symmetry

MDPI

Article

On the Alternative SOR-like Iteration Method for Solving Absolute Value Equations

Yiming Zhang, Dongmei Yu * and Yifei Yuan

Institute for Optimization and Decision Analytics, Liaoning Technical University, Fuxin 123000, China
* Correspondence: yudongmei1113@163.com

Abstract: In this paper, by equivalently reformulating the absolute value equation (AVE) into an alternative two-by-two block nonlinear equation, we put forward an alternative SOR-like (ASOR-like) iteration method to solve the AVE. The convergence of the ASOR-like iteration method is established, subjecting to specific restrictions placed on the associated parameter. The selection of the optimal iteration parameter is investigated theoretically. Numerical experiments are given to validate the feasibility and effectiveness of the ASOR-like iteration method.

Keywords: absolute value equations; alternative; SOR-like method; convergence analysis

MSC: 65F10; 65H10; 90C30

Citation: Zhang, Y.; Yu, D.; Yuan, Y. On the Alternative SOR-like Iteration Method for Solving Absolute Value Equations. *Symmetry* **2023**, *15*, 589. https://doi.org/10.3390/sym15030589

Academic Editor: Aviv Gibali

Received: 29 January 2023
Revised: 15 February 2023
Accepted: 17 February 2023
Published: 24 February 2023

1. Introduction

The absolute value equation, denoted by AVE, is to find a vector $x \in \mathbb{R}^n$ such that

$$Ax - |x| - b = 0, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n$, $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^\top$ with x_l being the l-th entry of x and $|\cdot|$ denotes absolute value for real scalar. The AVE (1) is a special case of the generalized AVE (GAVE)

$$Ax + B|x| - b = 0, \tag{2}$$

with $A, B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, which was introduced in [1] and further studied in [2–4]. In fact, if B is nonsingular, then (2) can be converted into (1). The AVE (1) is closely interrelated to the linear complementarity problem (LCP), bimatrix games and others (see e.g., [1–8] and the references therein).

In general, solving the AVE (1) is NP-hard [2]. Furthermore, when the AVE (1) is solvable, it is NP-complete for testing whether the AVE (1) has a unique solution or multiple solutions [9]. The existence of its solutions have been studied in [1,10–14], and an outstanding and commonly used sufficient condition for solving the AVE (1) can be found in [11], as follows.

Lemma 1 ([11])**.** *Assume that $A \in \mathbb{R}^{n\times n}$ is invertible. If $\|A^{-1}\| < 1$, then the AVE (1) is uniquely solvable for any $b \in \mathbb{R}^n$.*

For solving the AVE (1), a great deal of numerical methods have been proposed, such as the Newton-type iteration methods [4,7,15–23], the Picard iteration method [24], the preconditioned AOR iteration method [25], the generalized Gauss–Seidel iteration method [26], the Levenberg–Marquardt methods [27,28], the exact and inexact Douglas–Rachford splitting methods [29], the dynamical systems [30–35], the modified multivariate spectral gradient algorithm [36], the modified HS conjugate gradient method [37], and others (see e.g., [3,38–41] and the references therein).

In recent years, the SOR-like iteration methods have attracted considerable attention. Ke and Ma [42] first developed an SOR-like iteration method (Algorithm 1) by converting the AVE (1) into a two-by-two block nonlinear equation to address the AVE (1), and proved the convergence of the Algorithm 1 under the sufficient condition that $\|A^{-1}\| < 1$ with $\omega \in (0, 2)$.

Algorithm 1 ([42]). (The SOR-like iteration method)

Let the matrix A be nonsingular. Given two initial guesses $x^0, y^0 \in \mathbb{R}^n$, for $k = 0, 1, \cdots$ until the generated sequence $\{x^k\}$ is convergent, compute

$$\begin{cases} x^{k+1} = (1-\omega)x^k + \omega A^{-1}(y^k + b), \\ y^{k+1} = (1-\omega)y^k + \omega|x^{k+1}|. \end{cases} \tag{3}$$

In order to further explore the convergence conditions of the SOR-like iteration method for solving the AVE (1) in [42], Guo et al. [43] proved the convergence of Algorithm 1 from the perspective of spectral radius and got the optimal relaxation parameter $\omega_0 = \frac{2}{1+\sqrt{1-\rho}}$ with $\rho = \rho(D(x^{k+1})A^{-1})$, $D(x) \doteq diag(sign(x))$. Herein, $diag(x)$ represents a diagonal matrix with x_i as its diagonal entries for every $i = 1, 2, \cdots, n$ and

$$sign(x_i) = \begin{cases} 1, \text{ if } x_i > 0, \\ 0, \text{ if } x_i = 0, \\ -1, \text{ if } x_i < 0. \end{cases}$$

In the sequel, Chen et al. [44] investigated the theoretical optimal parameter ω^*_{opt} and the approximate optimal parameter ω^*_{aopt} of Algorithm 1 for resolving the AVE (1), resulting in

$$\omega^*_{opt} = \begin{cases} 1, \text{ if } 0 < \|A^{-1}\| \leq \dfrac{1}{4}, \\ \omega_{opt}, \text{ if } \dfrac{1}{4} < \|A^{-1}\| < 1, \end{cases} \tag{4}$$

$$\omega^*_{aopt}(\nu) = \frac{\sqrt{4\nu + 1} - 1}{2\nu}. \tag{5}$$

Meanwhile, by reformulating the AVE (1) as a two-by-two block nonlinear equation, a fixed point iteration (FPI) method was suggested for solving the AVE (1) in [45], but the convergence of the FPI method is only guaranteed for the case that $0 < \|A^{-1}\| < \frac{\sqrt{2}}{2}$. Furthermore, Yu et al. [46] put forward a modified fixed point iteration (MFPI) method by introducing a nonsingular matrix Q, which guaranteed the convergence for solving the AVE (1) with $\frac{\sqrt{2}}{2} \leq \|A^{-1}\| < 1$ by selecting an appropriate parameter matrix Q. In addition, Dong et al. [47] proposed a new SOR-like (NSOR) iteration method by rewriting the AVE (1) into a new two-by-two block nonlinear system, and the convergence conditions of the NSOR iteration method were proven from the perspective of spectrum. In this paper, by reformulating the AVE (1) into a new alternative two-by-two block nonlinear system, we propose an alternative SOR-like (ASOR-like) iteration method for solving the AVE (1) and prove its convergence from the view of iteration error and spectrum, respectively. Furthermore, the optimal iteration parameter selection is also discussed. In addition, we use numerical experiments to demonstrate the feasibility and effectiveness of the ASOR-like iteration method.

The layout of this paper is organized below. Section 2 explains some of the mathematical notations and the lemmas that are used later in the proof. Section 3 and Section 4 propose the iterative format, the convergence conditions and the optimal iteration parameter selection of the ASOR-like iteration method. In Section 5, some numerical experiments are conducted to prove the effectiveness of the proposed method by comparing it with some existing algorithms. Finally, we give a brief conclusion in Section 6.

2. Preliminaries

In this section, we will present some notations, classical definitions, and some auxiliary results that lay the foundation of our developments.

We start by recalling some notations and definitions used in this paper. $\mathbb{R}^{n\times n}$ is the set of all $n \times n$ real matrices and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. I is the identity matrix with suitable dimension. $\rho(A)$ denotes the spectral radius of A and is defined by the formula $\rho(A) \doteq \max|\lambda(A)|$ where $\lambda(A)$ denotes the eigenvalue of A. $\|A\| \doteq \max\{\|Ax\| : x \in \mathbb{R}^n, \|x\| = 1\}$ denotes the spectral norm of A, where $\|x\|^2 = x^H x$. $\|A\|_2$ denotes the 2-norm of A. Based on this definition we can derive

$$\|Ax\| \le \|A\|\|x\|,\ \|A+B\| \le \|A\| + \|B\|,\ \|AB\| \le \|A\|\|B\|, \tag{6}$$

where $A, B \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$ (see Chapter 5 of [48]).

Lemma 2 ([49])**.** *For any vectors $x, y \in \mathbb{R}^n$, the following results hold:*

- $\||x| - |y|\| \le \|x - y\|$;
- *If* $0 \le x \le y$, *then* $\|x\| \le \|y\|$;
- *Assume that P is a nonnegative matrix. If $x \le y$, then $Px \le Py$.*

3. An Alternative SOR-like Iteration Method

In this section, we put forward an alternative two-by-two block nonlinear system of the AVE (1). Let $y = x$, and then the AVE (1) is equivalent to

$$\begin{cases} Ay - |x| = b, \\ \quad x - y = 0, \end{cases} \tag{7}$$

that is

$$\boldsymbol{\mathcal{A}z} := \begin{pmatrix} \boldsymbol{A} & -\boldsymbol{D}(\boldsymbol{x}) \\ -\boldsymbol{I} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix} := \boldsymbol{b},$$

where $D(x) := diag(sign(x)), x \in \mathbb{R}^n$.

