and $y^* \ne x^*$, we have

$$
Ay^* - B|y^*| - b = 0.
$$

Furthermore, we obtain

$$
y^* = (\varphi_1 M_1 + \Omega)^{-1}\big((N_1 + (\varphi_1 - 1)M_1 + \Omega)y^* + B|y^*| + b\big), \tag{4.9}
$$

$$
y^* = (\varphi_2 M_2 + \Omega)^{-1}\big((N_2 + (\varphi_2 - 1)M_2 + \Omega)y^* + B|y^*| + b\big). \tag{4.10}
$$

Subtracting (4.9) from (4.6) and (4.10) from (4.7) (keep $x^* = \tilde{x}^*$ in mind), then taking the Euclidean norm on both sides of the first and second equations, we get

$$
\begin{aligned}
\|x^* - y^*\| &\le \big(\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1 + \Omega)\| + \|(\varphi_1 M_1 + \Omega)^{-1} B\|\big) \cdot \|x^* - y^*\| \\
&= \big(\eta_1(\Omega) + \xi_1(\Omega)\big)\|x^* - y^*\|, \\
\|x^* - y^*\| &\le \big(\|(\varphi_2 M_2 + \Omega)^{-1}(N_2 + (\varphi_2 - 1)M_2 + \Omega)\| + \|(\varphi_2 M_2 + \Omega)^{-1} B\|\big) \cdot \|x^* - y^*\| \\
&= \big(\eta_2(\Omega) + \xi_2(\Omega)\big)\|x^* - y^*\|.
\end{aligned}
$$

Under condition (4.1), we conclude that

$$
\|x^* - y^*\| \le \big(\eta_1(\Omega) + \xi_1(\Omega)\big)\big(\eta_2(\Omega) + \xi_2(\Omega)\big)\|x^* - y^*\| < \|x^* - y^*\|,
$$

which contradicts to the assumption $y^* \ne x^*$. This completes the proof. $\square$

The next corollary follows from the estimates

$$\begin{aligned}&\|(\varphi_1 M_1+\Omega)^{-1}(N_1+(\varphi_1-1)M_1+\Omega)\|+\|(\varphi_1 M_1+\Omega)^{-1}B\|\\ \le\ &\|(\varphi_1 M_1+\Omega)^{-1}\|(\|N_1+(\varphi_1-1)M_1+\Omega\|+\|B\|),\\ &\|(\varphi_2 M_2+\Omega)^{-1}(N_2+(\varphi_2-1)M_2+\Omega)\|+\|(\varphi_2 M_2+\Omega)^{-1}B\|\\ \le\ &\|(\varphi_2 M_2+\Omega)^{-1}\|(\|N_2+(\varphi_2-1)M_2+\Omega\|+\|B\|).\end{aligned}$$

Corollary 4.1. *Let $A,B\in\mathbb{R}^{n\times n}$ and $A=M_1-N_1=M_2-N_2$ be two splittings of the matrix A. Assume that $\varphi_1,\varphi_2\ge 0$ and $\Omega\in\mathbb{R}^{n\times n}$ is such that the matrices $\varphi_1 M_1+\Omega$, $\varphi_2 M_2+\Omega$ and $\varphi_1 M_1+\varphi_2 M_2+2\Omega-A$ are nonsingular. Define*

$$\begin{aligned}\tilde{\delta}=\ &\|(\varphi_1 M_1+\Omega)^{-1}\|\cdot\|(\varphi_2 M_2+\Omega)^{-1}\|\\ &\times\big(\|N_1+(\varphi_1-1)M_1+\Omega\|+\|B\|\big)\\ &\times\big(\|N_2+(\varphi_2-1)M_2+\Omega\|+\|B\|\big).\end{aligned}$$

If

$$\max\{\tilde{\delta},\rho\}<1,$$

then for any $b\in\mathbb{R}^n$, CCGAVE (1.2) *has a unique solution x^* and the sequence generated by Algorithm* 3.1 *converges to x^* for any initial vector $x^{(0)}\in\mathbb{R}^n$. Here, ρ is defined as in Theorem* 4.1.

According to the Banach perturbation lemma [7, Lemma 2.3.3.], we have the following corollary as well.

Corollary 4.2. *Let $A,B\in\mathbb{R}^{n\times n}$ and $A=M_1-N_1=M_2-N_2$ be two splittings of the matrix A, where M_1 and M_2 are nonsingular, φ_1, $\varphi_2>0$ are positive relaxation parameters. Assume that the matrix $\Omega\in\mathbb{R}^{n\times n}$ is a positive semi-definite matrix such that $\varphi_1 M_1+\Omega$, $\varphi_2 M_2+\Omega$ and $\varphi_1 M_1+\varphi_2 M_2+2\Omega-A$ are nonsingular. If*

$$\begin{aligned}\|(\varphi_1 M_1)^{-1}\|&<\frac{1}{\|\Omega\|+\|N_1+(\varphi_1-1)M_1+\Omega\|+\|B\|},\\ \|(\varphi_2 M_2)^{-1}\|&<\frac{1}{\|\Omega\|+\|N_2+(\varphi_2-1)M_2+\Omega\|+\|B\|},\end{aligned}\tag{4.11}$$

and $\rho<1$, then for any $b\in\mathbb{R}^n$, CCGAVE (1.2) *has a unique solution x^* and the sequence generated by Algorithm* 3.1 *converges to x^* for any initial vector $x^{(0)}\in\mathbb{R}^n$. Here, ρ is defined as in Theorem* 4.1.

