Computational and Applied Mathematics (2024) 43:313
https://doi.org/10.1007/s40314-024-02811-6

New iterative methods for solving generalized absolute value equations

Ghodrat Ebadi[1] · **Somayeh Seifollahzadeh**[1] · **Cornelis Vuik**[2]

Received: 11 January 2024 / Revised: 2 June 2024 / Accepted: 3 June 2024
© The Author(s) under exclusive licence to Sociedade Brasileira de Matemática Aplicada e Computacional 2024

Abstract
In this paper, two new iterative methods for solving generalized absolute value equations (GAVEs) are proposed and investigated using the single-step iteration (SSI) approach. The proposed iterative methods are Picard-SSI and nonlinear SSI-like methods. In the implementation of the Picard-SSI method, we have used the SSI method as an inner solver. The convergence of the proposed method for solving GAVE is analyzed under reasonable constraints. Several numerical examples are given to illustrate the efficiency and implementation of the proposed methods.

Keywords Absolute value equation · Picard-SSI method · Nonlinear SSI-like method · SSI method

Mathematics Subject Classification 65F10 · 65B99 · 65F30

1 Introduction

For given matrices $A, B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, let's consider the problem of finding a vector $x \in \mathbb{R}^n$ that satisfies the equation:

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

This problem, known as the Generalized Absolute Value Equations (GAVE), was introduced by Rohn (2004). In the special case where $B = I$, the equation reduces to:

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

✉ Ghodrat Ebadi
gebadi@tabrizu.ac.ir

Somayeh Seifollahzadeh
s.sifaleh@tabrizu.ac.ir

Cornelis Vuik
c.vuik@tudelft.nl

1 Faculty of mathematics, Statistics and Computer Science, University of Tabriz, Tabriz, Iran

2 Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

The GAVE finds applications in various fields including engineering, scientific computing, and game theory. For instance, bimatrix games, linear programs, and convex quadratic programs can be formulated as a linear complementarity problem (LCP) and can also be reduced to a GAVE (Mangasarian et al. 2006; Cottle et al. 1992). The GAVE can also be reduced to an LCP (Murty 1988; Prokopyev 2009). Additionally, the GAVE is a special case of the system of weakly nonlinear equations studied by Bai (1997). In Bai (1997), a two-stage iterative method is introduced, providing a comprehensive framework for applying matrix splitting iteration techniques to weakly nonlinear systems, which includes the methods presented in (Salkuyeh 2014; Miao et al. 2021; Zhou et al. 2021) as specific instances. For a deeper understanding of the interconnections among these approaches, refer to (Li et al. 2023).

In recent years, significant research has been dedicated to solving GAVEs. Several methods have been developed, and some studies (Mangasarian et al. 2006; Rohn 2009; Wu 2020) have established sufficient conditions for the existence and uniqueness of solutions to the GAVE and its special case, the AVE. The generalized Newton (GN) method was suggested by Mangasarian (2009) for solving AVE, and further generalizations of the GN method can be found in (Zhou et al. 2021; Noor et al. 2018; Wang et al. 2019).

The Picard-HSS method, proposed by Salkuyeh (2014), and the nonlinear HSS-like iteration method, put forth by Zhu and Qi (2018), were initially designed for weakly nonlinear problems in (Bai and Yang 2009). Other numerical methods based on the Picard-type method and generalizations of the nonlinear HSS-like method can be found in (Dehghan and Shirilord 2020; Miao et al. 2021; Zang 2015). Classical matrix-splitting iterative methods such as Gauss-Seidel, SOR, and AOR have also been utilized to solve GAVEs (Dong et al. 2020; Seifollahzadeh and Ebadi 2024).

When $B = 0$, the problem reduces to a system of linear equations, which has numerous applications in scientific computation (Bai et al. 2003; Idema and Vuik 2023; Ebadi et al. 2016; Li and Wu 2015; Golub and Van Loan 2013; Trefethen and Bau 1997).

The Hermitian and skew-Hermitian splitting (HSS) iteration method (Bai et al. 2003) is emphasized in this paper, which involves solving two linear subsystems. One of these subsystems is the shifted skew-Hermitian linear subsystem, which was studied by Idema and Vuik (2023). However, in some cases, solving one of the coefficient matrices of the linear subsystems, as it is skew-Hermitian, can be challenging (Benzi 2009). As a remedy to avoid solving a shifted skew-Hermitian linear subsystem, Li and Wu (2015) introduced the single-step HSS iteration method (SHSS) and Wu et al. (2017) presented a non-alternating preconditioned HSS (NPHSS) iteration method for non-Hermitian positive definite linear systems, and Wang et al. (2019) proposed a single-step iteration method for non-Hermitian positive definite linear systems, which is similar to NPHSS method and was referred to in Miao et al. (2020) with SSI.

In this article, we propose new Picard-SSI and nonlinear SSI-like methods for solving GAVEs based on the SSI technique. The Picard-SSI method combines the Picard and SSI methods as outer and inner iteration methods, respectively. It is worth mentioning that the method proposed by Miao et al. (2021) is a special case of our Picard-SSI method. Unlike the Picard-SSI method, the nonlinear SSI-like method does not require the computation of inner iterations. We provide convergence theorems for both methods and demonstrate their effectiveness through numerical examples.

The organization of the remainder of this paper is as follows: Some prerequisites are given in Sect. 2. The Picard-SSI and nonlinear SSI-like methods and their convergence are described in Sect. 3. Numerical examples to illustrate the effectiveness of the proposed methods are given in Sect. 4.

