Jingyong Tang

Inexact Newton-type method for solving large-scale absolute value equation $Ax - |x| = b$

Applications of Mathematics, Vol. 69 (2024), No. 1, 49–66

Persistent URL: `http://dml.cz/dmlcz/152252`

INEXACT NEWTON-TYPE METHOD FOR SOLVING LARGE-SCALE ABSOLUTE VALUE EQUATION $Ax - |x| = b$

Jingyong Tang, Xinyang

Received July 25, 2022. Published online August 31, 2023.

Abstract. Newton-type methods have been successfully applied to solve the absolute value equation $Ax - |x| = b$ (denoted by AVE). This class of methods usually solves a system of linear equations exactly in each iteration. However, for large-scale AVEs, solving the corresponding system exactly may be expensive. In this paper, we propose an inexact Newton-type method for solving the AVE. In each iteration, the proposed method solves the corresponding system only approximately. Moreover, it adopts a new line search technique, which is well-defined and easy to implement. We prove that the proposed method has global and local superlinear convergence under the condition that the interval matrix $[A-I, A+I]$ is regular. This condition is much weaker than those used in some Newton-type methods. Numerical results show that our method has fairly good practical efficiency for solving large-scale AVEs.

Keywords: absolute value equation; inexact Newton method; regularity of interval matrices; superlinear convergence

MSC 2020: 90C05, 90C33

1. Introduction

We consider the absolute value equation (AVE) of the type

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

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $|x|$ denotes the vector with absolute values of each component of x. As it was shown in [13], linear programs, quadratic programs, bimatrix games and other problems can all be reduced to a general linear complementarity

This research is partially supported by the Natural Science Foundation of Henan Province (222300420520) and Key Scientific Research Projects of Higher Education of Henan Province (22A110020).

DOI: 10.21136/AM.2023.0171-22

problem (LCP) which in turn is equivalent to the AVE (1.1). Thus, the AVE (1.1) formulation, which is simpler to state than an LCP, subsumes major fundamental problems of mathematical programming.

Recently, some computational methods have been proposed for solving the AVE (1.1) (e.g., [2], [5], [10], [11], [12], [15], [20], [24], [27], [29]). Among them, Newton-type methods is one class of the most effective methods. Mangasarian [10] proposed a generalized Newton method for solving the AVE (1.1) and proved that the method converges linearly from any starting point to the unique solution of the AVE (1.1) under the condition that $\|A^{-1}\| < \frac{1}{4}$. Caccetta et al. [2] studied a smoothing Newton method for solving the AVE (1.1) and proved that the method is globally convergent and the convergence rate is quadratic under the condition that $\|A^{-1}\| < 1$. Lately, many authors studied smoothing Newton methods to solve the general AVE of the type $Ax + B|x| = b$, where $B \in \mathbb{R}^{n\times n}$, and showed that their methods are globally and locally quadratically convergent under the condition that the minimal singular value of the matrix A is strictly greater than the maximal singular value of the matrix B (see [6], [21]).

The concept of interval matrix arises from the linear interval equations [19]. Given two matrices $A = (a_{ij})$, $B = (b_{ij})$, the matrix inequality $A \leqslant B$ refers to $a_{ij} \leqslant b_{ij}$ for any i, j. Given two matrices $\underline{A} = (\underline{a}_{ij})$ and $\overline{A} = (\overline{a}_{ij})$, an interval matrix $A^I := [\underline{A}, \overline{A}] = \{A;\ \underline{A} \leqslant A \leqslant \overline{A}\}$ is called *regular* if each $A \in A^I$ is nonsingular. Zhang and Wei [27] studied a generalized Newton method for solving the AVE (1.1) and established its global and finite convergence under the condition that $[A - I, A + I]$ is regular, where I denotes the identity matrix. Wang et al. [24] proposed a smoothing Newton method for solving the AVE (1.1) and showed that their method is globally convergent under the condition that $[A - I, A + I]$ is regular. Lately, Tang and Zhou [22] proposed a descent method to solve $Ax + B|x| = b$ and proved that their method is globally and locally quadratically convergent under the condition that $[A - |B|, A + |B|]$ is regular, which includes the cases in [24], [27]. It is worth pointing out that Newton-type methods in [2], [6], [21], [22], [24], [27] solve a system of linear equations exactly. These methods may be cumbersome if one is solving a large-scale AVE. The inexact approach is a one way to overcome this difficulty. For example, Chen et al. [3] proposed an inexact Douglas-Rachford splitting method for solving the AVE (1.1) and established its global linear convergence rate under the condition that $\|A^{-1}\| \leqslant 1$. Yu et al. [26] developed an inexact framework of the Newton-based matrix splitting iterative method to solve the generalized AVEs and proved that it converges linearly from any starting point under suitable conditions.

In this paper, we propose an inexact Newton-type method to solve the AVE (1.1). The proposed method solves the system of linear equations only approximately by using an inexact Newton method. Moreover, it adopts a new line search technique which is well-defined and easy to implement. We prove that the proposed method has global and local superlinear convergence under the condition that the interval matrix $[A-I, A+I]$ is regular. Since the regularity of $[A-I, A+I]$ includes the case $\|A^{-1}\| < 1$ (see [18], Theorem 4.1), it is weaker than those used in [2] and [10], where the methods require that $\|A^{-1}\| < 1$ and $\|A^{-1}\| < \frac{1}{4}$, respectively. We also compare the proposed method with some (inexact) Newton-type methods by numerical experiments. Numerical results show that our method has fairly good practical efficiency for solving large-scale AVEs.

This paper is organized as follows. In Section 2, we propose an inexact Newton-type method for solving the AVE (1.1). In Section 3, we investigate the global and local superlinear convergence of the proposed method. Some numerical results and conclusions are given in Sections 4 and 5, respectively.

In our notations, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors, and $\mathbb{R}^n_+$ (or $\mathbb{R}^n_{++}$) denotes the nonnegative (or positive) orthant in $\mathbb{R}^n$. Moreover, $\|\cdot\|$ denotes the 2-norm. For any $\alpha \in \mathbb{R}$ we define

$$\operatorname{sgn}(\alpha) := \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \\ -1 & \text{if } \alpha < 0. \end{cases}$$

For any $x \in \mathbb{R}^n$ we denote the diagonal matrix whose ith diagonal element is x_i by $\operatorname{diag}(x_i)$ and define $D(x) := \operatorname{diag}(\operatorname{sgn}(x_i))$. It is easy to see that $D(x) \in [-I, I]$ and $|x| = D(x)x$.

