Numerical Algorithms (2023) 93:695–710
https://doi.org/10.1007/s11075-022-01435-3

ORIGINAL PAPER

Shift-splitting fixed point iteration method for solving generalized absolute value equations

Xu Li[1] · Yi-Xin Li[1] · Yan Dou[1]

Received: 11 August 2022 / Accepted: 5 October 2022 / Published online: 25 October 2022

Abstract
Using the shift-splitting strategy, we propose a shift-splitting fixed point iteration (FPI-SS) method for solving large sparse generalized absolute value equations (GAVEs). The FPI-SS method is based on reformulating the GAVE as a two-by-two block nonlinear equation. Several different types of convergence conditions of the FPI-SS method are presented under suitable restrictions. Through numerical experiments, we demonstrate that the FPI-SS method is superior to the fixed point iteration method and the SOR-like iteration method in computing efficiency.

Keywords Generalized absolute value equation · Shift-splitting · Fixed point iteration · Convergence analysis

Mathematics Subject Classification 2010 65F10 · 65H10 · 90C05 · 90C30

1 Introduction

The generalized absolute value equation (GAVE) is formulated as:

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

where A, $B \in \mathbb{R}^{n\times n}$ are given large sparse matrices, $b \in \mathbb{R}^n$, and $|x| = (|x_1|, \ldots, |x_n|)^{\mathrm{T}} \in \mathbb{R}^n$ denotes the componentwise absolute value of an unknown $x \in \mathbb{R}^n$. If $B = I$, where I stands for an identity matrix of suitable dimension, the GAVE (1) can be simplified to the following absolute value equation (AVE)

✉ Xu Li
lixu@lut.edu.cn

Yi-Xin Li
leeyx11@163.com

Yan Dou
douy@lut.edu.cn

[1] Department of Applied Mathematics, Lanzhou University of Technology, 730050 Lanzhou, People's Republic of China

$$Ax - |x| = b. \tag{2}$$

GAVEs have arisen in various scientific and engineering fields and been presented in enormous applications since they were first introduced by Rohn [1]. Among many important applications, a well-known example is the linear complementarity problem (LCP) [2–6]. Besides LCPs, many other optimization problems can be transformed into the GAVEs (1), including linear programming and convex quadratic programming [1, 7].

Due to the existence of the nonlinear term $B|x|$, the GAVE (1) can be regarded as a weakly nonlinear system

$$Ax = G(x), \quad \text{with} \quad G(x) = B|x| + b. \tag{3}$$

For solving the general weakly nonlinear systems

$$Ax = G(x), \tag{4}$$

where the nonlinear function $G : \mathbb{R}^n \to \mathbb{R}^n$ is B-differentiable, through the two-stage splitting $A = E - F$ and $E = M - N$, Bai for the first time introduced and studied the following two-stage iterative method [8]:

$$Mx^{(k,\ell+1)} = Nx^{(k,\ell)} + Fx^{(k)} + G(x^{(k)}), \quad \text{for} \quad \ell = 0, 1, \dots, l_k - 1, \tag{5}$$

with $x^{(k,0)} := x^{(k)}$ and $x^{(k+1)} := x^{(k,l_k)}$. See also [9–11] for related methods. It is noted that the two-stage iterative method provides a general framework of matrix splitting iteration methods for solving the weakly nonlinear systems (4). For the GAVE (1), i.e., the case when $G(x) = B|x| + b$, the two-stage iterative method includes a series of existing matrix splitting iteration methods [12–16] as its special cases. For example, when $E = A$, $F = 0$, $M = E$, $N = 0$, and $l_k \equiv 1$, the two-stage iterative method reduces to the well-known Picard iteration method [12]

$$Ax^{(k+1)} = B|x^{(k)}| + b. \tag{6}$$

Recently, by reformulating the AVE (2) as a two-by-two block nonlinear equation, Ke et al. proposed an SOR-like iteration method [17] for solving the AVE (2). This method was also analyzed in [18]. The SOR-like iteration method received wide attentions and obtained considerable achievements in recent years. Using the similar technology, other SOR-like-based methods [19–21] are presented to solve the AVE (2). In order to further improve computational efficiency, Ke proposed an efficient fixed point iteration (FPI) method [22] to solve the AVE (2), which can be described as

Algorithm 1 (The FPI Method for AVE). Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular matrix and $b \in \mathbb{R}^n$. Given the initial vectors $x^{(0)}, y^{(0)} \in \mathbb{R}^n$, 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:

$$\begin{cases} x^{(k+1)} = A^{-1}(y^{(k)} + b), \\ y^{(k+1)} = (1-\omega)y^{(k)} + \omega|x^{(k+1)}|, \end{cases} \tag{7}$$

where ω is a positive constant.