2 Preliminaries

In this section, we give necessary lemmas and a few notations which are used throughout the paper and recall the HSS iteration method (Bai et al. 2003), the SHSS iteration method (Li and Wu 2015) and the SSI method (Wang et al. 2019). We use $\sigma_i(.)$ and $\lambda_i(.)$ to denote the i-th singular value and eigenvalue of a matrix, respectively, especially $\sigma_{max}(.)(\sigma_{min}(.))$ and $\lambda_{max}(.)(\lambda_{min}(.))$ denotes the maximum (minimum) absolute singular values and maximum (minimum) absolute eigenvalues of the given matrix. $\rho(A)$ represents the spectral radius of given matrix A, I denotes the identity matrix, $\|.\|$ is the 2-norm, and $\otimes$ represents the Kronecker product.

Lemma 1 (Mangasarian et al. 2006) *The AVE* (2) *is uniquely solvable for any* b, *if* $\|A^{-1}\| < 1$.

Lemma 2 (Wu 2020) *The GAVE (2) has a unique solution for any vector* b, *if matrices* A *and* B *satisfy*

$$\sigma_{max}(B) < \sigma_{min}(A). \tag{3}$$

Within this document, it is presumed that the GAVE possesses a unique solution.

Bai et al. (2003) presented the following HSS method for solving $Ax = b$.

2.1 The HSS iteration method

Suppose that $x^{(0)} \in \mathbb{R}^n$ is an arbitrary initial guess, compute $\{x^{(k)}\}_{k=0}^{\infty}$ for $k = 0, 1, \cdots$, with

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+\frac{1}{2})} + b, \end{cases} \tag{4}$$

in which $S = \frac{1}{2}(A - A^T)$ and $H = \frac{1}{2}(A + A^T)$ are the skew-Hermitian and Hermitian parts of non-Hermitian positive definite matrix $A \in \mathbb{R}^{n\times n}$, respectively, and α is a positive constant.

To avoid solving a shifted skew-Hermitian linear subsystem with the coefficient matrix $\alpha I + S$ in (4), the single-step HSS method was proposed by Li and Wu (2015) as below

$$(\alpha I + H)x^{(k+1)} = (\alpha I - S)x^{(k)} + b, \tag{5}$$

where α is a positive constant.

The following method is presented under the title of a new method by Wang et al. (2019) which is similar to NPHSS method and was referred to in (Miao et al. 2020) with SSI.

2.2 The SSI method

Suppose that P is a given Hermitian positive definite matrix, and $x^{(0)} \in \mathbb{R}^n$ is a given initial guess, compute $\{x^{(k)}\}_{k=0}^{\infty}$ for $k = 0, 1, \ldots$ by

$$(P + H)x^{(k+1)} = (P - S)x^{(k)} + b. \tag{6}$$

The method covers several methods with various choices of the matrix P. For instance, if $P = \alpha I$, the method reduces to the SHSS method. The matrix P can also be taken as $P = \alpha H$, $P = \Delta$, where $\Delta = diag(d_1, d_2, ..., d_n)$, $d_i > 0$, $i = 1, 2, ..., n$), or other different Hermitian matrices. Based on the theoretical and numerical findings presented in

(Wang et al. 2019), the new method is more efficient at solving non-Hermitian positive definite linear problems when the Hermitian part of the coefficient matrix is dominant.

3 SSI based methods for GAVE

In this section, the Picard-HSS (Salkuyeh 2014) and nonlinear HSS-like (Zhu and Qi 2018) iteration methods are briefly reviewed to solve AVEs, and we also introduce the Picard-SSI and nonlinear SSI-like methods, which are inspired from the Picard-HSS (Bai and Yang 2009; Salkuyeh 2014) and nonlinear HSS-like methods (Bai and Yang 2009; Zhu and Qi 2018), respectively. We also study their convergence properties.

3.1 The Picard-SSI method for solving GAVE

The Picard iteration method (Ortega and Rheinboldt 1970) for solving (2) can be written in the form

$$Ax^{(k+1)} = B|x^{(k)}| + b, k = 0, 1, 2, \cdots . \tag{7}$$

The Picard-HSS iteration method was presented by Salkuyeh (2014) to solve AVE (2), which used the HSS method as an inner iterative process in the Picard method.

The Picard-HSS iteration method: Assume $S = \frac{1}{2}(A - A^T)$ and $H = \frac{1}{2}(A + A^T)$ are the skew-Hermitian and Hermitian parts of non-Hermitian positive definite matrix $A \in \mathbb{R}^{n\times n}$, respectively, $\{l_k\}_{k=0}^{\infty}$ as a sequence of positive integers, α is a given positive constant, and $x^{(0)} \in \mathbb{R}^n$ is an initial guess for solution (2), for $k = 0, 1, 2, \cdots$, compute next iterate $x^{(k+1)}$ according to the following algorithm until the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ converges:

(I) Set $x^{(k,0)} = x^{(k)}$;
(II) For $l = 0, 1, 2, \ldots, l_k - 1$, solve

$$\begin{cases} (\alpha I + H)x^{(k,l+\frac{1}{2})} = (\alpha I - S)x^{(k,l)} + |x^{(k)}| + b, \\ (\alpha I + S)x^{(k,l+1)} = (\alpha I - H)x^{(k,l+\frac{1}{2})} + |x^{(k)}| + b; \end{cases} \tag{8}$$

(III) Set $x^{(k+1)} = x^{(k,l_k)}$.

To avoid solving the second subsystem with $(\alpha I + S)$ matrix in (8), which is as difficult as that of the original system, we employ the single-step HSS iteration method as the inner solver for the method (7) to approximate the solution of GAVEs and present the following Picard-SSI method.