Proof. In the light of the Banach perturbation lemma [7, Lemma 2.3.3.] and (4.11), we have

$$\begin{aligned}&\|(\varphi_1 M_1+\Omega)^{-1}\|\|(\varphi_2 M_2+\Omega)^{-1}\|\\ \le\ &\|(\varphi_1 M_1)^{-1}\|\|(\varphi_2 M_2)^{-1}\|\|(I+(\varphi_1 M_1)^{-1}\Omega)^{-1}\|\|(I+(\varphi_2 M_2)^{-1}\Omega)^{-1}\|\\ \le\ &\frac{\|(\varphi_1 M_1)^{-1}\|\|(\varphi_2 M_2)^{-1}\|}{(1-\|(\varphi_1 M_1)^{-1}\|\|\Omega\|)(1-\|(\varphi_2 M_2)^{-1}\|\|\Omega\|)}\\ <\ &\frac{1}{(\|N_1+(\varphi_1-1)M_1+\Omega\|+\|B\|)(\|N_2+(\varphi_2-1)M_2+\Omega\|+\|B\|)}.\end{aligned}$$

Therefore, the conclusion is drawn from Corollary 4.1. This completes the proof. $\square$

In particular, when $M_1, M_2 \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices and $\Omega = \omega I$ is a positive scalar matrix, then the following theorem can be obtained.

Theorem 4.2. *Let $A, B \in \mathbb{R}^{n\times n}$ and $A = M_1 - N_1 = M_2 - N_2$ be two splittings of the matrix A, where M_1, M_2 are symmetric positive definites. Assume that $\Omega = \omega I$ with $\omega > 0$ such that $\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A$ is nonsingular, $\varphi_1, \varphi_2 > 0$ are positive relaxation parameters. Let $\iota_{\min}, \iota_{\max}$ be the smallest and the largest eigenvalues of the matrix M_1, and $\kappa_{\min}, \kappa_{\max}$ be the smallest and the largest eigenvalues of the matrix M_2. Define $\|B\| = \vartheta$, $\psi_1 = \|M_1^{-1}N_1\|$, $\psi_2 = \|M_2^{-1}N_2\|$ and*

$$\hat{\delta} = \left(\frac{\iota_{\max}|\psi_1 + \varphi_1 - 1| + \omega + \vartheta}{\omega + \varphi_1 \iota_{\min}}\right)\left(\frac{\kappa_{\max}|\psi_2 + \varphi_2 - 1| + \omega + \vartheta}{\omega + \varphi_2 \kappa_{\min}}\right).$$

If

$$\max\{\hat{\delta}, \rho\} < 1, \tag{4.12}$$

then for any $b \in \mathbb{R}^n$, CCGAVE (1.2) *has a unique solution x^* and the sequence generated by Algorithm* 3.1 *converges to x^* for any initial vector $x^{(0)} \in \mathbb{R}^n$. Here, ρ is defined as in Theorem* 4.1.

Proof. According to Theorem 4.1, we just need to verify the sufficient condition (4.1). Since

$$\begin{aligned}
&\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1 + \Omega)\| \\
\le\ &\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1)\| + \|(\varphi_1 M_1 + \Omega)^{-1}\Omega\|, \\
&\|(\varphi_2 M_2 + \Omega)^{-1}(N_2 + (\varphi_2 - 1)M_2 + \Omega)\| \\
\le\ &\|(\varphi_2 M_2 + \Omega)^{-1}(N_2 + (\varphi_2 - 1)M_2)\| + \|(\varphi_2 M_2 + \Omega)^{-1}\Omega\|,
\end{aligned}$$

one has to show that

$$\begin{aligned}
&\big(\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1)\| + \|(\varphi_1 M_1 + \Omega)^{-1}\Omega\| + \|(\varphi_1 M_1 + \Omega)^{-1}B\|\big) \\
&\quad \times \big(\|(\varphi_2 M_2 + \Omega)^{-1}(N_2 + (\varphi_2 - 1)M_2)\| + \|(\varphi_2 M_2 + \Omega)^{-1}\Omega\| + \|(\varphi_2 M_2 + \Omega)^{-1}B\|\big) < 1.
\end{aligned}$$

Since M_1 is a symmetric positive definite matrix and $\Omega = \omega I$ a positive scalar matrix, simple calculations give

$$\begin{aligned}
&\|(\varphi_1 M_1 + \Omega)^{-1}(N_1 + (\varphi_1 - 1)M_1)\| \\
=\ &\|(\varphi_1 M_1 + \Omega)^{-1} M_1 M_1^{-1}(N_1 + (\varphi_1 - 1)M_1)\| \\
\le\ &\|(\varphi_1 M_1 + \Omega)^{-1} M_1\| \|M_1^{-1}(N_1 + (\varphi_1 - 1)M_1)\| \\
=\ &\|(\varphi_1 M_1 + \omega I)^{-1} M_1\| \|M_1^{-1}N_1 + \varphi_1 - 1\| \\
=\ &\max_{\iota \in \mathrm{sp}(M_1)} \frac{\iota|\psi_1 + \varphi_1 - 1|}{\omega + \varphi_1 \iota} \le \frac{\iota_{\max}|\psi_1 + \varphi_1 - 1|}{\omega + \varphi_1 \iota_{\min}},
\end{aligned}$$

$$\begin{aligned}
&\|(\varphi_1 M_1+\Omega)^{-1}\Omega\|+\|(\varphi_1 M_1+\Omega)^{-1}B\|\\
\le\ &\|(\varphi_1 M_1+\Omega)^{-1}\|(\|\Omega\|+\|B\|)\\
=\ &(\omega+\vartheta)\|(\varphi_1 M_1+\omega I)^{-1}\|\\
=\ &\max_{\iota\in \mathrm{sp}(M_1)}\frac{\omega+\vartheta}{\omega+\varphi_1\iota}=\frac{\omega+\vartheta}{\omega+\varphi_1\iota_{\min}}.
\end{aligned}$$

