AIMS Mathematics

AIMS Mathematics, 8(2): 4714–4725.
DOI: 10.3934/math.2023233
Received: 22 September 2022
Revised: 03 November 2022
Accepted: 18 November 2022
Published: 07 December 2022

http://www.aimspress.com/journal/Math

Research article

Relaxed modified Newton-based iteration method for generalized absolute value equations

Xin-Hui Shao* **and Wan-Chen Zhao**

Department of Mathematics, College of Sciences, Northeastern University, Shenyang 110819, China

* **Correspondence:** Email: xinhui1002@126.com.

Abstract: Many problems in different fields may lead to solutions of absolute value equations, such as linear programming problems, linear complementarity problems, quadratic programming, mixed integer programming, the bimatrix game and so on. In this paper, by introducing a nonnegative real parameter to the modified Newton-based iteration scheme, we present a new relaxed modified Newton-based (RMN) iteration method for solving generalized absolute value equations. The famous Picard iteration method and the modified Newton-type iteration method are the exceptional cases of the RMN iteration method. The convergence property of the new method is discussed. Finally, the validity and feasibility of the RMN iteration method are verified by experimental examples.

Keywords: generalized absolute value equation; relaxation; Newton-based method; convergence
Mathematics Subject Classification: 65F10, 90C05, 90C30

1. Introduction

We focus on the following generalized absolute value equations (GAVE):

$$Ax - B|x| = b, \tag{1.1}$$

where $A,\ B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Here, '$|x|$' denotes the vector in $\mathbb{R}^n$ with absolute values of components of x. Particularly, if $B = I$ or B is invertible, the GAVE (1.1) is simplified to the following absolute value equations (AVE):

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

Especially, when B is a zero matrix, the GAVE (1.1) is simplified to the linear system $Ax = b$, which plays a significant role in scientific computing problems. The main significance of the GAVE (1.1) and the AVE (1.2) is that many problems in different fields may be transformed into the form of an AVE. These include linear programming problems, linear complementarity problems (LCP), quadratic programming, mixed integer programming, the bimatrix game, and so on; [1–7] for more details. For example, given a matrix $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$, the linear complementarity problems (LCP) is to find $z \in \mathbb{R}^n$ so that

$$z \geq 0, w := Mz + q \geq 0, and\ z^T(Mz + q) = 0. \tag{1.3}$$

Over the last 20 years, to obtain the numerical solutions of the GAVE (1.1) and the AVE (1.2), people have done a lot of research and established many effective numerical methods to solve the GAVE (1.1) and the AVE (1.2), such as the Newton-based methods [8–18], the SOR-like iteration methods [19–21], the neural network methods [22–24], the matrix multisplitting Picard-iterative method [25] and so on.

Some ideas of our new work were inspired by the research on the GAVE (1.1) [10,13]. By separating the GAVE (1.1) from the sum of the differentiable function and Lipschitz continuous function, a common framework of the modified Newton-based (MN) iterative method [10] to solve the GAVE (1.1) has been established. The convergence conditions are given, and numerical experiments are used to show the availability of the MN method. At every iteration, we need to calculate the linear system with the coefficient matrix $\Omega + A$, where Ω is a given positive semi-definite matrix. However, if $\Omega + A$ is ill-conditioned, solving this linear system may be costly or impossible in practice. To solve this problem, we introduce a nonnegative real parameter $\theta \geq 0$ in the MN iteration frame and propose a new relaxed iterative method to solve the GAVE (1.1). The new plan includes the famous Picard iteration method [26,27] and the MN iteration method [10]. Two general sufficient conditions to ensure the convergence of the relaxed modified Newton-based (RMN) method are given. Further, there are some specific conditions, that is, the coefficient matrix A is symmetric positive definite or an H_+-matrix. Moreover, the experimental results verify the effectiveness of the RMN method.

The layout of the rest of this work is as follows. In Section 2, we establish a new relaxed iterative method to solve the GAVE (1.1). The associated convergence analysis is given in Section 3. The numerical results are provided in Section 4, and some remarks are given in Section 5.

2. Relaxed modified Newton-based iteration method

We will introduce a new MN iterative approach to acquire the numerical solutions of the GAVE (1.1) in this section.