2. An inexact Newton-type method

Lemma 2.1 ([27], Proposition 2.1)**.** *The* AVE (1.1) *is uniquely solvable for any* $b \in \mathbb{R}^n$ *if the interval matrix* $[A-I, A+I]$ *is regular.*

Let $w := (\varepsilon, x) \in \mathbb{R} \times \mathbb{R}^n$. We define the function $E(w)\colon \mathbb{R}^{1+n} \to \mathbb{R}^n$ as

$$E(w) := Ax - \sqrt{\varepsilon^2 + x^2} - b, \tag{2.1}$$

where

$$\sqrt{\varepsilon^2 + x^2} := \left(\sqrt{\varepsilon^2 + x_1^2}, \ldots, \sqrt{\varepsilon^2 + x_n^2}\right)^\top.$$

Obviously, x is a solution of the AVE (1.1) if and only if $\varepsilon = 0$ and $E(\varepsilon, x) = 0$.

Lemma 2.2. *The function $E(w)$ is continuously differentiable at every point $w \in \mathbb{R}_{++} \times \mathbb{R}^n$ with*

$$E'(w) = \begin{pmatrix} -d_\varepsilon & A - D_x \end{pmatrix}, \tag{2.2}$$

where

$$d_\varepsilon := \Big(\frac{\varepsilon}{\sqrt{\varepsilon^2 + x_1^2}}, \dots, \frac{\varepsilon}{\sqrt{\varepsilon^2 + x_n^2}} \Big)^\top, \quad D_x := \operatorname{diag}\Big(\frac{x_i}{\sqrt{\varepsilon^2 + x_i^2}} \Big). \tag{2.3}$$

By using $E(w)$, we define the merit function $\psi(w)\colon \mathbb{R}^{1+n} \to \mathbb{R}_+$ as

$$\psi(w) := \varepsilon^2 + \|E(w)\|^2. \tag{2.4}$$

We now describe our algorithm as follows.

Algorithm 2.1. (An inexact Newton-type method)

Step 0: Choose $\delta \in (0,1)$ and $\varepsilon_0 > 0$. Choose $\eta \in (0,1)$ and a sequence $\{\eta_k\}$ such that $\eta_k \in [0, \eta]$. Choose $\gamma \in (0,1)$ such that $\frac{1}{2} < \gamma + \eta < 1$ and $\varepsilon_0 \geqslant \gamma$. Choose $t \in (0,1)$ such that $t < 2(1 - \gamma - \eta)$. Choose $\theta > 0$ and a sequence $\{\theta_k\}$ such that $\sum_{k=0}^{\infty} \theta_k \leqslant \theta < \infty$. Choose $x^0 \in \mathbb{R}^n$ and let $w^0 := (\varepsilon_0, x^0)$. Set $k := 0$.

Step 1: Terminate if $\|Ax^k - |x^k| - b\| = 0$. Else, let

$$\tau_k := \gamma \min\big\{1, \psi(w^k), \dots, \psi(w^0)\big\}. \tag{2.5}$$

Step 2: Let

$$\Delta\varepsilon_k := -\varepsilon_k + \tau_k, \tag{2.6}$$

and choose $\Delta x^k \in \mathbb{R}^n$ such that

$$\|E'(w^k)(\Delta\varepsilon_k, \Delta x^k) + E(w^k)\| \leqslant \eta_k \|E(w^k)\|. \tag{2.7}$$

Set $\Delta w^k := (\Delta\varepsilon_k, \Delta x^k)$.

Step 3: Let l_k be the smallest nonnegative integer l satisfying

$$\psi(w^k + \delta^l \Delta w^k) \leqslant (1 + \theta_k - t\delta^l)\psi(w^k). \tag{2.8}$$

Set $\alpha_k := \delta^{l_k}$.

Step 4: Set $w^{k+1} := w^k + \alpha_k \Delta w^k$. Set $k := k + 1$ and go back to Step 1.

The Newton-type methods in [2], [6], [21], [22], [24], [27] compute the search direction by solving the system of equations

$$E'(w^k)(\Delta\varepsilon_k, \Delta x^k) = -E(w^k) \tag{2.9}$$

exactly. These methods may be cumbersome if one is solving a large-scale AVE. Algorithm 2.1 solves (2.9) only approximately in Step 2 though. Moreover, Algorithm 2.1 adopts a new line search technique in Step 3, which is much simpler than those used in Newton-type methods. As it will be shown later, by solving (2.9) approximately and adopting the new line search, Algorithm 2.1 has very encouraging convergence properties and numerical performances.

To show the well-definedness of Algorithm 2.1, we define the function $F(w)$: $\mathbb{R}^{1+n} \to \mathbb{R}^{1+n}$ as

$$F(w) := \begin{pmatrix} \varepsilon \\ E(w) \end{pmatrix}, \tag{2.10}$$

where $E(w)$ is defined by (2.1). Then $\psi(w) = \|F(w)\|^2$. For all $k \geqslant 0$, let

$$r_k := E'(w^k)(\Delta\varepsilon_k, \Delta x^k) + E(w^k). \tag{2.11}$$

Then it is easy to see that solving (2.6) and (2.7) is equivalent to solving

$$F'(w^k)(\Delta\varepsilon_k, \Delta x^k) = -F(w^k) + \begin{pmatrix} \tau_k \\ r_k \end{pmatrix} \quad \text{with } \|r_k\| \leqslant \eta_k \|E(w^k)\|. \tag{2.12}$$

Lemma 2.3.

(i) *The function $F(w)$ is continuously differentiable at any $w \in \mathbb{R}_{++} \times \mathbb{R}^n$ with*

$$F'(w) = \begin{pmatrix} 1 & 0 \\ -d_\varepsilon & A - D_x \end{pmatrix},$$

where d_ε and D_x are given in (2.3).

(ii) *If the interval matrix $[A-I, A+I]$ is regular, then $F'(w)$ is nonsingular for any $w \in \mathbb{R}_{++} \times \mathbb{R}^n$.*

(iii) *There exists $M > 0$ such that $\|F'(w)\| \leqslant M$ for any $w \in \mathbb{R}_{++} \times \mathbb{R}^n$.*

Proof. The result (i) obviously holds. For any $w \in \mathbb{R}_{++} \times \mathbb{R}^n$, since $-1 \leqslant x_i/\sqrt{\varepsilon^2 + x_i^2} \leqslant 1$ $(i = 1, \ldots, n)$, we have $D_x \in [-I, I]$ and hence $A - D_x \in [A-I, A+I]$. This together with the regularity of $[A-I, A+I]$ implies that $A - D_x$ is nonsingular and so is $F'(w)$. The result (iii) holds, since $\|d_\varepsilon\| \leqslant n$ and $\|D_x\| \leqslant n$ for any $w \in \mathbb{R}_{++} \times \mathbb{R}^n$. □