Similarly, since M_2 is a symmetric positive definite matrix, we have

$$\begin{aligned}
\|(\varphi_2 M_2+\Omega)^{-1}(N_2+(\varphi_2-1)M_2)\|&\le\frac{\kappa_{\max}|\psi_2+\varphi_2-1|}{\omega+\varphi_2\kappa_{\min}},\\
\|(\varphi_2 M_2+\Omega)^{-1}\Omega\|+\|(\varphi_2 M_2+\Omega)^{-1}B\|&\le\frac{\omega+\vartheta}{\omega+\varphi_2\kappa_{\min}}.
\end{aligned}$$

Hence, we just require that

$$\left(\frac{\iota_{\max}|\psi_1+\varphi_1-1|}{\omega+\varphi_1\iota_{\min}}+\frac{\omega+\vartheta}{\omega+\varphi_1\iota_{\min}}\right)\left(\frac{\kappa_{\max}|\psi_2+\varphi_2-1|}{\omega+\varphi_2\kappa_{\min}}+\frac{\omega+\vartheta}{\omega+\varphi_2\kappa_{\min}}\right)<1.$$

Under condition (4.12), we can obtain the condition (4.1). The proof is complete. □

Remark 4.1. Since some spectral norms or eigenvalues need to be computed, the convergence conditions provided in the above theorems and corollaries are generally not easy to check in practice, especially for large-scale problems. The following example demonstrates that the conditions of Theorem 4.1 can be satisfied. Let

$$A=\begin{pmatrix}6&-2\\2&6\end{pmatrix},\quad B=\begin{pmatrix}1&1\\-1&2\end{pmatrix},$$

then consider two methods reported in Remark 3.1. We have

$$D=\begin{pmatrix}6&0\\0&6\end{pmatrix},\quad L=\begin{pmatrix}0&0\\-2&0\end{pmatrix},\quad U=\begin{pmatrix}0&2\\0&0\end{pmatrix}.$$

In addition, let $\varphi_1=1.2$, $\varphi_2=0.8$ and

$$\Omega=\begin{pmatrix}2&1\\1&2\end{pmatrix}.$$

(1) We take $A=M_1-N_1=M_2-N_2$ with

$$\begin{aligned}
M_1&=D-L=\begin{pmatrix}6&0\\2&6\end{pmatrix}, &N_1&=U=\begin{pmatrix}0&2\\0&0\end{pmatrix},\\
M_2&=D-U=\begin{pmatrix}6&-2\\0&6\end{pmatrix}, &N_2&=L=\begin{pmatrix}0&0\\-2&0\end{pmatrix}.
\end{aligned}$$

Then it can be checked that $\varphi_1 M_1+\Omega$, $\varphi_2 M_2+\Omega$ and $\varphi_1 M_1+\varphi_2 M_2+2\Omega-A$ are nonsingular. Furthermore, we obtain $\max\{\delta,\rho\}=\max\{0.4174,0.2317\}<1$. Hence, the conditions of Theorem 4.1 are satisfied.

(2) We take $A = M_1 - N_1 = M_2 - N_2$ with $\alpha = \beta = 1.3$ and

$$M_1 = \frac{1}{\alpha}(D - \beta L) = \begin{pmatrix} 4.6145 & 0 \\ 2 & 4.6145 \end{pmatrix},$$
$$M_2 = \frac{1}{\alpha}(D - \beta U) = \begin{pmatrix} 4.6154 & -2 \\ 0 & 4.6154 \end{pmatrix},$$
$$N_1 = \frac{1}{\alpha}[(1-\alpha)D + (\alpha - \beta)L + \alpha U] = \begin{pmatrix} -1.3846 & 2 \\ 0 & -1.3846 \end{pmatrix},$$
$$N_2 = \frac{1}{\alpha}[(1-\alpha)D + (\alpha - \beta)U + \alpha L] = \begin{pmatrix} -1.3846 & 0 \\ -2 & -1.3846 \end{pmatrix}.$$

Then it can be checked that $\varphi_1 M_1 + \Omega$, $\varphi_2 M_2 + \Omega$ and $\varphi_1 M_1 + \varphi_2 M_2 + 2\Omega - A$ are nonsingular. In addition, we obtain $\max\{\delta, \rho\} = \max\{0.4825, 0.3484\} < 1$. Hence, the conditions of Theorem 4.1 are also satisfied.

Before ending this section, we present a convergence theorem for the RNMS iteration method (see Remark 3.2 for the details). The proof is similar to that of Theorem 4.1 and is omitted.

Theorem 4.3. *Let $A, B \in \mathbb{R}^{n\times n}$ and $A = M - N$ be a splitting of the matrix A. Assume that $\varphi \geq 0$ and $\Omega \in \mathbb{R}^{n\times n}$ is such that the matrix $\varphi M + \Omega$ is nonsingular. Define*

$$\eta(\Omega) = \|(\varphi M + \Omega)^{-1}(N + (\varphi - 1)M + \Omega)\|, \quad \xi(\Omega) = \|(\varphi M + \Omega)^{-1} B\|.$$

If

$$\delta' = \eta(\Omega) + \xi(\Omega) < 1, \tag{4.13}$$

then for any $b \in \mathbb{R}^n$, CCGAVE (1.2) has a unique solution x^* and the sequence generated by RNMS iteration scheme (3.2) converges to x^* for any initial vector $x^{(0)} \in \mathbb{R}^n$.

Remark 4.2. In the proof of [29, Theorem 3.1], under the solvable assumption, the RNMS iteration method is proved to converge to a solution of the GAVE if

$$\|(\varphi M + \Omega)^{-1}\|(\|N + (\varphi - 1)M + \Omega\| + \|B\|) < 1. \tag{4.14}$$

However, the unique solvability of the GAVE is not explored in [29]. It can be checked that (4.14) implies (4.13) but the converse is generally not true. Hence, comparing with [29, Theorem 3.1], we get a stronger conclusion under a weaker condition here. Moreover, by recalling Remark 3.2, the convergence theorem is available for the NMS iteration method, the Picard iteration method, the MN iteration method and the SS iteration method.