For this reason, the following symbols and definitions are introduced. Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$. If $a_{ij} \leq 0$ for any $i \neq j$, then A is a Z-matrix. If $A^{-1} \geq 0$ and A is a Z-matrix, then A is a nonsingular M-matrix. If the comparison matrix $\langle A\rangle$ of A is an M-matrix, then A is an H-matrix, where the form of $\langle A\rangle = (\langle a\rangle_{ij})$ is as follows:

$$\langle a\rangle_{ij} = \begin{cases} |a_{ii}|, & if\ i = j, \\ -|a_{ij}|, & if\ i \neq j. \end{cases}$$

If A is an H-matrix with positive diagonal terms, then A is an H_+-matrix. If A is symmetric and satisfies $x^T Ax > 0$ for all nonzero vectors x, then A is symmetric positive definite. Note that $|A| = \left(|a_{ij}|\right)$ and $\rho(A)$ represent the absolute value matrix and the spectral radius, respectively. $\|A\|_2$ denotes the Euclidean norm.

To begin, we start with a brief review of the MN iteration method proposed in [10]. Because the nonlinear term $B|x|$ exists, solving the GAVE (1.1) can be transformed into finding the solution of the nonlinear function $F(x)$:

$$F(x) \coloneqq Ax - B|x| - b = 0. \tag{2.1}$$

The MN method in [10] regards the nonlinear function $F(x)$ as the sum of the differentiable function $H(x)$ and Lipschitz continuous function $G(x)$. A positive semi-definite matrix $\Omega \in R^{n\times n}$ is introduced and $H(x) = \Omega x + Ax$ and $G(x) = -\Omega x - B|x| - b$ are substituted into the MN iteration scheme proposed in [28]. If the Jacobian matrix $H'(x) = \Omega + A$ is invertible, the MN iteration method to solve the GAVE (1.1) is established with more detail as follows:

Algorithm 2.1. [10] (MN iteration method)
Step 1. Choose an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$, and let $k \coloneqq 0$;
Step 2. For $k = 0,1,2,\cdots,$ compute $x^{(k+1)} \in \mathbb{R}^n$ by using

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - (\Omega + A)^{-1}\left(Ax^{(k)} - B\left|x^{(k)}\right| - b\right) \\ &= (\Omega + A)^{-1}\left(\Omega x^{(k)} + B\left|x^{(k)}\right| + b\right), \end{aligned} \tag{2.2}$$

where $\Omega + A$ is nonsingular and Ω is a known positive semi-definite matrix;
Step 3. If the iteration sequence $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ is convergent, then the iteration stops. Otherwise, let $k + 1$ replace k and go to Step 2.

According to Algorithm 2.1, the coefficient matrix of the linear system is $\Omega + A$, which must be calculated at every step of the MN iteration method. However, if $\Omega + A$ is ill-conditioned, solving the linear system may be costly or impossible in actual applications.

In this paper, we give a new method to solve the GAVE (1.1). We introduce a nonnegative real parameter $\theta \geq 0$ in the MN iteration scheme proposed in [28] and obtain an RMN iteration method (the RMN method for short) for solving the GAVE (1.1):

$$F(x^{(k)}) + \left(H'\left(x^{(k)}\right) + (\theta - 1)A\right)\left(x^{(k+1)} - x^{(k)}\right) = 0. \tag{2.3}$$

By substituting (2.1) and the Jacobian matrix $H'\left(x^{(k)}\right) = \Omega + A$ into (2.3). It can be understood as $\widetilde{H}(x) = \Omega x + \theta Ax$ and $\tilde{G}(x) = -\Omega x - (\theta - 1)Ax - B|x| - b$. Therefore, we can obtain the new RMN iteration method as described below:

Algorithm 2.2. (The relaxed modified Newton-based (RMN) iteration method)
Step 1. Choose an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$, and let $k \coloneqq 0$;
Step 2. For $k = 0,1,2,\cdots,$ compute $x^{(k+1)} \in \mathbb{R}^n$ by using

$$x^{(k+1)} = (\Omega + \theta A)^{-1}\left(\Omega x^{(k)} + (\theta - 1)Ax^{(k)} + B\left|x^{(k)}\right| + b\right), \tag{2.4}$$

where $\Omega + \theta A$ is nonsingular, Ω is a known positive semi-definite matrix and $\theta \geq 0$ is a nonnegative relaxation parameter;
Step 3. If the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ is convergent, then the iteration stops. Otherwise, let $k+1$ replace k and go to Step 2.

Comparing Algorithms 2.1 and 2.2, the coefficient matrix of the former is $\Omega + A$, and that of the latter is $\Omega + \theta A$. We can change the relaxation parameter θ to avoid the ill-conditioned coefficient matrix.