Theorem 2.1. *Let the interval matrix $[A-I, A+I]$ be regular. Then Algorithm 2.1 is well-defined and its generated sequence $\{w^k\} = \{(\varepsilon_k, x^k)\}$ satisfies $\varepsilon_k > 0$ and $\varepsilon_k \geqslant \tau_k$ for all $k \geqslant 0$.*

P r o o f. Suppose that $w^k = (\varepsilon_k, x^k) \in \mathbb{R}_{++} \times \mathbb{R}^n$ for some k. Since the interval matrix $[A-I, A+I]$ is regular, from Lemma 2.3 (ii), $F'(w^k)$ is nonsingular. Hence, there exists $(\Delta\varepsilon_k, \Delta x^k) \in \mathbb{R} \times \mathbb{R}^n$ satisfying

$$F'(w^k)(\Delta\varepsilon_k, \Delta x^k) = -F(w^k) + \begin{pmatrix} \tau_k \\ 0 \end{pmatrix},$$

which is equivalent to

$$\Delta\varepsilon_k = -\varepsilon_k + \tau_k, \quad E'(w^k)(\Delta\varepsilon_k, \Delta x^k) = -E(w^k).$$

Obviously, this $(\Delta\varepsilon_k, \Delta x^k)$ satisfies (2.6) and (2.7). Thus, we can get a search direction $\Delta w^k = (\Delta\varepsilon_k, \Delta x^k)$ in Step 2. Moreover, since

$$\lim_{l\to\infty} \psi(w^k + \delta^l \Delta w^k) = \psi(w^k) < (1+\theta_k)\psi(w^k) = \lim_{l\to\infty} (1+\theta_k - t\delta^l)\psi(w^k),$$

there must exist a nonnegative integer $\bar{l} > 0$ such that

$$\psi(w^k + \delta^{\bar{l}} \Delta w^k) \leqslant (1+\theta_k - t\delta^{\bar{l}})\psi(w^k).$$

This shows that the line search in Step 3 is feasible. So, we can find a step-size $\alpha_k \in (0,1]$ in Step 3 and get the $(k+1)$th iteration point $w^{k+1} = w^k + \alpha_k \Delta w^k$ in Step 4. Since $\varepsilon_k > 0$, from (2.6) we have

$$\varepsilon_{k+1} = \varepsilon_k + \alpha_k \Delta\varepsilon_k = (1-\alpha_k)\varepsilon_k + \alpha_k\tau_k > 0. \tag{2.13}$$

Thus, we can conclude that if $w^k = (\varepsilon_k, x^k) \in \mathbb{R}_{++} \times \mathbb{R}^n$ for some k, then w^{k+1} can be generated by Algorithm 2.1 with $w^{k+1} = (\varepsilon_{k+1}, x^{k+1}) \in \mathbb{R}_{++} \times \mathbb{R}^n$. Since $w^0 = (\varepsilon_0, x^0) \in \mathbb{R}_{++} \times \mathbb{R}^n$, by mathematical induction we prove that Algorithm 2.1 is well-defined and generates an infinite sequence $\{w^k\} = \{(\varepsilon_k, x^k)\}$ with $\varepsilon_k > 0$. Moreover, by (2.13), if $\varepsilon_k \geqslant \tau_k$ for some k, then $\varepsilon_{k+1} \geqslant (1-\alpha_k)\tau_k + \alpha_k\tau_k = \tau_k \geqslant \tau_{k+1}$, where the second inequality holds because $\{\tau_k\}$ is monotonically decreasing by its definition in (2.5). This together with $\varepsilon_0 \geqslant \gamma \geqslant \gamma\min\{1, \psi(w^0)\} = \tau_0$ gives $\varepsilon_k \geqslant \tau_k$ for all $k \geqslant 0$. The proof is completed. □

3.1. Global convergence.

Lemma 3.1 ([9], Lemma 2.2). *Let* $\{a_k\}$ *and* $\{\varrho_k\}$ *be positive sequences satisfying* $a_{k+1} \leqslant (1+\varrho_k)a_k + \varrho_k$ *and* $\sum\limits_{k=0}^{\infty} \varrho_k < \infty$. *Then* $\{a_k\}$ *is convergent.*

Lemma 3.2. *Suppose that the interval matrix* $[A-I, A+I]$ *is regular. Let* $\{w^k\}$ *be the iteration sequence generated by Algorithm* 2.1. *Then for all* $k \geqslant 0$,

$$\psi(w^k) \leqslant \psi(w^0)\mathrm{e}^{\theta}. \tag{3.1}$$

P r o o f. By Steps 3 and 4 in Algorithm 2.1, $\psi(w^{k+1}) \leqslant (1+\theta_k)\psi(w^k)$ for all $k \geqslant 0$. Hence, by following the proof of Lemma 2.1 in [9], we have for all $k \geqslant 0$

$$\begin{aligned}
\psi(w^{k+1}) &\leqslant (1+\theta_k)\psi(w^k) \leqslant \ldots \leqslant \psi(w^0)\prod_{i=0}^{k}(1+\theta_i) \\
&\leqslant \psi(w^0)\Big(\frac{1}{k+1}\sum_{i=0}^{k}(1+\theta_i)\Big)^{k+1} = \psi(w^0)\Big(1+\frac{1}{k+1}\sum_{i=0}^{k}\theta_i\Big)^{k+1} \\
&\leqslant \psi(w^0)\Big(1+\frac{\theta}{k+1}\Big)^{k+1} \leqslant \psi(w^0)\mathrm{e}^{\theta}.
\end{aligned}$$

This together with $\psi(w^0) \leqslant \psi(w^0)\mathrm{e}^{\theta}$ proves the lemma. □

Theorem 3.1 (Global convergence). *Let the interval matrix* $[A-I, A+I]$ *be regular and let* $\{w^k\} = \{(\varepsilon_k, x^k)\}$ *be the iteration sequence generated by Algorithm* 2.1. *Then the following results hold.*

(i) $\{w^k\}$ *is bounded.*
(ii) $\lim\limits_{k\to\infty} \psi(w^k) = 0$.
(iii) *Any accumulation point of* $\{w^k\}$ *is a solution of* $F(w) = 0$.