5. Numerical Experiments

In this section, we present three numerical examples to demonstrate the effectiveness of the RTNMS iteration method for solving CCGAVE (1.2). In our computations, all runs are implemented in MATLAB R2014b with a machine precision 2.22×10^{-16} on a personal computer with 1.60 GHz central processing unit (Intel(R) Core(TM) i5−8265), 8 GB memory and Windows 10 operating system.

As special cases, the Picard iteration method [24], the MN iteration method [26] and the NMS iteration method [30] are extended to solve CCGAVE (1.2). Among which, the NGS and NSOR iteration methods are tested as representatives of the NMS iteration method. At the same time, the RTNGS and RTNSOR iteration methods are tested as representatives of the RTNMS iteration method. We compare the above mentioned methods with the GN method [19]. In the numerical results, we report the number of iteration steps (denoted by 'IT'), the elapsed CPU time in seconds (denoted by 'CPU') and the relative residual error (denoted by 'RES'). Here, RES is set to be

$$\mathrm{RES} = \frac{\|Ax^{(k)} - B|x^{(k)}| - b\|}{\|b\|},$$

where $x^{(k)}$ is the k-th approximate solution to CCGAVE (1.2). The initial vector is chosen to be

$$x^{(0)} = (1,0,1,0,\ldots,1,0,\ldots)^{\intercal} \in \mathbb{R}^n,$$

and all iterations are terminated once RES $< 10^{-6}$ or the number of the prescribed iteration steps $k_{\max} = 500$ is exceeded.

For the sake of fairness, we use the same matrix $\Omega = (1/2)D$ in the MN, NGS, NSOR, RTNGS and RTNSOR iteration methods, where D is the diagonal part of A. For the NSOR, RTNGS and RTNSOR iteration methods, the experimentally optimal parameters are selected, which leads to the smallest iteration step. For the GN method, the generalized Jacobian matrix of $|x|$ is specialized by t [19], and we choose $t = 0$. The LU factorization is utilized to solve all the linear systems. We choose $\mathscr{L}_\theta = \mathscr{L}_\theta^{n_1} \times \mathscr{L}_\theta^{n_2} \times \cdots \times \mathscr{L}_\theta^{n_r}$ and set $n_1 = \cdots = n_r = n/r$. Note that $\sigma_{\max}(B) < \sigma_{\min}(A)$ is satisfied in the following three numerical examples, CCGAVE (1.2) has a unique solution for any $b \in \mathbb{R}^n$.

Example 5.1. Consider CCGAVE (1.2) with

$$A = \begin{pmatrix} 8 & -1 & -1 & & \\ & 8 & -1 & \ddots & \\ & & 8 & \ddots & -1 \\ & & & \ddots & -1 \\ & & & & 8 \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad B = \begin{pmatrix} 4 & & & & \\ & 4 & & & \\ & & 4 & & \\ & & & \ddots & \\ & & & & 4 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

and $b = Ax^* - B|x^*|$, where $x^* = (-1,1,-1,1,\ldots,-1,1,\ldots)^{\intercal} \in \mathbb{R}^n$.

For this example, we take $r = 1$, $\theta = \pi/16$ and $r = 4$, $\theta = \pi/12$, respectively. Tables 1-2 list the numerical results. According to Tables 1-2, we can observe that the elapsed CPU time of all these test methods increase when the problem size n increases. In addition, we find that all methods are convergent, but the six test methods proposed in this

Table 1: Numerical results for Example 5.1 with $r = 1$, $\theta = \pi/16$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	3	3	3	3
	CPU	2.3811	9.6990	33.4310	122.9172
	RES	5.0929×10^{-7}	3.9990×10^{-7}	3.2869×10^{-7}	2.7883×10^{-7}
Picard	IT	27	27	27	27
	CPU	0.0068	0.0105	0.0252	0.0442
	RES	9.2806×10^{-7}	8.9308×10^{-7}	8.7256×10^{-7}	8.5908×10^{-7}
MN	IT	24	24	24	25
	CPU	0.0057	0.0094	0.0213	0.0384
	RES	6.9032×10^{-7}	8.5012×10^{-7}	9.4612×10^{-7}	6.7344×10^{-7}
NGS	IT	23	24	24	24
	CPU	0.0047	0.0079	0.0176	0.0299
	RES	9.0751×10^{-7}	7.7916×10^{-7}	8.8375×10^{-7}	9.5352×10^{-7}
NSOR	$\alpha_{\exp}$	1.2	1.2	1.2	1.2
	IT	18	16	15	15
	CPU	0.0042	0.0063	0.0103	0.0142
	RES	9.9203×10^{-7}	9.6722×10^{-7}	6.4599×10^{-7}	9.7145×10^{-7}
RTNGS	$\varphi_{1\exp}$	0.9	0.9	0.9	0.9
	$\varphi_{2\exp}$	0.3	0.3	0.3	0.3
	IT	6	6	7	7
	CPU	**0.0033**	**0.0054**	**0.0098**	**0.0133**
	RES	3.6382×10^{-7}	8.6086×10^{-7}	2.8110×10^{-7}	3.3520×10^{-7}
RTNSOR	$\alpha_{\exp}$	2.0	2.0	0.6	1.7
	$\varphi_{1\exp}$	1.8	1.7	0.5	1.4
	$\varphi_{2\exp}$	0.7	1.0	0.4	1.0
	IT	6	5	5	5
	CPU	**0.0029**	**0.0049**	**0.0092**	**0.0122**
	RES	8.2383×10^{-7}	7.6161×10^{-7}	1.3230×10^{-7}	5.8776×10^{-7}

paper are better than the GN method in terms of the elapsed CPU time. For the test methods proposed in this paper, the number of iteration steps and the elapsed CPU time of the RTNGS and RTNSOR iteration methods are more efficient than the Picard, MN, NGS and NSOR iteration methods. Furthermore, we check the convergence condition (4.1) in Theorem 4.1 (for RTNGS and RTNSOR) and the convergence condition (4.13) in Theorem 4.3 (for Picard, MN, NGS and NSOR)[‡]. Numerical results in Table 3 demonstrate the convergence condition (4.1) in Theorem 4.1 and the convergence condition (4.13) in Theorem 4.3 are satisfied, where RTNSOR_1 and RTNSOR_2 represent the RTNSOR iteration method for $r = 1$, $\theta = \pi/16$ and $r = 4$, $\theta = \pi/12$, respectively.