3.1.1 The Picard-SSI method

Let $H = \frac{1}{2}(A + A^T)$, $S = \frac{1}{2}(A - A^T)$ are the Hermitian and skew-Hermitian parts of non-Hermitian positive definite matrix $A \in \mathbb{R}^{n\times n}$, respectively, P is a Hermitian positive definite matrix, and $x^{(0)}$ is a given initial guess and $\{l_k\}_{k=0}^{\infty}$ be a sequence of positive integers. Compute $x^{(k+1)}$ for $k = 0, 1, 2, \ldots$, until $\{x^{(k)}\}$ converges:

(I) Choose $x^{(k,0)} = x^{(k)}$;
(II) For $l = 0, 1, 2, \cdots , l_k - 1$, solve

$$(P + H)x^{(k,l+1)} = (P - S)x^{(k,l)} + B|x^{(k)}| + b; \tag{9}$$

(III) Consider $x^{(k+1)} = x^{(k,l_k)}$.

When $B = I$ and $P = \alpha I$, the above scheme reduces to the Picard-SHSS iteration scheme presented by Miao et al. (2021). If we take $B = 2\Omega - A$, where Ω is a positive diagonal parameter matrix, the Picard-SSI method reduces to the modulus-based non-alternating preconditioned splitting (MINPS) method (Wu et al. 2022). It is worth noting that the modulus-based matrix splitting iteration techniques for LCPs originally proposed by Bai (2010). The linear equation (9) can be effectively resolved using either Cholesky factorization or the conjugate gradient (CG) method. In a reformulated form, Picard-SSI method can be used as

$$x^{(k+1)} = T^{l_k} x^{(k)} + \sum_{j=0}^{l_k-1} T^j (GB|x^{(k)}| + Gb), \tag{10}$$

or

$$x^{(k+1)} = T^{l_k} x^{(k)} + (I - T^{l_k})(A^{-1}B|x^{(k)}| + A^{-1}b), \tag{11}$$

where $T = (P+H)^{-1}(P-S) = (I + P^{-1}H)^{-1}(I - P^{-1}S)$ and $G = (P+H)^{-1}$.

In the next theorem, we prove the convergence of the Picard-SSI method.

Theorem 3 *Let matrix A be a non-Hermitian positive definite matrix, $B \in \mathbb{R}^{n\times n}$ be an arbitrary matrix, $\|A^{-1}\|\|B\| = \eta < 1$, and*

$$\sigma_{max}(\tilde{S}) < \lambda_{min}(\tilde{H}), \tag{12}$$

where $\tilde{S} = P^{-1}S$ and $\tilde{H} = P^{-1}H$. Then for given a sequence of positive integers $\{l_k\}$ and initial guess $x^{(0)}$, the sequence $\{x^{(k)}\}$ produced by iterative method (9), converges to a unique solution of GAVE (2) provided that $\liminf_{k\to\infty} l_k \geq L$, where $L \in \mathbb{N}$ satisfying

$$\|T^r\| < \frac{1-\eta}{1+\eta} \quad \forall r \geq L. \tag{13}$$

Proof Let x^* be the unique solution of GAVE (2), so x^* satisfies (11) and we get

$$\begin{aligned} \|x^* - x^{(k+1)}\| &\leq \|x^* - x^{(k)}\|\|T^{l_k}\| + \|I - T^{l_k}\|\|B\|\|A^{-1}\|\||x^*| - |x^{(k)}|\|, \\ &\leq \left(\|T^{l_k}\| + (1 + \|T^{l_k}\|)\|A^{-1}\|\|B\|\right)\|x^{(k)} - x^*\|. \end{aligned} \tag{14}$$

Since H, S, and P^{-1} are Hermitian, skew-Hermitian and Hermitian positive definite matrices, respectively, and we know that all eigenvalues of $\tilde{H}$ are real and positive, then we have

$$\begin{aligned} \rho(T) &\leq \|(I + P^{-1}H)^{-1}\|\|(I - P^{-1}S)\| \\ &\leq \max_{1\leq i\leq n} \frac{1}{1+\lambda_i(\tilde{H})} \max_{1\leq i\leq n} \sqrt{1 + \sigma_i^2(\tilde{S})} \end{aligned}$$

$$\rho(T) \leq \|T\| \leq \frac{\sqrt{1+\sigma_{max}^2(\tilde{S})}}{1+\lambda_{min}(\tilde{H})}. \tag{15}$$

Therefore, the assumption (12) yields $\rho(T) < 1$. Then $T^r \to 0$ as r tends to infinity and

$$\exists L \in \mathbb{N}, \forall k = 0, 1, \cdots, \ s.t \qquad \|T^r\| < \frac{1-\eta}{1+\eta} \quad \forall r \geq L.$$

With setting $\liminf_{k\to\infty} l_k \geq L$, using the above inequality and (14) gives $\|x^{(k+1)} - x^*\| < \|x^{(k)} - x^*\|$, which completes the proof. □

In the following part, a residual-updating variant of the Picard-SSI method is defined.

The Picard-SSI method (residual-updating variant): Let $x^{(0)}$ be an initial guess, P is a Hermitian positive definite matrix, and $\{l_k\}$ is a sequence of positive integers. Apply the following iterative scheme to compute $x^{(k+1)}$ for $k = 0, 1, 2, \cdots$, until $\{x^{(k)}\}$ converges.

(I) Define $r^{(k)} = B|x^{(k)}| + b - Ax^{(k)}$ and choose $y^{(k,0)} = 0$;

(II) For $l = 0, 1, 2, \ldots, l_k - 1$, solve

$$(P + H)y^{(k,l+1)} = (P - S)y^{(k,l)} + r^{(k)};$$

(III) Set $x^{(k+1)} = x^{(k)} + y^{(k,l_k)}$.