Let $\mathcal{A} = \mathcal{D} - \mathcal{L} - \mathcal{U}$, where

$$\mathcal{D} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix},\ \mathcal{L} = \begin{pmatrix} 0 & 0 \\ I & 0 \end{pmatrix},\ \mathcal{U} = \begin{pmatrix} 0 & D(x) \\ 0 & 0 \end{pmatrix},$$

and then the following fixed point equation can be gained,

$$(\mathcal{D} - \omega\mathcal{L})z = [(1-\omega)\mathcal{D} + \omega\mathcal{U}]z + \omega b,$$

where the parameter $\omega > 0$. That is,

$$\begin{pmatrix} A & 0 \\ -\omega I & I \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} (1-\omega)A & \omega D(x) \\ 0 & (1-\omega)I \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} + \begin{pmatrix} \omega b \\ 0 \end{pmatrix}. \tag{8}$$

Based on (8), we establish the following matrix splitting iteration method to solve the AVE (1), called the alternative SOR-like (ASOR-like) iteration method. The algorithmic framework for this method is as follows.

Algorithm 2 (The ASOR-like iteration method)

Let the matrix A be nonsingular. Given two initial guesses $x^0, y^0 \in \mathbb{R}^n$, for $k = 0, 1, \cdots$ until the generated sequence $\{x^k\}$ is convergent, compute

$$\begin{cases} y^{k+1} = (1-\omega)y^k + \omega A^{-1}(|x^k| + b), \\ x^{k+1} = (1-\omega)x^k + \omega y^{k+1}. \end{cases} \tag{9}$$

In the following, we demonstrate the main outcomes of this paper. Theorems 1 and 2 are inspired by that of Theorem 3.1 in [42] and Theorem 2.1 in [44], respectively. Let (y^*, x^*) be the solution pair of the nonlinear system (7), then we have

$$y^* = (1-\omega)y^* + \omega A^{-1}(|x^*| + b), \tag{10}$$
$$x^* = (1-\omega)x^* + \omega y^*. \tag{11}$$

Let the vector pair (y^k, x^k) be generated by (9), and define the iteration errors as

$$e_k^y = y^* - y^k \text{ and } e_k^x = x^* - x^k. \tag{12}$$

Then, the convergence results of the ASOR-like iteration method can be obtained as follows.

Theorem 1. *Let the matrix A be invertible. Denote*

$$\nu = \|A^{-1}\| \quad \textit{and} \quad T = \begin{pmatrix} |1-\omega| & \omega\nu \\ \omega|1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix},$$

if $\|T\| < 1$, then $\|E_{k+1}\| \le \|E_k\|$, where $\|E_{k+1}\| = (\|e_{k+1}^y\|, \|e_{k+1}^x\|)^T$.

Proof. From (9), (10), (11), and (12), we have

$$e_{k+1}^y = (1-\omega)e_k^y + \omega A^{-1}(|x^*| - |x^k|), \tag{13}$$
$$e_{k+1}^x = (1-\omega)e_k^x + \omega e_{k+1}^y. \tag{14}$$

According to (13), (14) and Lemma 2, we can get

$$\begin{aligned} \|e_{k+1}^y\| &\le |1-\omega|\|e_k^y\| + \omega\nu\||x^*| - |x^k|\| \\ &\le |1-\omega|\|e_k^y\| + \omega\nu\|e_k^x\|, \\ \|e_{k+1}^x\| &\le |1-\omega|\|e_k^x\| + \omega\|e_{k+1}^y\|. \end{aligned}$$

Thus, we can derive that

$$\begin{pmatrix} 1 & 0 \\ -\omega & 1 \end{pmatrix} \begin{pmatrix} \|e_{k+1}^y\| \\ \|e_{k+1}^x\| \end{pmatrix} \le \begin{pmatrix} |1-\omega| & \omega\nu \\ 0 & |1-\omega| \end{pmatrix} \begin{pmatrix} \|e_k^y\| \\ \|e_k^x\| \end{pmatrix}. \tag{15}$$

Let

$$P = \begin{pmatrix} 1 & 0 \\ \omega & 1 \end{pmatrix} \ge 0.$$

According to Lemma 2, multiplying (15) from left by the nonnegative matrix P, it holds that

$$\begin{pmatrix} \|e_{k+1}^y\| \\ \|e_{k+1}^x\| \end{pmatrix} \le \begin{pmatrix} |1-\omega| & \omega\nu \\ \omega|1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix} \begin{pmatrix} \|e_k^y\| \\ \|e_k^x\| \end{pmatrix}. \tag{16}$$

Denote

$$\|E_{k+1}\| = \begin{pmatrix} \|e^y_{k+1}\| \\ \|e^x_{k+1}\| \end{pmatrix} \text{ and } T = \begin{pmatrix} |1-\omega| & \omega\nu \\ \omega|1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix} \geq 0.$$

In the light of (16), it follows that

$$\|E_{k+1}\| \leq \|TE_k\| \leq \|T\|\|E_k\|.$$

If $\|T\| < 1$, then we can obtain

$$\|E_{k+1}\| \leq \|E_k\|.$$

This completes the proof. □

Theorem 2. *Let the matrix A be invertible. Denote $\nu = \|A^{-1}\|$, $\varphi = |1-\omega|$, $\psi = \omega^2\nu$, if*

$$0 \leq 3\varphi^2 + 2\psi^2 + 2\varphi\psi < min\{1+\varphi^4, 2\}, \tag{17}$$

then the following inequality holds,

$$|||(e^y_{k+1}, e^x_{k+1})||| \leq |||(e^y_k, e^x_k)|||, \tag{18}$$

where $|||\cdot|||$ is defined by

$$|||(e^y, e^x)||| = \sqrt{\|e^y\|^2 + \omega^{-2}\|e^x\|^2}.$$

Proof. According to the proof of Theorem 1, we get

$$\begin{pmatrix} \|e^y_{k+1}\| \\ \|e^x_{k+1}\| \end{pmatrix} \leq \begin{pmatrix} |1-\omega| & \omega\nu \\ \omega|1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix} \begin{pmatrix} \|e^y_k\| \\ \|e^x_k\| \end{pmatrix}. \tag{19}$$

Denote

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & \omega^{-1} \end{pmatrix} \geq 0.$$

Multiplying (19) from left by the nonnegative matrix Q, we get

$$\begin{pmatrix} \|e^y_{k+1}\| \\ \omega^{-1}\|e^x_{k+1}\| \end{pmatrix} \leq \begin{pmatrix} |1-\omega| & \omega^2\nu \\ |1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix} \begin{pmatrix} \|e^y_k\| \\ \omega^{-1}\|e^x_k\| \end{pmatrix}.$$

Then it can be concluded that

$$|||(e^y_{k+1}, e^x_{k+1})||| \leq \|\hat{T}\| \cdot |||(e^y_k, e^x_k)|||,$$

where

$$\hat{T} = \begin{pmatrix} |1-\omega| & \omega^2\nu \\ |1-\omega| & |1-\omega| + \omega^2\nu \end{pmatrix} := \begin{pmatrix} \varphi & \psi \\ \varphi & \varphi+\psi \end{pmatrix} \geq 0.$$

Next, we discuss the selection of the iteration parameter ω such that $\|\hat{T}\|^2 < 1$, thus the inequality (18) holds.

Because

$$\hat{T}^\top\hat{T} = \begin{pmatrix} 2\varphi^2 & \varphi^2 + 2\varphi\psi \\ \varphi^2 + 2\varphi\psi & \varphi^2 + 2\psi^2 + 2\varphi\psi \end{pmatrix}$$

is a symmetric positive semidefinite matrix, then we have $\|\hat{T}\|^2 = \rho(\hat{T}^\top\hat{T}) = \kappa_{max}(\hat{T}^\top\hat{T})$, where κ is an eigenvalue of $\hat{T}^\top\hat{T}$, and then it holds that

$$(\kappa - 2\varphi^2)[\kappa - (\varphi^2 + 2\psi^2 + 2\varphi\psi)] - (\varphi^2 + 2\varphi\psi)^2 = 0,$$

namely,

$$\kappa^2 - (3\varphi^2 + 2\psi^2 + 2\varphi\psi)\kappa + \varphi^4 = 0,$$

from which we obtain

$$\kappa = \frac{3\varphi^2 + 2\psi^2 + 2\varphi\psi \pm \sqrt{(3\varphi^2 + 2\psi^2 + 2\varphi\psi)^2 - 4\varphi^4}}{2}.$$

