



Research article

A generalization of the AOR iteration method for solving absolute value equations

Cui-Xia Li*

School of Mathematics, Yunnan Normal University, Kunming 650500, China

* **Correspondence:** Email: lixiatk@126.com; Tel: +8618211679797.

Abstract: In this paper, based on the accelerated over relaxation (AOR) iteration method, a generalization of the AOR iteration method is presented to solve the absolute value equations (AVE), which is called the GAOR method. The convergence conditions of the GAOR method are obtained. Numerical experiments are presented in order to verify the feasibility of the GAOR method.

Keywords: absolute value equation; AOR method; convergence

1. Introduction

In this paper, based on the AOR iteration method, we consider the numerical solution of the absolute value equations (AVE)

$$Ax - |x| = b, \quad (1.1)$$

where $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ and $|x|$ denotes all the components of the vector $x \in \mathbb{R}^n$ by absolute value. If ‘ $|x|$ ’ is replaced by ‘ $B|x|$ ’ in (1.1), then the general AVE is obtained, see [1, 2]. Nowadays, many optimization problems including linear programming, convex quadratic programming and linear complementarity problem [3–5] can be formulated as the AVE (1.1) such that the AVE (1.1) has been attracted considerable attention.

To efficiently solve the AVE (1.1), in recent years, a great deal of effort has been committed to develop iteration methods. For example, in [6], Mangasarian presented a generalized Newton method for solving the AVE (1.1) and simply described below

$$x^{(k+1)} = (A - D(x^{(k)}))^{-1}b, k = 0, 1, \dots, \quad (1.2)$$

where $D(x^{(k)})$ is a diagonal matrix of the form $D(x^{(k)}) = \text{diag}(\text{sign}(x^{(k)}))$. One can see [7–10] and find other versions of the generalized Newton method. Clearly, using the generalized Newton method to solve the AVE (1.1), the inverse of the matrix $A - D(x^{(k)})$ should be computed. Noting that the matrix

$A - D(x^{(k)})$ is changed with the iteration index k , the computation cost of the generalized Newton method may be highly expensive. To remain the iteration matrix unchanged, the following Picard iteration method was considered in [11]

$$x^{(k+1)} = A^{-1}(|x^{(k)}| + b), k = 0, 1, \dots \quad (1.3)$$

When the matrix A in (1.3) is ill-conditioned, in each iteration of the Picard method, we have to deal with this ill-conditioned linear system. To overcome the inverse of the matrix A , by making use of the Hermitian and skew-Hermitian splitting (HSS) of the matrix A in [12], Zhu et al. in [13] presented the nonlinear HSS-like method and discussed the convergence property of the nonlinear HSS-like method. Other versions of the nonlinear HSS-like method, one can see [14–16] for more details. In addition, in [17], the classical AOR iteration method has been expanded to solve the AVE. Other relative works are [18–24].

Recently, based on the methodology of the Gauss-Seidel method, together with the matrix splitting $A = D - L - U$ of matrix A , where $D = \text{diag}(A)$, L and U are strictly lower and upper triangular matrices obtained from $-A$, respectively, in [25], the generalized Gauss-Seidel (GGS) iteration method was proposed and adopted to solve the AVE (1.1), which is of form

$$(D - L)x^{(k+1)} - |x^{(k+1)}| = U|x^{(k)}| + b, k = 0, 1, \dots \quad (1.4)$$

Numerical experiments showed the efficiency of the GGS method.

In this paper, inspired by the work in [25], based on the AOR iteration method, a generalization of the AOR iteration method (GAOR) is presented to solve the AVE (1.1) and its convergence conditions are discussed in detail. By making use of some numerical experiments, we present the effectiveness of the GAOR method.

The remainder of the paper is organized as follows: Section 2 goes over some preparatory knowledge. Section 3 presents the GAOR iteration method and its convergence conditions. Section 4 reports some numerical results to show the efficiency of the GAOR method. Finally, Section 5 draws some conclusions.