The numbers $l_k, k = 0, 1, 2, ...,$ of the inner SSI steps in the Picard-SSI method are problem-dependent and can be difficult to determine in actual computations (Bai and Yang 2009). Consequently, to avoid the need for an explicit inner iteration process, we propose the following nonlinear SSI-like method based on the nonlinear fixed-point equations: $(P + H)x = (P - S)x + B|x| + b$.

3.2 The nonlinear SSI-like method

In this subsection, we review the nonlinear HSS-like method to solve AVEs. In order to improve the efficiency of this method, we propose a nonlinear SSI-like method. Let $H = \frac{1}{2}(A + A^T)$, $S = \frac{1}{2}(A - A^T)$ be the Hermitian and skew-Hermitian parts of non-Hermitian positive definite matrix $A \in \mathbb{R}^{n\times n}$, respectively. In the nonlinear HSS-like iteration method (Zhu and Qi 2018), assume that $x^{(0)} \in \mathbb{R}^n$ be an initial guess, according to the following procedure, compute $x^{(k+1)}$ for $k = 0, 1, 2, \cdots$ until $\{x^{(k)}\}_{k=0}^{\infty}$ converges.

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + |x^{(k)}| + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+\frac{1}{2})} + |x^{(k+\frac{1}{2})}| + b, \end{cases} \tag{16}$$

where $\alpha > 0$.

Below, we present the nonlinear SSI-like method to solve GAVEs.

3.2.1 The nonlinear SSI-like method

Let $x^{(0)}$ and P be given as an initial guess and a Hermitian positive definite matrix. Using the following procedure, compute $x^{(k+1)}$ for $k = 0, 1, 2, \cdots$, until $\{x^{(k)}\}$ converges.

$$(P + H)x^{(k+1)} = (P - S)x^{(k)} + B|x^{(k)}| + b. \tag{17}$$

where $\alpha > 0$. In particular, when $P = I$ and $B = I$, the iteration method (17) is given by

$$(\alpha I + H)x^{(k+1)} = (\alpha I - S)x^{(k)} + |x^{(k)}| + b. \tag{18}$$

Now, we prove the convergence of the proposed method.

Theorem 4 *Assume H and S are the Hermitian and skew-Hermitian parts of non-Hermitian positive definite matrix A in (2). Let P be a Hermitian positive definite matrix and $B \in \mathbb{R}^{n\times n}$ be an arbitrary matrix. Then the new iteration method (17) is convergent if*

$$\delta = \frac{\sqrt{1 + \sigma_{max}^2(\tilde{S})} + \kappa}{1 + \lambda_{min}(\tilde{H})} < 1,$$

where $\tilde{H} = P^{-1}H$, $\tilde{S} = P^{-1}S$ and $\kappa = \frac{\sigma_{max}(B)}{\lambda_{min}(P)}$.

Proof Let $x^* \in \mathbb{R}^n$ be unique solution of GAVE. Using (15) for $T = (P + H)^{-1}(P - S)$ yields

$$\begin{aligned}\|x^{(k+1)} - x^*\| &\le \|(I + P^{-1}H)^{-1}(I - P^{-1}S)\| \|x^{(k)} - x^*\| \\ &\quad + \|(I + P^{-1}H)^{-1}\| \|P^{-1}\| \|B\| \||x^k| - |x^*|\| \\ &\le \frac{\sqrt{1 + \sigma_{max}^2(\tilde{S})} + \kappa}{1 + \lambda_{min}(\tilde{H})} \|x^{(k)} - x^*\|.\end{aligned}$$

Now, since $\delta < 1$, the proof is complete. □

At the end of this section, we reformulated the nonlinear SSI-like method into residual-updating form as follows.

The nonlinear SSI-like method (residual-updating variant): Let $x^{(0)}$ be a given initial guess, P be a given Hermitian Positive definite matrix. Using the following iterative method, compute $\{x^{(k)}\}_{k=0}^{\infty}$:

(I) Define $r^{(k)} = B|x^{(k)}| + b - Ax^{(k)}$;
(II) Solve $(P + H)\nu = r^{(k)}$;
(III) Set $x^{(k+1)} = x^{(k)} + \nu$.

4 Numerical experiments

In this section, we illustrate the performance of the nonlinear SSI-like and Picard-SSI methods by solving three GAVE examples. We compare the proposed methods with the nonlinear HSS-like method (Zhu and Qi 2018) and Picard-HSS (Salkuyeh 2014) with respect to computing times (denoted by CPU) and iteration steps (denoted by IT). All numerical experiments are performed in double precision using MATLAB 2017 (64-bit) on Intel® Core^{TM} i5-10210U processor @ 1.60GHz 2.11 GHz, 8GB RAM. In examples, we choose $x^{(0)} = 0$ and all numerical computations are terminated if the current iteration satisfies

$$\frac{\|Ax^{(k)} - B|x^{(k)}| - b\|}{\|b\|} \le 10^{-7},$$

or the maximum number of iterations $k_{max} = 1000$ is exceeded. We will denote " − " in the tables below if the iteration method cannot converge within k_{max} iterations. The tolerance for controlling the accuracy of the inner iterations is set to be 10^{-2} or a maximum number of iterations of 5 ($l_k = 5$). The experimentally optimal parameters α for all presented methods are used in the implementations, with the least CPU times for these iterative methods yielded; see Table 1 and 3. The subsystems are solved exactly making use of Cholesky or LU factorizations.

Example 1 (Salkuyeh 2014) Consider

$$\begin{cases} -(\nu_{xx} + \nu_{yy}) + q(\nu_x + \nu_y) + p\nu = f(x, y), & (x, y) \in \omega, \\ \nu(x, y) = 0, & (x, y) \in \partial\omega, \end{cases}$$

where q is a positive constant, $p \in \mathbb{R}$, $\omega = (0, 1) \times (0, 1)$ and $\partial\omega$ is boundary of ω.