[‡]The same goes to the following two examples.

Table 2: Numerical results for Example 5.1 with $r=4$, $\theta=\pi/12$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	4	4	4	4
	CPU	2.7461	11.3115	38.3681	136.7964
	RES	2.2218×10^{-8}	2.5530×10^{-8}	2.7686×10^{-8}	2.9197×10^{-8}
Picard	IT	25	24	24	24
	CPU	0.0089	0.0102	0.0163	0.0378
	RES	6.1412×10^{-7}	9.5237×10^{-7}	9.0013×10^{-7}	8.6667×10^{-7}
MN	IT	24	25	25	25
	CPU	0.0080	0.0117	0.0182	0.0409
	RES	7.2114×10^{-7}	6.8807×10^{-7}	8.1281×10^{-7}	8.9612×10^{-7}
NGS	IT	23	24	25	25
	CPU	0.0071	0.0093	0.0138	0.0269
	RES	8.5061×10^{-7}	9.1274×10^{-7}	7.4707×10^{-7}	8.3960×10^{-7}
NSOR	$\alpha_{\exp}$	1.2	1.2	1.2	1.2
	IT	20	18	14	16
	CPU	0.0048	0.0080	0.0109	0.0185
	RES	8.0920×10^{-7}	8.6258×10^{-7}	8.8741×10^{-7}	8.9638×10^{-7}
RTNGS	$\varphi_{1\exp}$	0.9	0.9	0.9	0.9
	$\varphi_{2\exp}$	0.3	0.3	0.3	0.3
	IT	6	6	7	7
	CPU	**0.0042**	**0.0072**	**0.0103**	**0.0172**
	RES	5.8215×10^{-7}	8.0411×10^{-7}	3.2845×10^{-7}	4.3465×10^{-7}
RTNSOR	$\alpha_{\exp}$	2.0	2.0	2.0	1.8
	$\varphi_{1\exp}$	1.8	1.7	1.7	1.5
	$\varphi_{2\exp}$	2.0	0.8	0.8	0.7
	IT	4	6	6	6
	CPU	**0.0033**	**0.0059**	**0.0096**	**0.0143**
	RES	7.4587×10^{-7}	9.3362×10^{-7}	8.0629×10^{-7}	7.8458×10^{-7}

Table 3: Values of $\max\{\delta,\rho\}$ in (4.1) and δ' in (4.13) for Example 5.1.

Method	n			
	60^2	80^2	100^2	120^2
Picard	0.6667	0.6667	0.6667	0.6667
MN	0.8000	0.8000	0.8000	0.8000
NGS	0.8333	0.8333	0.8333	0.8333
NSOR	0.8125	0.8125	0.8125	0.8125
RTNGS	0.8621	0.8621	0.8621	0.8621
RTNSOR_1	0.7758	0.5432	0.6221	0.5692
RTNSOR_2	0.7071	0.6856	0.6856	0.6964

The CCLCP is to find two vectors $\mu, \nu \in \mathbb{R}^n$ such that

$$\begin{aligned} &P\mu - Q\nu = d, \\ &\mu \in \mathscr{L}_\theta, \quad \nu \in \mathscr{L}_\theta^*, \quad \langle \mu, \nu \rangle = 0, \end{aligned} \tag{5.1}$$

where $P, Q \in \mathbb{R}^{n\times n}$ and $d \in \mathbb{R}^n$. According to [19, Theorem 2.1], if we let $A = P + Q$, $B = Q - P$, $b = 2d$, then CCLCP (5.1) converts to CCGAVE (1.2). Based on this fact, we give the following two examples.

Example 5.2. Consider CCLCP (5.1), let m be a positive integer, $n = m^2$, $Q = \hat{Q} + 2I_n$, $P = \text{blktridiag}(-I_m, W, -I_m) \in \mathbb{R}^{n\times n}$, and $d = (1/2)(P(x^* + |x^*|) + Q(x^* - |x^*|))$, where $\hat{Q} = I_m \otimes W \in \mathbb{R}^{n\times n}$, $W = \text{tridiag}(-1, 5, -1) \in \mathbb{R}^{m\times m}$ and $x^* = (-1, 1, -1, 1, \ldots, -1, 1, \ldots)^\intercal \in \mathbb{R}^n$.

Table 4: Numerical results for Example 5.2 with $r = 1$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	4	4	4	4
	CPU	3.3993	18.2178	53.3196	149.8582
	RES	5.2394×10^{-12}	5.2502×10^{-12}	5.2572×10^{-12}	5.2609×10^{-12}
Picard	IT	17	17	17	17
	CPU	0.0443	0.1543	0.3086	0.5395
	RES	8.1387×10^{-7}	7.9178×10^{-7}	7.7834×10^{-7}	7.6932×10^{-7}
MN	IT	15	15	15	15
	CPU	0.0417	0.1385	0.2896	0.4916
	RES	7.1650×10^{-7}	7.4849×10^{-7}	7.6784×10^{-7}	7.8080×10^{-7}
NGS	IT	15	15	15	15
	CPU	0.0045	0.0081	0.0167	0.0256
	RES	8.3969×10^{-7}	9.0838×10^{-7}	9.5006×10^{-7}	9.7804×10^{-7}
NSOR	α_{exp}	1.3	1.3	1.3	1.3
	IT	10	10	10	10
	CPU	0.0036	0.0069	0.0129	0.0204
	RES	6.3585×10^{-7}	7.4418×10^{-7}	8.3740×10^{-7}	9.0772×10^{-7}
RTNGS	$\varphi_{1\text{exp}}$	0.9	0.9	0.9	0.9
	$\varphi_{2\text{exp}}$	0.4	0.4	0.4	0.4
	IT	4	4	4	4
	CPU	**0.0030**	**0.0062**	**0.0110**	**0.0181**
	RES	7.9232×10^{-7}	7.7208×10^{-7}	7.6633×10^{-7}	7.6529×10^{-7}
RTNSOR	α_{exp}	1.4	1.4	1.4	1.4
	$\varphi_{1\text{exp}}$	1.1	1.1	1.1	1.1
	$\varphi_{2\text{exp}}$	0.6	0.6	0.6	0.6
	IT	4	4	4	4
	CPU	**0.0027**	**0.0058**	**0.0105**	**0.0167**
	RES	5.4835×10^{-7}	6.2186×10^{-7}	6.6973×10^{-7}	7.0294×10^{-7}