2. Preparatory knowledge

Let $C = (c_{ij}) \in \mathbb{R}^{n \times n}$. ‘diag(C)’ denotes the diagonal part of matrix C . For $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{n \times n}$, we call $A \geq B$ if $a_{ij} \geq b_{ij}$ for $i, j = 1, 2, \dots, n$. Matrix A is called non-negative if $A \geq 0$; further, $A - B \geq 0$ if and only if $A \geq B$. These definitions carry immediately over to vectors by identifying them with $n \times 1$ matrices. Matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is called a Z -matrix if $a_{ij} \leq 0$ for $i \neq j$; an L -matrix if A is a Z -matrix and $a_{ii} > 0$ for $i = 1, \dots, n$; an M -matrix if A is a Z -matrix and $A^{-1} \geq 0$; and an H -matrix if its comparison matrix $\langle A \rangle = (\langle a \rangle_{ij}) \in \mathbb{R}^{n \times n}$ is an M -matrix, where

$$\langle a \rangle_{ij} = \begin{cases} |a_{ij}| & \text{for } i = j, \\ -|a_{ij}| & \text{for } i \neq j, \end{cases} \quad i, j = 1, 2, \dots, n.$$

If A is an M -matrix and B is a Z -matrix, then $A \leq B$ implies that B is an M -matrix [26]. A matrix A is irreducible if the directed graph associated to A is strongly connected [27].

Let $A = M - N$. It is called a matrix splitting of A if $\det(M) \neq 0$; called regular if $M^{-1} \geq 0$ and $N \geq 0$; an M -splitting of A if M is an M -matrix and $N \geq 0$. Evidently, if $A = M - N$ is an M -splitting and A is a nonsingular M -matrix, then $\rho(M^{-1}N) < 1$, where $\rho(\cdot)$ denotes the spectral radius of the matrix.

Lemma 2.1. [28] *Let A be an H -matrix. Then $|A^{-1}| \leq \langle A \rangle^{-1}$.*

Lemma 2.2. [29] *Let $x, y \in \mathbb{R}^n$. Then $\| |x| - |y| \|_2 \leq \|x - y\|_2$.*

3. GAOR iteration method

In this section, the generalized AOR (GAOR) method is introduced to solve the AVE (1.1). For this purpose, we split A into

$$A = \frac{1}{\omega}(M - N), \quad (3.1)$$

with

$$M = \Omega + D - rL \text{ and } N = \Omega + (1 - \omega)D + (\omega - r)L + \omega U,$$

where Ω is a positive diagonal matrix, ω and r are real parameters with $\omega \neq 0$, $D = \text{diag}(A)$, L and U are the previously mentioned. Based on the matrix splitting (3.1), the AVE (1.1) is rewritten as the fixed-point equations

$$\frac{1}{\omega}(\Omega + D - rL)x - |x| = \frac{1}{\omega}[\Omega + (1 - \omega)D + (\omega - r)L + \omega U]x + b$$

or

$$(\Omega + D - rL)x - \omega|x| = [\Omega + (1 - \omega)D + (\omega - r)L + \omega U]x + \omega b, \quad (3.2)$$

then for $k = 0, 1, \dots$, we can define the GAOR method for solving (1.1) below

$$(\Omega + D - rL)x^{(k+1)} - \omega|x^{(k+1)}| = [\Omega + (1 - \omega)D + (\omega - r)L + \omega U]x^{(k)} + \omega b. \quad (3.3)$$

Lemma 3.1. *Let $\alpha > \beta > 0$. Then $\alpha x - \beta|x| = b$ with $b \in \mathbb{R}$ has only one solution in \mathbb{R} . If $b \geq 0$, then the solution is $x = \frac{b}{\alpha - \beta}$, otherwise $x = \frac{b}{\alpha + \beta}$.*

Proof. The results in Lemma 3.1 are valid, whose proof is omitted here.

Next, based on Lemma 3.1, we present Algorithm 1 to solve each step of the GAOR iteration method (3.3) when all the diagonal entries of the matrix $\Omega + D$ are greater than ω .

Algorithm 1:

For $k=0, 1, \dots$, until convergence, do

Set $s = \omega b_1, x_i^{(0)} = 0$.

For $i = 1, 2, \dots, n$, do

If $s \geq 0$, then

$$x_i^{(k+1)} := \frac{s}{a_{ii} + \omega_{ii} - \omega};$$

Else

$$x_i^{(k+1)} := \frac{s}{a_{ii} + \omega_{ii} + \omega};$$

Endif

set

$$s := \omega b_i + (\omega_{ii} + (1 - \omega)a_{ii})x_i^{(k)} - (\omega - r) \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \omega \sum_{j=i+1}^n a_{ij}x_j^{(k)} - r \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)}.$$

Enddo

Enddo

Theorem 3.1. *Let Ω be a positive diagonal matrix and all singular values of the matrix $\Omega + D - rL$ exceed ω with $\omega, r > 0$. If*

$$\|I - \omega(\Omega + D - rL)^{-1}A\|_2 < 1 - \omega\|(\Omega + D - rL)^{-1}\|_2, \quad (3.4)$$

then the GAOR method (3.3) converges to the unique solution of the AVE (1.1) for an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$.