Remark 2.1. Obviously, if we put $\theta = 1$, then the RMN method (2.4) is simplified to the famous MN method [10]. If we set $\theta = 1$ and Ω as a zero matrix, then the RMN method (2.4) is simplified to the Picard method [26,27].

3. Convergence property

We will discuss the convergence analysis of the RMN method to solve the GAVE (1.1) in this section. First, we give the general conditions of convergence. Second, for the case that the coefficient matrix A is symmetric positive definite or an H_+-matrix, then some sufficient convergence theorems of the RMN method are provided. The RMN method can be simplified to the MN method [10] and the Picard method [26,27], so their convergence conditions can be obtained immediately.

3.1. General sufficient convergence property

Theorem 3.1 and Theorem 3.2 present the general sufficient convergence of Algorithm 2.2 when the related matrix is invertible.

Theorem 3.1. Let $A, B \in \mathbb{R}^{n\times n}$, $\theta \geq 0$ be a nonnegative relaxation parameter and Ω be a positive semi-definite matrix which makes $\Omega + \theta A$ is invertible. If

$$\|(\Omega + \theta A)^{-1}\|_2 < \frac{1}{\|\Omega + (\theta - 1)A\|_2 + \|B\|_2}, \tag{3.1}$$

then the iterative sequence $\{x^{(k)}\}_{k=1}^{+\infty}$ created by Algorithm 2.2 is convergent.

Proof. Suppose that the GAVE (1.1) has a solution x^*; then, x^* satisfies the following equation:

$$Ax^* - B|x^*| = b, \tag{3.2}$$

which is equal to

$$(\Omega + \theta A)x^* = \Omega x^* + (\theta - 1)Ax^* + B|x^*| + b. \tag{3.3}$$

Subtracting (3.3) from (2.4) gives the error expression as follows:

$$(\Omega + \theta A)(x^{(k+1)} - x^*) = \Omega(x^{(k)} - x^*) + (\theta - 1)A(x^{(k)} - x^*) + B(|x^{(k)}| - |x^*|). \tag{3.4}$$

Noticing that $\Omega + \theta A$ is nonsingular, we have

$$x^{(k+1)} - x^* = (\Omega + \theta A)^{-1}((\Omega + (\theta - 1)A)(x^{(k)} - x^*) + B(|x^{(k)}| - |x^*|)). \tag{3.5}$$

Using the 2-norm for (3.5), we get

$$\begin{aligned}\|x^{(k+1)}-x^*\|_2 &= \left\|(\Omega+\theta A)^{-1}\left((\Omega+(\theta-1)A)\left(x^{(k)}-x^*\right)+B\left(\left|x^{(k)}\right|-|x^*|\right)\right)\right\|_2\\ &\le \|(\Omega+\theta A)^{-1}\|_2\cdot\|(\Omega+(\theta-1)A)\left(x^{(k)}-x^*\right)+B\left(\left|x^{(k)}\right|-|x^*|\right)\|_2\\ &\le \|(\Omega+\theta A)^{-1}\|_2\cdot(\|\ \Omega+(\theta-1)A\ \|_2+\|B\|_2)\|x^{(k)}-x^*\|_2.\end{aligned}\tag{3.6}$$

According to the condition (3.1), $\left\{x^{(k)}\right\}_{k=1}^{+\infty}$, as created by Algorithm 2.2, is convergent.

Theorem 3.2. Let $A\in\mathbb{R}^{n\times n}$ be nonsingular, $B\in\mathbb{R}^{n\times n}$, $\theta>0$ be a positive relaxation parameter and Ω be a positive semi-definite matrix which makes $\Omega+\theta A$ is nonsingular. If

$$\|(\theta A)^{-1}\|_2<\frac{1}{\|\Omega\|_2+\|\Omega+(\theta-1)A\|_2+\|B\|_2},\tag{3.7}$$

then the iterative sequence $\left\{x^{(k)}\right\}_{k=1}^{+\infty}$ created by Algorithm 2.2 is convergent.

Proof. From the Banach lemma [29], we can get

$$\begin{aligned}\|(\Omega+\theta A)^{-1}\|_2 &\le \frac{\|(\theta A)^{-1}\|_2}{1-\|(\theta A)^{-1}\|_2\|\Omega\|_2}\\ &< \frac{\frac{1}{\|\Omega\|_2+\|\Omega+(\theta-1)A\|_2+\|B\|_2}}{1-\frac{\|\Omega\|_2}{\|\Omega\|_2+\|\Omega+(\theta-1)A\|_2+\|B\|_2}}\\ &= \frac{1}{\|\Omega+(\theta-1)A\|_2+\|B\|_2}.\end{aligned}\tag{3.8}$$

Then the conclusion is drawn from Theorem 3.1.