Table 5: Numerical results for Example 5.2 with $r = m$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	4	4	4	4
	CPU	4.8509	21.4432	68.5549	205.6763
	RES	7.5635×10^{-8}	9.5964×10^{-8}	1.1182×10^{-7}	1.2460×10^{-7}
Picard	IT	19	19	19	19
	CPU	0.0638	0.1581	0.3206	0.6366
	RES	9.1765×10^{-7}	7.4831×10^{-7}	6.4714×10^{-7}	5.7957×10^{-7}
MN	IT	17	19	21	21
	CPU	0.0606	0.1530	0.3182	0.6226
	RES	9.6039×10^{-7}	9.6874×10^{-7}	7.2883×10^{-7}	9.8270×10^{-7}
NGS	IT	24	23	22	19
	CPU	0.0320	0.0422	0.0535	0.0662
	RES	8.7011×10^{-7}	7.9381×10^{-7}	6.6987×10^{-7}	9.9007×10^{-7}
NSOR	$\alpha_{\exp}$	1.9	2.0	2.0	2.0
	IT	16	15	15	15
	CPU	0.0219	0.0281	0.0386	0.0544
	RES	6.4907×10^{-7}	9.3739×10^{-7}	8.9013×10^{-7}	8.5596×10^{-7}
RTNGS	$\varphi_{1\exp}$	1.9	1.4	1.2	1.1
	$\varphi_{2\exp}$	0.7	0.6	0.6	0.6
	IT	7	6	6	6
	CPU	**0.0158**	**0.0197**	**0.0277**	**0.0406**
	RES	9.0094×10^{-7}	8.6502×10^{-7}	7.6970×10^{-7}	9.1843×10^{-7}
RTNSOR	$\alpha_{\exp}$	1.5	2.0	1.7	1.7
	$\varphi_{1\exp}$	1.9	1.9	1.7	1.6
	$\varphi_{2\exp}$	1.0	1.1	0.8	0.8
	IT	6	5	5	5
	CPU	**0.0136**	**0.0172**	**0.0230**	**0.0353**
	RES	7.7104×10^{-7}	8.1813×10^{-7}	5.1160×10^{-7}	7.1089×10^{-7}

Table 6: Values of $\max\{\delta, \rho\}$ in (4.1) and δ' in (4.13) for Example 5.2.

Method	n			
	60^2	80^2	100^2	120^2
Picard	0.6653	0.6659	0.6662	0.6663
MN	0.8326	0.8329	0.8331	0.8332
NGS	0.8660	0.8663	0.8664	0.8665
NSOR_1	0.8357	0.8360	0.8362	0.8363
NSOR_2	0.7844	0.7773	0.7774	0.7775
RTNGS_1	0.7853	0.7859	0.7862	0.7863
RTNGS_2	0.7638	0.7352	0.7529	0.7205
RTNSOR_1	0.6826	0.6831	0.6834	0.6835
RTNSOR_2	0.7115	0.6573	0.6617	0.6579

In Example 5.2, we choose $\theta = \pi/8$ and pick two scenarios for the parameter $r = 1, m$. Numerical results are listed in Tables 4-5, which indicate that each test method converges to the solution x^* of CCGAVE (1.2), and the elapsed CPU time also increases with the increase of the dimension n of the coefficient matrix. Furthermore, we can find that the RTNGS and RTNSOR iteration methods outperform the Picard, MN, NGS and NSOR iteration methods in terms of the number of iteration steps and the elapsed CPU time. Although the GN method requires fewer iteration steps, our methods take less the elapsed CPU time. In addition, numerical results in Table 6 illustrate the convergence condition (4.1) in Theorem 4.1 and the convergence condition (4.13) in Theorem 4.3 are satisfied, where NSOR_1, RTNGS_1, RTNSOR_1 and NSOR_2, RTNGS_2, RTNSOR_2 represent the corresponding iteration methods for $r = 1$ and $r = m$, respectively.

Example 5.3. Consider CCLCP (5.1), let m be a positive integer, $n = m^2$, $Q = \hat{Q} + 3I_n$, $P = \text{blktridiag}(-1.5I_m, W, -0.5I_m) \in \mathbb{R}^{n\times n}$ and $d = (1/2)(P(x^*+|x^*|)+Q(x^*-|x^*|))$, where $\hat{Q} =$