Note that the FPI method reduces to the Picard iteration method for $\omega = 1$. Owing to the simplicity and effectiveness of FPI method for solving the AVE (2), Yu et al. developed a modified FPI (MFPI) method [23], which is a generalized version of the FPI method.

Clearly, at each step of the FPI method, a linear system $Au = f$ needs to be solved. Since A is always large and sparse, a computationally efficient way is to use matrix splitting iteration methods to obtain the approximate solution of this linear system. For solving non-Hermitian positive definite linear systems, Bai et al. first proposed the shift-splitting (SS) iteration method [24]. Motivated by its promising performance, the SS method was extended to solve many linear systems with special structure such as the saddle point problems [25], block 3×3 saddle point problems [26], and time-harmonic eddy current problems [27]. In this paper, using the shift-splitting [24] of the coefficient matrix A, we propose a shift-splitting fixed point iteration (FPI-SS) method for solving the GAVE (1). Compared with the FPI method, the coefficient matrix of the first sub-iteration scheme of our method is more diagonally dominant. Our method is more efficient than the FPI method and the SOR-like iteration method as shown in our numerical experiments.

In what follows, some notations in this work are described. For $x \in \mathbb{R}^n$, x_i stands for the ith entry of vector x for all $i = 1, 2, \ldots, n$. $\mathrm{sgn}(x) \in \mathbb{R}^n$ denotes a vector with components equal to 1, 0, or -1 depending on whether the corresponding component of the vector x is positive, zero, or negative, respectively. Let $\mathrm{diag}(x) \in \mathbb{R}^{n\times n}$ represent a diagonal matrix with x_i as its ith diagonal entry for $i = 1, 2, \ldots, n$. For matrix $M \in \mathbb{R}^{n\times n}$, $\|M\|$ denotes the spectral norm defined by $\|M\| := \max\{\|Mx\| : x \in \mathbb{R}^n, \|x\| = 1\}$, where $\|x\|$ is the 2-norm.

The organization of the remaining parts is the following. In Section 2, we present a brief introduction of the FPI method and establish the FPI-SS method for solving the GAVE (1). In Section 3, the convergence theories for the FPI-SS method are presented in detail. In Section 4, we give two numerical examples in Section 4 to verify the effectiveness of our method. Finally, the conclusions are given in Section 5.

2 The shift-splitting fixed point iteration (FPI-SS) method

Let $y = |x|$, then the GAVE (1) is equivalent to

$$\begin{cases} Ax - By = b, \\ -|x| + y = 0, \end{cases} \tag{8}$$

which can be reformulated as the following two-by-two block nonlinear equation

$$\begin{pmatrix} A & -B \\ -H(x) & I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \tag{9}$$

where $H(x) = \mathrm{diag}(\mathrm{sign}(x))$.

If A is a nonsingular matrix, (9) yields the following fixed point equation

$$\begin{cases} x^* = A^{-1}(By^* + b), \\ y^* = (1-\omega)y^* + \omega|x^*|, \end{cases} \tag{10}$$

where the relaxation parameter $\omega > 0$.

Then, we can obtain the following fixed point iteration (FPI) method for the GAVE (1).

Algorithm 2 (The FPI Method for GAVE). Let $A \in \mathbb{R}^{n\times n}$ be nonsingular, $B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Given the initial vectors $x^{(0)}, y^{(0)} \in \mathbb{R}^n$, 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:

$$\begin{cases} x^{(k+1)} = A^{-1}(By^{(k)} + b), \\ y^{(k+1)} = (1-\omega)y^{(k)} + \omega|x^{(k+1)}|, \end{cases} \tag{11}$$

where ω is a positive constant.

It is evident that Algorithm 2 reduces to Algorithm 1 when we take $B = I$. Similarly, if we set $\omega = 1$ in Algorithm 2, the Picard iteration method (6) for solving the GAVE (1) can be obtained. Since the convergence analyses of Algorithm 2 are analogous to those of Algorithm 1 discussed in detail in [22], we do not give them here.

Importantly, by employing the following shift-splitting of the matrix A [24]

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

where the parameter α is a positive constant and the matrix $\alpha I + A$ is invertible, we get the following fixed point equation from (9)

$$\begin{cases} x^* = (\alpha I + A)^{-1}(\alpha I - A)x^* + 2(\alpha I + A)^{-1}(By^* + b), \\ y^* = (1-\omega)y^* + \omega|x^*|, \end{cases} \tag{12}$$

which leads to the following FPI-SS method for the GAVE (1).

Algorithm 3 (The FPI-SS Method for GAVE). Let A, $B \in \mathbb{R}^{n\times n}$ 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 $x^{(0)}, y^{(0)} \in \mathbb{R}^n$, 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:

$$\begin{cases} x^{(k+1)} = (\alpha I + A)^{-1}(\alpha I - A)x^{(k)} + 2(\alpha I + A)^{-1}(By^{(k)} + b), \\ y^{(k+1)} = (1-\omega)y^{(k)} + \omega|x^{(k+1)}|, \end{cases} \tag{13}$$

where ω is a positive constant.