Consequently,

$$\kappa_{max}(\hat{T}^\top \hat{T}) = \frac{3\varphi^2 + 2\psi^2 + 2\varphi\psi + \sqrt{(3\varphi^2 + 2\psi^2 + 2\varphi\psi)^2 - 4\varphi^4}}{2}.$$

In particular,

$$\begin{aligned}\kappa_{max}(\hat{T}^\top \hat{T}) < 1 \quad &\Longleftrightarrow 3\varphi^2 + 2\psi^2 + 2\varphi\psi + \sqrt{(3\varphi^2 + 2\psi^2 + 2\varphi\psi)^2 - 4\varphi^4} < 2 \\ &\Longleftrightarrow \sqrt{(3\varphi^2 + 2\psi^2 + 2\varphi\psi)^2 - 4\varphi^4} < 2 - (3\varphi^2 + 2\psi^2 + 2\varphi\psi).\end{aligned}$$

Hence, a sufficient condition for the convergence is

$$\begin{cases} 3\varphi^2 + 2\psi^2 + 2\varphi\psi \in (0, 2), \\ 3\varphi^2 + 2\psi^2 + 2\varphi\psi \in (0, 1 + \varphi^4). \end{cases} \tag{20}$$

From (20), we have $\kappa_{max}(\hat{T}^\top \hat{T}) < 1$ provide (17), which completes the proof. □

Note that if the conditions of Theorem 2 are satisfied, then we obtain

$$0 \leq |||(e^y_{k+1}, e^x_{k+1})||| \leq \|\hat{T}\| \cdot |||(e^y_k, e^x_k)||| \leq \cdots \leq \|\hat{T}\|^{k+1} \cdot |||(e^y_0, e^x_0)|||.$$

Hence, $\lim_{k\to\infty} \|e^y_k\| = 0$ and $\lim_{k\to\infty} \|e^x_k\| = 0$. Therefore, the iteration sequence $\{x^k\}_{k=0}^{\infty}$ generated by (9) will convergent to the solution of the AVE (1).

In order to further study the existence of parameter ω for solving AVE (1), from the perspective of spectrum, we analyze the range and the optimal choice of parameter ω under the convergence condition of Algorithm 2. To determine the spectrum of iteration matrix, we consider the following eigenvalue problem

$$\lambda \begin{pmatrix} A & 0 \\ -\omega I & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} (1-\omega)A & \omega D(x) \\ 0 & (1-\omega)I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

where λ is an arbitrary eigenvalue of $T(\omega)$. This means that we can provide a good approximation for optimal choice of parameter ω with $D(x) \to D$. Then we focus on the following eigenvalue equation

$$\lambda \begin{pmatrix} A & 0 \\ -\omega I & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} (1-\omega)A & \omega D \\ 0 & (1-\omega)I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{21}$$

It is important to be able to find the optimal parameter ω (hereafter abbreviated as ω^*_{opt}) to minimize $\rho(T(\omega))$ for Algorithm 2; that is

$$\omega^*_{opt} = argmin\{\rho(T(\omega))\},$$

where

$$\rho(T(\omega)) = max|\lambda|.$$

To this end, we need the following auxiliary lemmas.

Lemma 3 ([50])**.** *Consider the quadratic equation $x^2 - bx + c = 0$, where b and c are real numbers. Both roots of the equation are less than one in modulus if and only if $|c| < 1$ and $|b| < 1 + c$.*

Lemma 4. *If $z^H z = 1$ and $\|z\| = 1$, there exists z_0 satisfying $z_0^H B z_0 = \|B\|$ for any matrix B.*

Proof. Due to $z^H B z = \sqrt{(z^H B z)^H (z^H B z)} = \sqrt{z^H B^H z z^H B z} = \sqrt{z^H B^H B z}$, then there exists z_0 satisfying $\sqrt{z_0^H B^H B z_0} = \max\limits_{\|z\|=1} \sqrt{z^H B^H B z} = \|B\|$. □

The following proof is inspired by [51]. Notice that $D^2 = I$ where D is a diagonal matrix. Without loss of generality, suppose that $z_1^H z_1 = 1$. From (21), it holds that

$$\lambda^2 z_1 = \{(\omega^2 A^{-1} D + (2 - 2\omega) I)\lambda - (\omega - 1)^2 I]\} z_1. \tag{22}$$

There exists a vector z_1 satisfying $z_1^H A^{-1} D z_1 = \|A^{-1} D\|$. Multiplying both sides of (22) by z_1^H from left and using Lemma 4, we obtain

$$\lambda^2 - (\omega^2 \mu + 2 - 2\omega)\lambda + (\omega - 1)^2 = 0, \tag{23}$$

where $\mu = \|A^{-1} D\|$. The roots of (23) are given by

$$\lambda = \frac{(\omega^2 \mu + 2 - 2\omega) \pm \sqrt{(\omega^2 \mu + 2 - 2\omega)^2 - 4(\omega - 1)^2}}{2}. \tag{24}$$

According to Lemma 3, we obtain a sufficient condition such that the two roots of (23) are both less than one, that is

$$\begin{cases} |(\omega - 1)^2| < 1, & (25) \\ F := |\omega^2 \mu - (2\omega - 2)| < 1 + (\omega - 1)^2 := G. & (26) \end{cases}$$

It is easy to check that (25) is equivalent to $\omega \in (0, 2)$. Equation (26) seems harder to be verified at first sight. Hence, we will proceed to discuss more about it. Notice

$$F < \omega^2 \nu + |2\omega - 2| := \hat{F} \text{ and } G = \omega^2 - 2\omega + 2,$$

a sufficient condition for (26) is $\hat{F} < G$ for $\omega \in (0, 2)$. Let $f_\nu(\omega) \doteq \hat{F} - G$, and then $f_\nu(\omega) < 0$ holds for $\omega \in (0, 1]$ when $\nu < 1$. For $\omega \in (1, 2)$, we have $f_\nu(\omega) \doteq (\nu - 1)\omega^2 + 4\omega - 4 < 0$. The roots of $f_\nu(\omega)$ are

$$\omega_1 = \frac{-2 - 2\sqrt{\nu}}{\nu - 1} \text{ and } \omega_2 = \frac{-2 + 2\sqrt{\nu}}{\nu - 1}.$$

Thus, we can obtain $1 < \omega_2 < 2 < \omega_1$ if $\nu < 1$, which leads to the solution set of $f_\nu(\omega) < 0$ being $\omega \in (1, \omega_2)$.

In conclusion, when $\nu \in (0, 1)$, if

$$\omega \in (0, \frac{2 - 2\sqrt{\nu}}{1 - \nu}) \doteq \Omega, \tag{27}$$

the roots of (23) are strictly lower than one in modulus.

Remark 1. *It is well-known that $\omega \in (0, 2)$ is the selection of parameter ω for the classical SOR iteration method and the SOR-like iteration method in [42], which is also the basic necessary convergent condition. Considering the relationship between the convergence conditions of the ASOR-like method from the two perspectives, it is easy to check that (25) is equivalent to $1 + \varphi^4 < 2$, which is a sufficient condition of (20). This also shows that the convergence condition from the spectral perspective based on [51] is tighter than those from the norm perspective based on [42,44].*

4. Optimal Parameter for the ASOR-like Iteration Method

In this section, we consider the choice of the iteration parameter ω. Let $\varrho(\omega) \doteq \omega^2\mu + 2 - 2\omega$, $\tau(\omega) \doteq (\omega-1)^2$. According to (24), we get

$$max|\lambda| = \begin{cases} \dfrac{\varrho(\omega) + \sqrt{\varrho^2(\omega) - 4\tau(\omega)}}{2}, \text{ if } \varrho(\omega) > 0, \\ \dfrac{|\varrho(\omega) - \sqrt{\varrho^2(\omega) - 4\tau(\omega)}|}{2}, \text{ if } \varrho(\omega) \leq 0, \end{cases}$$

from which we minimize $max|\lambda|$ to approximately obtain the following condition:

$$\begin{cases} \varrho^2(\omega) - 4\tau(\omega) = 0, \text{ if } \varrho(\omega) > 0, \\ \varrho(\omega) - \sqrt{\varrho^2(\omega) - 4\tau(\omega)} = 0, \text{ if } \varrho(\omega) \leq 0. \end{cases} \tag{28}$$

In fact, due to $\mu \leq \nu$, $\varrho(\omega) \leq 0$ can shrink to a sufficient condition $\hat{\varrho}(\omega) \doteq \omega^2\nu + 2 - 2\omega \leq 0$, which means $\omega \in [\frac{1-\sqrt{1-2\nu}}{\nu}, 2)$ for $\nu \in (0, \frac{1}{2})$ and ω is an empty set for $\nu \in [\frac{1}{2}, 1)$. However, from (28), we only need to prove $\tau(\omega) = 0$ when $\hat{\varrho}(\omega) \leq 0$ that obtains $\omega^*_{opt} = 1 < \frac{1-\sqrt{1-2\nu}}{\nu}$ for $\nu \in (0, \frac{1}{2})$ and $\hat{\varrho}(\omega^*_{opt}) = \nu > 0$. This is a contradictory inequality.