Proof. Since all singular values of the matrix $\Omega + D - rL$ exceed ω with $\omega, r > 0$,

$$\omega\|(\Omega + D - rL)^{-1}\|_2 = \omega\sigma_{\max}((\Omega + D - rL)^{-1}) = \frac{\omega}{\sigma_{\min}(\Omega + D - rL)} < 1.$$

Let x^* be a solution of the AVE (1.1). Then from (3.2) we have

$$(\Omega + D - rL)x^* - \omega|x^*| = [\Omega + (1 - \omega)D + (\omega - r)L + \omega U]x^* + \omega b. \quad (3.5)$$

After subtracting (3.3) from (3.5), we obtain

$$(\Omega + D - rL)(x^{(k+1)} - x^*) = \omega(|x^{(k+1)}| - |x^*|) + (\Omega + (1 - \omega)D + (\omega - r)L + \omega U)(x^{(k)} - x^*),$$

which is equal to

$$x^{(k+1)} - x^* = (\Omega + D - rL)^{-1}[\omega(|x^{(k+1)}| - |x^*|) + (\Omega + (1 - \omega)D + (\omega - r)L + \omega U)(x^{(k)} - x^*)].$$

Taking 2-norm in the latter equation yields

$$\begin{aligned} \|x^{(k+1)} - x^*\|_2 &= \|(\Omega + D - rL)^{-1}[\omega(|x^{(k+1)}| - |x^*|) + (\Omega + (1 - \omega)D \\ &\quad + (\omega - r)L + \omega U)(x^{(k)} - x^*)]\|_2 \\ &\leq \omega\|(\Omega + D - rL)^{-1}(|x^{(k+1)}| - |x^*|)\|_2 + \|(\Omega + D - rL)^{-1} \\ &\quad (\Omega + (1 - \omega)D + (\omega - r)L + \omega U)(x^{(k)} - x^*)\|_2 \\ &\leq \omega\|(\Omega + D - rL)^{-1}\|_2\|x^{(k+1)} - |x^*|\|_2 + \|(\Omega + D - rL)^{-1} \\ &\quad (\Omega + (1 - \omega)D + (\omega - r)L + \omega U)\|_2\|x^{(k)} - x^*\|_2 \\ &\leq \omega\|(\Omega + D - rL)^{-1}\|_2\|x^{(k+1)} - x^*\|_2 + \|(\Omega + D - rL)^{-1} \end{aligned}$$

$$\begin{aligned} & (\Omega + (1 - \omega)D + (\omega - r)L + \omega U)\|_2 \|x^{(k)} - x^*\|_2 \\ &= \omega \|(\Omega + D - rL)^{-1}\|_2 \|x^{(k+1)} - x^*\|_2 \\ & \quad + \|I - \omega(\Omega + D - rL)^{-1}A\|_2 \|x^{(k)} - x^*\|_2. \end{aligned}$$

Further,

$$\|x^{(k+1)} - x^*\|_2 \leq \frac{\|I - \omega(\Omega + D - rL)^{-1}A\|_2}{1 - \omega\|(\Omega + D - rL)^{-1}\|_2} \|x^{(k)} - x^*\|_2.$$

To show the uniqueness of the solution, let y^* be another solution of the AVE (1.1). Then, from Eq (3.2) we have

$$(\Omega + D - rL)y^* - \omega|y^*| = [\Omega + (1 - \omega)D + (\omega - r)L + \omega U]y^* + \omega b. \quad (3.6)$$

After subtracting (3.6) from (3.5), similar to the above discussion, we obtain

$$\begin{aligned} \|y^* - x^*\|_2 &\leq \frac{\|I - \omega(\Omega + D - rL)^{-1}A\|_2}{1 - \omega\|(\Omega + D - rL)^{-1}\|_2} \|y^* - x^*\|_2 \\ &< \|y^* - x^*\|_2, \end{aligned}$$

which is contradiction. Therefore, $y^* = x^*$. This completes the proof.

When $w = r$ in (3.3), the GAOR method reduces to the GSOR method. Its convergence theory is given in Corollary 3.1.

