



An inexact fixed point iteration method for solving absolute value equation

Xin-Mei Lv¹ · Shu-Xin Miao¹

Received: 6 August 2023 / Revised: 22 November 2023 / Accepted: 6 December 2023 /
Published online: 3 January 2024

© The JJIAM Publishing Committee and Springer Nature Japan KK, part of Springer Nature 2024

Abstract

The fixed point iteration method is an effective method for solving absolute value equation via equivalent two-by-two block form. To further improve the computational efficiency of the fixed point iteration method, by using the preconditioned shift-splitting strategy, we propose an inexact fixed point iteration method for solving absolute value equation in this paper. We obtain some convergence conditions for the proposed method. The effectiveness of the proposed method are shown by three examples.

Keywords Absolute value equation · Matrix-splitting · Fixed point iteration · Convergence

Mathematics Subject Classification JIAM-D-23-00181R1

1 Introduction

Considering the absolute value equation(AVE)

$$Ax - |x| = b, \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$ is an unknown vector to be determined, $|x|$ denotes the vector with absolute values of each component of x . System (1) is a special case of the generalized absolute value equation (GAVE) [16]:

$$Ax - B|x| = b, \quad (2)$$

✉ Xin-Mei Lv
lxm20222023@163.com

Shu-Xin Miao
miaosx205@163.com

¹ College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China

where $B \in \mathbb{R}^{n \times n}$, which was firstly introduced by Rohn [16] and studied in a more general background in [12, 14, 15]. AVE (1) arises in a variety of scientific computing and engineering applications such as linear programming [10, 12], the quasi-complementarity problems [18], quadratic programming, the general linear complementarity problem [3], and so on.

Recently, many scholars have studied the unique solvability of AVE (1) and GAVE (2), for example, Wu and Li presented two necessary and sufficient conditions and some sufficient conditions for the unique solvability of AVE (1) in [19]. More solvability conditions of AVE (1) can be found in [7] and references therein. In order to approximate its numerical solution, a large number of methods have been proposed to solve AVE (1) or GAVE (2), including modified or generalized Newton method [15, 20], matrix splitting iterative method [1], Picard-type method [17], the neural network model methods [4, 13], and the methods based on the equivalent two-by-two block form, such as the SOR-like method [6, 8], the fixed point iteration (FPI) method [9], the modified fixed point iteration (MFPI) method [21] and the shift-splitting fixed point iteration method [11].

By reformulating the AVE (1) to an equivalent two-by-two block form, Ke [9] proposed the FPI method for solving the AVE (1), which can be described as

Method 1 (FPI Method [9]) *Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix and $b \in \mathbb{R}^n$. Given the initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^n$, for $k = 0, 1, 2, \dots$ until the iteration sequence $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ is convergent, compute*

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

where the relaxation parameter $\omega > 0$.

Note that there is a linear system with coefficient matrix A need to be solved at each step of the FPI method, we prefer to use iterative method to approximate its solution since matrix A is always large and sparse. If we split A as

$$A = \frac{1}{2}(\alpha I + A) - \frac{1}{2}(\alpha I - A),$$

where α is a positive parameter, and approximate $x^{(k+1)}$ in the FPI method by the shift-splitting method [2], then we have the following shift-splitting FPI method (abbreviated as FPI-SS method) for solving AVE (1)

Method 2 (FPI-SS Method for AVE (1)) *Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix and $b \in \mathbb{R}^n$. Let α be a positive constant such that $\alpha I + A \in \mathbb{R}^{n \times n}$ is nonsingular. Given the initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^n$, compute $(x^{(k+1)}, y^{(k+1)})$ for $k = 0, 1, 2, \dots$ until the iteration sequence $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ is convergent, compute*

$$\begin{cases} x^{(k+1)} = (\alpha I + A)^{-1}(\alpha I - A)x^{(k)} + 2(\alpha I + A)^{-1}(y^{(k)} + b), \\ y^{(k+1)} = (1 - \omega)y^{(k)} + \omega|x^{(k+1)}|, \end{cases} \quad (4)$$

where the parameter ω is a positive constant.