In addition, when $\varrho(\omega) > 0$, according to (28), we get

$$\varrho^2_{max}(\omega) - 4\tau(\omega) = 0 \Longleftrightarrow h_\nu(\omega) \doteq \nu^2\omega^4 - 4\nu\omega^3 + 4\omega^2 - 8\omega + 4 = 0, \tag{29}$$

which implies $\max|\lambda| = \frac{\varrho_{max}(\omega) + \sqrt{\varrho^2_{max}(\omega) - 4\tau(\omega)}}{2}$ with $\varrho_{max}(\omega) = \omega^2\nu + 2 - 2\omega > 0$ and $\mu \leq \nu$. For $\varrho_{max}(\omega) > 0$, after some simple algebraic operations, we get the existence of ω that $\omega \in (0, \frac{1-\sqrt{1-2\nu}}{\nu}) \in (0,2)$ for $\nu \in (0, \frac{1}{2})$ and $\omega \in (0,2)$ for $\nu \in [\frac{1}{2}, 2)$. The roots of $h_\nu(\omega)$ can be solved by the function **roots** in Matlab to get the theoretical optimal parameter ω^*_{opt}, expressed as

$$\begin{aligned} \omega_1(\nu) &= \frac{\sqrt{\nu} + 1 - \sqrt{2\sqrt{\nu} + 1 - \nu}}{\nu}, \omega_2(\nu) = \frac{\sqrt{\nu} + 1 + \sqrt{2\sqrt{\nu} + 1 - \nu}}{\nu}, \\ \omega_3(\nu) &= \frac{-\sqrt{\nu} + 1 + \sqrt{-2\sqrt{\nu} - 1 + \nu}}{\nu}, \omega_4(\nu) = \frac{-\sqrt{\nu} + 1 - \sqrt{-2\sqrt{\nu} - 1 + \nu}}{\nu}. \end{aligned} \tag{30}$$

In order to explore the characteristics of the roots of the quadratic Equation (29), we plot the contour for $h_\nu(\omega)$ and the $\omega_i(\nu)$ for $i = 1,2,3,4$ with $\nu \in (0,1)$ in Figure 1. In fact, $\omega_1(\nu)$ and $\omega_2(\nu)$ with $\nu \in (0,1)$ are both real values when $2\sqrt{\nu} > 0 > \nu - 1$, $\omega_3(\nu)$ and $\omega_4(\nu)$ with $\nu \in (0, 3-2\sqrt{2}) \approx (0, 0.172)$ are both real values when $2\sqrt{\nu} < 1 - \nu$. In this case, the complex roots are not considered. Therefore, it is obvious that $\nu_1 = 1 - \frac{\sqrt{3}}{2} \approx 0.134$ for $\omega^*_{opt} \in (0,2)$ and

$$\lim_{\nu \to 0^+} \omega_1(\nu) = 1, \lim_{\nu \to 0^+} \omega_4(\nu) = 1, \lim_{\nu \to \nu_1} \omega_4(\nu) = \omega_4(\nu_1) = 2, \lim_{\nu \to 1} \omega_1(\nu) = \omega_1(1) = 2 - \sqrt{2} \approx 0.5858.$$

However, due to $\omega_4(\nu) \notin (0, \frac{1-\sqrt{1-2\nu}}{\nu})$ in $\nu \in (0, \nu_1)$, from Figure 1, we know that

$$\omega^*_{opt} = \omega_1(\nu), \text{ if } \nu \in (0,1). \tag{31}$$

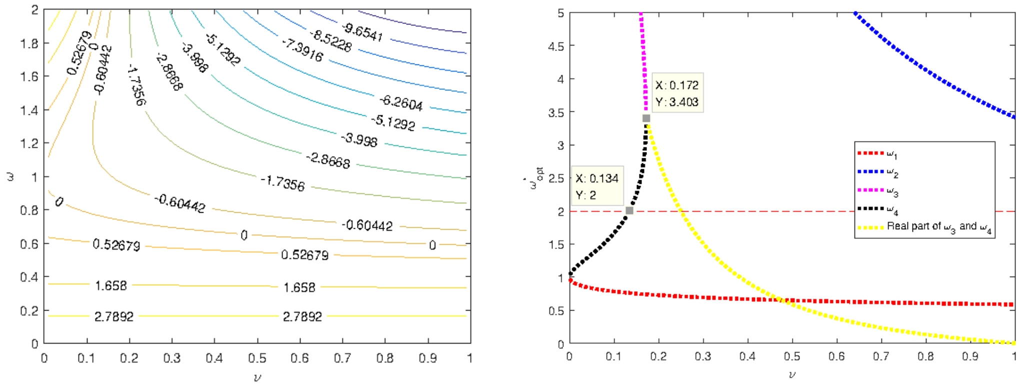

Figure 1. The contours of $h_\nu(\omega)$ with $\nu = [0.01 : 0.01 : 0.99]$ and $\omega = [0.01 : 0.01 : 1.99]$ (**left**) and the curve of $\omega_i(\nu)$ for $i = 1,2,3,4$ with $\nu = [0.001 : 0.001 : 0.999]$ (**right**).

Now, we devote our attention to investigating the approximate optimal parameter ω^*_{aopt}. Let $l_\nu(\omega) = max\{\varphi(\omega), \psi(\omega)\}$, and then we have

$$\hat{T}(\omega) \leq \begin{pmatrix} l_\nu(\omega) & l_\nu(\omega) \\ l_\nu(\omega) & 2l_\nu(\omega) \end{pmatrix} = l_\nu(\omega)\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \doteq l_\nu(\omega)H.$$

It follows that

$$\|\hat{T}(\omega)\|_2 \leq \|l_\nu(\omega)H\|_2 = l_\nu(\omega)\|H\|_2 = l_\nu(\omega)\frac{3+\sqrt{5}}{2}.$$

Let $\delta = \frac{2}{3+\sqrt{5}}$, and this $l_\nu(\omega)$ satisfies $\|\hat{T}(\omega)\|_2 \leq \frac{l_\nu(\omega)}{\delta}$, where $\frac{l_\nu(\omega)}{\delta}$ is an upper bound of $\|\hat{T}(\omega)\|$ with $\omega \in (0,2)$. This is the reason that we find ω^*_{aopt} in minimizing $l_\nu(\omega)$.

It is not difficult to find that $\varphi(\omega)$ is strictly monotonously decreasing for $\omega \in (0,1)$ and is strictly monotonously increasing for $\omega \in (1,2)$. In addition, $\psi(\omega)$ is strictly monotonously increasing in $\omega \in (0,2)$. By simply drawing and analyzing function $\varphi(\omega)$ and $\psi(\omega)$, we derive that

$$\omega^*_{aopt} = arg\ min\{l_\nu(\omega)\} = \frac{-1+\sqrt{1+4\nu}}{2\nu} > 0. \tag{32}$$

It notices that ω^*_{aopt} is obtained by $\varphi(\omega) = 1-\omega = \omega^2 - \nu = \psi(\omega)$ with $\omega \in (0,2)$ and $\nu \in (0,1)$.

Remark 2. *Consider the range of values of ω obtained by the above convergence conditions, according to (27), (30), (31), and (32), we plot the Figure 2. It is easy to see that the blue curve divides the green area into two parts; the top part is actually the condition of $\hat{\varrho}(\omega) \leq 0$, and the bottom part is actually the condition of $\varrho_{max}(\omega) > 0$. According to the condition of (27), when $\nu = \frac{1}{3}$, it holds $\frac{2-2\sqrt{\nu}}{1-\nu} = \frac{1-\sqrt{1-2\nu}}{\nu}$. Therefore, it leads to the new convergence conditions that if $\varrho_{max}(\omega) > 0$, $\omega \in (0, \frac{1-\sqrt{1-2\nu}}{\nu})$ for $\nu \in (0, \frac{1}{3})$ and $\omega \in (0, \frac{2-2\sqrt{\nu}}{1-\nu})$ for $\nu \in [\frac{1}{3}, 1)$; if $\hat{\varrho}(\omega) \leq 0$, $\omega \in (\frac{1-\sqrt{1-2\nu}}{\nu}, \frac{2-2\sqrt{\nu}}{1-\nu})$ for $\nu \in (0, \frac{1}{3})$. Furthermore, $\omega^*_{opt}, \omega^*_{aopt} \in \Omega$.*