Corollary 3.1. *Let Ω be a positive diagonal matrix and all singular values of the matrix $\Omega + D - \omega L$ exceed ω with $\omega > 0$. If*

$$\|I - \omega(\Omega + D - \omega L)^{-1}A\|_2 < 1 - \omega\|(\Omega + D - \omega L)^{-1}\|_2, \quad (3.7)$$

then the GSOR method (3.3) converges to the unique solution of the AVE (1.1) for an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$.

Further, in Corollary 3.1, if $\omega = 1$, then the convergence theory of GGS is obtained and described in Corollary 3.2.

Corollary 3.2. *Let Ω be a positive diagonal matrix and all singular values of the matrix $\Omega + D - L$ exceed 1. If*

$$\|I - (\Omega + D - L)^{-1}A\|_2 < 1 - \|(\Omega + D - L)^{-1}\|_2, \quad (3.8)$$

then the GGS method (3.3) converges to the unique solution of the AVE (1.1) for an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$.

Remark 3.1. In Corollary 3.2, if Ω is a zero matrix, then the results of Corollary 3.2 are similar to Theorem 3 in [25]. In fact, if we use ‘all singular values of the matrix $D - L$ exceed 1’ instead of ‘Let the diagonal entries of A be greater than one and the matrix $D - L - I$ be strictly row diagonally dominant’ in Theorem 3 in [25], then Theorem 3 in [25] holds as well.

It is not difficult to find that the above theoretical results including Theorem 3.1, Corollary 3.1 and Corollary 3.2 are a litter too general and not easy to verify. To overcome these negative factors, an effective approach is to limit the type of matrix A . Next, we restrict our attention to this situation where A is an H -matrix with positive diagonals. For A being an H -matrix with positive diagonals, a simple convergence condition for the GAOR method (3.3) can be obtained, see Theorem 3.2.

Theorem 3.2. Let A be an H -matrix with positive diagonals and the positive diagonal matrix $\Omega \geq \omega I$. If ω and r satisfy

$$0 < r \leq \omega \leq 1, \quad (3.9)$$

then the GAOR method (3.3) converges to the unique solution of (1.1) for an arbitrary initial guess $x^{(0)} \in \mathbb{R}^n$.

Proof. Since A is an H -matrix with positive diagonals, we have

$$\langle A \rangle = |D| - |L| - |U|.$$

Further,

$$\langle A \rangle \leq \langle D - L \rangle \leq \langle D - rL \rangle.$$

It follows that matrix $\Omega + D - rL$ is an H -matrix with positive diagonals. It holds that

$$|(\Omega + D - rL)^{-1}| < \langle \Omega + D - rL \rangle^{-1}.$$

From the proof of Theorem 3.1, we get

$$\begin{aligned} x^{(k+1)} - x^* &= (\Omega + D - rL)^{-1} [\omega(|x^{(k+1)}| - |x^*|) + (\Omega + (1 - \omega)D \\ &\quad + (\omega - r)L + \omega U)(x^{(k)} - x^*)]. \end{aligned}$$

Further,

$$\begin{aligned} |x^{(k+1)} - x^*| &= |(\Omega + D - rL)^{-1} [\omega(|x^{(k+1)}| - |x^*|) + (\Omega + (1 - \omega)D \\ &\quad + (\omega - r)L + \omega U)(x^{(k)} - x^*)]| \\ &\leq \omega |(\Omega + D - rL)^{-1}| \cdot \| |x^{(k+1)}| - |x^*| \| + |(\Omega + D - rL)^{-1}| \\ &\quad | \Omega + (1 - \omega)D + (\omega - r)L + \omega U \| |x^{(k)} - x^*| \\ &\leq \omega \langle \Omega + D - rL \rangle^{-1} |x^{(k+1)} - x^*| + \langle \Omega + D - rL \rangle^{-1} \\ &\quad | \Omega + (1 - \omega)D + (\omega - r)L + \omega U \| |x^{(k)} - x^*|. \end{aligned} \quad (3.10)$$

Since the matrix $\Omega + D - rL$ is an H -matrix with positive diagonals, the matrix $\langle \Omega + D - rL \rangle$ is an M -matrix and $\langle \Omega + D - rL \rangle^{-1} > 0$. Noting that $\Omega \geq \omega I$, it is easy to obtain that

$$\rho(\omega \langle \Omega + D - rL \rangle^{-1}) = \max \frac{\omega}{a_{ii} + \omega_{ii}} < 1,$$

where $a_{ii} + \omega_{ii}$ denotes the diagonal elements of matrix $\Omega + D$. Hence, from (3.10), we get

$$\begin{aligned} |x^{(k+1)} - x^*| &\leq (I - \omega \langle \Omega + D - rL \rangle^{-1})^{-1} \langle \Omega + D - rL \rangle^{-1} \\ &\quad | \Omega + (1 - \omega)D + (\omega - r)L + \omega U \| |x^{(k)} - x^*| \\ &= (\langle \Omega + D - rL \rangle - \omega I)^{-1} | \Omega + (1 - \omega)D + (\omega - r)L + \omega U \| |x^{(k)} - x^*|. \end{aligned}$$