The mesh Reynolds number can be calculated using the formula $\mathcal{L} = \frac{qh}{2}$, where q is the flow velocity and h is the equidistant step size. The value of h is determined by the equation $h = \frac{1}{m+1}$, where m represents the number of intervals between two adjacent points on the

Table 1 The optimal values α for Example 1

Method	m	$p=0$			$p=0.5$			$p=1$		
		50	80	100	50	80	100	50	80	100
Nonlinear SSI-like	$q=1$	0.6	0.6	0.6	0.145	0.145	0.145	0.01	0.01	0.01
	$q=10$	0.7	0.75	0.65	0.15	0.15	0.15	0.01	0.01	0.01
	$q=100$	3.4	1.6	1	2.09	0.21	0.16	1.27	0.1	0.1
	$q=200$	15.1	5.4	3.8	12.04	3.24	1.65	8.7	2.3	1.14
Nonlinear HSS-like	$q=1$	1.7	1.7	1.7	1.65	1.65	1.65	2.4	2.4	2.4
	$q=10$	1.5	1.6	1.6	1.65	1.6	1.6	2.5	1.6	1.6
	$q=100$	2.2	2	2	2.05	1.72	1.72	2.6	2.6	2.5
	$q=200$	2.6	2.4	2.3	2.15	1.87	1.75	2.7	2.6	2.6
Picard-SSI	$q=1$	3.1	3.1	3.1	1.43	1.44	1.45	.5	0.5	0.5
	$q=10$	6.15	8.0	7.0	1.42	1.44	1.45	2	1	1
	$q=100$	3.7	2.3	2.3	1.7	1.1	1.4	2.1	0.5	0.8
	$q=200$	13.3	5.2	3.6	10.7	3.62	2	8.4	2	0.9
Picard-HSS	$q=1$	6.2	6.2	6.2	3.5	3.5	3.5	2.3	2.3	2.3
	$q=10$	7.2	7.5	7	3.5	3.5	3.5	2.3	2.3	2.3
	$q=100$	4.6	4.6	4.6	3	3	3	3	3	3
	$q=200$	4	4	4	3.5	3.5	3.5	4	4	4

mesh. The central difference scheme and five-point finite difference scheme are used to the convective terms and diffusive terms, respectively, to acquire the linear equations system $Cx = \mathsf{d}$ where

$$C = \mathcal{T}_x \otimes I_m + I_m \otimes \mathcal{T}_{\mathsf{y}} + pI_n \in \mathbb{R}^{n\times n}, \tag{19}$$

with $n = m^2$, $\mathcal{T}_x = tridiag(-\mathcal{L}-1, 4, \mathcal{L}-1)$ and $\mathcal{T}_y = tridiag(-\mathcal{L}-1, 0, \mathcal{L}-1)$. In AVE (2), consider $A = C$ and $b = Ax - |x|$, where $x = (i(-1)^i)_{n\times 1}$ be an exact solution. For this example, the parameter matrix P is taken as αI. Note that the matrix A is nonsymmetric positive definite for every nonnegative number q.

CPU times and iteration numbers for all methods are presented in Table 2, where the optimal parameters presented in Table 1 are used in these experiments.

In Fig 1, the iteration steps and different values of parameter α are presented for methods in Example 4.1 with $p = 0, q = 1, m = 100$. From the figure, we can find that the optimal parameters α for the nonlinear SSI-like, nonlinear HSS-like, Picard-HSS, and Picard-SSI methods are 0.6, 1.7, 6.2 and 3.1, respectively.

Table 2 shows that all tested iterative methods can successfully compute an approximate solution for Example 4.1. Regarding CPU time in Table 2, for $q = 1, 10, 100$, the SSI-like method performs better than others. The second method that causes less CPU time is the nonlinear HSS-like method and then the Picard-SSI method and Picard-HSS method, respectively. For $q = 200$, the superiority of the nonlinear SSI-like method disappears and the nonlinear HSS-like method becomes better.

In the following example, we consider the GAVE (2), which can be found by linear complementarity problem (LCP). The LCP$(q, \mathcal{A})$ consists of finding unknown vectors $z, w \in \mathbb{R}^n$ such that

$$w = \mathcal{A}z + q \geq 0, \quad z \geq 0 \quad and \quad z^T w = 0, \tag{20}$$

Table 2 Numerical results of different iterative methods for Example 1

q	Method	n	$p = 0$			$p = 0.5$			$p = 1$		
			2500	6400	10000	2500	6400	10000	2500	6400	10000
	Nonlinear SSI-like	IT	36	35	35	30	30	30	19	19	19
		CPU	0.0131	0.0644	0.1537	0.0102	0.0464	0.1068	0.0092	0.0318	0.0665
1	Nonlinear HSS-like	IT	25	26	26	39	40	40	21	21	21
		CPU	0.0195	0.0873	0.1912	0.0306	0.1271	0.2657	0.0162	0.0735	0.1328
	Picard-SSI	IT	30	29	29	28	28	28	19	20	20
		CPU	0.0574	0.2188	0.5452	0.0550	0.2016	0.4643	0.0213	0.1093	0.3053
	Picard-HSS	IT	28	28	28	32	32	32	20	20	20
		CPU	0.0961	0.5552	1.1188	0.1251	0.6254	1.2578	0.0778	0.3856	0.8051
	Nonlinear SSI-like	IT	43	44	35	30	30	30	19	19	19
		CPU	0.0148	0.0735	0.1372	0.0106	0.0467	0.1156	0.0068	0.0321	0.0624
10	Nonlinear HSS-like	IT	37	34	29	39	40	40	23	22	22
		CPU	0.0271	0.1244	0.2197	0.0291	0.1481	0.2773	0.0173	0.0867	0.1342
	Picard-SSI	IT	37	37	29	29	28	28	21	19	19
		CPU	0.0656	0.5554	0.9358	0.0498	0.2139	0.4471	0.03605	0.1318	0.3167
	Picard-HSS	Iter	36	37	28	36	37	38	20	20	20
		CPU	0.1474	0.7344	1.1421	0.1453	0.7267	1.5614	0.0714	0.4465	0.7625
	Nonlinear SSI-like	IT	172	226	293	75	35	30	35	20	20
		CPU	0.1013	0.3516	1.0521	0.0224	0.0547	0.1150	0.0266	0.0352	0.0751