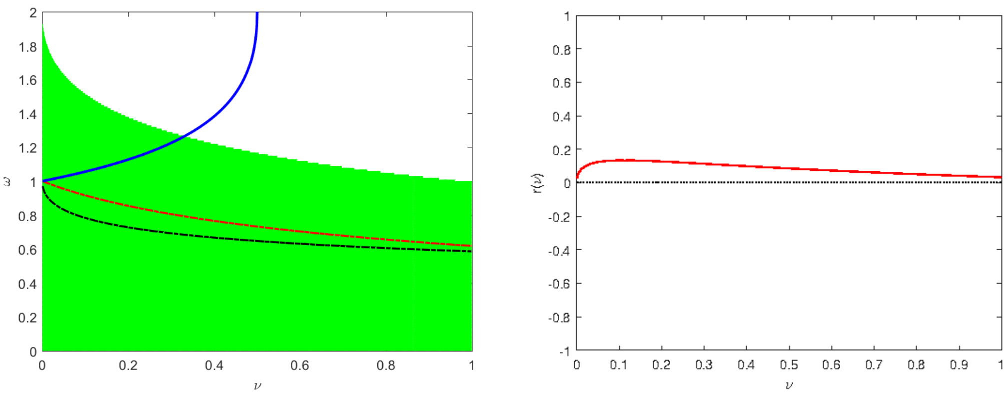

Figure 2. Left: The range and curves of the parameter $\omega \in (0, 2)$ with $\nu \in [0.001 : 0.001 : 0.999]$ (the black line : ω^*_{opt}; the red line : ω^*_{aopt}; the blue line : $\omega(\nu) = \frac{1-\sqrt{1-2\nu}}{\nu}$ for $\nu \in (0, \frac{1}{2})$; the green area: Ω); **Right**: the curve of $r(\nu)$ with $\nu = [0.001 : 0.001 : 0.999]$.

Comparing ω^*_{opt} and ω^*_{aopt}, we have

$$\lim_{\nu\to 1} \omega^*_{aopt}(\nu) = \frac{\sqrt{5}-1}{2} \approx 0.618, \lim_{\nu\to 1} \omega^*_{opt}(\nu) = 2 - \sqrt{2} \approx 0.586.$$

The right of Figure 2 illustrates the gap of the ω^*_{opt} and ω^*_{aopt} where $r(\nu) = \omega^*_{aopt} - \omega^*_{opt}$.

5. Numerical Results

In this section, we will present three numerical examples to compare the ASOR-like iteration method with the previous algorithms to illustrate the feasibility and effectiveness of the ASOR-like iteration method. The following six algorithms will be tested.

1. SOR-like-exp method [42]: namely, the iteration format is (3). We choose the experimental optimal parameter ω^*_{exp} with the smallest iteration step of the corresponding method in $\omega = [0.001 : 0.001 : 1.999]$ (in Example 1) and $\omega = [0.01 : 0.01 : 1.99]$ (in Example 2 and Example 3).
2. ASOR-like-exp method: its iteration format is (9). The optimal parameter selection of the ASOR-like-exp method is consistent with the SOR-like-exp method.
3. SOR-like-opt method [44]: its iteration format is consistent with the SOR-like-exp method where the theoretical optimal parameter ω^*_{opt} follows (4). ω^*_{opt} can be calculated by the classical bisection method with the termination criterion is $|g^1_\nu(\omega)| \leq 10^{-10}$ or the updated ends of the interval $b_2 - b_1 \leq 10^{-10}$, see [44] for specific operations.
4. ASOR-like-opt method: its iteration format is consistent with the ASOR-like-exp method, and ω^*_{opt} is calculated in accordance with (31).
5. SOR-like-aopt method [44]: its iteration format is consistent with the SOR-like-exp method where the approximate optimal parameter ω^*_{aopt} follows (5).
6. ASOR-like-aopt method: its iteration format is consistent with the ASOR-like-exp method and ω^*_{aopt} is calculated in accordance with (32).

The numerical experiments are explained in several aspects in the following. On the one hand, the choice of parameters ω are particularly important, which greatly affects the CPU time of numerical experiments. On other hand, in order to facilitate the comparison of algorithms, we select the following three experiments that satisfy the unique solution property of the AVE (1) for comparison.

All test problems are conducted under MATLAB R2016a on a personal computer with 1.19 GHz central processing unit (Intel(R) Core(TM) i5-1035U), 8.00 GB memory and Windows 10 operating system. The description of each method includes the number of iteration steps (denoted by "IT"), the CPU time (denoted by "CPU") and residual relative error (denoted by "RES"). The stopping criterion of iteration is

$$RES(x^k) \doteq \|Ax^k - |x^k| - b\|_2 < 10^{-5}$$

or the prescribed maximal iteration number $k_{max} = 1000$ is exceeded ("–" is used in the following tables to illustrate this case). All tests are started from the initial zero vector.

Example 1. *Considering the random AVE* (1) *with* $\|A^{-1}\| < 1$ *in [16,44], the influence of the condition number and the density of A (abbreviation for* $cond(A)$ *and* $density(A)$*) on the tests will be discussed during the numerical implements.*

Let $min(cond(A))$ be 1, 10, or 10^2, respectively, and the results are used to analyze the superiority of the ASOR-like method in different optimal parameter ω. Let $x^* = -100 + 200 \times rand(n, 1)$ and $b = Ax^* - |x^*|$ is generated. For Example 1, the information (the order n, the approximate density of A (abbreviation for $a.density(A)$), $cond(A)$ and $\|A^{-1}\|$) of random AVE problems under specific conditions obtained by numerical experiments are shown in Tables 1–3.

Table 1. Numerical results for Example 1 with $min(cond(A)) = 1$.

Method	n	**256**	**512**	**1024**	**2048**	**4096**
	$a.density(A)$	**0.003**	**0.003**	**0.0003**	**0.00003**	**0.000003**
	$density(A)$	**0.0039**	**0.0029**	$\mathbf{9.7656 \times 10^{-4}}$	$\mathbf{4.8828 \times 10^{-4}}$	$\mathbf{2.4414 \times 10^{-4}}$
	$cond(A)$	**2.5059**	**2.8172**	**3.5639**	**1.5041**	**2.5778**
	$\|A^{-1}\|$	**0.4024**	**0.9875**	**0.7948**	**0.6119**	**0.6153**
SOR-like-exp	ω^*_{exp}	0.972	0.977	0.926	0.973	0.995
	IT	15	34	26	29	25
	CPU	18.5880	230.0134	100.1782	221.7795	457.9501
	RES	9.8549×10^{-6}	9.7743×10^{-6}	9.5922×10^{-6}	9.7838×10^{-6}	9.7806×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.973	0.977	0.927	0.973	0.995
	IT	15	34	26	29	25
	CPU	22.1293	226.7375	97.8316	211.2889	464.9240
	RES	9.8864×10^{-6}	9.8190×10^{-6}	9.9883×10^{-6}	9.9058×10^{-6}	9.8234×10^{-6}
SOR-like-opt	ω^*_{opt}	0.924	0.623	0.705	0.8	0.798
	IT	18	78	46	44	41
	CPU	0.0156	0.0249	0.0135	0.0185	0.0323
	RES	5.3732×10^{-6}	8.1300×10^{-6}	6.6885×10^{-6}	8.1115×10^{-6}	8.4195×10^{-6}
ASOR-like-opt	ω^*_{opt}	0.667	0.587	0.606	0.629	0.629
	IT	36	86	59	67	62
	CPU	0.0067	0.0227	0.0071	0.0164	0.0308
	RES	6.7158×10^{-6}	8.4438×10^{-6}	8.4476×10^{-6}	7.8581×10^{-6}	8.6371×10^{-6}
SOR-like-aopt	ω^*_{aopt}	0.765	0.620	0.657	0.7	0.699
	IT	27	78	51	56	52
	CPU	0.0066	0.0200	0.0082	0.0137	0.0252
	RES	9.4714×10^{-6}	9.1027×10^{-6}	8.9346×10^{-6}	7.8581×10^{-6}	7.8337×10^{-6}
ASOR-like-aopt	ω^*_{aopt}	0.765	0.620	0.657	0.7	0.699
	IT	28	79	52	56	52
	CPU	0.0045	0.0188	0.0070	0.0151	0.0248
	RES	6.8235×10^{-6}	8.7466×10^{-6}	8.0426×10^{-6}	8.9334×10^{-6}	9.2929×10^{-6}

Table 2. Numerical results for Example 1 with $min(cond(A)) = 10$.