Let

$$\bar{M} = \langle \Omega + D - rL \rangle, \bar{N} = \omega I + | \Omega + (1 - \omega)D + (\omega - r)L + \omega U | \text{ and } \bar{A} = \bar{M} - \bar{N}.$$

Evidently, the GAOR method (3.3) converges to the unique solution of the AVE (1.1) if $\rho(\bar{M}^{-1}\bar{N}) < 1$. Noting that \bar{A} is an M -matrix, matrix \bar{M} is an M -matrix and $\bar{N} \geq 0$. Then, $\bar{A} = \bar{M} - \bar{N}$ is an M -splitting. Therefore, $\rho(\bar{M}^{-1}\bar{N}) < 1$.

From Theorem 3.2, the convergence conditions of the corresponding GSOR and GGS methods for A being an H -matrix with positive diagonals are obtained, which are omitted.

4. Numerical experiments

In this section, we give some numerical experiments to assess the efficiency of the GAOR method for solving the AVE (1.1). We compare GAOR with AOR in [17] from two aspects: the number of iterations (denoted as IT) and the CPU time in seconds (denoted as CPU). Meanwhile, we also investigate the generalized Newton method, the nonlinear HSS-like method [13] and the Picard-HSS method [14]. The starting iterate is zero vector, all iterations are terminated once the relative residual error satisfies

$$\frac{\|Ax^{(k)} - |x^{(k)}| - b\|_2}{\|b\|_2} \leq 10^{-6}$$

or if the number of the prescribed iteration $k_{\max} = 500$ is exceeded. All the tests are performed in MATLAB R2016B.

Example 4.1. ([25,30]) Let m be a prescribed positive integer and $n = m^2$. Consider the AVE (1.1), in which $A \in \mathbb{R}^{n \times n}$ is given by $A = \hat{M} + \mu I$, where

$$\hat{M} = \text{tridiag}(-I, S, -I) = \begin{bmatrix} S & -I & 0 & \cdots & 0 & 0 \\ -I & S & -I & \cdots & 0 & 0 \\ 0 & -I & S & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & S & -I \\ 0 & 0 & 0 & \cdots & -I & S \end{bmatrix} \in \mathbb{R}^{n \times n}$$

with

$$S = \text{tridiag}(-1, 4, -1) = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 4 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 4 \end{bmatrix} \in \mathbb{R}^{m \times m},$$

and $b = Ax^* - |x^*|$ with $x^* = (-1, 1, -1, 1, \dots, -1, 1)^T \in \mathbb{R}^n$.

Example 4.1 can be induced by using the finite difference to discretize the flow of water through a porous dam under the equidistant grid [30].

Example 4.2. ([31]) Let m be a prescribed positive integer and $n = m^2$. Consider the AVE (1.1), in which $A \in \mathbb{R}^{n \times n}$ is given by $A = \hat{M} + \mu I$, where

$$\hat{M} = \text{tridiag}(-1.5I, S, -0.5I) = \begin{bmatrix} S & -0.5I & 0 & \cdots & 0 & 0 \\ -1.5I & S & -0.5I & \cdots & 0 & 0 \\ 0 & -1.5I & S & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & S & -0.5I \\ 0 & 0 & 0 & \cdots & -1.5I & S \end{bmatrix} \in \mathbb{R}^{n \times n}$$

with

$$S = \text{tridiag}(-1.5, 4, -0.5) = \begin{bmatrix} 4 & -0.5 & 0 & \cdots & 0 & 0 \\ -1.5 & 4 & -0.5 & \cdots & 0 & 0 \\ 0 & -1.5 & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & -0.5 \\ 0 & 0 & 0 & \cdots & -1.5 & 4 \end{bmatrix} \in \mathbb{R}^{m \times m},$$

and $b = Ax^* - |x^*|$ with $x^* = (-1, 1, -1, 1, \dots, -1, 1)^T \in \mathbb{R}^n$.