Table 2 continued

q	Method	n	$p=0$			$p=0.5$			$p=1$		
			2500	6400	10000	2500	6400	10000	2500	6400	10000
100	Nonlinear HSS-like	IT	64	147	313	30	38	39	22	22	23
		CPU	0.1197	0.4652	2.2147	0.0249	0.1358	0.2781	0.0352	0.0833	0.1638
	Picard-SSI	IT	56	135	270	33	33	36	19	19	19
		CPU	0.2187	0.9874	4.0305	0.0587	0.2507	0.5441	0.0726	0.1035	0.2515
	Picard-HSS	Iter	52	146	295	28	30	32	19	19	19
		CPU	0.2248	2.6224	12.1094	0.1052	0.5722	1.3397	0.0741	0.3526	0.7541
	Nonlinear SSI-like	IT	327	303	321	206	122	90	118	53	36
		CPU	0.0965	0.4858	1.067	0.0682	0.1935	0.3027	0.0464	0.0891	0.1253
200	Nonlinear HSS-like	IT	49	83	116	25	32	36	20	22	22
		CPU	0.0388	0.2823	0.7283	0.01870	0.1078	0.2487	0.0258	0.0776	0.1214
	Picard-SSI	IT	57	74	100	40	40	37	28	20	19
		CPU	0.1011	0.5083	1.5905	0.0698	0.3064	0.6176	0.0662	0.1510	0.3193
	Picard-HSS	Iter	28	58	136	21	30	32	17	19	19
		CPU	0.1060	1.2821	5.1406	0.0974	0.5789	1.3289	0.0665	0.3662	0.7751

Table 3 The optimal values α for Example 2

Method	P	$n = 2500$	$n = 10000$	$n = 40000$	$n = 90000$
nonlinear SSI-like	αI	2.3	2.3	2.3	2.3
	αH	0.4	0.4	0.4	0.4
Picard-SSI	αI	25	25	25	25
	αH	4	4	4	4
nonlinear HSS-like	αI	6	6	6	6
	αH	1.2	1.2	1.2	1.2
Picard-HSS	αI	60	60	60	60
	αH	8.5	8.5	8.5	8.5

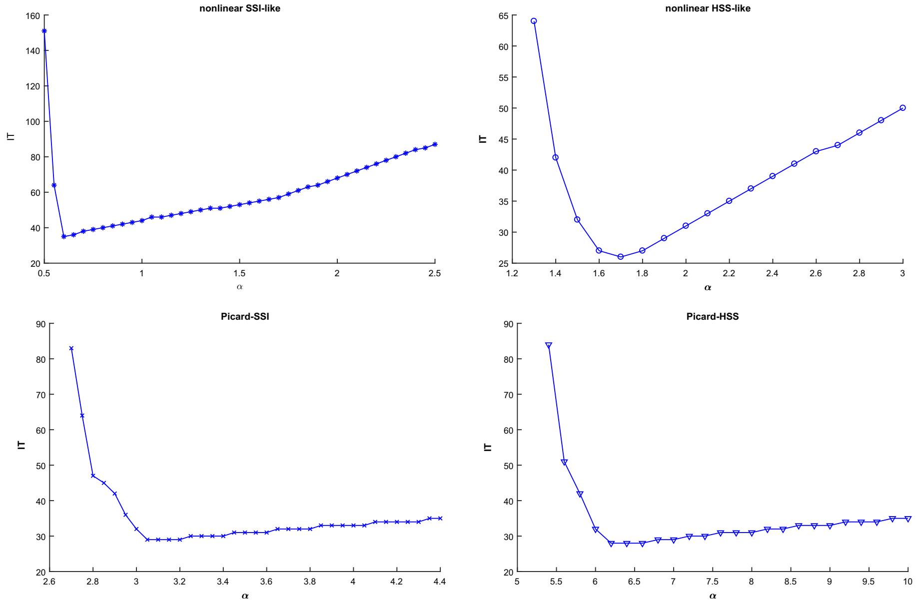

Fig. 1 The number of iterations and various α values for nonlinear SSI-like, nonlinear HSS-like, Picard-HSS and Picard-SSI methods in Example 1 with parameters set as $p = 0$, $q = 1$, and $m = 100$

where $\mathcal{A} \in \mathbb{R}^{n\times n}$ and vector $q \in \mathbb{R}^n$. Let Ω, and γ are a positive diagonal matrix and positive constant, respectively. As we know from (Bai 2010; Mangasarian et al. 2006), if x is a solution of

$$(\Omega + \mathcal{A})x - (\Omega - \mathcal{A})|x| = -\gamma q, \tag{21}$$

$z = \frac{1}{\gamma}(|x| + x)$ is a solution of the LCP$(q, \mathcal{A})$ (20).

Example 2 (Bai 2010) Consider the GAVE (21), in which $\Omega = \frac{1}{2}diag(\mathcal{A})$, $\mathcal{A} = tridiag(-1.5I, S, -0.5I) + 4I \in \mathbb{R}^{n\times n}$, $q = -\mathcal{A}z^* \in \mathbb{R}^n$, $S = tridiag(-1.5, 4, -0.5) \in \mathbb{R}^{m\times m}$, $n = m^2$, $\gamma = 2$ and $z^* = (1, 2, 1, 2, ..., 1, 2, ...)^T \in \mathbb{R}^n$. In this example, the parameter matrix P is taken αI and αH.