Method	n	256	512	1024	2048	4096
	$a.density(A)$	**0.003**	**0.003**	**0.0003**	**0.00003**	**0.000003**
	$density(A)$	**0.0039**	**0.0029**	$\mathbf{9.7656 \times 10^{-4}}$	$\mathbf{4.8828 \times 10^{-4}}$	$\mathbf{2.4414 \times 10^{-4}}$
	$cond(A)$	**14.7244**	**16.3457**	**35.1532**	**19.9552**	**43.1216**
	$\|\|A^{-1}\|\|$	**0.5628**	**0.8446**	**0.7157**	**0.5137**	**0.7003**
SOR-like-exp	ω^*_{exp}	0.974	0.979	0.983	0.986	0.993
	IT	11	13	10	10	9
	CPU	2.0967	16.7058	6.5141	15.7872	34.3956
	RES	9.5213×10^{-6}	9.7642×10^{-6}	9.4063×10^{-6}	9.1467×10^{-6}	9.3907×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.975	0.98	0.984	0.987	0.994
	IT	11	13	10	10	9
	CPU	2.0719	16.9688	6.5851	15.9789	34.7201
	RES	9.9142×10^{-6}	9.6326×10^{-6}	9.3416×10^{-6}	8.8497×10^{-6}	8.4067×10^{-6}
SOR-like-opt	ω^*_{opt}	0.829	0.682	0.744	0.858	0.752
	IT	19	30	23	17	23
	CPU	0.0106	0.0182	0.0129	0.0129	0.0203
	RES	3.7751×10^{-6}	5.6117×10^{-6}	6.0539×10^{-6}	5.1048×10^{-6}	4.6180×10^{-6}
ASOR-like-opt	ω^*_{opt}	0.636	0.6	0.615	0.645	0.617
	IT	32	37	33	31	34
	CPU	0.0053	0.0108	0.0061	0.0093	0.0182
	RES	7.0505×10^{-6}	9.4759×10^{-6}	8.9738×10^{-6}	8.0716×10^{-6}	5.3617×10^{-6}
SOR-like-aopt	ω^*_{aopt}	0.713	0.647	0.674	0.728	0.678
	IT	25	32	27	24	28
	CPU	0.0036	0.0125	0.0069	0.0085	0.0173
	RES	8.5072×10^{-6}	8.8675×10^{-6}	9.9723×10^{-6}	8.0679×10^{-6}	4.6255×10^{-6}
ASOR-like-aopt	ω^*_{aopt}	0.713	0.647	0.674	0.728	0.678
	IT	26	33	29	25	29
	CPU	0.0032	0.0119	0.0065	0.0080	0.0163
	RES	7.6108×10^{-6}	8.3079×10^{-6}	4.7287×10^{-6}	6.9463×10^{-6}	4.9004×10^{-6}

Table 3. Numerical results for Example 1 with $min(cond(A)) = 100$.

Method	n	256	512	1024	2048	4096
	$a.density(A)$	**0.003**	**0.003**	**0.0003**	**0.00003**	**0.000003**
	$density(A)$	**0.0039**	**0.0029**	$\mathbf{9.7656 \times 10^{-4}}$	$\mathbf{4.8828 \times 10^{-4}}$	$\mathbf{2.4414 \times 10^{-4}}$
	$cond(A)$	**120.3861**	**307.0414**	**153.6908**	**109.2455**	**200.7276**
	$\|\|A^{-1}\|\|$	**0.3826**	**0.6618**	**0.3243**	**0.9690**	**0.9450**
SOR-like-exp	ω^*_{exp}	0.994	0.995	0.996	0.998	1
	IT	6	6	6	8	7
	CPU	1.6663	14.5714	5.9188	15.1130	33.1430
	RES	7.8386×10^{-6}	8.9158×10^{-6}	7.4674×10^{-6}	7.6907×10^{-6}	5.8261×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.995	0.996	0.996	0.998	1
	IT	6	6	6	10	7
	CPU	1.7731	14.7743	6.2905	15.1825	33.6529
	RES	6.6006×10^{-6}	7.1726×10^{-6}	9.1736×10^{-6}	8.1919×10^{-6}	5.8261×10^{-6}

Table 3. *Cont.*

Method	n	256	512	1024	2048	4096
	$a.density(A)$	**0.003**	**0.003**	**0.0003**	**0.00003**	**0.000003**
	$density(A)$	**0.0039**	**0.0029**	$\mathbf{9.7656 \times 10^{-4}}$	$\mathbf{4.8828 \times 10^{-4}}$	$\mathbf{2.4414 \times 10^{-4}}$
	$cond(A)$	**120.3861**	**307.0414**	**153.6908**	**109.2455**	**200.7276**
	$\lVert A^{-1} \rVert$	**0.3826**	**0.6618**	**0.3243**	**0.9690**	**0.9450**
SOR-like-opt	ω^*_{opt}	0.936	0.773	0.967	0.630	0.639
	IT	10	18	9	29	28
	CPU	0.0096	0.0149	0.0151	0.0165	0.0255
	RES	3.5974×10^{-6}	3.5650×10^{-6}	9.8522×10^{-7}	9.2673×10^{-6}	8.4378×10^{-6}
ASOR-like-opt	ω^*_{opt}	0.671	0.622	0.686	0.588	0.591
	IT	25	29	25	34	34
	CPU	0.0035	0.0103	0.0057	0.0104	0.0190
	RES	4.7060×10^{-6}	4.9047×10^{-6}	4.0254×10^{-6}	8.9792×10^{-6}	6.3144×10^{-6}
SOR-like-aopt	ω^*_{aopt}	0.772	0.687	0.795	0.623	0.628
	IT	18	22	17	30	29
	CPU	0.0033	0.0096	0.0050	0.0110	0.0193
	RES	3.0545×10^{-6}	8.0103×10^{-6}	5.4605×10^{-6}	6.4254×10^{-6}	7.6955×10^{-6}
ASOR-like-aopt	ω^*_{aopt}	0.772	0.687	0.795	0.623	0.628
	IT	19	24	18	31	31
	CPU	0.0031	0.0081	0.0049	0.0103	0.0150
	RES	3.5584×10^{-6}	6.2406×10^{-6}	5.6879×10^{-6}	8.5925×10^{-6}	5.2066×10^{-6}

From the numerical results displayed in Tables 1–3, we find that the "CPU" of the ASOR-like-opt iteration method and the ASOR-like-aopt iteration method are less than the SOR-like-opt iteration method and the SOR-like-aopt iteration method in general, but the ASOR-like-opt iteration method compared to the SOR-like-opt iteration method requires much iteration steps, the two methods for selecting the approximate optimal parameter ω^*_{aopt} or the experimental optimal parameter ω^*_{exp} basically keep the same iteration steps. In brief, the ASOR-like iteration method is superior to the SOR-like iteration method under choosing appropriate optimal parameter.

Example 2 ([24])**.** *Consider the two-dimensional convection diffusion equation*

$$\begin{aligned} -(u_{xx}+u_{yy})+q(u_x+u_y)+pu=f(x,y),\ (x,y)\in \Upsilon,\\ u(x,y)=0,\ (x,y)\in\partial\Upsilon, \end{aligned}$$

where q is a nonnegative constant, p is a real number, $\Upsilon=(0,1)\times(0,1)$, and $\partial\Upsilon$ is its boundary. By using the five-point finite difference scheme and the central difference scheme to the diffusive terms and the convective terms, respectively. The equidistant step $h=\frac{1}{m+1}$ and the mesh Reynolds number $r=\frac{qh}{2}$ are denoted. Then we acquire the system of linear equations $Rx=d$, where the matrix of $R=T_x\otimes I_m+I_m\otimes T_y+pI_n\in\mathbb{R}^{m^2\times m^2}$, $I_m\in\mathbb{R}^{m\times m}$ and $I_n\in\mathbb{R}^{n\times n}$ are two identity matrices, $\otimes$ means the Kronecker product symbol, $T_x=tridiag(t_1,t_2,t_3)$ and $T_y=tridiag(t_1,0,t_3)$ are the tridiagonal matrices with $t_1=-1-r$, $t_2=4$, $t_3=-1+r$. For our numerical experiments, we define the matrix A in AVE (1) by making use of the matrix R as follows.

For any positive number p and q, the matrix R is nonsymmetric positive definite. When $q=0$, the matrix R provided is symmetric positive definite. We set $A=R+5(L-L^\top)$, where L is the strictly lower part of R. It is not hard to find that the matrix A is nonsymmetric positive definite. Let $x^*_i=(-1)^i i$, $i=1,2,\cdots$, and $b=Ax^*-|x^*|$ is generated. We present the numerical results for different values of m, p, q in Tables 4 and 5.

Table 4. Numerical results for Example 2 with $m = 10$ and $p = 0$.