Table 1. IT and CPU for Example 4.1 with $\omega = 0.9$ and $\Omega = 0.1I$.

		r	0.9	0.5	0.1
$n = 32^2$	GAOR	IT	20	23	27
		CPU	0.3092	0.3764	0.4366
	AOR	IT	25	30	35
		CPU	0.0065	0.0083	0.0095
$n = 64^2$	GAOR	IT	20	23	27
		CPU	4.0596	4.6910	5.4559
	AOR	IT	26	30	35
		CPU	0.0122	0.0150	0.0163
$n = 128^2$	GAOR	IT	20	24	27
		CPU	58.8267	70.3705	79.2503
	AOR	IT	26	30	35
		CPU	0.0474	0.0551	0.0719

Table 2. IT and CPU for Example 4.1 with $\omega = 0.9$ with $\Omega = 0.5I$.

		r	0.9	0.5	0.1
$n = 32^2$	GAOR	IT	23	26	30
		CPU	0.4073	0.4193	0.5326
	AOR	IT	25	30	35
		CPU	0.0038	0.0096	0.0098
$n = 64^2$	GAOR	IT	23	27	31
		CPU	4.6584	5.4570	6.2490
	AOR	IT	26	30	35
		CPU	0.0113	0.0132	0.0150
$n = 128^2$	GAOR	IT	23	27	31
		CPU	67.7754	79.5942	93.9553
	AOR	IT	26	30	35
		CPU	0.0400	0.0546	0.0738

Table 3. IT and CPU for Example 4.2 with $\omega = 0.9$ with $\Omega = 0.1I$.

		r	0.9	0.5	0.1
$n = 32^2$	GAOR	IT	15	20	25
		CPU	0.2489	0.3082	0.4191
	AOR	IT	20	26	33
		CPU	0.0021	0.0040	0.0046
$n = 64^2$	GAOR	IT	15	21	26
		CPU	3.1600	4.3562	5.3924
	AOR	IT	20	27	34
		CPU	0.0080	0.0114	0.0207
$n = 128^2$	GAOR	IT	15	21	27
		CPU	46.3959	68.4617	79.7561
	AOR	IT	20	27	34
		CPU	0.0330	0.0689	0.1058

Table 4. IT and CPU for Example 4.2 with $\omega = 0.9$ with $\Omega = 0.5I$.

		r	0.9	0.5	0.1
$n = 32^2$	GAOR	IT	17	23	29
		CPU	0.3110	0.3740	0.4757
	AOR	IT	20	26	33
		CPU	0.0021	0.0031	0.0060
$n = 64^2$	GAOR	IT	18	24	30
		CPU	3.6997	4.8956	6.0457
	AOR	IT	20	27	34
		CPU	0.0081	0.0114	0.0164
$n = 128^2$	GAOR	IT	18	24	30
		CPU	52.7953	70.6259	97.6267
	AOR	IT	20	27	34
		CPU	0.0332	0.0523	0.0686

To fairly compare the GAOR method with the AOR method, we choose the same parameters to test the GAOR method and the AOR method. Under this consideration, in Tables 1–4, for Examples 4.1 and 4.2 with $\mu = 2$, $\Omega = 0.1I$ and $\Omega = 0.5I$, we list some iteration results to illustrate the convergence behaviors of the GAOR and AOR methods for the different problem sizes of n . We observe that the GAOR and AOR methods can calculate a satisfactory approximation to the solution of the AVE. Fixing ω and n with r increasing, the iteration steps of the GAOR and AOR methods descend. Fixing ω and r with n increasing, the iteration steps of the GAOR and AOR methods may be hardly sensitive to change. Further, we find that the GAOR method requires less iteration steps than the AOR method, but the GAOR method requires more CPU times than the AOR method. The time-consuming of the GAOR method is due to the code only edited by all the components of the matrix. To make the AOR method more competitive, an effective approach is to optimize the edited code, which is an interesting work in the future.

Table 5. Numerical results of GN for Example 4.1.

n	32^2	64^2	128^2
IT	3	3	3
CPU	0.2344	4.5520	235.1104

Table 6. Numerical results of GN for Example 4.2.

n	32^2	64^2	128^2
IT	2	3	3
CPU	0.5260	8.2994	508.0766

Tables 5 and 6 list the numerical results of the generalized Newton method, where ‘GN’ denote the generalized Newton method. Comparing the GAOR method, the AOR method with the generalized Newton method, the generalized Newton method require the least iteration steps, but it also takes the most time. Among three methods, the computational efficiency of the generalized Newton method is worst.

Table 7. IT and CPU for Example 4.2 with $\Omega = 0.5I$.