As we all know, the GAVE (1.1) can be simplified to the AVE (1.2). Therefore, let $B=I$, the RMN method is also suitable for solving the AVE (1.2).

Corollary 3.1. Let $A\in\mathbb{R}^{n\times n}$, $\theta\ge 0$ be a nonnegative relaxation parameter and Ω be a positive semi-definite matrix which makes $\Omega+\theta A$ is nonsingular. If

$$\|(\Omega+\theta A)^{-1}\|_2<\frac{1}{\|\Omega+(\theta-1)A\|_2+1},\tag{3.9}$$

then the iterative sequence $\left\{x^{(k)}\right\}_{k=1}^{+\infty}$ created by Algorithm 2.2 to solve the AVE (1.2) is convergent.

Corollary 3.2. Let $A\in\mathbb{R}^{n\times n}$ be nonsingular, $\theta>0$ be a positive relaxation parameter and Ω be a positive semi-definite matrix which makes $\Omega+\theta A$ is nonsingular. If

$$\|(\theta A)^{-1}\|_2<\frac{1}{\|\Omega\|_2+\|\Omega+(\theta-1)A\|_2+1},\tag{3.10}$$

then the iterative sequence $\left\{x^{(k)}\right\}_{k=1}^{+\infty}$ created by Algorithm 2.2 to solve the AVE (1.2) is convergent.

3.2. Special sufficient convergence property

If A is symmetric positive definite or an H_+-matrix, we can respectively obtain Theorem 3.3 and Theorem 3.4, for $\Omega = \omega I$ with $\omega > 0$.

Theorem 3.3. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix, $\theta > 0$ be a positive relaxation parameter and $\Omega = \omega I$ with $\omega > 0$. Further denote that λ_{min} and λ_{max} are the minimum and the maximum eigenvalues of the matrix A, respectively, and assume $\|B\|_2 = \tau$. If $\tau > \lambda_{max}$ or $\tau < \lambda_{min}$, then the iterative sequence $\{x^{(k)}\}_{k=1}^{+\infty}$ created by Algorithm 2.2 is convergent.

Proof. In fact, according to Theorem 3.1, we only need to get

$$\|(\Omega + \theta A)^{-1}\|_2 \|\Omega + (\theta - 1)A + B\|_2 < 1. \tag{3.11}$$

By the assumptions, we have

$$\begin{aligned} \|(\Omega + \theta A)^{-1}\|_2 \|\Omega + (\theta - 1)A + B\|_2 &= \max_{\lambda \in \mathrm{sp}\,(A)} \frac{|\omega + (\theta - 1)\lambda + \tau|}{\omega + \theta\lambda} \\ &= \max_{\lambda \in \mathrm{sp}(A)} \frac{|\omega + \theta\lambda - \lambda + \tau|}{\omega + \theta\lambda} = 1 + \max_{\lambda \in \mathrm{sp}\,(A)} \frac{|\tau - \lambda|}{\omega + \theta\lambda} \\ &= \max\left\{1 + \frac{|\tau - \lambda_{min}|}{\omega + \theta\lambda_{min}}, 1 + \frac{|\tau - \lambda_{max}|}{\omega + \theta\lambda_{max}}\right\} \\ &= \begin{cases} 1 + \frac{\tau - \lambda_{min}}{\omega + \lambda_{min}} & \text{for } \tau \leqslant \sqrt{\lambda_{min}\lambda_{max}} \\ 1 + \frac{\lambda_{max} - \tau}{\omega + \theta\lambda_{max}} & \text{for } \tau \geqslant \sqrt{\lambda_{min}\lambda_{max}} \end{cases}. \end{aligned} \tag{3.12}$$

It follows that, if $\tau > \lambda_{max}$ or $\tau < \lambda_{min}$, then $\{x^{(k)}\}_{k=1}^{+\infty}$, as created by Algorithm 2.2, is convergent.

Similarly, let $B = I$, and the following is the corresponding inference.

Corollary 3.3. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix, $\theta > 0$ be a positive relaxation parameter and $\Omega = \omega I$ with $\omega > 0$. Further denote that λ_{min} and λ_{max} are the minimum and the maximum eigenvalues of the matrix A, respectively. If $\lambda_{max} < 1$ or $\lambda_{min} > 1$, then $\{x^{(k)}\}_{k=1}^{+\infty}$, as created by Algorithm 2.2 to solve the AVE (1.2), is convergent.