Method	q	**0**	**1**	**10**	**100**	**1000**
	$\|\|A^{-1}\|\|$	**0.6836**	**0.6568**	**0.4955**	**0.2682**	**0.2502**
SOR-like-exp	ω^*_{exp}	0.99	0.99	0.99	0.99	1
	IT	14	14	13	10	7
	CPU	17.3447	16.7893	15.9677	13.5858	10.6116
	RES	6.1270×10^{-6}	5.2623×10^{-6}	5.4657×10^{-6}	4.4779×10^{-6}	1.8263×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.99	0.99	0.99	0.99	1
	IT	14	14	13	10	7
	CPU	16.5011	16.5607	15.8689	13.2042	10.3419
	RES	6.3024×10^{-6}	5.4242×10^{-6}	5.6905×10^{-6}	4.7519×10^{-6}	1.8263×10^{-6}
SOR-like-opt	ω^*_{opt}	0.761	0.775	0.869	0.993	1
	IT	27	26	19	10	7
	CPU	0.0207	0.0148	0.0118	0.0140	0.0103
	RES	5.4247×10^{-6}	5.3442×10^{-6}	9.1631×10^{-7}	3.2788×10^{-6}	1.9915×10^{-6}
ASOR-like-opt	ω^*_{opt}	0.619	0.623	0.648	0.702	0.708
	IT	39	38	35	26	23
	CPU	0.0095	0.0106	0.0086	0.0098	0.0078
	RES	6.4887×10^{-6}	8.2543×10^{-6}	6.6371×10^{-6}	4.9962×10^{-6}	7.0446×10^{-6}
SOR-like-aopt	ω^*_{aopt}	0.682	0.689	0.733	0.820	0.828
	IT	32	32	27	18	16
	CPU	0.0091	0.0095	0.0090	0.0068	0.0057
	RES	8.4176×10^{-6}	6.1208×10^{-6}	9.5221×10^{-6}	7.4910×10^{-6}	3.6549×10^{-6}
ASOR-like-aopt	ω^*_{aopt}	0.682	0.689	0.733	0.820	0.828
	IT	33	32	28	19	16
	CPU	0.0085	0.0093	0.0080	0.0062	0.0056
	RES	7.2919×10^{-6}	8.8531×10^{-6}	7.2922×10^{-6}	4.2199×10^{-6}	9.4326×10^{-6}

Table 5. Numerical results for Example 2 with $m = 10$ and $p = 1$.

Method	q	**0**	**1**	**10**
	$\|\|A^{-1}\|\|$	**0.4535**	**0.4419**	**0.3633**
SOR-like-exp	ω^*_{exp}	0.99	0.99	0.99
	IT	12	12	12
	CPU	15.6642	15.6092	15.2944
	RES	8.1549×10^{-6}	8.2374×10^{-6}	5.4467×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.99	0.99	0.99
	IT	12	12	12
	CPU	15.9589	15.4252	15.2944
	RES	8.3692×10^{-6}	8.4516×10^{-6}	5.6546×10^{-6}
SOR-like-opt	ω^*_{opt}	0.894	0.901	0.947
	IT	17	17	14
	CPU	0.0157	0.0118	0.0138
	RES	6.8516×10^{-6}	3.9006×10^{-6}	6.5650×10^{-7}
ASOR-like-opt	ω^*_{opt}	0.656	0.658	0.676
	IT	33	33	30
	CPU	0.0091	0.0089	0.0081
	RES	9.4174×10^{-6}	7.1217×10^{-6}	8.6655×10^{-6}

Table 5. *Cont.*

Method	q	**0**	**1**	**10**
	$\|A^{-1}\|$	**0.4535**	**0.4419**	**0.3633**
SOR-like-aopt	ω^*_{aopt}	0.747	0.751	0.779
	IT	26	25	23
	CPU	0.0079	0.0079	0.0068
	RES	5.3883×10^{-6}	8.4835×10^{-6}	5.0107×10^{-6}
ASOR-like-aopt	ω^*_{aopt}	0.747	0.751	0.779
	IT	26	26	23
	CPU	0.0075	0.0074	0.0066
	RES	9.2671×10^{-6}	6.3792×10^{-6}	7.6512×10^{-6}

From Tables 4 and 5 we can see that all iteration methods can successfully produce an approximately unique solution to the AVE (1) for selecting appropriate problem scales $n = m^2$ and the convective measurements q ($q = 0, 1, 10, 100, 1000$ when $p = 0$ and $m = 10$; $q = 0, 1, 10$ when $p = 1$ and $m = 10$). In the case where it converges to the unique solution of AVE (1), the ASOR-like-opt iteration method and the ASOR-like-aopt iteration method are superior to the SOR-like-opt iteration method and the SOR-like-aopt iteration method in "CPU", respectively, and the numerical results with theoretical optimal parameters are much better than the numerical results with experimental optimal parameters.

Example 3. *Consider the AVE (1), where the sparse, symmetry matrix A with $\|A^{-1}\| < 1$ comes from five different test problems in [42]. Let $x^* = (-1, 1, -1, 1, \cdots, -1, 1, \cdots)$ and $b = Ax^* - |x^*|$ is generated.*

From Table 6, we present the numerical results on the ASOR-like iteration method incorporated with the SOR-like iteration method, corresponding to these optimal parameters. Obviously, all iteration methods can compute an approximate solution of the problem in [42]. In particular, the ASOR-like-opt iteration method outperforms the SOR-like-opt iteration method for all small-scale full data matrix problems.

Table 6. Numerical results for Example 3.

Method	*problem*	*mesh1e1*	*mesh1em1*	*mesh2e1*	*Trefethen_20b*	*Trefethen_200b*
	$\|A^{-1}\|$	**0.5747**	**0.6397**	**0.7615**	**0.4244**	**0.4265**
SOR-like-exp	ω^*_{exp}	0.94	0.93	0.94	0.95	0.95
	IT	15	15	17	10	10
	CPU	2.8584	2.9256	26.1986	1.5056	47.5188
	RES	9.2893×10^{-6}	9.8022×10^{-6}	8.0935×10^{-6}	7.8939×10^{-6}	7.9623×10^{-6}
ASOR-like-exp	ω^*_{exp}	0.95	0.91	0.94	0.95	0.95
	IT	15	16	17	10	10
	CPU	2.8557	2.8761	27.4489	1.6236	46.2290
	RES	7.0764×10^{-6}	9.3454×10^{-6}	8.6726×10^{-6}	9.0674×10^{-6}	9.1348×10^{-6}
SOR-like-opt	ω^*_{opt}	0.822	0.785	0.721	0.911	0.91
	IT	21	22	29	12	12
	CPU	0.0080	0.0112	0.0188	0.0103	0.0191
	RES	6.6983×10^{-6}	8.4588×10^{-6}	9.1045×10^{-7}	3.8820×10^{-6}	4.1161×10^{-6}
ASOR-like-opt	ω^*_{opt}	0.635	0.625	0.61	0.662	0.661
	IT	33	33	39	23	23
	CPU	0.0032	0.0033	0.0125	0.0028	0.0148
	RES	9.8908×10^{-6}	9.8762×10^{-6}	8.6726×10^{-6}	7.3721×10^{-6}	7.8951×10^{-6}

6. Conclusions

The ASOR-like iteration method is developed to solve the AVE (1) by reformulating equivalently the AVE (1) as an alternative two-by-two block nonlinear system. The convergence results of the ASOR-like iteration method are proven under proper conditions imposed on the involved parameter. The optimal parameter and the approximate optimal parameter are explored. Numerical results are presented to demonstrate that the ASOR-like iteration method with the optimal parameter is feasible and effective in the case of small-scale problems. However, for large-scale problems, designing an efficient algorithm is still to be further studied. In addition, the choice of the optimal iteration parameter in theory is also worth considering from different perspectives.