P r o o f. By Lemma 3.2, we have for all $k \geqslant 0$,

$$\psi(w^k) = \varepsilon_k^2 + \|Ax^k - \sqrt{\varepsilon_k^2 + (x^k)^2} - b\|^2 \leqslant \psi(w^0)\mathrm{e}^{\theta}. \tag{3.2}$$

This implies that $\{\varepsilon_k\}$ is bounded. So, to show $\{w^k\}$ is bounded, we only need to prove that $\{x^k\}$ is bounded. On the contrary, we may assume $\lim\limits_{k\to\infty} \|x^k\| = \infty$. Since the sequence $\{x^k/\|x^k\|\}$ is bounded, it has at least one accumulation point $\widetilde{x}$. By passing to a subsequence $\{k_n\}$, we may assume that $\lim\limits_{k_n\to\infty} x^{k_n}/\|x^{k_n}\| = \widetilde{x}$. Then it follows from the continuity of the 2-norm that $\|\widetilde{x}\| = 1$. From (3.2), we have

$$\left\| A\frac{x^{k_n}}{\|x^{k_n}\|} - \sqrt{\frac{\varepsilon_{k_n}^2}{\|x^{k_n}\|^2} + \Big(\frac{x^{k_n}}{\|x^{k_n}\|}\Big)^2} - \frac{b}{\|x^{k_n}\|}\right\|^2 \leqslant \frac{\psi(w^0)\mathrm{e}^\theta}{\|x^{k_n}\|^2}. \tag{3.3}$$

By letting $k_n \to \infty$ in (3.3), we have $A\widetilde{x} - |\widetilde{x}| = 0$, i.e., $(A - D(\widetilde{x}))\widetilde{x} = 0$. Since $D(\widetilde{x}) \in [-I, I]$, we have $A - D(\widetilde{x}) \in [A - I, A + I]$. This together with the regularity of $[A - I, A + I]$ implies that $A - D(\widetilde{x})$ is nonsingular. So, we have $\widetilde{x} = 0$, which contradicts $\|\widetilde{x}\| = 1$. Thus, $\{x^k\}$ is bounded and so is $\{w^k\}$.

Next we prove the result (ii). By Steps 3 and 4 in Algorithm 2.1, we have

$$\psi(w^{k+1}) \leqslant (1 + \theta_k - t\alpha_k)\psi(w^k) \leqslant (1 + \theta_k)\psi(w^k)$$

for all $k \geqslant 0$. Also notice that $\sum\limits_{k=0}^{\infty} \theta_k < \infty$. Thus, by Lemma 3.1, there exists a constant $\psi^* \geqslant 0$ such that

$$\lim_{k\to\infty} \psi(w^k) = \psi^*. \tag{3.4}$$

Since the sequence $\{\tau_k\}$ is monotonically decreasing and bounded from below by zero, there exists a constant $\tau^* \geqslant 0$ such that $\lim\limits_{k\to\infty} \tau_k = \tau^*$. Now we suppose $\tau^* > 0$ and will derive a contradiction. By the definition of τ_k, we have $\tau_k \leqslant \gamma\psi(w^k)$ for all $k \geqslant 0$, which gives $0 < \tau^* \leqslant \gamma\psi^*$ and hence $\psi^* > 0$. Since the sequence $\{w^k\} = \{(\varepsilon_k, x^k)\}$ is bounded, it has at least one accumulation point, denoted by $w^* := (\varepsilon^*, x^*)$. Then there exists a subsequence $\{w^k\}_{k\in K}$ such that $\lim\limits_{k\in K, k\to\infty} w^k = w^*$, where $K \subset \{0, 1, \ldots\}$. In what follows, we divide the proof into the following two cases.

Case 1 $\alpha_k \geqslant c$ for all $k \in K$, where $c > 0$ is a constant. Then from Steps 3 and 4 in Algorithm 2.1 we have

$$\psi(w^{k+1}) \leqslant (1 + \theta_k - t\alpha_k)\psi(w^k) \leqslant (1 + \theta_k - tc)\psi(w^k) \quad \forall\, k \in K. \tag{3.5}$$

Since $\sum\limits_{k=0}^{\infty} \theta_k < \infty$, we have $\lim\limits_{k\to\infty} \theta_k = 0$. Hence, by letting $k \to \infty$ with $k \in K$ in (3.5), we have $\psi^* \leqslant (1 - tc)\psi^*$, which contradicts $\psi^* > 0$.

Case 2 $\{\alpha_k\}_{k\in K}$ has a subsequence converging to 0. Without loss of generality, we assume $\lim\limits_{k\in K, k\to\infty} \alpha_k = 0$. From Theorem 2.1 we get

$$\varepsilon^* = \lim_{k\in K, k\to\infty} \varepsilon_k \geqslant \lim_{k\in K, k\to\infty} \tau_k = \tau^* > 0.$$

So, $F(w)$ is continuously differentiable at $w^* = (\varepsilon^*, x^*) \in \mathbb{R}_{++} \times \mathbb{R}^n$ and $F'(w^*)$ is nonsingular by Lemma 2.3 (ii). Thus, from Proposition 3.1 in [17], there exists a constant $C > 0$ such that for any sufficiently large $k \in K$,

$$\|F'(w^k)^{-1}\| \leqslant C. \tag{3.6}$$

By (2.5), (2.12), (3.1) and (3.6), also using $\|r_k\| \leqslant \eta_k \|E(w^k)\| \leqslant \eta \|F(w^k)\|$, we have for any sufficiently large $k \in K$,

$$\begin{aligned}\|\Delta w^k\| &= \|F'(w^k)^{-1}[-F(w^k) + (\tau_k, r_k^\top)^\top]\| \\ &\leqslant \|F'(w^k)^{-1}\|(\|F(w^k)\| + \tau_k + \|r_k\|) \\ &\leqslant C\Big[(1+\eta)\sqrt{\psi(w^0)\mathrm{e}^{\theta}} + \gamma\Big].\end{aligned}$$

Thus, $\{\Delta w^k\}_{k\in K}$ is bounded and it has one convergent subsequence. Without loss of generality, we assume $\lim\limits_{k\in K, k\to\infty} \Delta w^k = \Delta w^*$. From Step 3 in Algorithm 2.1, for any sufficiently large $k \in K$,

$$\psi(w^k + \delta^{-1}\alpha_k \Delta w^k) > (1 + \theta_k - t\delta^{-1}\alpha_k)\psi(w^k) \geqslant (1 - t\delta^{-1}\alpha_k)\psi(w^k),$$

which yields

$$\frac{\psi(w^k + \delta^{-1}\alpha_k \Delta w^k) - \psi(w^k)}{\delta^{-1}\alpha_k} > -t\psi(w^k). \tag{3.7}$$