The FPI-SS method is an inexact FPI method, and is firstly proposed for solving the GAVE (2) in [11]. When $B = I$, the GAVE (2) is the AVE (1), and Algorithm 3 in [11] reduces to Method 2. In this paper, based on the preconditioned shift-splitting technique, we propose another inexact FPI method for solving the AVE (1). This paper is organized as follows. In Sect. 2, new inexact FPI method for solving the AVE (1) is established. The convergence analysis of the proposed method is studied in Sect. 3. In Sect. 4, numerical experiments are present to illustrate the effectiveness and feasibility of the proposed method. Finally, a brief conclusion is given in Sect. 5.

2 The FPI-PSS method

Similar to [5], assume that A is splitted as

$$A = \frac{1}{2}(\alpha P + A) - \frac{1}{2}(\alpha P - A)$$

with positive parameter α and symmetric positive definite matrix P , then the $x^{(k+1)}$ in the FPI method (3) can be solved by the following preconditioned shift-splitting (PSS) method

$$x^{(k+1)} = (\alpha P + A)^{-1}(\alpha P - A)x^{(k)} + 2(\alpha P + A)^{-1}(y^{(k)} + b).$$

Hence, we have the following inexact FPI method, termed as the FPI-PSS method, for solving the AVE (1)

Method 3 (FPI-PSS Method for AVE (1)) *Let $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Given the initial vectors $x^{(0)} \in \mathbb{R}^n$ and $y^{(0)} \in \mathbb{R}^n$, compute $\{x^{(k+1)}, y^{(k+1)}\}$ for $k = 0, 1, 2, \dots$ using the following iteration scheme until $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ satisfies the stopping criterion:*

$$\begin{cases} x^{(k+1)} = (\alpha P + A)^{-1}(\alpha P - A)x^{(k)} + 2(\alpha P + A)^{-1}(y^{(k)} + b), \\ y^{(k+1)} = (1 - \omega)y^{(k)} + \omega|x^{(k+1)}|, \end{cases} \tag{5}$$

where α is a positive iteration parameter and P is symmetric positive definite matrix.

Clearly, the iteration matrix of the FPI-PSS method is

$$M = \begin{bmatrix} \alpha P + A & 0 \\ -\omega D(x) & I \end{bmatrix}^{-1} \begin{bmatrix} \alpha P - A & 2I \\ 0 & (1 - \omega)I \end{bmatrix},$$

where $D(x)$ is a diagonal matrix of the form $D(x) = \text{diag}(\text{sign}(x))$ wherein $\text{sign}(x)$ denotes a vector with components equal to 1, 0 or -1 depending on whether the corresponding component of x is positive, zero or negative, respectively.

Especially, when $P = I$, the FPI-PSS method becomes the FPI-SS method. Therefore, the proposed FPI-PSS method is a generalization of Method 2. Moreover, we can see that the FPI-PSS method has the same computational processes as the shift-splitting fixed point iteration method in [11], so the FPI-PSS method can also be used to solve the GAVE (2).

3 Convergence of the FPI-PSS method

In this section, the convergence of the FPI-PSS method for solving the AVE (1) is studied. Let $\rho(M)$ denotes the spectral radius of the iteration matrix M , then the FPI-PSS method is convergent if and only if $\rho(M) < 1$. Assume that λ is an eigenvalue of matrix M and $[u, v]^T$ is the corresponding eigenvector, we have

$$M \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix},$$

where is equivalent to

$$\begin{cases} (\alpha P - A)u + 2v = \lambda(\alpha P + A)u, \\ (1 - \omega)v = \lambda(-\omega D(x)u + v). \end{cases} \quad (6)$$

Next, we will study the convergence of the FPI-PSS method. For this purpose, several helpful lemmas are presented as follows.