Remark 1 If matrix A is positive semi-definite, the condition that $\alpha I + A$ is nonsingular naturally holds. Even if matrix A is singular, we can always find some sufficiently large parameters α to ensure that $\alpha I + A$ is nonsingular. Therefore, the FPI-SS method has a broader range of application than the FPI method. In addition, owing to the positive scalar matrix αI, the matrix $\alpha I + A$ is expected to be strictly diagonally dominant and better conditioned than the matrix A. Thus, our FPI-SS method may have better computing efficiency than the FPI method.

3 Convergence of the FPI-SS method

We first give some lemmas that will be used in convergence analysis of the FPI-SS method for solving the GAVE (1).

Lemma 1 [28–30] *For any vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, the following results hold*:

(1) $\||x| - |y|\| \le \|x - y\|$;
(2) *if* $0 \le x \le y$, *then* $\|x\|_p \le \|y\|_p$, *with* $\|\cdot\|_p$ *standing for p-norm of vector*;
(3) *if* $x \le y$ *and* P *is a nonnegative matrix*, *then* $Px \le Py$.

Lemma 2 [28, 29] *For any matrices $A, B \in \mathbb{R}^{n\times n}$, if $0 \le A \le B$, then $\|A\|_p \le \|B\|_p$, with $\|\cdot\|_p$ standing for p-norm of matrix.*

Lemma 3 [28, 31] *Both roots of the real quadratic equation $x^2 - ax + b = 0$ are less than one in modulus if and only if $|b| < 1$ and $|a| < 1 + b$.*

In the remainder of this section, we assume that the GAVE (1) has a unique solution. Let (x^*, y^*) be the solution pair of (12) and $(x^{(k)}, y^{(k)})$ be generated by the FPI-SS iteration (13). The iteration errors are denoted by

$$e_k^x = x^* - x^{(k)} \quad \text{and} \quad e_k^y = y^* - y^{(k)}.$$

Then, we can get the following convergence theorem by estimating the above two iteration errors.

Theorem 1 *Let $A, B \in \mathbb{R}^{n\times n}$ 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. Denote*

$$\delta = \|(\alpha I + A)^{-1}(\alpha I - A)\|, \beta = 2\|(\alpha I + A)^{-1}B\|, \gamma = |1 - \omega|,$$

and

$$E^{(k+1)} = \begin{pmatrix} \|e^x_{k+1}\| \\ \|e^y_{k+1}\| \end{pmatrix}.$$

Then, we have

$$\|E^{(k+1)}\|_\infty \leq \|L(\alpha, \omega)\|_\infty \cdot \|E^{(k)}\|_\infty, \tag{14}$$

where $\|\cdot\|_\infty$ *denotes the* ∞*-norm of vector or matrix and*

$$L(\alpha, \omega) := \begin{pmatrix} \delta & \beta \\ \omega\delta & \omega\beta + \gamma \end{pmatrix}.$$

Furthermore, $\|L(\alpha, \omega)\|_\infty < 1$ *if and only if parameters* α *and* ω *satisfy*

$$\delta + \beta < 1 \quad and \quad 0 < \omega < \frac{2}{1 + \delta + \beta}, \tag{15}$$

i.e., if the conditions (15) hold, the iteration sequence $\{x^{(k)}\}_{k=0}^{+\infty}$ *generated by the FPI-SS iteration converges to the unique solution* x^* *of the GAVE (1) for any initial vector.*

Proof Subtracting (13) from (12), we get

$$e^x_{k+1} = (\alpha I + A)^{-1}(\alpha I - A)e^x_k + 2(\alpha I + A)^{-1}Be^y_k, \tag{16}$$

$$e^y_{k+1} = (1 - \omega)e^y_k + \omega(|x^*| - |x^{(k+1)}|). \tag{17}$$

According to (16), we can obtain

$$\|e^x_{k+1}\| \leq \delta\|e^x_k\| + \beta\|e^y_k\|. \tag{18}$$

From (17) and Lemma 1, we have

$$\begin{aligned} \|e^y_{k+1}\| \leq& \gamma \cdot \|e^y_k\| + \omega\||x^*| - |x^{(k+1)}|\| \\ \leq& \gamma \cdot \|e^y_k\| + \omega\|x^* - x^{(k+1)}\| \\ =& \gamma \cdot \|e^y_k\| + \omega\|e^x_{k+1}\|. \end{aligned} \tag{19}$$