Since $\psi(w)$ is continuously differentiable at $w^* = (\varepsilon^*, x^*) \in \mathbb{R}_{++} \times \mathbb{R}^n$, by letting $k \to \infty$ with $k \in K$ in (3.7), we have

$$\psi'(w^*)\Delta w^* \geqslant -t\psi^*. \tag{3.8}$$

On the other hand, since $\psi(w) = \|F(w)\|^2$, by (2.12) we have for all $k \geqslant 0$,

$$\begin{aligned}\frac{1}{2}\psi'(w^k)\Delta w^k &= F(w^k)^\top F'(w^k)\Delta w^k = -\psi(w^k) + \varepsilon_k \tau_k + E(w^k)^\top r_k \\ &\leqslant -\psi(w^k) + \gamma\psi(w^k) + \eta_k \|E(w^k)\|^2 \leqslant -(1-\gamma-\eta)\psi(w^k),\end{aligned}$$

where the first inequality holds since $\varepsilon_k \leqslant \sqrt{\psi(w^k)}$ and $\tau_k \leqslant \gamma \min\{1, \psi(w^k)\} \leqslant \gamma\sqrt{\psi(w^k)}$. This yields

$$\psi'(w^*)\Delta w^* \leqslant -2(1-\gamma-\eta)\psi^*. \tag{3.9}$$

By (3.8) and (3.9), we have $2(1-\gamma-\eta)\psi^* \leqslant t\psi^*$. This together with $\psi^* > 0$ gives $2(1-\gamma-\eta) \leqslant t$, which contradicts $t < 2(1-\gamma-\eta)$. Thus, we have $\lim\limits_{k\to\infty} \tau_k = \tau^* = 0$. Furthermore, by the definition of τ_k in (2.5), there exists a subsequence $\{w^{k_n}\}$ such that $\lim\limits_{k_n\to\infty} \psi(w^{k_n}) = 0$. This together with (3.4) proves the result (ii).

The result (iii) holds by the result (ii) and the continuity of ψ. We complete the proof. □

3.2. Local superlinear convergence. For any $w \in \mathbb{R}^{1+n}$, let $\partial F(w)$ be the Clarke generalized Jacobian of F at w (see [17]), which can be characterized by the convex hull of the set

$$\Big\{V\,;\; V = \lim_{w^k \to w} F'(w^k),\ F \text{ is differentiable at } w^k \in \mathbb{R}^{1+n}\Big\}.$$

Then we have the following lemma.

Lemma 3.3. *Suppose that the interval matrix $[A-I, A+I]$ is regular. Let w^* be any accumulation point of the iteration sequence $\{w^k\}$ generated by Algorithm* 2.1. *Then all $V \in \partial F(w^*)$ are nonsingular.*

P r o o f. The proof can be found in Theorem 2.4 in [22]. □

Lemma 3.4. *Suppose that the interval matrix $[A-I, A+I]$ is regular. Let τ_k, $F(w)$ and r_k be defined by* (2.5), (2.10) *and* (2.11), *respectively. Let w^* be any accumulation point of the iteration sequence $\{w^k\}$ generated by Algorithm* 2.1. *If $\eta_k \to 0$ as $k \to \infty$, then for all w^k sufficiently close to w^*,*

$$\|F(w^k) - F(w^*) - F'(w^k)(w^k - w^*)\| = o(\|w^k - w^*\|), \tag{3.10}$$

$$\|r_k\| = o(\|w^k - w^*\|), \tag{3.11}$$

$$\tau_k = o(\|w^k - w^*\|). \tag{3.12}$$

P r o o f. Equality (3.10) holds because F is semismooth at w^* (see, [6], Proposition 2.2 (iii)). Moreover, since $F(w^*) = 0$, by Taylor formula, for all w^k sufficiently close to w^*,

$$-F(w^k) = F(w^*) - F(w^k) = F'(w^k)(w^* - w^k) + o(\|w^* - w^k\|),$$

which together with Lemma 2.3 (iii) yields

$$\|F(w^k)\| = O(\|w^k - w^*\|).$$

Thus, for all w^k sufficiently close to w^*,

$$\|r_k\| \leqslant \eta_k \|E(w^k)\| \leqslant \eta_k \|F(w^k)\| = o(\|w^k - w^*\|),$$

$$\tau_k \leqslant \gamma\psi(w^k) = \gamma\|F(w^k)\|^2 = O(\|w^k - w^*\|^2) = o(\|w^k - w^*\|).$$

This completes the proof. □

Theorem 3.2 (Local superlinear convergence). *Suppose that the interval matrix $[A-I, A+I]$ is regular. Let w^* be any accumulation point of the iteration sequence $\{w^k\}$ generated by Algorithm* 2.1. *If $\eta_k \to 0$ as $k \to \infty$, then $\{w^k\}$ converges to w^* superlinearly with*

$$\|w^{k+1} - w^*\| = o(\|w^k - w^*\|).$$

Proof. By Lemma 3.3 and Lemma 3.4, similarly to the proof of Theorem 8 in [16], we can obtain that for all w^k sufficiently close to w^*,

$$\psi(w^k + \Delta w^k) = o(\psi(w^k)), \tag{3.13}$$

$$\|w^k + \Delta w^k - w^*\| = o(\|w^k - w^*\|). \tag{3.14}$$

By (3.13), we have for all w^k sufficiently close to w^*,

$$\psi(w^k + \Delta w^k) \leqslant (1 + \theta_k - t)\psi(w^k).$$

Thus, for all w^k sufficiently close to w^*, $\alpha_k = 1$ satisfies (2.8) and hence $w^{k+1} = w^k + \Delta w^k$. This together with (3.14) proves the theorem. □

4. Numerical results

In this section, we report some numerical results on Algorithm 2.1. All experiments were performed on a PC with CPU of Inter(R) Core(TM)i7-7700 CPU @ 3.60GHz and RAM of 8.00GB. The program codes are written in MATLAB and run in MATLAB R2018a environment. The parameters used in Algorithm 2.1 are chosen as $\varepsilon_0 = 10^{-2}$, $\delta = 0.8$, $\gamma = 10^{-3}$, $\eta_k = 1/(2^k + 1)$, $\theta^k = 0.9^k$, $t = 0.1$. Moreover, we use $\|Ax^k - |x^k| - b\| \leqslant 10^{-7}$ as the stopping criterion. In our tests, system (2.7) in Step 2 is solved by using the GMRES (m) package with m=20, allowing a maximum of 100 cycles (2000 iterations).

4.1. Algorithm 2.1 for solving the AVE (1.1).

In this subsection, we test the following five AVEs.

Example 4.1 ([15]). Let the matrix $A \in \mathbb{R}^{n\times n}$ be given by

$$a_{ii} = 4n, \quad a_{i,i+1} = a_{i+1,i} = n, \quad a_{ij} = 0.5, \quad i = 1, 2, \ldots, n.$$

Let $b := (A - I)e$, where $e = (1, \ldots, 1)^\top$. This example has an exact solution $x^* = (1, \ldots, 1)^\top$.