Lemma 1 [6] *Let $A \in \mathbb{R}^{n \times n}$, if the smallest singular value of the A exceed 1 and η is an eigenvalue of the matrix $D(x)A^{-1}$, then $|\eta| < 1$.*

Lemma 2 [22] *Consider the real quadratic equation $x^2 + bx + d = 0$, where b and d are real numbers. Both roots of the equation are less than one in modulus if and only if $|d| < 1$ and $|b| < 1 + d$.*

Lemma 3 *Let $A \in \mathbb{R}^{n \times n}$, if the smallest singular value of the A exceed 1 and λ is an eigenvalue of the matrix M , then $\lambda \neq 1$.*

Proof If $\lambda = 1$ is an eigenvalue of matrix M , then (6) is equivalent to

$$\begin{cases} (\alpha P - A)u + 2v = (\alpha P + A)u, \\ (1 - \omega)v = -\omega D(x)u + v. \end{cases} \quad (7)$$

From (7), we can get that

$$(I - D(x)A^{-1})u = 0.$$

It follows from Lemma 1 that $I - D(x)A^{-1}$ is nonsingular, so we have $u = 0$. Then, from (7) we can get that $v = 0$. We have the contradictory conclusion with properties of eigenvector. Hence $\lambda \neq 1$. \square

Lemma 4 *Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix and $\omega > 0$, if λ satisfies*

$$(\lambda - 1)(\lambda + \omega - 1)\alpha Pu + (\lambda + 1)(\lambda + \omega - 1)Au = 2\lambda\omega D(x)u, \quad (8)$$

then λ is an eigenvalue of the matrix M . Conversely, if λ is an eigenvalue of the matrix M such that $\lambda \neq 1 - \omega$, then λ satisfies (8).

Proof Let $[u, v]^T$ be the eigenvector of M corresponding to the eigenvalue λ . Then it follows from (6) that

$$\begin{cases} (\lambda - 1)\alpha Pu + (\lambda + 1)Au = 2v \\ (\lambda + \omega - 1)v = \lambda\omega D(x)u. \end{cases} \tag{9}$$

Combining the two equality in (9), we get (8). □

We can prove the other assertion by reversing the process.

Theorem 1 *Let A be a symmetric positive definite matrix. Assume that λ is an eigenvalue of iteration matrix M and $[u, v]^T \in \mathbb{C}^{n \times n}$ is the corresponding eigenvector. Denote $a = \frac{u^* Au}{u^* Pu}$, $c = \frac{u^* D(x)u}{u^* Pu}$. Then the FPI-PSS method is convergence if and only if the following conditions are satisfied*

$$\begin{cases} 0 < \omega < \frac{2\alpha}{\alpha - c}, \alpha > a > c, \\ 0 < \omega < \frac{2a}{a - \alpha}, c < \alpha < \sqrt{ac} < a, \\ 0 < \omega < \frac{2\alpha}{\alpha - c}, c < \sqrt{ac} < \alpha < a, \\ 0 < \omega < \frac{2a}{a - \alpha}, \alpha < c < a. \end{cases}$$

Proof From Lemma 4 we know that λ satisfies (8). Multiplying $\frac{u^*}{u^* Pu}$ on both sides of (8), we get

$$(\lambda - 1)(\lambda + \omega - 1)\alpha \frac{u^* Pu}{u^* Pu} + (1 + \lambda)(\lambda + \omega - 1) \frac{u^* Au}{u^* Pu} - 2\lambda\omega \frac{u^* D(x)u}{u^* Pu} = 0,$$

that is

$$(\lambda - 1)(\lambda + \omega - 1)\alpha + (1 + \lambda)(\lambda + \omega - 1)a - 2\lambda\omega c = 0,$$

or equivalently,

$$\lambda^2 + \frac{(\omega - 2)\alpha + \omega a - 2\omega c}{a + \alpha} \lambda + \frac{(1 - \omega)(\alpha - a)}{a + \alpha} = 0.$$

From Lemma 2 and Lemma 3, we know that the FPI-PSS method is convergent if and only if

$$\left| \frac{(1 - \omega)(\alpha - a)}{a + \alpha} \right| < 1$$

and

$$\left| \frac{(\omega - 2)\alpha + \omega a - 2\omega c}{a + \alpha} \right| < 1 + \frac{(1 - \omega)(\alpha - a)}{a + \alpha}.$$

In what follows, we divide our discussion into three cases for solving above inequalities.