Table 7: Numerical results for Example 5.3 with $\theta = \pi/6$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	5	5	5	5
	CPU	6.4909	29.2790	95.4782	268.7339
	RES	3.7213×10^{-12}	4.9194×10^{-12}	5.8987×10^{-12}	6.7146×10^{-12}
Picard	IT	15	15	15	15
	CPU	0.0671	0.1633	0.3192	0.5611
	RES	6.8355×10^{-7}	5.8115×10^{-7}	5.2374×10^{-7}	4.8741×10^{-7}
MN	IT	24	25	25	25
	CPU	0.0775	0.1745	0.3539	0.5984
	RES	8.7387×10^{-7}	6.8298×10^{-7}	7.6271×10^{-7}	8.1961×10^{-7}
NGS	IT	24	25	25	25
	CPU	0.0313	0.0435	0.0589	0.0724
	RES	8.3724×10^{-7}	7.3617×10^{-7}	8.5909×10^{-7}	9.4674×10^{-7}
NSOR	α_{exp}	2.0	1.9	1.4	1.4
	IT	15	15	13	14
	CPU	0.0208	0.0279	0.0326	0.0426
	RES	8.2382×10^{-7}	9.1130×10^{-7}	8.3775×10^{-7}	7.1415×10^{-7}
RTNGS	$\varphi_{1\text{exp}}$	1.1	0.9	0.8	0.8
	$\varphi_{2\text{exp}}$	0.4	0.4	0.5	0.4
	IT	6	7	6	6
	CPU	**0.0139**	**0.0226**	**0.0258**	**0.0342**
	RES	8.8692×10^{-7}	4.4225×10^{-7}	7.3338×10^{-7}	4.5955×10^{-7}
RTNSOR	α_{exp}	1.2	1.3	1.2	1.3
	$\varphi_{1\text{exp}}$	1.1	1.0	0.9	0.9
	$\varphi_{2\text{exp}}$	0.6	0.7	0.5	0.8
	IT	5	5	5	5
	CPU	**0.0118**	**0.0173**	**0.0215**	**0.0295**
	RES	7.2275×10^{-7}	2.2092×10^{-7}	2.4798×10^{-7}	4.0209×10^{-7}

Table 8: Numerical results for Example 5.3 with $\theta = \pi/8$.

Method		n			
		60^2	80^2	100^2	120^2
GN	IT	4	4	4	4
	CPU	4.6144	21.1618	75.2917	196.4198
	RES	2.2809×10^{-7}	2.8966×10^{-7}	3.3776×10^{-7}	3.7661×10^{-7}
Picard	IT	22	21	21	21
	CPU	0.0722	0.1659	0.3335	0.6370
	RES	6.3344×10^{-7}	8.9509×10^{-7}	7.6311×10^{-7}	6.7575×10^{-7}
MN	IT	21	19	22	23
	CPU	0.0677	0.1535	0.3210	0.5827
	RES	7.4631×10^{-7}	9.9809×10^{-7}	7.8363×10^{-7}	7.5054×10^{-7}
NGS	IT	24	21	18	22
	CPU	0.0296	0.0400	0.0492	0.0732
	RES	7.6135×10^{-7}	9.1681×10^{-7}	7.0439×10^{-7}	6.7360×10^{-7}
NSOR	$\alpha_{\exp}$	2.0	1.9	1.9	1.9
	IT	16	16	16	16
	CPU	0.0206	0.0307	0.0410	0.0524
	RES	6.9173×10^{-7}	9.1798×10^{-7}	8.5678×10^{-7}	8.1202×10^{-7}
RTNGS	$\varphi_{1\exp}$	1.5	1.2	1.1	1.1
	$\varphi_{2\exp}$	0.8	0.8	0.7	0.5
	IT	7	7	7	6
	CPU	**0.0150**	**0.0217**	**0.0320**	**0.0428**
	RES	3.4363×10^{-7}	4.8120×10^{-7}	5.0508×10^{-7}	2.3280×10^{-7}
RTNSOR	$\alpha_{\exp}$	2	1.8	1.1	1.8
	$\varphi_{1\exp}$	2.2	1.7	1.2	1.6
	$\varphi_{2\exp}$	1.1	1.2	0.7	0.8
	IT	6	6	6	5
	CPU	**0.0128**	**0.0198**	**0.0270**	**0.0327**
	RES	7.6750×10^{-7}	7.1534×10^{-7}	7.1383×10^{-7}	7.2166×10^{-7}

$I_m \otimes W \in \mathbb{R}^{n\times n}$, $W = \text{tridiag}(-1.5, 5, -0.5) \in \mathbb{R}^{m\times m}$ and $x^* = (-1, 1, -1, 1, \ldots, -1, 1, \ldots)^\intercal \in \mathbb{R}^n$.

For Example 5.3, we choose $r = m$ and take two cases for the parameter $\theta = \pi/6$, $\pi/8$. In Tables 7-8, we report the numerical results of $\theta = \pi/6$ and $\theta = \pi/8$, respectively. We can find that all of the test methods can converge to the solution x^* of CCGAVE (1.2). Furthermore, Tables 7-8 also show that the RTNGS and RTNSOR iteration methods are superior to the GN, Picard, MN, NGS and NSOR iteration methods with respect to the elapsed CPU time. In addition, the RTNGS and RTNSOR iteration methods have fewer iteration steps than the Picard, MN, NGS and NSOR iteration methods. In addition, numerical results in Table 9 indicate the convergence condition (4.1) in Theorem 4.1 and the convergence con-

Table 9: Values of $\max\{\delta, \rho\}$ in (4.1) and δ' in (4.13) for Example 5.3.

Method	n			
	60^2	80^2	100^2	120^2
Picard	0.7130	0.7136	0.7138	0.7140
MN	0.8511	0.8514	0.8516	0.8517
NGS	0.8660	0.8663	0.8664	0.8665
NSOR_1	0.7636	0.7732	0.8224	0.8226
NSOR_2	0.7636	0.7732	0.7734	0.7736
RTNGS_1	0.8684	0.8538	0.7105	0.8452
RTNGS_2	0.7810	0.7675	0.7533	0.7309
RTNSOR_1	0.7860	0.6988	0.7919	0.7009
RTNSOR_2	0.7678	0.7155	0.7421	0.7147

dition (4.13) in Theorem 4.3 are satisfied, where NSOR_1, RTNGS_1, RTNSOR_1 and NSOR_2, RTNGS_2, RTNSOR_2 denote the corresponding iteration methods for $\theta = \pi/6$ and $\theta = \pi/8$, respectively.