Rearranging (18) and (19), we find

$$\begin{pmatrix} 1 & 0 \\ -\omega & 1 \end{pmatrix} \begin{pmatrix} \|e^x_{k+1}\| \\ \|e^y_{k+1}\| \end{pmatrix} \leq \begin{pmatrix} \delta & \beta \\ 0 & \gamma \end{pmatrix} \begin{pmatrix} \|e^x_k\| \\ \|e^y_k\| \end{pmatrix}. \tag{20}$$

Let

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

Multiplying (20) from left by the nonnegative matrix P and according to Lemma 1, we have

$$\begin{pmatrix} \|e^x_{k+1}\| \\ \|e^y_{k+1}\| \end{pmatrix} \leq \begin{pmatrix} \delta & \beta \\ \omega\delta & \omega\beta+\gamma \end{pmatrix} \begin{pmatrix} \|e^x_k\| \\ \|e^y_k\| \end{pmatrix}, \tag{21}$$

which can be rewritten as

$$E^{(k+1)} \leq L(\alpha,\omega)\cdot E^{(k)}. \tag{22}$$

Taking the ∞-norm on both sides of inequality (22) and according to (2) of Lemma 1, the estimation (14) is obtained. Since

$$\|L(\alpha,\omega)\|_\infty = \max\{\delta+\beta, (\delta+\beta)\omega+\gamma\},$$

we have

$$\begin{aligned} \|L(\alpha,\omega)\|_\infty < 1 &\Leftrightarrow \begin{cases} \delta+\beta<1 \\ (\delta+\beta)\omega+\gamma<1 \end{cases} \Leftrightarrow \begin{cases} \delta+\beta<1 \\ |1-\omega|<1-(\delta+\beta)\omega \end{cases} \\ &\Leftrightarrow \begin{cases} \delta+\beta<1 \\ 1-(\delta+\beta)\omega>0 \\ (\delta+\beta)\omega-1<1-\omega<1-(\delta+\beta)\omega \end{cases} \Leftrightarrow \begin{cases} \delta+\beta<1 \\ \omega<\frac{1}{\delta+\beta} \\ 0<\omega<\frac{2}{1+\delta+\beta} \end{cases} \\ &\Leftrightarrow \begin{cases} \delta+\beta<1 \\ 0<\omega<\frac{2}{1+\delta+\beta}. \end{cases} \end{aligned}$$

From (14), we deduce that

$$0 \leq \|E^{(k)}\|_\infty \leq \|L(\alpha,\omega)\|_\infty \cdot \|E^{(k-1)}\|_\infty \leq \cdots \leq \|L(\alpha,\omega)\|^k_\infty \cdot \|E^{(0)}\|_\infty.$$

Hence if the conditions (15) are satisfied, then we have $\lim\limits_{k\to\infty} \|E^{(k)}\|_\infty = 0$.

As

$$\|E^{(k)}\|_\infty = \max\{\|e^x_k\|, \|e^y_k\|\},$$

it follows that

$$\lim_{k\to\infty} \|e^x_k\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|e^y_k\| = 0,$$

which mean that the iteration sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{+\infty}$ is convergent to (x^*, y^*) under the conditions (15). This proves the theorem.

Using a different error estimate by a new weighted norm, we can obtain another convergence theorem as follows.

Theorem 2 *Let the assumptions of Theorem* 1 *hold*, δ, β, *and* γ *be defined as in Theorem* 1. *Denote*

$$E_{\omega}^{(k+1)} = \begin{pmatrix} \|e_{k+1}^{x}\| \\ \omega^{-1}\|e_{k+1}^{y}\| \end{pmatrix}.$$

Then, we have

$$\|E_{\omega}^{(k+1)}\| \leq \|T(\alpha,\omega)\| \cdot \|E_{\omega}^{(k)}\|, \tag{23}$$

where

$$T(\alpha,\omega) := \begin{pmatrix} \delta & \omega\beta \\ \delta & \omega\beta+\gamma \end{pmatrix}.$$

Furthermore, $\|T(\alpha,\omega)\| < 1$ *if and only if parameters* α *and* ω *satisfy*

$$\delta\gamma < 1 \tag{24}$$

and

$$2\delta^2 + \omega^2\beta^2 + (\omega\beta+\gamma)^2 - \delta^2\gamma^2 - 1 < 0, \tag{25}$$

i.e., if the conditions (24)–(25) hold, the iteration sequence $\{x^{(k)}\}_{k=0}^{+\infty}$ *generated by the FPI-SS iteration converges to the unique solution* x^{*} *of the GAVE (1) for any initial vector.*

Proof Denote

$$D = \begin{pmatrix} 1 & 0 \\ 0 & \omega^{-1} \end{pmatrix} > 0.$$

According to Lemma 1, we multiply left (21) by matrix D to obtain

$$\begin{pmatrix} \|e_{k+1}^{x}\| \\ \omega^{-1}\|e_{k+1}^{y}\| \end{pmatrix} \leq \begin{pmatrix} \delta & \omega\beta \\ \delta & \omega\beta+\gamma \end{pmatrix} \begin{pmatrix} \|e_{k}^{x}\| \\ \omega^{-1}\|e_{k}^{y}\| \end{pmatrix},$$

which can be rewritten as

$$E_{\omega}^{(k+1)} \leq T(\alpha,\omega) \cdot E_{\omega}^{(k)}.$$

From the above, it follows that (23) holds.