Author Contributions: Conceptualization, methodology, software, original draft preparation, Y.Z.; guidance, review and revision, D.Y.; translation, editing and review, Y.Z., D.Y. and Y.Y.; validation, Y.Z., D.Y. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No.12201275) and the China Postdoctoral Science Foundation (No.2021M701537).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Dr. Cairong Chen from Fujian Normal University for his helpful discussions and guidance. The authors would like to thank the four anonymous reviewers and the editor for their helpful comments and suggestions that have greatly improved the paper.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Rohn, J. A theorem of the alternatives for the equation $Ax + B|x| = b$. *Linear Multilinear Algebra* **2004**, *52*, 421–426. [CrossRef]
2. Mangasarian, O.L. Absolute value programming. *Comput. Optim. Appl.* **2007**, *36*, 43–53. [CrossRef]
3. Mangasarian, O.L. Absolute value equation solution via concave minimization. *Optim. Lett.* **2019**, *1*, 3–8. [CrossRef]
4. Mangasarian, O.L. A generalized Newton method for absolute value equations. *Optim. Lett.* **2009**, *3*, 101–108. [CrossRef]
5. Bai, Z.-Z. Modulus–based matrix splitting iteration methods for linear complementarity problems. *Numer. Linear Algebra Appl.* **2010**, *17*, 917–933. [CrossRef]
6. Cottle, R.W.; Pang, J.-S.; Stone, R.E. *The Linear Complementarity Problem*; Academic Press: New York, NY, USA, 1992.
7. Hu, S.-L.; Huang, Z.-H.; Zhang, Q. A generalized Newton method for absolute value equations associated with second order cones. *J. Comput. Appl. Math.* **2019**, *39*, 1490–1501. [CrossRef]
8. Miao, X.-H.; Yang, J.-T.; Saheya, B.; Chen, J.-S. A smoothing Newton method for absolute value equation associated with second-order cone. *Appl. Numer. Math.* **2017**, *120*, 82–96. [CrossRef]
9. Prokopyev, O. On equivalent reformulations for absolute value equations. *Comput. Optim. Appl.* **2009**, *44*, 363–372. [CrossRef]
10. Mezzadri, F. On the solution of general absolute value equations. *Appl. Math. Lett.* **2020**, *107*, 106462. [CrossRef]
11. Mangasarian, O.L.; Meyer, R.R. Absolute value equations. *Linear Algebra Appl.* **2006**, *419*, 359–367. [CrossRef]
12. Rohn, J. On unique solvability of the absolute value equation. *Optim. Lett.* **2009**, *3*, 603–606. [CrossRef]
13. Wu, S.-L.; Li, C.-X. A note on unique solvability of the absolute value equation. *Optim. Lett.* **2020**, *14*, 1957–1960. [CrossRef]
14. Wu, S.-L.; Shen, S.-Q. On the unique solution of the generalized absolute value equation. *Optim. Lett.* **2021**, *15*, 2017–2024. [CrossRef]
15. Cao, Y.; Shi, Q.; Zhu, S.-L. A relaxed generalized Newton iteration method for generalized absolute value equations. *AIMS Math.* **2021**, *6*, 1258–1275. [CrossRef]
16. Cruz, J.Y.B.; Ferreira, O.P.; Prudente, L.F. On the global convergence of the inexact semi-smooth Newton method for absolute value equation. *Comput. Optim. Appl.* **2016**, *65*, 93–108. [CrossRef]
17. Feng, J.; Liu, S. A new two-step iterative method for solving absolute value equations. *J. Inequal. Appl.* **2019**, *39*, 1–8. [CrossRef]
18. Huang, B.-H.; Ma, C.-F. Convergent conditions of the generalized Newton method for absolute value equation over second order cones. *Appl. Math. Lett.* **2019**, *92*, 151–157. [CrossRef]
19. Lian, Y.-Y.; Li, C.-X.; Wu, S.-L. Stronger convergent results of the generalized Newton method for the generalized absolute value equations. *J. Comput. Appl. Math.* **2018**, *2018*, 221–226. [CrossRef]
20. Li, C.-X. A modified generalized newton method for absolute value equations. *J. Optim. Theory Appl.* **2016**, *170*, 1055–1059. [CrossRef]
21. Shi, L.; Iqbal, J.; Arif, M.; Khan, A. A two-step Newton-type method for solving system of absolute value equations. *Math. Probl. Eng.* **2020**, *2020*, 2798080. [CrossRef]

22. Wang, A.; Cao, Y.; Chen, J.-X. Modified Newton-type iteration methods for generalized absolute value equations. *J. Optim. Theory Appl.* **2019**, *2020*, 216–230. [CrossRef]
23. Wu, S.-L.; Li, C.-X.; Zhou, H.-Y. Newton-based matrix splitting method for generalized absolute value equation. *J. Comput. Appl. Math.* **2021**, *394*, 113578.
24. Salkuyeh, D.K. The Picard–HSS iteration method for absolute value equations. *Optim. Lett.* **2014**, *8*, 2191–2202. [CrossRef]
25. Li, C.-X. A preconditioned AOR iterative method for the absolute value equations. *Int. J. Comput. Methods* **2017**, *14*, 1750016. [CrossRef]
26. Edalatpour, V.; Hezari, D.; Salkuyeh, D.K. A generalization of the Gauss–Seidel iteration method for solving absolute value equations. *Appl. Math. Comput.* **2017**, *293*, 156–167. [CrossRef]
27. Huang, B.-H.; Ma, C.-F. The Modulus-based Levenberg-Marquardt method for solving linear complementarity problem. *Numer. Math. Theor. Meth. Appl.* **2019**, *12*, 154–168.
28. Iqbal, J.; Iqbal, A.; Arif, M. Levenberg-Marquardt method for solving systems of absolute value equations. *J. Comput. Appl. Math.* **2015**, *282*, 134–138. [CrossRef]
29. Chen, C.-R.; Yu, D.-M.; Han, D.-R. Exact and inexact Douglas-Rachford splitting methods for solving large-scale sparse absolute value equations. *IMA J. Numr. Anal.* **2022**, *online first*. [CrossRef]
30. Chen, C.-R.; Yu, D.-M.; Han, D.-R. An inverse-free dynamical system for solving the absolute value equations. *Appl. Numer. Math.* **2021**, *168*, 170–181. [CrossRef]
31. Huang, X.-J.; Cui, B.-T. Neural network-based method for solving absolute value equations. *ICIC Express Lett.* **2017**, *11*, 853–861.
32. Mansoori, A.; Eshaghnezhad, M.; Effati, S. An efficient neural network model for solving the absolute value equations. *IEEE Trangs. Circuits Syst. II* **2017**, *65*, 391–395. [CrossRef]
33. Mansoori, A.; Erfanian, M. A dynamic model to solve the absolute value equations. *J. Comput. Appl. Math.* **2019**, *333*, 28–35. [CrossRef]
34. Saheya, B.; Nguyen, C.-T.; Chen, J.-S. Neural network based on systematically generated smoothing functions for absolute value equation. *J. Appl. Math. Comput.* **2019**, *61*, 533–558. [CrossRef]
35. Yu, D.-M.; Chen, C.-R.; Yang, Y.-N.; Han, D.-R. An inertial inverse-free dynamical system for solving absolute value equations. *J. Ind. Manag. Optim.* **2023**, *19*, 2549–2559. [CrossRef]
36. Yu, Z.-S.; Li, L.; Yuan, Y. A modified multivariate spectral gradient algorithm for solving absolute value equations. *Appl. Math. Lett.* **2021**, *121*, 107461. [CrossRef]
37. Li, Y.; Du, S. Modified HS conjugate gradient method for solving generalized absolute value equations. *J. Inequal. Appl.* **2019**, *68*, 68. [CrossRef]
38. Rohn, J.; Hooshyarbakhsh, V.; Farhadsefat, R. An iterative method for solving absolute value equations and sufficient conditions for unique solvability. *Optim. Lett.* **2014**, *8*, 35–44. [CrossRef]
39. Saheya, B.; Yu, C.-H.; Chen, J.-S. Numerical comparisons based on four smoothing functions for absolute value equation. *J. Appl. Math. Comput.* **2018**, *56*, 131–149. [CrossRef]
40. Zamani, M.; Hladík, M. A new concave minimization algorithm for the absolute value equation solution. *Optim. Lett.* **2021**, *15*, 2241–2254. [CrossRef]
41. Zamani, M.; Hladík, M. Error bounds and a condition number for the absolute value equations. *Math. Program.* **2022**, 1–29. [CrossRef]
42. Ke, Y.-F.; Ma, C.-F. SOR-like iteration method for solving absolute value equations. *Appl. Math. Comput.* **2017**, *311*, 195–202. [CrossRef]
43. Guo, P.; Wu, S.-L.; Li, C.-X. On the SOR-like iteration method for solving absolute value equations. *Appl. Math. Lett.* **2019**, *97*, 107–113. [CrossRef]
44. Chen, C.-R.; Yu, D.-M.; Han, D.-R. Optimal parameter for the SOR-like iteration method for solving the system of absolute value equations. *arXiv* **2020**, arXiv:2001.05781.
45. Ke, Y.-F. The new iteration algorithm for absolute value equation. *Appl. Math. Lett.* **2020**, *99*, 105990. [CrossRef]
46. Yu, D.-M.; Chen, C.-R.; Han, D.-R. A modified fixed point iteration method for solving the system of absolute value equations. *Optimization* **2022**, *71*, 449–461. [CrossRef]
47. Dong, X.; Shao, X.-H.; Shen, H.-L. A new SOR-like method for solving absolute value equations. *Appl. Numer. Math.* **2020**, *156*, 410–421. [CrossRef]
48. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 1990.
49. Berman, A.; Plemmons, R. *Nonnegative Matrices in the Mathematical Sciences*; Academic Press: New York, NY, USA, 1979.
50. Young, D.M. *Iterative Solution for Large Linear Systems*; Academic Press: New York, NY, USA, 1971.
51. Shams, N.N.; Jahromi, A.F.; Beik, F.P.A. Iterative schemes induced by block splittings for solving absolute value equations. *Filomat* **2020**, *34*, 4171–4188. [CrossRef]