Theorem 3.4. Let $A \in \mathbb{R}^{n\times n}$ be an H_+-matrix, $\theta > 0$ be a positive relaxation parameter and $\Omega = \omega I$ with $\omega > 0$. If

$$\|(\Omega + \langle\theta A\rangle)^{-1}\|_2 < \frac{1}{\|\Omega + |\theta - 1||A| + |B|\|_2}, \tag{3.13}$$

then the iterative sequence $\{x^{(k)}\}_{k=1}^{+\infty}$ created by Algorithm 2.2 converges.

Proof. According to the assumptions and [30], we have

$$|(\Omega + \theta A)^{-1}| \leq (\Omega + \langle\theta A\rangle)^{-1}. \tag{3.14}$$

Using the absolute value of (3.5), we get

$$\begin{aligned}\left|x^{(k+1)}-x^*\right| &= \left|(\Omega+\theta A)^{-1}\left(\Omega\left(x^{(k)}-x^*\right)+(\theta-1)A\left(x^{(k)}-x^*\right)+B\left(\left|x^{(k)}\right|-|x^*|\right)\right)\right| \\ &\leq |(\Omega+\theta A)^{-1}|\left(|\Omega|\left|x^{(k)}-x^*\right|+|\theta-1||A|\left|x^{(k)}-x^*\right|\right)+|B|\left||x^{(k)}|-|x^*|\right|\Big) \\ &\leq (\Omega+\langle\theta A\rangle)^{-1}(\Omega+|\theta-1||A|+|B|)\left|x^{(k)}-x^*\right|.\end{aligned} \tag{3.15}$$

Since

$$\begin{aligned}\rho((\Omega+\langle\theta A\rangle)^{-1}(\Omega+|\theta-1||A|+|B|)) &\leq \|(\Omega+\langle\theta A\rangle)^{-1}(\Omega+|\theta-1||A|+|B|)\|_2 \\ &\leq \|(\Omega+\langle\theta A\rangle)^{-1}\|_2\|\Omega+|\theta-1||A|+|B|\|_2.\end{aligned} \tag{3.16}$$

It follows that, if the condition (3.13) is satisfied, then the sequence $\{x^{(k)}\}_{k=1}^{+\infty}$ created by Algorithm 2.2 converges.

Corollary 3.4. Let $A\in\mathbb{R}^{n\times n}$ be an H_+-matrix, $\theta>0$ be a positive relaxation parameter and $\Omega=\omega I$ with $\omega>0$. If

$$\|(\Omega+\langle\theta A\rangle)^{-1}\|_2<\frac{1}{\|\Omega+|\theta-1||A|+1\|_2}, \tag{3.17}$$

then the sequence $\{x^{(k)}\}_{k=1}^{+\infty}$ created by Algorithm 2.2 to solve the AVE (1.2) converges.

4. Numerical experiments

This section describes the use of two numerical examples to compare the RMN method and the MN method [10] in terms of the iteration step number (indicated as ‘IT’), the amount of CPU time (indicated as ‘CPU’) and the norm of relative residual vector (indicated as ‘RES’). Here, ‘RES’ was set to be

$$\mathrm{RES}=\frac{\|Ax^{(k)}-B\left|x^{(k)}\right|-b\|_2}{\|\ b\ \|_2}.$$

Here, we used MATLAB R2020B for all the experiments. All numerical computations were started from the initial vector

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

The iteration was terminated once RES$<10^{-7}$ or the largest number of iteration step k_{max} exceeds 500. As we all know, different values of Ω will affect the performances for the MN method [10], while the new RMN method has two influencing factors: Ω and θ. Therefore, selecting the appropriate Ω and θ is a significant issue that needs further research. As described in [10], let $\Omega=\omega I$, and the best experimental parameter is recorded as ω_{exp}, which minimizes the iteration step number of the MN method. When the iteration step numbers are the same, take the minimum value of RES. Similarly, the optimal experimental parameters of the new RMN method are recorded as ω_{exp} and θ_{exp}. In addition, the Cholesky and LU factorizations are utilized to solve all the subsystems when $\omega I+\theta A$ are symmetric positive definite and nonsymmetric, respectively.

As pointed out in [4,9], if the eigenvalue of M is not 1, the LCP (1.3) can lead to

$$(M+I)x-(M-I)|x| = q \; with \; x = \frac{1}{2}[(M-I)z+q]. \tag{4.1}$$

If we let $A = M + I$, $B = M - I$ and $b = q$, (4.1) converts to the form of the GAVE (1.1). According to this, we give the following examples.