Case 1: $c = 0$

In this case, we have $\lambda = \frac{\alpha - a}{\alpha + a}$, obviously, $|\lambda| < 1$.

Case 2: $c > 0$

Now, when $\alpha > a > c$, we get that

$$0 < \omega < \frac{2\alpha}{\alpha - c},$$

while when $a > \alpha > c$, we obtain that

$$\begin{cases} 0 < \omega < \frac{2a}{a - \alpha}, & c < \alpha < \sqrt{ac} < a \\ 0 < \omega < \frac{2\alpha}{\alpha - c}, & c < \sqrt{ac} < \alpha \leq a. \end{cases}$$

and when $\alpha < c < a$, we have

$$0 < \omega < \frac{2a}{a - \alpha}.$$

Case 3: $c < 0$

In this case, we have the same results as in the Case 2.

According to Case 1, 2 and 3, the proof is completed. \square

4 Numerical experiments

In this section, three example are given to illustrate the feasibility and efficiency of the FPI-PSS method proposed in this work. To this end, we compare the FPI-PSS method with the FPI method [9], the FPI-SS method (4) and two new fixed point iteration method [1] from aspects of the numbers of iteration steps (denoted as “IT”), elapsed CPU time in seconds (denoted as “CPU”), and relative residual error (denoted as “RES”) which is defined by

$$\text{RES} := \|Ax^{(k)} - |x^{(k)}| - b\|_2.$$

In our implementation, we choose $P = I + H$ with $H = \frac{A+A^T}{2}$, all initial guess vectors $x^{(0)}$ and $y^{(0)}$ are selected to zero vectors and all iterations are terminated if $\text{RES} \leq 10^{-6}$ or the maximum number of iteration step k_{\max} exceeds 500. All computations

Table 1 Numerical results for Example 1 with $\mu = 1$

Method	n	50 ²	100 ²	150 ²	200 ²
FPI	ω_{opt}	0.6	0.6	0.6	0.6
	IT	15	15	15	15
	CPU	0.0355	0.2741	0.6384	1.3510
	RES	3.0053e-07	5.7495e-07	8.4868e-07	3.1642e-07
FPI-SS	ω_{opt}	0.9	0.9	0.9	0.9
	α_{opt}	5.5	5.5	5.3	5.3
	IT	14	14	15	15
	CPU	0.0337	0.2500	0.6002	1.2043
FPI-PSS	RES	5.5162e-07	8.2088e-07	6.8092e-07	8.7699e-07
	ω_{opt}	1.0	1.0	1.0	1.0
	α_{opt}	1.4	1.3	1.3	1.3
	IT	9	9	9	9
FPI-PSS	CPU	0.0285	0.1639	0.3763	0.7114
	RES	1.2348e-07	3.4811e-07	5.4170e-07	7.3529e-07

are performed in MATLAB R2022b on a personal computer with 1.80GHZ central processing unit (Intel(R) Core(TM) i7-8550U) and 8GB memory.

Example 1 Let the coefficient matrix $A \in \mathbb{R}^{n \times n}$ of AVE (1) be defined by $A = \widehat{A} + \mu I \in \mathbb{R}^{n \times n}$, where

$$\widehat{A} = \text{Tridiag}(-I, S, -I) = \begin{bmatrix} S & -I & 0 & \dots & 0 & 0 \\ -I & S & -I & \dots & 0 & 0 \\ 0 & -I & S & \dots & 0 & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & S & -I \\ 0 & 0 & \dots & \dots & -I & S \end{bmatrix} \in \mathbb{R}^{n \times n}$$

is a block-tridiagonal matrix,

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

is a tridiagonal matrix, $n = m^2$. Let $x^* = (-0.5, -1, -0.5, \dots, -0.5, -1, \dots)^T \in \mathbb{R}^n$ be the exact solution of the AVE (1).