		ω	1	0.9	0.8
$n = 32^2$	GSOR	IT	13	17	23
		CPU	0.2578	0.3110	0.4375
	NHSS	IT	112	346	–
		CPU	0.4957	1.5667	–
	PHSS	IT	12(240)	26(520)	–
		CPU	0.9727	2.2767	–
$n = 64^2$	GSOR	IT	13	18	24
		CPU	3.0469	3.6997	5.2188
	NHSS	IT	170	471	–
		CPU	3.3429	9.0310	–
	PHSS	IT	20(400)	–	–
		CPU	2.8327	–	–
$n = 128^2$	GSOR	IT	13	18	24
		CPU	40.0088	52.7953	78.6563
	NHSS	IT	286	–	–
		CPU	36.3121	–	–
	PHSS	IT	30(600)	–	–
		CPU	54.0620	–	–

Finally, we compare the GAOR method with the nonlinear HSS-like method and the Picard-HSS method under the same parameter, see Table 7. The explanation is that the nonlinear HSS-like method and the Picard-HSS method only contain a parameter, here, we consider the GAOR method with $\omega = r$, i.e., the GSOR method. When the Picard-HSS method is applied, the stopping criterion for its inner iterations is adopted in [13]. In Table 7, ‘GSOR’, ‘NHSS’ and ‘PHSS’, respectively, denote the GSOR

method, the nonlinear HSS-like method and the Picard-HSS method, ‘–’ denotes that the iteration steps exceed 500, and ‘·(·)’ denotes the outer (inner) iteration steps. From Table 7, the numerical results show that the GSOR method is more effective than the nonlinear HSS-like method and the Picard-HSS method.

Overall, based on the numerical results, the GAOR method displays the good performance when the above presented five testing methods are applied to solve the absolute value equations.

5. Conclusions

In this paper, we have designed a generalization AOR (GAOR) method to solve the absolute value equations. Some convergence properties for the proposed GAOR method are gained. Numerical experiments have been reported to verify the efficiency of the proposed method.

Acknowledgements

The author thanks five anonymous referees for their constructive suggestions and helpful comments, which led to significant improvement of the original manuscript of this paper. The authors are partially supported by National Natural Science Foundation of China (11961082).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. J. Rohn, A theorem of the alternatives for the equation $Ax + B|x| = b$, *Linear Multilinear A.*, **52** (2004), 421–426. <https://doi.org/10.1080/0308108042000220686>
2. O. L. Mangasarian, Absolute value programming, *Comput. Optim. Appl.*, **36** (2007), 43–53. <https://doi.org/10.1007/s10589-006-0395-5>
3. O. L. Mangasarian, Absolute value equations via concave minimization, *Optim. Lett.*, **1** (2007), 1–8. <https://doi.org/10.1007/s11590-006-0005-6>
4. R. W. Cottle, G. B. Dantzig, Complementary pivot theory of mathematical programming, *Linear Algebra Appl.*, **1** (1968), 103–125. [https://doi.org/10.1016/0024-3795\(68\)90052-9](https://doi.org/10.1016/0024-3795(68)90052-9)
5. R. W. Cottle, J. S. Pang, R. E. Stone, *The Linear Complementarity Problem*, Academic, San Diego, 1992. https://doi.org/10.1007/978-94-015-8330-5_3
6. O. L. Mangasarian, A generalized Newton method for absolute value equations, *Optim. Lett.*, **3** (2009), 101–108. <https://doi.org/10.1007/s11590-008-0094-5>
7. L. Caccetta, B. Qu, G. L. Zhou, A globally and quadratically convergent method for absolute value equations, *Comput. Optim. Appl.*, **48** (2011), 45–58. <https://doi.org/10.1007/s10589-009-9242-9>
8. S. L. Hu, Z. H. Huang, Q. Zhang, A generalized Newton method for absolute value equations associated with second order cones, *J. Comput. Appl. Math.*, **235** (2011), 1490–1501. <https://doi.org/10.1016/j.cam.2010.08.036>