Example 1. ([6]) Let $n = m^2$. We consider the LCP (1.3), where $M = \widehat{M} + \mu I \in \mathbb{R}^{n\times n}$ $and\; q = -Mz^* \in \mathbb{R}^n$ with

$$\widehat{M} = tridiag(-I, S, -I) \in \mathbb{R}^{n\times n}, S = tridiag(-1,4,-1) \in \mathbb{R}^{m\times m},$$

and $z^* = (1.2,1.2,1.2, \dots ,1.2, \dots)^T \in \mathbb{R}^n$ is the unique solution of the LCP. In this situation, the unique solution of the GAVE (1.1) is $x^* = (-0.6, -0.6, -0.6, \dots, -0.6, \dots)^T \in \mathbb{R}^n$.

To demonstrate the superiority of the new RMN iteration method, we took $\mu = -4$, $\mu = -1$ and $\mu = 4$ in our actual experiments. By computation, the matrices M and A are symmetric positive indefinite and symmetric positive definite when $\mu = -4$ and $\mu = 4$. When $\mu = -1$, the former is symmetric indefinite, and the latter is symmetric positive definite.

Table 1. Experimental results for Example 1 with $\mu = -4$.

	$\boldsymbol{n}$	3600	4900	6400	8100	10000	12100
MN	ω_{exp}	4.21	4.21	4.21	4.21	4.21	4.21
	IT	42	42	41	41	41	41
	CPU	0.0257	0.0313	0.0423	0.0632	0.0945	0.1439
	RES	8.7911e-08	8.1628e-08	9.9938e-08	9.4510e-08	8.9924e-08	8.5983e-08
RMN	ω_{exp}	2	2	2	2	2	2
	θ_{exp}	0	0	0	0	0	0
	IT	1	1	1	1	1	1
	CPU	0.0011	0.0014	0.0016	0.0016	0.0022	0.0024
	RES	3.6161e-16	3.6281e-16	3.6372e-16	3.6443e-16	3.6499e-16	3.6545e-16

Table 2. Experimental results for Example 1 with $\mu = -1$.

	$\boldsymbol{n}$	3600	4900	6400	8100	10000	12100
MN	ω_{exp}	1.18	1.18	1.18	1.18	1.18	1.18
	IT	45	44	44	44	44	44
	CPU	0.0206	0.0314	0.0499	0.0715	0.1003	0.1548
	RES	8.1234e-08	9.8868e-08	9.2976e-08	8.8063e-08	8.3888e-08	8.0286e-08
RMN	ω_{exp}	3.07	3.07	3.06	3.06	3.05	3.05
	θ_{exp}	0.46	0.46	0.46	0.46	0.46	0.46
	IT	33	33	33	33	33	32
	CPU	0.0169	0.0277	0.0389	0.0582	0.0861	0.1282
	RES	8.5725e-08	7.9222e-08	7.4120e-08	6.9804e-08	6.6263e-08	9.7359e-08

Table 3. Experimental results for Example 1 with $\mu = 4$.

	$\boldsymbol{n}$	3600	4900	6400	8100	10000	12100
MN	ω_{exp}	4.96	4.95	4.94	4.93	4.93	4.92
	IT	12	12	12	12	12	12
	CPU	0.0093	0.0151	0.0215	0.0370	0.0519	0.0913
	RES	3.7076e-08	3.6153e-08	3.5311e-08	3.4548e-08	3.3845e-08	3.3194e-08
RMN	ω_{exp}	0	0	0	0	0	0
	θ_{exp}	1.72	1.72	1.72	1.72	1.72	1.72
	IT	8	8	8	8	8	8
	CPU	0.0073	0.0120	0.0167	0.0259	0.0361	0.0648
	RES	2.0827e-08	2.1162e-08	2.1414e-08	2.1611e-08	2.1769e-08	2.1898e-08

In Tables 1–3, we give the experimental results of different values of μ. It is easy to see that when the grid size n increases, the iteration step number and CPU time of the MN and RMN methods also increase. Compared with the former, the latter has less iteration steps and CPU time. Therefore, we can conclude that the new RMN iteration method has better computational performance.