For different problem scales $n = m^2$, the optimal experimental parameters, IT, CPU and RES of the FPI, FPI-SS and FPI-PSS methods for Example 1 are listed in Table 1

Table 2 Numerical results for Example 1 with $\mu = 4$

Method	n	50 ²	100 ²	150 ²	200 ²
FPI	ω_{opt}	0.8	0.8	0.8	0.8
	IT	9	9	9	9
	CPU	0.0216	0.1743	0.3791	0.7390
	RES	1.5153e-07	2.9252e-07	4.3328e-07	5.7397e-07
FPI-SS	ω_{opt}	0.9	0.9	0.9	0.9
	α_{opt}	9.1	8.2	8.3	8.0
	IT	11	12	12	12
	CPU	0.0293	0.2141	0.5045	0.9736
FPI-PSS	RES	6.0431e-07	2.8137e-07	6.5055e-07	4.6551e-07
	ω_{opt}	1.0	1.0	1.0	1.0
	α_{opt}	1.2	1.2	1.1	1.1
	IT	7	7	7	7
FPI-PSS	CPU	0.0136	0.1233	0.2837	0.5623
	RES	7.1382e-07	5.9957e-07	6.1103e-07	8.2080e-07

and Table 2 for $\mu = 1$ and $\mu = 4$, respectively. The optimal parameters are obtained through the numerical experiments, which result in the least number of iteration steps of each methods.

From Table 1 and Table 2, we can see that each of the tested methods can successfully converge to the exact solution of AVE (1), and the number of iteration steps decreases with μ . Among all tested iteration methods, the FPI-PSS method is the most efficient one as it requires the least iterative steps and the least computation time to achieve the terminated criterion.

Example 2 Let the coefficient matrix $A \in \mathbb{R}^{n \times n}$ of AVE (1) be defined by $A = \widehat{A} + \mu I \in \mathbb{R}^{n \times n}$, where

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

Table 3 Numerical results for Example 2 with $\mu = 1$

Method	n	50 ²	100 ²	150 ²	200 ²
FPI	ω_{opt}	0.6	0.6	0.6	0.6
	IT	15	15	15	16
	CPU	0.3389	1.7421	4.7607	8.2571
	RES	3.0285e-07	5.7750e-07	8.5132e-07	3.1771e-07
FPI-SS	ω_{opt}	0.9	0.9	0.9	0.9
	α_{opt}	5.6	5.4	5.5	5.6
	IT	15	15	16	16
	CPU	0.2508	1.5804	1.6736	2.9403
FPI-PSS	RES	9.6119e-07	9.7779e-07	4.2584e-07	7.9709e-07
	ω_{opt}	1.0	1.0	1.0	1.0
	α_{opt}	1.4	1.4	1.3	1.4
	IT	9	9	9	9
FPI-PSS	CPU	0.0716	0.3788	0.9415	1.5442
	RES	5.2439e-07	6.1553e-07	8.8475e-07	8.1297e-07

is a block-tridiagonal matrix,

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

is a tridiagonal matrix, $n = m^2$. Let $x^* = (-0.5, -1, -0.5, \dots, -0.5, -1, \dots)^T \in \mathbb{R}^n$ be the exact solution of the AVE (1).

In Table 3 and Table 4, we report the numerical results for Example 2 with $\mu = 1$ and $\mu = 4$, respectively. Notably, the FPI-PSS method requires the least iteration steps and costs the least computing time than the FPI method and the FPI-SS method.