6. Conclusions

In this paper, a convergent and efficient relaxation two-step Newton-based matrix splitting iteration method for solving the CCGAVE is developed. Meanwhile, it seems to be the first time that some sufficient conditions for the unique solvability of the CCGAVE are explicitly given. However, the properties of the solution set of the CCGAVE need further study.

Acknowledgments

The authors are grateful to the two anonymous reviewers and the editor for their comments and suggestions that have helped to improve the paper.

This work was partially supported by the Ministry of Science and Technology of China (Grant No. 2021YFA1003600). D. Yu was partially supported by the National Natural Science Foundation of China (Grant No. 12201275), by the Ministry of Education in China of Humanities and Social Science Project (Grant No. 21YJCZH204), by the Natural Science Foundation of Liaoning Province (Grant No. 2024-MS-206) and by the Liaoning Provincial Department of Education (Grant No. JYTZD2023072). C. Chen was partially supported by the Natural Science Foundation of Fujian Province (Grant No. 2021J01661) and by the Fujian Alliance of Mathematics (Grant No. 2023SXLMQN03). D. Han was partially supported by the National Natural Science Foundation of China (Grant Nos. 12126603, 12131004).

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Math. Program. **95**, 3–51 (2003).

[2] Y.L. Chang, C.Y. Yang and J.S. Chen, *Smooth and nonsmooth analysis of vector-valued functions associated with circular cones*, Nonlinear Anal. **85**, 160–173 (2013).

[3] C.R. Chen, D.M. Yu and D.R. Han, *Exact and inexact Douglas-Rachford splitting methods for solving large-scale sparse absolute value equations*, IMA J. Numer. Anal. **43**, 1036–1060 (2023).

[4] X. Chen, D. Sun and J. Sun, *Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems*, Comput. Optim. Appl. **25**, 39–56 (2003).

[5] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo Publishing (2005).

[6] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume I*, Springer (2003).

[7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press (2013).

[8] M. Hladík, *Properties of the solution set of absolute value equations and the related matrix classes*, SIAM J. Matrix Anal. Appl. **44**, 175–195 (2023).

[9] S.L. Hu, Z.H. Huang and Q. Zhang, *A generalized Newton method for absolute value equations associated with second order cones*, J. Comput. Appl. Math. **235**, 1490–1501 (2011).

[10] B.H. Huang and W. Li, *A modified SOR-like method for absolute value equations associated with second order cones*, J. Comput. Appl. Math. **400**, 113745 (2022).

[11] X.Q. Jiang and Y. Zhang, *A smoothing-type algorithm for absolute value equations*, J. Ind. Manag. Optim. **9**, 789–798 (2013).

[12] Y.F. Ke, C.F. Ma and H. Zhang, *The relaxation modulus-based matrix splitting iteration methods for circular cone nonlinear complementarity problems*, Comput. Appl. Math. **37**, 6795–6820 (2018).

[13] C.H. Ko and J.S. Chen, *Optimal grasping manipulation for multifingered robots using semismooth Newton method*, Math. Probl. Eng. **2013**, 681–710 (2013).

[14] C.X. Li and S.L. Wu, *A shift splitting iteration method for generalized absolute value equations*, Comput. Methods Appl. Math. **21**, 863–872 (2021).

[15] O.L. Mangasarian, *Absolute value programming*, Comput. Optim. Appl. **36**, 43–53 (2007).

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

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

[18] X.H. Miao, W.M. Hsu, C.T. Nguyn and J.S. Chen, *The solvabilities of three optimization problems associated with second-order cone*, J. Nonlinear Convex Anal. **22**, 937–967 (2021).

[19] X.H. Miao, J.T. Yang and S.L. Hu, *A generalized Newton method for absolute value equations associated with circular cones*, Appl. Math. Comput. **269**, 155–168 (2015).

[20] X.H. Miao, J.T. Yang, B. Saheya and J.S. Chen, *A smoothing Newton method for absolute value equation associated with second-order cone*, Appl. Numer. Math. **120**, 82–96 (2017).

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

[22] J. Rohn, *Systems of linear interval equations*, Linear Algebra Appl. **126**, 39–78 (1989).

[23] J. Rohn, *A theorem of the alternative for the equation $Ax+B|x|=b$*, Linear Multilinear Algebra **52**, 421–426 (2004).

[24] J. Rohn, V. Hooshyarbakhsh and R. Farhadsefat, *An iterative method for solving absolute value

equations and sufficient conditions for unique solvability, Optim. Lett. **8**, 35–44 (2014).
[25] X.H. Shao and W.C. Zhao, *Relaxed modified Newton-based iteration method for generalized absolute value equations*, AIMS Math. **8**, 4714–4725 (2023).
[26] A. Wang, Y. Cao and J.X. Chen, *Modified Newton-type iteration methods for generalized absolute value equations*, J. Optim. Theory Appl. **181**, 216–230 (2019).
[27] D.M. Yu, C.R. Chen and D.R. Han, *A modified fixed point iteration method for solving the system of absolute value equations*, Optimization **71**, 449–461 (2022).
[28] M. Zamani and M. Hladík, *Error bounds and a condition number for the absolute value equations*, Math. Program. **198**, 85–113 (2023).
[29] W.C. Zhao and X.H. Shao, *New matrix splitting iteration method for generalized absolute value equations*, AIMS Math. **8**, 10558–10578 (2023).
[30] H.Y. Zhou, S.L. Wu and C.X. Li, *Newton-based matrix splitting method for generalized absolute value equation*, J. Comput. Appl. Math. **394**, 113578 (2021).
[31] J.C. Zhou and J.S. Chen, *Properties of circular cone and spectral factorization associated with circular cone*, J. Nonlinear Convex Anal. **14**, 807–816 (2013).