9. S. Ketabchi, H. Moosaei, An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side, *Comput. Math. Appl.*, **64** (2012), 1882–1885. <https://doi.org/10.1016/j.camwa.2012.03.015>
10. 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. <https://doi.org/10.1007/s10957-009-9557-9>
11. J. Rohn, V. Hooshyarbakhsh, R. Farhadsefat, An iterative method for solving absolute value equations and sufficient conditions for unique solvability, *Optim. Lett.*, **8** (2014), 35–44. <https://doi.org/10.1007/s11590-012-0560-y>
12. Z. Z. Bai, G. H. Golub, M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM J. Matrix Anal. Appl.*, **24** (2003), 603–626. <https://doi.org/10.1137/S0895479801395458>
13. M. Z. Zhu, Y. E. Qi, The nonlinear HSS-like iteration method for absolute value equations, *IAENG Inter. J. Appl. Math.*, **48** (2018), 312–316.
14. D. K. Salkuyeh, The Picard-HSS iteration method for absolute value equations, *Optim. Lett.*, **8** (2014), 2191–2202. <https://doi.org/10.1007/s11590-014-0727-9>
15. J. J. Zhang, The relaxed nonlinear PHSS-like iteration method for absolute value equations, *Appl. Math. Comput.*, **265** (2015), 266–274. <https://doi.org/10.1016/j.amc.2015.05.018>
16. C. X. Li, On the modified Hermitian and skew-Hermitian splitting iteration methods for a class of the weakly absolute value equations, *J. Inequal. Appl.*, **2016** (2016), 260. <https://doi.org/10.1186/s13660-016-1202-1>
17. C. X. Li, A preconditioned AOR iterative method for the absolute value equations, *Int. J. Comp. Methods*, **14** (2017), 1750016. <https://doi.org/10.1142/S0219876217500165>
18. X. M. Gu, T. Z. Huang, H. B. Li, S. F. Wang, L. Li, Two CSCS-based iteration methods for solving absolute value equations, *J. Appl. Anal. Comput.*, **7** (2017), 1336–1356. <https://doi.org/10.11948/2017082>
19. J. He, Y. M. Liu, J. K. Tian, Two numerical iteration methods for solving absolute value equations, *ScienceAsia*, **44** (2018), 40–45. <https://doi.org/10.2306/scienceasia1513-1874.2018.44.040>
20. Z. S. Yu, L. Li, Y. Yuan, A modified multivariate spectral gradient algorithm for solving absolute value equations, *Appl. Math. Lett.*, **121** (2021), 107461. <https://doi.org/10.1016/j.aml.2021.107461>
21. C. R. Chen, Y. N. Yang, D. M. Yu, D. R. Han, An inverse-free dynamical system for solving the absolute value equations, *Appl. Numer. Math.*, **168** (2021), 170–181. <https://doi.org/10.1016/j.apnum.2021.06.002>
22. Y. F. Ke, C. F. Ma, On SOR-like iteration methods for solving weakly nonlinear systems, *Optim. Method. Softw.*, 2020. <https://doi.org/10.1080/10556788.2020.1755861>
23. Y. F. Ke, The new iteration algorithm for absolute value equation, *Appl. Math. Lett.*, **99** (2020), 105990. <https://doi.org/10.1016/j.aml.2019.07.021>
24. Y. F. Ke, C. F. Ma, SOR-like iteration method for solving absolute value equations, *Appl. Math. Comput.*, **311** (2017), 195–202. <https://doi.org/10.1016/j.amc.2017.05.035>

25. V. Edalatpour, D. Hezari, D. K. Salkuyeh, A generalization of the Gauss-Seidel iteration method for solving absolute value equations, *Appl. Math. Comput.*, **293** (2017), 156–167. <https://doi.org/10.1016/j.amc.2016.08.020>
26. A. Berman, R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic, New York, 1979. <https://doi.org/10.1016/C2013-0-10361-3>
27. R. S. Varga, *Matrix Iterative Analysis*, Springer, Berlin, 2000. <https://doi.org/10.2307/3620826>
28. S. L. Wu, C. X. Li, Two-sweep modulus-based matrix splitting iteration methods for linear complementarity problems, *J. Comput. Appl. Math.*, **302** (2016), 327–339. <https://doi.org/10.1016/j.cam.2016.02.011>
29. J. L. Dong, M. Q. Jiang, A modified modulus method for symmetric positive-definite linear complementarity problems, *Numer. Linear Algebra Appl.*, **16** (2009), 129–143. <https://doi.org/10.1002/nla.609>
30. C. M. Elliot, J. R. Ockenden, *Weak Variational Methods for Moving Boundary Value Problems*, Pitman, London, 1982. <https://doi.org/10.1137/1026023>
31. Z. Z. Bai, Modulus-based matrix splitting iteration methods for linear complementarity problems, *Numer. Linear Algebra Appl.*, **17** (2010), 917–933. <https://doi.org/10.1002/nla.680>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)