Example 3 [1] Let the coefficient matrix $A \in \mathbb{R}^{n \times n}$ of AVE (1) be defined by $A = \widehat{A} + \mu I \in \mathbb{R}^{n \times n}$, where

$$\widehat{A} = \text{Tridiag}(-1.5I, S, -0.5I) = \begin{bmatrix} S & -0.5I & 0 & \dots & 0 & 0 \\ -1.5I & S & -0.5I & \dots & 0 & 0 \\ 0 & -1.5I & S & \dots & 0 & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & S & -0.5I \\ 0 & 0 & \dots & \dots & -1.5I & S \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Table 4 Numerical results for Example 2 with $\mu = 4$

Method	n	50 ²	100 ²	150 ²	200 ²
FPI	ω_{opt}	0.8	0.8	0.8	0.8
	IT	9	9	9	9
	CPU	0.1645	1.3121	2.7513	3.6440
	RES	1.5219e-07	2.9329e-07	4.3409e-07	5.7480e-07
FPI-SS	ω_{opt}	0.9	0.9	0.9	0.9
	α_{opt}	9.2	8.6	9.1	8.1
	IT	12	12	12	12
	CPU	0.1958	0.9624	1.3512	2.2624
FPI-PSS	RES	4.1893e-07	9.4763e-07	9.5725e-07	5.9258e-07
	ω_{opt}	1.0	1.0	1.0	1.0
	α_{opt}	1.1	1.2	1.1	1.2
	IT	7	7	7	7
FPI-PSS	CPU	0.0590	0.2978	0.7214	1.2295
	RES	4.0899e-07	3.2193e-07	8.0123e-07	4.0477e-07

is a block-tridiagonal matrix,

$$S = \text{tridiag}(-1.5, 8, -0.5) = \begin{bmatrix} 8 & -0.5 & 0 & \dots & 0 & 0 \\ -1.5 & 8 & -0.5 & \dots & 0 & 0 \\ 0 & -1.5 & 8 & \dots & 0 & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & 8 & -0.5 \\ 0 & 0 & \dots & \dots & -1.5 & 8 \end{bmatrix} \in \mathbb{R}^{m \times m}$$

is a tridiagonal matrix, $n = m^2$. Let $x^* = (-1, 1, -1, \dots, -1, 1, \dots)^T \in \mathbb{R}^n$ be the exact solution of the AVE (1).

Table 5 presents the numerical results of the FPI-PSS method and two methods in [1], where Method I with parameter 1 and Method II with parameter 0.97, see [1] for more details. From Table 5, we can see that the number of iteration steps for Method I is the same as the FPI-PSS method, but the FPI-PSS method requires less time than other two methods to achieve the terminated criterion. Thus, the proposed FPI-PSS method is more effective and feasible for solving the AVE (1).

At the end of this section, we give the following remark. From the numerical results of this section, we can see that the numerical optimal parameter ω of the FPI-PSS method is $\omega_{opt} = 1$ in three tested examples with different problem scales. If $\omega = 1$, the iterative scheme of the FPI-PSS method becomes

$$x^{(k+1)} = (\alpha P + A)^{-1}(\alpha P - A)x^{(k)} + (\alpha P + A)^{-1}(|x^{(k)}| + b),$$

which is an inexact Picard method for solving the AVE (1).

Table 5 Numerical results for Example 3 with $\mu = 4$

Method	n	20^2	30^2	40^2	50^2
Method I	IT	9	9	9	9
	CPU	0.0637	0.1975	0.7459	2.0811
	RES	2.3228e-07	3.5240e-07	4.7177e-07	5.9086e-07
Method II	IT	10	10	10	10
	CPU	0.0761	0.4914	2.0010	6.3861
	RES	2.3109e-07	3.5759e-07	4.8409e-07	6.1059e-07
FPI-PSS	ω_{opt}	1.0	1.0	1.0	1.0
	α_{opt}	0.9	0.9	0.9	0.9
	IT	9	9	9	9
	CPU	0.0092	0.0196	0.0432	0.0693
	RES	2.8805e-07	4.4070e-07	5.9347e-07	7.4628e-07

5 Conclusions

In this paper, we propose an inexact fixed point iteration method, termed as FPI-PSS method, to solve the absolute value equation. The FPI-PSS method is constructed by combining the preconditioned shift-splitting iteration method with the fixed point iteration method. Some convergence conditions of FPI-PSS method are given. In addition, three examples show that the FPI-PSS method is superior to other comparison methods from the aspects of iteration steps and computing times. However, how to choose the optimal involved parameters in the FPI-PSS method need further study.