In Table 3, the optimal parameter α, and in Table 4, the CPU times and the iteration steps

Table 4 Numerical results of Example 2 for different matrix P

Method	P	n	2500	10000	40000	90000
	αI	IT	11	11	11	11
nonlinear SSI-like		CPU	0.0036	0.0379	0.3071	1.0727
	αH	IT	9	9	9	9
		CPU	0.0034	0.0321	0.2487	0.7692
	αI	IT	13	12	12	11
Picard-SSI		CPU	0.0201	0.2174	1.6063	7.0721
	αH	IT	8	8	8	8
		CPU	0.0131	0.1318	1.0647	4.2092
	αI	IT	7	7	7	7
nonlinear HSS-like		CPU	0.0081	0.0721	0.5063	1.9371
	αH	IT	6	6	6	6
		CPU	0.0052	0.0603	0.4041	1.2390
	αI	IT	14	14	14	13
Picard-HSS		CPU	0.0642	0.6328	5.8869	17.9012
	αH	IT	8	9	9	9
		CPU	0.0412	0.3903	2.8969	8.4767

of nonlinear SSI-like, nonlinear HSS-like, Picard-HSS, and Picard-SSI methods are listed for different values of m to solve Example 2.

Table 4 clearly indicates that the CPU time for the nonlinear SSI-like method is lower than that of all other three methods for both $P = \alpha H$ and $P = \alpha I$. Furthermore, it is evident that the CPU times for the nonlinear SSI-like, nonlinear HSS-like, Picard-HSS, and Picard-SSI methods with $P = \alpha H$ are all lower than those of these methods for $P = \alpha I$.

Example 3 (Yong 2015) Consider the following boundary value problem:

$$\begin{cases} -\dfrac{d^2u}{dx^2} + |u| = (x^2 - 1), 0 \leq x \leq 1 \\ u(0) = -1, u(1) = 0 \end{cases} \tag{22}$$

with exact solution

$$u(x) = 0.1916 \sin x - 4 \cos x - x^2 + 3,\ 0 \leq x \leq 1. \tag{23}$$

Let $h = \frac{1}{n+1}$ be the mesh size, $x_0 = 0,\ x_i = ih, i = 1, 2, \cdots, n,\ x_{n+1} = 1$. By replacing the interior mesh-points $x_i, i = 1, \cdots, n$ into (22), we obtain the following equations:

$$-u''(x_i) + |u(x_i)| = f(x_i), i = 1, 2, \cdots, n.$$

Using the five-point central difference formula for $u''(x_i)$, we have:

$$\begin{aligned} u''(x_i) &\approx \frac{-u_{i-2} + 16u_{i-1} - 30u_i + 16u_{i+1} - u_{i+2}}{12h^2}, i = 2, \cdots, n-1, \\ u''(x_1) &\approx \frac{11u_0 - 20u_1 + 6u_2 + 4u_3 - u_4}{12h^2}, i = 1, \\ u''(x_n) &\approx \frac{-u_{n-4} + 4u_{n-3} + 6u_{n-2} - 20u_{n-1} + 11u_n}{12h^2}, \end{aligned} \tag{24}$$

Table 5 Numerical results of Example 3 for different n

Method	n	10	100	200	500	5000
	IT	13	14	14	14	14
nonlinear SSI-like	CPU	0.0003	0.0004	0.0005	0.0008	0.0022
	α	3.5	4.7	5.4	6	6.5
	IT	61	507	973	–	–
nonlinear HSS-like	CPU	0.0024	0.0185	0.0373	–	–
	α	69.7	730	1487	–	–
	IT	7	6	6	6	6
Picard-SSI	CPU	0.0003	0.0005	0.0006	0.0011	0.0051
	α	10.9	13	13.7	13.7	39.75
	IT	11	94	181	429	–
Picard-HSS	CPU	0.0010	0.0145	0.0382	0.1241	–
	α	75	780	1600	4200	–

where u_i denotes the numerical approximation of $u(x_i)$. Thus, we obtain the system $Ax - B|x| = b$, where $B = -I$,

$$A = \frac{1}{12h^2}\begin{pmatrix} 20 & -6 & -4 & 1 & & & & & \\ -16 & 30 & -16 & 1 & & & & & \\ 1 & -16 & 30 & -16 & 1 & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -16 & 30 & -16 & 1 \\ & & & & & 1 & -16 & 30 & -16 \\ & & & & & 1 & -4 & -6 & 20 \end{pmatrix},$$

and

$$b = \begin{Bmatrix} f(x_1) + \frac{11u_0}{12h^2} \\ f(x_2) - \frac{u_0}{12h^2} \\ f(x_3) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{u_{n+1}}{12h^2} \\ f(x_n) + \frac{11u_{n+1}}{12h^2} \end{Bmatrix},$$

where $f(x_i) = (x_i^2 - 1)$, and $x_i = ih, i = 1, 2, \cdots, n$.

Table 5 presents the numerical results obtained using the nonlinear SSI-like, Picard-SSI, Picard-HSS, and nonlinear HSS-like methods for different values of n in Example 3.

Table 5 shows that the nonlinear HSS-like method fails to converge even after 1000 iterations for both $n = 500$ and $n = 5000$. Furthermore, the Picard-HSS method also fails to converge after 1000 iterations when $n = 5000$, whereas the nonlinear SSI-like and Picard-SSI methods show convergence. Based on a comparison of these methods, it was found that the nonlinear SSI-like and Picard-SSI methods consume less CPU time and require fewer iterations to obtain the numerical solution compared to the HSS-like and Picard-HSS methods. In terms of performance, the nonlinear SSI-like method outperforms the other three methods.