Let λ be an eigenvalue of the matrix $Q := T(\alpha,\omega)^{\mathrm{T}} T(\alpha,\omega)$. Since

$$Q = \begin{pmatrix} 2\delta^2 & 2\omega\beta\delta + \delta\gamma \\ 2\omega\beta\delta + \delta\gamma & \omega^2\beta^2 + (\omega\beta+\gamma)^2 \end{pmatrix},$$

we get

$$\mathrm{tr}(Q) = 2\delta^2 + \omega^2\beta^2 + (\omega\beta+\gamma)^2$$

and

$$\det(Q) = \delta^2\gamma^2.$$

Thus, λ is the root of the following real quadratic equation

$$\lambda^2 - (2\delta^2 + \omega^2\beta^2 + (\omega\beta + \gamma)^2)\lambda + \delta^2\gamma^2 = 0. \tag{26}$$

From Lemma 3, it follows that $\|T(\alpha, \omega)\| < 1$ if and only if

$$\delta^2\gamma^2 < 1,$$

and

$$2\delta^2 + \omega^2\beta^2 + (\omega\beta + \gamma)^2 < 1 + \delta^2\gamma^2.$$

From (23), we conclude that

$$0 \le \|E_\omega^{(k)}\| \le \|T(\alpha, \omega)\| \cdot \|E_\omega^{(k-1)}\| \le \cdots \le \|T(\alpha, \omega)\|^k \cdot \|E_\omega^{(0)}\|.$$

Hence, we have $\lim_{k\to\infty} \|E_\omega^{(k)}\| = 0$ when the conditons (24)–(25) are satisfied.

From the definition

$$\|E_\omega^{(k)}\| = \sqrt{\|e_k^x\|^2 + \omega^{-2}\|e_k^y\|^2},$$

we get

$$\lim_{k\to\infty} \|e_k^x\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|e_k^y\| = 0,$$

which mean that the iteration sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{+\infty}$ is convergent to (x^*, y^*) under the conditions (24)–(25). This completes the proof.

Theorem 2 shows that in order to obtain the convergence of the FPI-SS method, we need to find the conditions in which $\|T(\alpha, \omega)\| < 1$ holds. Here, we give convergence conditions that are simpler than those in Theorem 2.

Corollary 1 *Let the assumptions of Theorem* 1 *hold*, δ, β, *and* γ *be defined as in Theorem* 1, $E_\omega^{(k+1)}$ *and* $T(\alpha, \omega)$ *be defined as in Theorem* 2. *If*

$$\delta < \frac{3-\sqrt{5}}{2},\ \beta < \frac{\sqrt{5}-1}{2}, \tag{27}$$

and

$$\frac{\sqrt{5}-1}{2} < \omega < \min\{\frac{3-\sqrt{5}}{2\beta},\ \frac{5-\sqrt{5}}{2}\}, \tag{28}$$

then $\|T(\alpha, \omega)\| < 1$, *i.e., the FPI-SS method is convergent when the conditions* (27)–(28) *hold.*

Proof Let $\eta = \max\{\delta,\ \omega\beta,\ \gamma\}$, we can get

$$0 \leq T(\alpha,\omega) = \begin{pmatrix} \delta & \omega\beta \\ \delta & \omega\beta+\gamma \end{pmatrix} \leq \begin{pmatrix} \eta & \eta \\ \eta & 2\eta \end{pmatrix} = \eta\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} := \eta K,$$

where

$$K = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

From lemma 2, we obtain

$$\|T(\alpha,\omega)\| \leq \|\eta K\| = \eta\|K\| = \eta \cdot \frac{3+\sqrt{5}}{2}.$$

Let $\theta = \dfrac{3-\sqrt{5}}{2}$. Hence, we have $\|T(\alpha,\omega)\| < 1$ if $\eta < \theta$. Then,

$$\eta < \theta \Leftrightarrow \begin{cases} \delta < \theta \\ \omega\beta < \theta \\ \gamma = |1-\omega| < \theta \end{cases} \Leftrightarrow \begin{cases} \delta < \theta \\ \omega < \frac{\theta}{\beta} \\ 1-\theta < \omega < 1+\theta \end{cases}$$

$$\Leftrightarrow \begin{cases} \delta < \theta \\ 1-\theta < \omega < \min\{\frac{\theta}{\beta},\ 1+\theta\} \\ 1-\theta < \frac{\theta}{\beta} \end{cases}$$

$$\Leftrightarrow \begin{cases} \delta < \theta \\ \frac{\sqrt{5}-1}{2} < \omega < \min\{\frac{3-\sqrt{5}}{2\beta},\ \frac{5-\sqrt{5}}{2}\}. \\ \beta < \frac{\theta}{1-\theta} = \frac{\sqrt{5}-1}{2} \end{cases}$$

Therefore, if the conditions (27)–(28) are satisfied, the iteration sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{+\infty}$ is convergent to (x^*, y^*).