E x a m p l e 4.2 ([5], [24]). Choose $A = \mathrm{round}(100 * (\mathrm{eye}(n,n) - 0.02 * (2 * \mathrm{rand}\,(n,n) - 1)))$. Choose $\bar{x} = \mathrm{rand}\,(n,1)$ and let $b := A\bar{x} - |\bar{x}|$.

E x a m p l e 4.3. Let the matrix $A \in \mathbb{R}^{n\times n}$ be given by

$$A = \begin{pmatrix} 4 & -2 & 0 & \ldots & 0 & 0 \\ 1 & 4 & -2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 4 & -2 \\ 0 & 0 & 0 & \ldots & 1 & 4 \end{pmatrix}.$$

Choose $\bar{x} = \mathrm{rand}\,(n,1)$ and let $b := A\bar{x} - |\bar{x}|$.

E x a m p l e 4.4. The matrix A and the vector b are generated by the following MATLAB code:

```
%Sgen: Generate A and b in AVE (1.1)
D = diag(randperm(n)');
U = orth(rand (n));
A = U' * D * U;
A = 5 * round(A, 2);
x̄ = rand (n, 1) − rand (n, 1);
b = Ax̄ − |x̄|.
```

E x a m p l e 4.5 ([4], [21]). Consider the following ordinary differential equation (ODE):

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} - |x| = (1 - t^2), \quad x(0) = -1, \quad x(1) = 0, \quad t \in [0,1].$$

As explained in [4], Example 4.2, after discretization (by using finite difference method), the above ODE can be recast as the AVE (1.1), where the matrix A is given by

$$a_{i,j} = \begin{cases} -242, & i = j, \\ 121, & |i-j| = 1, \\ 0, & \text{otherwise}. \end{cases}$$

Choose $\bar{x} = \mathrm{rand}\,(n,1) - \mathrm{rand}\,(n,1)$ and set $b := A\bar{x} - |\bar{x}|$.

In the experiments, we first compare the following four methods:

▷ **INM**—Inexact Newton-type method studied in this paper.

▷ **SINM**—Smoothing inexact Newton-type method studied in [14]. This method is designed based on the well-known Zhang-Hager's nonmonotone line search technique [28].

▷ **GNM**—Generalized Newton method studied in [10].

▷ **DM**—Descent method studied in [22].

		INM		SINM		GNM		DM	
Ex	n	**aIT**	**aCPU**	**aIT**	**aCPU**	**aIT**	**aCPU**	**aIT**	**aCPU**
4.1	5000	3.0	0.36	3.0	0.36	1.0	2.65	4.0	11.42
	8000	4.0	1.57	4.0	1.63	1.0	8.32	6.0	60.13
	10000	4.0	2.91	4.0	2.82	1.0	14.50	5.0	96.13
	12000	4.0	4.67	4.0	5.25	1.0	24.41	6.0	184.51
	15000	4.0	10.56	4.0	10.78	1.0	45.46	5.0	318.83
4.2	5000	3.0	0.36	3.0	0.37	1.0	3.68	4.7	13.44
	8000	5.3	1.62	5.3	1.77	1.0	14.15	6.2	61.97
	10000	5.8	4.59	5.8	5.12	1.0	25.66	6.0	112.65
	12000	4.3	5.18	4.3	5.52	1.0	43.46	5.2	160.47
	15000	4.9	12.51	4.9	13.61	1.0	82.78	5.6	359.23
4.3	5000	3.0	0.36	3.0	0.36	1.0	0.54	4.1	12.16
	8000	3.0	1.23	3.0	1.25	1.0	2.42	4.5	48.04
	10000	3.0	2.17	3.0	2.08	1.0	2.77	4.6	92.10
	12000	3.0	3.74	3.0	4.02	1.0	5.10	4.7	160.16
	15000	3.2	7.15	3.2	7.77	1.0	7.25	5.0	338.49
4.4	2000	3.0	0.04	3.0	0.04	3.0	0.57	4.0	0.99
	5000	3.1	0.37	3.1	0.41	3.0	6.02	4.0	11.38
	8000	3.0	1.21	3.0	1.20	3.0	21.11	4.2	43.02
	10000	3.1	2.36	3.1	2.85	3.0	38.24	4.5	86.49
	12000	3.1	4.97	3.1	4.52	3.0	64.30	4.1	133.40
4.5	5000	7.4	0.98	7.4	0.94	7.0	15.34	7.9	21.95
	8000	7.4	3.44	7.4	3.45	8.0	57.32	8.0	80.56
	10000	7.2	6.50	7.2	7.42	7.0	89.97	8.0	149.24
	12000	7.5	13.92	7.5	13.94	7.0	146.72	8.0	265.23
	15000	7.4	36.19	7.4	35.01	8.0	311.45	8.0	508.73

Table 1. Comparison of **INM**, **SINM**, **GNM** and **DM** for AVEs with $x^0 = \text{rand}\,(n, 1)$

For every example we test 10 times by using a starting point $x^0 = \text{rand}\,(n, 1)$. Numerical results are listed in Table 1, in which **Ex** denotes the test example, n denotes the size of the problem, **aIT** and **aCPU** denote the average value of the iteration numbers and the average value of the CPU time in seconds, respectively. From Table 1, three observations can be made here.

(i) Although the line search technique adopted in **INM** is much simpler than that used in **SINM**, **INM** and **SINM** have similar computation effects.

(ii) **INM** can save large computation time compared to **DM**.

(iii) **INM** takes more iteration numbers than **GNM** does, while less CPU time is needed.

In the above experiments, we find that **GNM** only needs one iteration when we apply it to solve Examples 4.1–4.3. It may be because the starting point $x^0 = \mathrm{rand}\,(n,1)$ is close to the solution of these AVEs. Thus, in order to more accurately display the computation efficiency of algorithms, we apply **INM** and **GNM** to solve Examples 4.1–4.4 again by taking $x^0 = 5 * \mathrm{rand}\,(n,1)$. Numerical results are listed in Table 2, in which **aRES** denotes the average value of $\|Ax^k - |x^k| - b\|$ when the algorithm terminates. It is worth pointing out that we also apply **INM** and **GNM** with $x^0 = 5 * \mathrm{rand}\,(n,1)$ to solve Example 4.5, but two methods are all failed. From Table 2, we find that **INM** also takes more iteration numbers than **GNM** does, while less CPU time is needed.