Example 2. ([6,31]) Let $n = m^2$. We consider the LCP (1.3), where $M = \widehat{M} + 4I \in \mathbb{R}^{n\times n}$ $and\ q = -Mz^* \in \mathbb{R}^n$ with

$$\widehat{M} = tridiag(-1.5I, S, -0.5I) \in \mathbb{R}^{n\times n}, S = tridiag(-1.5,4,-0.5) \in \mathbb{R}^{m\times m},$$

and $z^* = (1.2,1.2,1.2,\dots,1.2,\dots)^T \in \mathbb{R}^n$ is the unique solution of the LCP. In this situation, the unique solution of the GAVE (1.1) is $x^* = (-0.6,-0.6,-0.6,\ \dots,-0.6,\dots)^T \in \mathbb{R}^n$.

In this case, the matrix A is a s.d.d H_+-matrix. The experimental results in Table 4 further demonstrate that the results obtained from Table 3, that is, the computational efficiency of the RMN method is better than the MN method.

Table 4. Experimental results for Example 2.

	$\boldsymbol{n}$	3600	4900	6400	8100	10000	12100
MN	ω_{exp}	5.05	5.04	5.03	5.02	5.01	5.01
	IT	12	12	12	12	12	12
	CPU	0.0363	0.0455	0.0670	0.1238	0.1735	0.2568
	RES	5.5099e-08	5.3683e-08	5.2389e-08	5.1208e-08	5.0132e-08	4.9137e-08
RMN	ω_{exp}	0	0	0	0	0	0
	θ_{exp}	1.72	1.72	1.72	1.72	1.72	1.72
	IT	8	8	8	8	8	8
	CPU	0.0197	0.0358	0.0581	0.0895	0.1337	0.1878
	RES	2.0511e-08	2.0890e-08	2.1175e-08	2.1398e-08	2.1577e-08	2.1724e-08

5. Conclusions

In this paper, the RMN method has been established to solve the GAVE (1.1) by introducing a nonnegative real parameter $\theta \geq 0$. To ensure the convergence of the RMN method, some sufficient theorems are given. We used two numerical experiments from the LCP to show that, as compared with the existing MN iteration method [10], the new RMN method is feasible under certain conditions.

Acknowledgments

The authors would like to thank the three anonymous referees for providing helpful suggestions, which greatly improved the paper. This research was supported by the Fundamental Research Funds for the Central Universities (N2224005-1).

Conflict of interest

The authors declare that they have no competing interests.

References

1. J. Rohn, A theorem of the alternatives for the equation $Ax + B|x| = b$, *Linear Multilinear A.*, **52** (2004), 421–426. https://doi.org/10.1080/0308108042000220686
2. O. L. Mangasarian, Absolute value programming, *Comput. Optim. Appl.*, **36** (2007), 43–53. https://doi.org/10.1007/s10589-006-0395-5

3. J. L. Dong, M. Q. Jiang, A modified modulus method for symmetric positive-definite linear complementarity problems, *Numer. Linear Algebra Appl.*, **16** (2009), 129–143. https://doi.org/10.1002/nla.609
4. O. L. Mangasarian, R. R. Meyer, Absolute value equations, *Linear Algebra Appl.*, **419** (2006), 359–367. https://doi.org/10.1016/j.laa.2006.05.004
5. L. Abdallah, M. Haddou, T. Migot, Solving absolute value equation using complementarity and smoothing functions, *J. Comput. Appl. Math.*, **327** (2018), 196–207. https://doi.org/10.1016/j.cam.2017.06.019
6. Z. Z. Bai, Modulus-based matrix splitting iteration methods for linear complementarity problems, *Numer. Linear Algebra Appl.*, **17** (2010), 917–933. https://doi.org/10.1002/nla.680
7. J. Rohn, Systems of linear interval equations, *Linear Algebra Appl.*, **126** (1989), 39–78. https://doi.org/10.1016/0024-3795(89)90004-9
8. L. Caccetta, B. Qu, G. L. Zhou, A globally and quadratically convergent method for absolute value equations, *Comput. Optim. Appl.*, **48** (2011), 45–58. https://doi.org/10.1007/s10589-009-9242-9
9. O. L. Mangasarian, A generalized Newton method for absolute value equations, *Optim. Lett.*, **3** (2009), 101–108. https://doi.org/10.1007/s11590-008-0094-5
10. A. Wang, Y. Cao, J. X. Chen, Modified Newton-type iteration methods for generalized absolute value equations, *J. Optim. Theory. Appl.*, **181** (2019), 216–230. https://doi.org/10.1007/s10957-018-1439-6
11. N. Zainali, T. Lotfifi, On developing a stable and quadratic convergent method for solving absolute value equation, *J. Comput. Appl. Math.*, **330** (2018), 742–747. https://doi.org/10.1016/j.cam.2017.07.009
12. H. Y. Zhou, S. L. Wu, C. X. Li, Newton-based matrix splitting method for generalized absolute value equation, *J. Comput. Appl. Math.*, **394** (2021), 113578. https://doi.org/10.1016/j.cam.2021.113578
13. Y. Cao, Q. Shi, S. L. Zhu, A relaxed generalized Newton iteration method for generalized absolute value equations, *AIMS Math.*, **6** (2021), 1258–1275. https://doi.org/10.3934/math.2021078
14. R. Ali, I. Khan, A. Ali, Two new generalized iteration methods for solving absolute value equations using *M*-matrix, *AIMS Math.*, **7** (2022), 8176–8187. https://doi.org/10.3934/math.2022455
15. A. Khan, J. Iqbal, A. Akgül, R. Alia, Y. Du, A. Hussain, et al., A Newton-type technique for solving absolute value equations, *Alexandria Eng. J.*, 2022. https://doi.org/10.1016/j.aej.2022.08.052
16. R. Ali, K. Pan, The solution of the absolute value equations using two generalized accelerated overrelaxation methods, *Asian-Eur. J. Math.*, **15** (2022), 2250154. https://doi.org/10.1142/S1793557122501546
17. R. Ali, K. Pan, The new iteration methods for solving absolute value equations, *Appl. Math.*, 2021, 1–14. https://doi.org/10.21136/AM.2021.0055-21
18. R. Ali, K. Pan, Two new fixed point iterative schemes for absolute value equations, *Japan J. Indust. Appl. Math.*, 2022, 1–12. https://doi.org/10.1007/s13160-022-00526-x
19. P. Guo, S. L Wu, C. X Li, On the SOR-like iteration method for solving absolute value equations, *Appl. Math. Lett.*, **97** (2019), 107–113. https://doi.org/10.1016/j.aml.2019.03.033