4 Numerical experiments

In this section, two examples from LCPs are presented to show the feasibility and effectiveness of the FPI-SS method. We compare the FPI-SS method with the FPI method [22] and the SOR-like iteration method [17, 18] 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

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

In our implementation, all initial guess vectors $x^{(0)}$ and $y^{(0)}$ are chosen to zero vectors and all iterations are terminated if RES $\leq 10^{-6}$ or the maximum number of iteration steps $k_{\max}$ exceeds 500. All computations are performed in MATLAB R2018b on a personal computer with 2.40GHz central processing unit (Intel(R) Core(TM) i5-6200U) and 8 GB memory.

Consider the following LCP(q, M) [2]: to derive two real vectors $z, \omega \in \mathbb{R}^n$ such that

$$z \geq 0, \quad \omega = Mz + q \geq 0, \quad z^{\mathrm{T}}\omega = 0, \tag{29}$$

where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ are given. From [3–6], the LCP(q, M) (29) can be formulated as the following GAVE:

$$(M + I)x - (M - I)|x| = q, \tag{30}$$

with

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

Example 1 ([5, 6]) The matrix $M \in \mathbb{R}^{n\times n}$ is defined by $M = \widehat{M} + \mu I \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ is defined by $q = -Mz^*$, where

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

is a block-tridiagonal matrix,

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

is a tridiagonal matrix, $n = m^2$, and $z^* = (1, 2, 1, 2, \ldots, 1, 2, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the unique solution of the LCP(q, M) (29). It can be derived that $x^* = (-0.5, -1, -0.5, -1, \ldots, -0.5, -1, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the exact solution after formulating the LCP(q, M) (29) as the GAVE (30).

For various problem sizes n, the optimal experimental parameters, the iteration steps, CPU time, and relative residual errors of three methods in the case of $\mu = 1$ and $\mu = 4$ are listed in Tables 1 and 2, respectively.

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

Method	n	50^2	100^2	150^2	200^2
SOR-like	ω_{exp}	1.0	1.0	1.0	1.0
	IT	65	65	65	65
	CPU	0.2185	1.3265	3.1923	6.3418
	RES	8.1731e-07	9.3586e-07	9.7624e-07	9.9659e-07
FPI	ω_{exp}	1.9	1.9	1.9	1.9
	IT	32	32	32	32
	CPU	0.1028	0.6670	1.5182	3.0045
	RES	8.6927e-07	8.1065e-07	7.8688e-07	7.7407e-07
FPI-SS	ω_{exp}	1.3	1.4	1.3	1.3
	α_{exp}	3.3	4.6	3.0	3.3
	IT	16	16	16	15
	CPU	0.0435	0.2931	0.7731	1.5064
	RES	5.5429e-07	8.3983e-07	6.0380e-07	9.9518e-07

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

Method	n	50^2	100^2	150^2	200^2
SOR-like	ω_{exp}	1.0	1.0	1.0	1.0
	IT	27	27	27	27
	CPU	0.0881	0.5479	1.3626	2.8915
	RES	7.5622e-07	8.0353e-07	8.1946e-07	8.2746e-07
FPI	ω_{exp}	1.7	1.7	1.7	1.7
	IT	13	13	13	13
	CPU	0.0410	0.2745	0.6650	1.3447
	RES	5.0722e-07	5.4099e-07	5.5243e-07	5.5818e-07
FPI-SS	ω_{exp}	1.1	1.1	1.1	1.1
	α_{exp}	4.5	6.9	7.5	7.5
	IT	10	10	9	9
	CPU	0.0324	0.1889	0.4332	0.9821
	RES	9.3877e-07	4.6551e-07	9.7719e-07	9.5782e-07

We find that each tested method converges to the exact solution and the number of iterative steps becomes smaller with the increase of μ. Notably, among these methods, the FPI-SS method requires the least iteration steps and costs the least computing time.

Example 2 ([5]) Consider the LCP(q, M) (29). The matrix $M \in \mathbb{R}^{n\times n}$ is defined by $M = \widehat{M} + \mu I \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ is defined by $q = -Mz^*$, where

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

is a block-tridiagonal matrix,

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

is a tridiagonal matrix, $n = m^2$, and $z^* = (1, 2, 1, 2, \ldots, 1, 2, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the unique solution of the LCP(q, M) (29). It can be derived that $x^* = (-0.5, -1, -0.5, -1, \ldots, -0.5, -1, \ldots)^{\mathrm{T}} \in \mathbb{R}^n$ is the exact solution after formulating the LCP(q, M) (29) as the GAVE (30).