		INM			**GNM**		
Ex	n	**aIT**	**aCPU**	**aRES**	**aIT**	**aCPU**	**aRES**
4.1	5000	3.0	0.36	8.0331e − 08	2.0	4.95	1.5101e − 09
	8000	4.0	1.61	3.0701e − 08	2.0	16.34	3.4913e − 09
	10000	4.0	2.91	5.6824e − 08	2.0	28.15	5.2873e − 09
	12000	4.0	4.94	7.5132e − 08	2.0	46.85	1.4122e − 08
	15000	4.0	11.27	8.1970e − 08	2.0	86.72	2.3114e − 08
4.2	2000	3.0	0.05	8.9597e − 09	3.0	0.96	6.2348e − 12
	5000	3.1	0.36	5.4121e − 09	3.0	10.60	2.2051e − 11
	8000	6.9	2.99	2.8298e − 11	7.6	99.20	1.1304e − 10
	10000	8.2	9.12	3.7558e − 11	7.5	184.95	6.5056e − 10
	12000	7.4	26.84	4.9505e − 11	6.4	268.55	1.7292e − 09
4.3	5000	4.1	0.46	2.1385e − 14	4.0	2.08	1.5768e − 14
	8000	4.1	1.63	2.7705e − 14	4.0	6.58	1.9632e − 14
	10000	4.3	3.55	2.6024e − 14	4.0	10.53	2.1928e − 14
	12000	4.0	5.49	3.3660e − 14	4.0	17.69	2.3977e − 14
	15000	4.0	10.87	4.2512e − 14	4.0	27.77	2.6524e − 14
4.4	2000	3.0	0.05	1.3546e − 08	3.0	0.58	8.7027e − 11
	5000	3.1	0.37	8.2002e − 09	3.0	6.07	9.9641e − 10
	8000	3.0	1.21	2.0361e − 08	3.0	20.41	2.5201e − 09
	10000	3.1	2.22	1.5487e − 08	3.0	39.11	3.8757e − 09
	12000	3.1	4.32	2.6078e − 08	3.0	62.83	5.6556e − 09

Table 2. Comparison of **INM** and **GNM** for AVEs with $x^0 = 5 * \mathrm{rand}\,(n,1)$.

Next, we compare **INM** with the inexact Douglas-Rachford splitting method studied in [3], denoted by **IDRSM**. We apply these two algorithms to solve Examples 4.1–4.4 by using the starting point $x^0 = 0.5 * \mathrm{rand}\,(n,1)$. Numerical results are listed in Table 3, which shows that **INM** converges faster than **IDRSM** does. This confirms the theoretical properties that the convergence rate of **INM** is superlinear while the convergence rate of **IDRSM** is linear.

		INM			**IDRSM**		
Ex	n	**aIT**	**aCPU**	**aRES**	**aIT**	**aCPU**	**aRES**
4.1	5000	3.0	0.37	$8.0265\text{e}-08$	6.0	3.61	$2.6344\text{e}-08$
	8000	4.0	1.75	$6.6018\text{e}-08$	6.0	6.61	$5.1136\text{e}-08$
	10000	4.0	3.14	$5.2037\text{e}-08$	6.0	11.69	$7.2468\text{e}-08$
	12000	4.0	6.28	$1.2478\text{e}-08$	6.0	19.43	$8.6660\text{e}-08$
	15000	3.0	13.38	$4.1268\text{e}-08$	7.0	30.71	$2.3896\text{e}-09$
4.2	1000	3.0	0.02	$9.0788\text{e}-09$	6.0	0.22	$2.9674\text{e}-08$
	2000	3.1	0.05	$6.5810\text{e}-09$	6.3	0.67	$6.6841\text{e}-08$
	3000	3.0	0.11	$3.9247\text{e}-09$	7.0	1.51	$1.0793\text{e}-08$
	4000	3.0	0.21	$7.3085\text{e}-09$	7.0	3.95	$7.1550\text{e}-08$
	5000	3.0	0.36	$1.2373\text{e}-08$	8.2	8.31	$6.1360\text{e}-08$
4.3	5000	3.0	0.46	$8.9701\text{e}-08$	6.2	6.68	$5.7786\text{e}-08$
	8000	3.0	1.88	$2.1453\text{e}-09$	6.0	9.44	$4.4760\text{e}-08$
	10000	3.0	3.96	$6.7820\text{e}-09$	6.2	14.54	$2.0340\text{e}-08$
	12000	3.0	5.23	$3.432\text{e}-08$	6.2	22.41	$7.1622\text{e}-08$
	15000	3.0	10.44	$6.9211\text{e}-08$	6.8	35.27	$5.4368\text{e}-08$
4.4	5000	3.0	0.56	$8.4798\text{e}-09$	6.0	5.52	$7.3209\text{e}-09$
	8000	3.0	1.45	$1.7765\text{e}-08$	6.0	7.62	$1.1249\text{e}-09$
	10000	3.0	3.11	$4.2214\text{e}-09$	6.0	11.21	$6.0074\text{e}-09$
	12000	3.0	6.12	$3.1359\text{e}-08$	6.0	17.37	$6.7502\text{e}-09$
	15000	3.1	11.46	$1.0998\text{e}-09$	6.0	27.53	$8.9934\text{e}-09$

Table 3. Comparison of **INM** and **IDRSM** for AVEs with $x^0 = 0.5 * \mathrm{rand}\,(n, 1)$.

4.2. Algorithm 2.1 for solving the LCP. As pointed out in [13], Proposition 2, every linear complementarity problem (LCP)

$$z \geqslant 0, \quad Mz + q \geqslant 0, \quad z^\top (Mz + q) = 0 \tag{4.1}$$

for a given $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$, can be reduced to the AVE

$$(M - I)^{-1}(M + I)x - |x| = (M - I)^{-1}q, \tag{4.2}$$

with

$$x = \frac{1}{2}((M - I)z + q),$$

provided 1 is not an eigenvalue of M, which can be easily achieved by rescaling M and q. In this subsection, we apply Algorithm 2.1 to solve the following two LCPs.

E x a m p l e 4.6. Consider the LCP (4.1), where

$$M = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 & 0 \\ -1 & 4 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4 & -1 \\ 0 & 0 & 0 & \dots & -1 & 4 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ -1 \end{pmatrix}.$$

E x a m p l e 4.7. Consider the LCP (4.1), where $M = 10*(B/\|B\| + \mathrm{eye}(n))$ with $B = \mathrm{rand}\,(n,n)$ and $q = (1,\dots,1)^\top$.

The corresponding AVE solved for each LCP was that specified by (4.2). We take $x^0 = \mathrm{rand}\,(n,1)$ as a starting point. Numerical results are listed in Table 4, in which **IT** and **CPU** denote the iteration numbers and the CPU time in seconds respectively, and **RES** denotes the value of $\|Ax^k - |x^k| - b\|$ when the algorithm terminates, where $A := (M-I)^{-1}(M+I)$ and $b := (M-I)^{-1}q$. From Table 4 we may see that our algorithm is also very effective for solving large-scale LCPs.