20. X. Dong, X. H. Shao, H. L. Shen, A new SOR-like method for solving absolute value equations, *Appl. Numer. Math.*, **156** (2020), 410–421. https://doi.org/10.1016/j.apnum.2020.05.013
21. Y. F. Ke, C. F. Ma, SOR-like iteration method for solving absolute value equations, *Appl. Math. Comput.*, **311** (2017), 195–202. https://doi.org/10.1016/j.amc.2017.05.035
22. C. R. Chen, Y. N. Yang, D. M. Yu, D. R. Han, An inverse-free dynamical system for solving the absolute value equations, *Appl. Numer. Math.*, **168** (2021), 170–181. https://doi.org/10.1016/j.apnum.2021.06.002
23. A. Mansoori, M. Erfanian, A dynamic model to solve the absolute value equations, *J. Comput. Appl. Math.*, **333** (2018), 28–35. https://doi.org/10.1016/j.cam.2017.09.032
24. A. Mansoori, M. Eshaghnezhad, S. Effati, An efficient neural network model for solving the absolute value equations, *IEEE T. Circuits-II*, **65** (2017), 391–395. https://doi.org/10.1109/TCSII.2017.2750065
25. M. Dehghan, A. Shirilord, Matrix multisplitting Picard-iterative method for solving generalized absolute value matrix equation, *Appl. Numer. Math.*, **158** (2020), 425–438. https://doi.org/10.1016/j.apnum.2020.08.001
26. D. K. Salkuyeh, The Picard–HSS iteration method for absolute value equations, *Optim. Lett.*, **8** (2014), 2191–2202. https://doi.org/10.1007/s11590-014-0727-9
27. J. Rohn, V. Hooshyarbakhsh, R. Farhadsefat, An iterative method for solving absolute value equations and sufficient conditions for unique solvability, *Optim. Lett.*, **8** (2014), 35–44. https://doi.org/10.1007/s11590-012-0560-y
28. D. F. Han, The majorant method and convergence for solving nondifferentiable equations in Banach space, *Appl. Math. Comput.*, **118** (2001), 73–82. https://doi.org/10.1016/S0096-3003(99)00183-6
29. G. H. Golub, C. F. Van Loan, *Matrix computations*, 3 Eds., Johns Hopkins University Press, 2013.
30. A. Frommer, G. Mayer, Convergence of relaxed parallel multisplitting methods, *Linear Algebra Appl.*, **119** (1989), 141–152. https://doi.org/10.1016/0024-3795(89)90074-8
31. S. L. Wu, C. X. Li, Two-sweep modulus-based matrix splitting iteration methods for linear complementarity problems, *J. Comput. Appl. Math.*, **302** (2016), 327–339. https://doi.org/10.1016/j.cam.2016.02.011