In Tables 3 and 4, we list the numerical results of three methods by using experimental optimal parameters in the case of $\mu = 1$ and $\mu = 4$, respectively. From those results, we get the same conclusions as Example 1.

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

Method	n	50^2	100^2	150^2	200^2
SOR-like	$\omega_{\exp}$	1.0	1.0	1.0	1.0
	IT	64	65	65	65
	CPU	0.7538	3.3268	8.2104	15.4580
	RES	8.5588e-07	8.7416e-07	9.3491e-07	9.6552e-07
FPI	$\omega_{\exp}$	1.8	1.9	1.9	1.9
	IT	33	32	32	32
	CPU	0.3669	1.6016	4.5292	7.5732
	RES	7.3824e-07	9.9926e-07	9.2406e-07	8.8200e-07
FPI-SS	$\omega_{\exp}$	1.2	1.2	1.2	1.2
	$\alpha_{\exp}$	2.2	2.3	2.3	2.4
	IT	22	21	21	21
	CPU	0.2111	0.9085	2.6978	5.0868
	RES	9.9926e-07	8.7943e-07	7.0811e-07	6.3176e-07

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

Method	n	50^2	100^2	150^2	200^2
SOR-like	$\omega_{\exp}$	1.0	1.0	1.0	1.0
	IT	27	27	27	27
	CPU	0.3075	1.3550	3.5566	6.9559
	RES	7.4004e-07	7.9538e-07	8.1402e-07	8.2338e-07
FPI	$\omega_{\exp}$	1.7	1.7	1.7	1.7
	IT	13	13	13	13
	CPU	0.1435	0.6567	1.6867	3.2636
	RES	5.0817e-07	5.4126e-07	5.5257e-07	5.5827e-07
FPI-SS	$\omega_{\exp}$	1.1	1.1	1.1	1.1
	$\alpha_{\exp}$	5.7	7.0	6.0	5.8
	IT	11	10	10	10
	CPU	0.1027	0.4260	1.3000	2.5901
	RES	4.4188e-07	8.4124e-07	8.2384e-07	9.0426e-07

5 Conclusion

In this paper, by combining the shift-splitting of the coefficient matrix with the fixed point iteration (FPI) method, we proposed a shift-splitting fixed point iteration (FPI-SS) method to solve the generalized absolute value equation (GAVE). We have given several different types of convergence conditions of the FPI-SS method by introducing two different norms of the iteration error. Furthermore, using two numerical examples from linear complementarity problems, we have demonstrated that the FPI-SS method outperforms the FPI method and the SOR-like iteration method in terms of iteration steps and computing times.

Finally, we should mention that the FPI-SS method can be seen as an inexact version of the FPI method. If we replace the shift-splitting in the FPI-SS algorithm with other matrix splitting such as SOR-based splitting [28, 29, 31] and HSS-based splitting [28, 32–34], we can establish a series of inexact FPI methods which may have similar convergence results. In real applications of inexact FPI algorithms, how to choose the optimal (or quasi-optimal) parameters is an interesting and practical topic, which is left as our future work.

Acknowledgements The authors would like to thank the two referees for their constructive suggestions which greatly improve the presentation of the paper. This research was funded by the Natural Science Foundation of Gansu Province (No. 20JR5RA464) and the National Natural Science Foundation of China (Nos. 11501272 and 11901267).

Author contribution Xu Li and Yi-Xin Li wrote the main manuscript text and Yi-Xin Li performed the numerical experiments. All authors reviewed the manuscript.

Funding The research of Xu Li was supported by the Natural Science Foundation of Gansu Province (No. 20JR5RA464) and the National Natural Science Foundation of China (No. 11501272). The research of Yi-Xin Li was supported by the Natural Science Foundation of Gansu Province (No. 20JR5RA464). The research of Yan Dou was supported by the National Natural Science Foundation of China (No. 11901267).

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

Declarations

Competing interests The authors declare no competing interests.

Ethics approval and consent to participate Not applicable.

Consent for publication The authors agree to publication of the article in English by Springer in Springer's corresponding English-language journal.

Human and animal ethics Not applicable.