Ex	n	**IT**	**CPU**	**RES**
4.6	5000	3	5.89	1.4145e − 10
	8000	3	16.25	1.7890e − 10
	10000	3	27.59	2.0001e − 10
	12000	3	47.10	2.1910e − 10
	15000	3	96.26	2.4495e − 10
4.7	5000	3	3.61	7.0713e − 11
	8000	3	12.36	8.9446e − 11
	10000	3	29.02	9.9997e − 11
	12000	3	55.13	1.0955e − 10
	15000	3	187.31	1.2248e − 10

Table 4. Numerical results of Algorithm 2.1 for solving LCPs.

5. Conclusions

We proposed an inexact Newton-type method for solving the AVE. The proposed method has global and local superlinear convergence under the condition that the interval matrix $[A-I, A+I]$ is regular. This condition is much weaker than those used in some Newton-type methods. More sufficient conditions for the regularity of $[A-I, A+I]$ can be found in [18]. In each iteration, the proposed method solves the corresponding system only approximately and adopts a new line search technique. Numerical results show that our method is very effective for solving large-scale AVEs. It is worth pointing out that, although the inexact Newton-type method saves a lot of computational work in practical computation compared to Newton-type method, it still needs to calculate the Jacobian matrix at every iteration, which may be cumbersome if one is solving a large-scale problem or the Jacobian matrix is difficult to calculate. This drawback may be overcame by inexact quasi-Newton method (e.g., [1]). Lately, Tang et al. [23] proposed an accelerated smoothing Newton method (ASNM) which has cubic convergence rate. As a future research issue, it is worth investigating the inexact version of ASNM and applying it to solve large-scale AVEs. It is also worth pointing out that many new types of AVEs have been proposed in recent years (e.g., [7], [8], [25], [30]). As another interesting issue, it is worth investigating the convergence properties and numerical performance of the (inexact) Newton-type methods for solving these new AVEs.

Acknowledgement. We are very grateful to the referees for their valuable comments on the paper, which have considerably improved the paper.

References

[1] *C. A. Arias, H. J. Martínez, R. Pérez*: Global inexact quasi-Newton method for nonlinear system of equations with constraints. Appl. Numer. Math. *150* (2020), 559–575. zbl MR doi

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

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

[4] *F. K. Haghani*: On generalized Traub's method for absolute value equations. J. Optim. Theory Appl. *166* (2015), 619–625. zbl MR doi

[5] *J. Iqbal, A. Iqbal, M. Arif*: Levenberg-Marquardt method for solving systems of absolute value equations. J. Comput. Appl. Math. *282* (2015), 134–138. zbl MR doi

[6] *X. Jiang, Y. Zhang*: A smoothing-type algorithm for absolute value equations. J. Ind. Manag. Optim. *9* (2013), 789–798. zbl MR doi

[7] *S. Kumar, Deepmala*: A note on unique solvability of the generalized absolute value matrix equation. Natl. Acad. Sci. Lett. *46* (2023), 129–131. MR doi

[8] *S. Kumar, Deepmala*: The unique solvability conditions for a new class of absolute value equation. To appear in Yugosl. J. Oper. Res. doi

[9] *D. Li, M. Fukushima*: A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optim. Methods Softw. *13* (2000), 181–201. zbl MR doi

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

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

[12] *O. L. Mangasarian*: A hybrid algorithm for solving the absolute value equation. Optim. Lett. *9* (2015), 1469–1474. zbl MR doi

[13] *O. L. Mangasarian, R. R. Meyer*: Absolute value equations. Linear Algebra Appl. *419* (2006), 359–367. zbl MR doi

[14] *T. Ni, W.-Z. Gu, J. Zhai*: An inexact smoothing-type algorithm for support vector machines. Neurocomputing *129* (2014), 127–135. doi

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

[16] *L. Qi, D. Sun, G. Zhou*: A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. Math. Program. *87* (2000), 1–35. zbl MR doi

[17] *L. Qi, J. Sun*: A nonsmooth version of Newton's method. Math. Program. *58* (1993), 353–367. zbl MR doi

[18] *G. Rex, J. Rohn*: Sufficient conditions for regularity and singularity of interval matrices. SIAM J. Matrix Anal. Appl. *20* (1998), 437–445. zbl MR doi

[19] *J. Rohn*: Systems of linear interval equations. Linear Algebra Appl. *126* (1989), 39–78. zbl MR doi

[20] *J. Rohn*: An algorithm for computing all solutions of an absolute value equation. Optim. Lett. *6* (2012), 851–856. zbl MR doi

[21] *B. Saheya, C.-H. Yu, J.-S. Chen*: Numerical comparisons based on four smoothing functions for absolute value equation. J. Appl. Math. Comput. *56* (2018), 131–149. zbl MR doi

[22] *J. Tang, J. Zhou*: A quadratically convergent descent method for the absolute value equation $Ax + B|x| = b$. Oper. Res. Lett. *47* (2019), 229–234. zbl MR doi

[23] *J. Tang, J. Zhou, H. Zhang*: An accelerated smoothing Newton method with cubic convergence for weighted complementarity problems. J. Optim. Theory Appl. *196* (2023), 641–665. zbl MR doi

[24] *A. Wang, H. Wang, Y. Deng*: Interval algorithm for absolute value equations. Cent. Eur. J. Math. *9* (2011), 1171–1184. zbl MR doi

[25] *S.-L. Wu*: The unique solution of a class of the new generalized absolute value equation. Appl. Math. Lett. *116* (2021), Article ID 107029, 6 pages. zbl MR doi

[26] *D. Yu, C. Chen, D. Han*: An inexact framework of the Newton-based matrix splitting iterative method for the generalized absolute value equation. Available at `https://arxiv.org/abs/2103.10129` (2022), 15 pages. doi

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

[28] *H. Zhang, W. W. Hager*: A nonmonotone line search technique and its application to unconstrained optimization. SIAM. J. Optim. *14* (2004), 1043–1056. zbl MR doi

[29] *J.-J. Zhang*: The relaxed nonlinear PHSS-like iteration method for absolute value equations. Appl. Math. Comput. *265* (2015), 266–274. zbl MR doi

[30] *H. Zhou, S. Wu*: On the unique solution of a class of absolute value equations $Ax - B|Cx| = d$. AIMS Math. *6* (2021), 8912–8919. zbl MR doi

Author's address: *Jingyong Tang*, College of Mathematics and Statistics, Xinyang Normal University, 464000 Xinyang, P. R. China, e-mail: `tangjy@xynu.edu.cn`.