Acknowledgements The author thank editor and the anonymous referees for their constructive suggestions and helpful comments, which greatly improved the quality of this paper. First author is supported by the Excellent Postgraduate Innovation Star Scientific Research Project of Gansu Province (No. 2023CXZX-327).

Data availability Data sharing is not applicable to this article as no datasets were generated or analyzed in this study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ali, R., Pan, K.-J.: Two new fixed point iteration schemes for absolute value equations. *Jpn. J. Ind. Appl. Math.* **40**, 303–314 (2023)
2. Bai, Z.-Z., Yin, J.-F., Su, Y.-F.: A shift-splitting preconditioner for non-Hermitian positive definite matrices. *J. Comput. Math.* **24**, 539–552 (2006)
3. Cottle, R.W., Pang, J.-S., Stone, R.E.: *The Linear Complementarity Problem*. Academic Press (1992)
4. Cui, L.-B., Hu, Q.: A chord-Zhang neural network model for solving absolute value equations. *Pac. J. Optim.* **18**, 77–89 (2022)

5. Dou, Y., Yang, A.-L., Wu, Y.-J.: A new Uzawa-type iteration method for non-Hermitian saddle-point problems. *East Asian J. Appl. Math.* **7**, 211–226 (2017)
6. Guo, P., Wu, S.-L., Li, C.-X.: On the SOR-like iteration method for solving absolute value equations. *Appl. Math. Lett.* **97**, 107–113 (2019)
7. Hladik, M., Moosaei, H.: Some notes on the solvability conditions for absolute value equations. *Optim. Lett.* **17**, 211–218 (2023)
8. Ke, Y.-F., Ma, C.-F.: SOR-like iteration method for solving absolute value equations. *Appl. Math. Comput.* **311**, 195–202 (2017)
9. Ke, Y.-F.: The new iteration algorithm for absolute value equation. *Appl. Math. Lett.* **99**, 105990 (2020)
10. Ketabchi, S., Moosaei, H.: An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side. *Comput. Math. Appl.* **64**, 1882–1885 (2012)
11. Li, X., Li, Y.-X., Dou, Y.: Shift-splitting fixed point iteration method for solving generalized absolute value equations. *Numer. Algorithms.* **93**, 695–710 (2023)
12. Mangasarian, O.L.: Absolute value programming. *Comput. Optim. Appl.* **36**, 43–53 (2007)
13. Mansoori, A., Eshaghnezhad, M., Effati, S.: An efficient neural network model for solving the absolute value equations. *IEEE Trans. Circuits Syst. II Express Briefs* **65**, 391–395 (2018)
14. Mangasarian, O.L.: Absolute value equation solution via concave minimization. *Optim. Lett.* **1**, 3–8 (2007)
15. Noor, M.A., Iqbal, J., Noor, K.I., Al-Said, E.: On an iterative method for solving absolute value equations. *Optim. Lett.* **6**, 1027–1033 (2012)
16. Rohn, J.: A theorem of the alternatives for the equation $Ax + B|x| = b$. *Linear Multilinear Algebra* **52**, 421–426 (2004)
17. Salkuyeh, D.K.: The Picard-HSS iteration method for absolute value equations. *Optim. Lett.* **8**, 2191–2202 (2014)
18. Wu, S.-L., Guo, P.: Modulus-based matrix splitting algorithms for the quasi-complementarity problems. *Appl. Numer. Math.* **132**, 127–137 (2018)
19. Wu, S.-L., Li, C.-X.: The unique solution of the absolute value equations. *Appl. Math. Lett.* **76**, 195–200 (2018)
20. Wang, A., Cao, Y., Chen, J.-X.: Modified Newton-type iteration methods for generalized absolute value equations. *J. Optim. Theory Appl.* **181**, 216–230 (2019)
21. Yu, D.-M., Chen, C.-R., Han, D.-R.: A modified fixed point iteration method for solving the system of absolute value equations. *Optimization* **71**, 449–461 (2022)
22. Young, D.M.: *Iterative Solution of Large Linear Systems*. Academic Press (1971)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.