References

1. Rohn, J.: A theorem of the alternatives for the equation $Ax + B|x| = b$. Linear Multilinear Algebra **52**(6), 421–426 (2004)
2. Cottle, R.W., Pang, J.-S., Stone, R.E.: The Linear Complementarity Problem. Academic Press, (1992)
3. Schäfer, U.: On the modulus algorithm for the linear complementarity problem. Oper. Res. Lett. **32**(4), 350–354 (2004)
4. Mangasarian, O.L., Meyer, R.R.: Absolute value equations. Linear Algebra Appl. **419**(2–3), 359–367 (2006)
5. Bai, Z.-Z.: Modulus-based matrix splitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **17**(6), 917–933 (2010)
6. Bai, Z.-Z., Zhang, L.-L.: Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **20**(3), 425–439 (2013)
7. Mangasarian, O.L.: Absolute value programming. Comput. Optim. Appl. **36**(1), 43–53 (2007)
8. Bai, Z.-Z.: A class of two-stage iterative methods for systems of weakly nonlinear equations. Numer. Algorithms **14**(4), 295–319 (1997)
9. Bai, Z.-Z.: Parallel multisplitting two-stage iterative methods for large sparse systems of weakly nonlinear equations. Numer. Algorithms **15**(3–4), 347–372 (1997)
10. Bai, Z.-Z., Migallón, V., Penadés, J., Szyld, D.B.: Block and asynchronous two-stage methods for mildly nonlinear systems. Numer. Math. **82**(1), 1–20 (1999)
11. Bai, Z.-Z., Yang, X.: On HSS-based iteration methods for weakly nonlinear systems. Appl. Numer. Math. **59**(12), 2923–2936 (2009)
12. Rohn, J., Hooshyarbakhsh, V., Farhadsefat, R.: An iterative method for solving absolute value equations and sufficient conditions for unique solvability. Optim. Lett. **8**(1), 35–44 (2014)
13. Wang, A., Cao, Y., Chen, J.-X.: Modified Newton-type iteration methods for generalized absolute value equations. J. Optim. Theory Appl. **181**(1), 216–230 (2019)
14. Zhou, H.-Y., Wu, S.-L., Li, C.-X.: Newton-based matrix splitting method for generalized absolute value equation. J. Comput. Appl. Math. **394**, 113578–15 (2021)
15. Li, C.-X., Wu, S.-L.: A shift splitting iteration method for generalized absolute value equations. Comput. Methods Appl. Math. **21**(4), 863–872 (2021)
16. Salkuyeh, D.K.: The Picard-HSS iteration method for absolute value equations. Optim. Lett. **8**(8), 2191–2202 (2014)
17. Ke, Y.-F., Ma, C.-F.: SOR-like iteration method for solving absolute value equations. Appl. Math. Comput. **311**, 195–202 (2017)
18. 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)
19. Huang, B., Li, W.: A modified SOR-like method for absolute value equations associated with second order cones. J. Comput. Appl. Math. **400**, 113745–20 (2022)
20. Dong, X., Shao, X.-H., Shen, H.-L.: A new SOR-like method for solving absolute value equations. Appl. Numer. Math. **156**, 410–421 (2020)

21. Zhang, J.-L., Zhang, G.-F., Liang, Z.-Z.: A modified generalized SOR-like method for solving an absolute value equation. Linear Multilinear Algebra (2022). https://doi.org/10.1080/03081087.2022.2066614
22. Ke, Y.-F.: The new iteration algorithm for absolute value equation. Appl. Math. Lett. **99**, 105990–7 (2020)
23. Yu, D., Chen, C., Han, D.: A modified fixed point iteration method for solving the system of absolute value equations. Optimization **71**(3), 449–461 (2022)
24. Bai, Z.-Z., Yin, J.-F., Su, Y.-F.: A shift-splitting preconditioner for non-Hermitian positive definite matrices. J. Comput. Math. **24**(4), 539–552 (2006)
25. Cao, Y., Du, J., Niu, Q.: Shift-splitting preconditioners for saddle point problems. J. Comput. Appl. Math. **272**, 239–250 (2014)
26. Cao, Y.: Shift-splitting preconditioners for a class of block three-by-three saddle point problems. Appl. Math. Lett. **96**, 40–46 (2019)
27. Cao, Y.: A general class of shift-splitting preconditioners for non-Hermitian saddle point problems with applications to time-harmonic eddy current models. Comput. Math. Appl. **77**(4), 1124–1143 (2019)
28. Bai, Z.-Z., Pan, J.-Y.: Matrix Analysis and Computations. SIAM, (2021)
29. Varga, R.S.: Matrix Iterative Analysis. Springer, (2000)
30. Bai, Z.-Z., Evans, D.J.: Matrix multisplitting relaxation methods for linear complementarity problems. Int. J. Comput. Math. **63**(3–4), 309–326 (1997)
31. Young, D.M.: Iterative Solution of Large Linear Systems. Academic Press, (1971)
32. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**(3), 603–626 (2003)
33. Bai, Z.-Z., Golub, G.H., Ng, M.K.: On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. Numer. Linear Algebra Appl. **14**(4), 319–335 (2007)
34. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput. **26**(3), 844–863 (2005)

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