5 Conclusion

In this paper, we have considered the nonlinear SSI-like and Picard-SSI methods that utilize the single-step iteration method for solving GAVEs. We have discussed their convergence properties and conducted numerical experiments to confirm that the proposed methods are efficient and feasible for solving GAVEs. Our findings demonstrate that the nonlinear SSI-like method is more efficient than the nonlinear HSS-like, Picard-HSS, and Picard-SSI methods.

Data Availability No new data was created in this work.

References

Bai Z-Z (1997) A class of two-stage iterative methods for systems of weakly nonlinear equations. Numer Algorithms 14:295–319

Bai Z-Z (2010) Modulus-based matrix splitting iteration methods for linear complementarity problems. Numer Linear Algebra Appl 17(6):917–933

Bai Z-Z, Yang X (2009) On HSS-based iteration methods for weakly nonlinear systems. Appl Numer Math 59(12):2923–2936

Bai Z-Z, Golub GH, Ng MK (2003) Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J Matrix Anal Appl 24(3):603–626

Benzi M (2009) A generalization of the Hermitian and skew-Hermitian splitting iteration. SIAM J Matrix Anal Appl 31(2):360–374

Cottle RW, Pang J-S, Stone RE (1992) The linear complementarity problem. Academic Press, San Diego

Dehghan M, Shirilord A (2020) Matrix multisplitting Picard-iterative method for solving generalized absolute value matrix equation. Appl Numer Math 158:425–438

Dong X, Shao X-H, Shen H-L (2020) A new SOR-like method for solving absolute value equations. Appl Numer Math 156:410–421

Ebadi G, Alipour N, Vuik C (2016) Deflated and augmented global Krylov subspace methods for the matrix equations. Appl Numer Math 99:137–150

Golub GH, Van Loan CF (2013) Matrix computations. Johns Hopkins University Press, Baltimore

Idema R, Vuik C (2023) A comparison of Krylov methods for shifted skew-symmetric systems. https://doi.org/10.48550/arXiv.2304.04092

Li C-X, Wu S-L (2012) A modified GHSS method for non-Hermitian positive definite linear systems. Jpn J Ind Appl Math 29(2):253–268

Li C-X, Wu S-L (2015) A single-step HSS method for non-Hermitian positive definite linear systems. Appl Math Lett 44:26–29

Li X, Li Y-X, Dou Y (2023) Shift-splitting fixed point iteration method for solving generalized absolute value equations. Numer Algorithms 93:695–710

Mangasarian OL (2009) A generalized Newton method for absolute value equations. Optim Lett 3:101–108

Mangasarian OL, Meyer RR (2006) Absolute value equations. Linear Algebra Appl 419:359–367

Miao S-X, Zhang J (2020) On Uzawa-SSI method for non-Hermitian saddle point problems. AIMS Math 5(6):7301–7315

Miao S-X, Xiong X-T, Wen J (2021) On Picard-SHSS iteration method for absolute value equation. AIMS Math 6(2):1743–1753

Murty KG, Yu FT (1998) Linear complementarity, linear and nonlinear programming, vol 3. Heldermann, Berlin

Noor MA, Noor KI, Batool S (2018) On generalized absolute value equations. UPB Sci Bull Ser A Appl Math Phys 80(4):63–70

Ortega JM, Rheinbolt WC (1970) Iterative solution of nonlinear equations in several variables. Academic Press, New York

Prokopyev O (2009) On equivalent reformulations for absolute value equations. Comput Optim Appl 44:363–372

Rohn J (2004) A theorem of the alternatives for the equation $Ax + B|x| = b$. Linear Multilinear Algebra 52(6):421–426

Rohn J (2009) On unique solvability of the absolute value equation. Optim Lett 3:603–606

Salkuyeh DK (2014) The Picard-HSS iteration method for absolute value equations. Optim Lett 8:2191–2202

Seifollahzadeh S, Ebadi G (2024) On diagonal and off-diagonal splitting-based iteration method to solve absolute value equations. Filomat 38(6):2001–2013

Trefethen LN, Bau D (1997) Numerical linear algebra. SIAM, Philadelphia

Wang A, Cao Y, Chen J-X (2019) Modified Newton-type iteration methods for generalized absolute value equations. J Optim Theory Appl 181:216–230

Wang X, Xiao X-Y, Zheng Q-Q (2019) A single-step iteration method for non-Hermitian positive definite linear systems. J Comput Appl Math 346:471–482

Wu S-L, Li C-X (2020) A note on unique solvability of the absolute value equation. Optim Lett 14(7):1957–1960

Wu Y-J, Li X, Yuan J-Y (2017) A non-alternating preconditioned HSS iteration method for non-Hermitian positive definite linear systems. Comput Appl Math 36(1):367–381

Wu Y-J, Zhang W-H, Yang A-L (2022) Modulus-based inexact non-alternating preconditioned splitting method for linear complementarity problems. Linear Multilinear Algebra 70(22):7414–7432

Yong L (2015) Iteration method for absolute value equation and applications in two-point boundary value problem of linear differential equation. J Interdiscip Math 18:355–374

Zhang J-J (2015) The relaxed nonlinear PHSS-like iteration method for absolute value equations. Appl Math Comput 265:266–274

Zhou H-Y, Wu S-L, Li C-X (2021) Newton-based matrix splitting method for generalized absolute value equation. J Comput Appl Math 394:113578

Zhu M-Z, Qi Y-E (2018) The nonlinear HSS-like iteration method for absolute value equations. IAENG Int J Appl Math 48(3):75–79

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.