AIMS Mathematics, 11(5): 14302–14322.
DOI: 10.3934/math.2026587
Received: 06 March 2026
Revised: 24 April 2026
Accepted: 08 May 2026
Published: 20 May 2026

https://www.aimspress.com/journal/Math

Research article

A new two step iterative method based on Newton and fixed point for solving absolute value equations

Nifatamah Makaje, Asma Yafad and Aniruth Phon-On*

Department of Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Thailand

* **Correspondence:** Email: aniruth.p@psu.ac.th; Tel: +66824245993.

Abstract: The absolute value equation (AVE) is particularly important in numerical mathematics, mathematical programming, and optimization. Due to its close connection to linear complementarity problems, significant research efforts have been devoted to developing efficient numerical solvers for AVEs. In this study, we presented a novel iterative methodology that combines Newton's method with fixed-point iteration, known as the Newton technique fixed-point iteration (NTFPI) method, to solve AVEs. The proposed solution integrated the optimal elements of Newton methods and fixed-point iterations to enhance the convergence speed and stabilize calculations. A theoretical examination was performed to determine the convergence characteristics of the approach. In addition, numerical assessments were performed to determine the efficacy of the NTFPI method in relation to other recognized Newton and fixed-point iterative techniques. The findings indicated that the proposed method enhanced accuracy and reduced repetitions, particularly for complex and large-scale tasks. These findings improved the numerical approaches for AVEs and demonstrate their potential applications in engineering, computational economics, and numerical optimization.

Keywords: absolute value equation; Newton's method; fixed-point method; linear complementarity problem; optimization problem
Mathematics Subject Classification: 15A06, 49M15, 65F10, 65H10, 90C05, 90C33

1. Introduction

Consider the system of absolute value equation (AVE) represented by:

$$Ax - |x| - b = 0 \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, x denotes the vector of unknowns, and $|x|$ represents the vector of absolute values of each component of x. A more general version of this system, referred to as the generalized

absolute value equation (GAVE), is given by

$$Ax - B|x| - b = 0,$$

where $A, B \in \mathbb{R}^{n\times n}$. This generalized form was first introduced by Rohn in 2004 [1] and has since been widely studied by numerous researchers. If B is nonsingular, the system can be easily transformed into the AVE form

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

However, to avoid matrix inversion and to accommodate the singular case, many researchers prefer to analyze the GAVE directly (see [2, 3]). The significance of the absolute value equations lies in their capacity to simplify complex problems, such as linear programs, quadratic programs, bimatrix games, and various other applications, by reducing them to linear complementarity problems (LCPs). In 2006, Mangasarian and Meyer [4] demonstrated that the general NP-hard LCP, originally proposed by Cottle and Dantzig in 1968 [5], encompasses a broad class of mathematical programming problems, many of which are more tractable when formulated as AVEs. Furthermore, Mangasarian [4, 6] has shown the equivalence between AVEs and LCPs, which has led to increased interest in studying the properties and solutions of AVEs.

In recent years, a wide range of numerical techniques have been developed to solve AVEs. Haghani [7] proposed and analyzed the convergence of a generalized Traub's approach, while Prokopyev [8] investigated the unique solvability of AVEs and their connection to LCPs. Li [9] proposed and analyzed a modified generalized Newton iterative method to solve AVEs under certain singular-value conditions on the matrix of the system, and Li [10] employed the accelerated over-relaxation (AOR) method. Abdallah et al. [11] applied complementarity functions to address AVEs. Fakharzadeh and Sham [12] introduced a mixed-type splitting technique, and Ali et al. [13] developed a fixed-point iteration method. Ke and Ma [14] suggested a successive over relaxation (SOR)-like approach, while Feng and Liu [15] proposed a two-step iterative scheme along with an improved generalized Newton method, and [16] proposed a globally convergent predictor–corrector scheme with demonstrated performance in benchmark AVEs . The matrix multisplitting method was explored by Dehghan and Shirilord [17], and Khan et al. [18] introduced a two-step technique using a generalized Newton method as the predictor step and Simpson's method as the corrector. Several other techniques have also been proposed in the literature. In particular, fixed-point-type iterations and their variants have been investigated in [19–21]. Closely related iterative schemes and refinements can be found in [13, 22, 23]. Further developments for AVEs and generalized models are reported in [2, 24, 25].

Among those methods, two-step iterative strategies have attracted considerable attention due to their potential for faster convergence and improved stability. In particular, authors in [16, 18, 26] laid an important foundation by designing hybrid techniques that combine the strengths of Newton-type and fixed-point methods. These developments provide a key motivation for further refinement and generalization of two-step iterative frameworks.

In 2024, Yafad et al. [27] proposed a new two-step iterative method to solve AVEs, drawing inspiration from the Newton-based scheme introduced by Khan et al. [18] and the fixed-point iteration method of Mann [28]. This method, which we refer to as the Newton technique fixed-point iteration (NTFPI) method, employs the classical Newton technique as the predictor step followed by a

modified fixed-point iteration as the corrector step. The simplicity of this approach, coupled with its rapid convergence, makes it a promising tool for solving AVEs.

In this paper, we extend the work of Yafad et al. [27] by providing a rigorous convergence analysis of the NTFPI method, which was not established in [27]. We also strengthen the numerical study by adding a comparative evaluation against additional state-of-the-art algorithms, including the TSI method, together with several benchmark test problems. Furthermore, we include practical application examples, such as LCPs and equilibrium problems, to demonstrate the robustness and efficiency of the proposed NTFPI method.

2. Preliminary

We consider the AVE in an equivalent form of (1.1), represented by the following expression:

$$f(x) = 0 \tag{2.1}$$

where $f(x) = Ax - |x| - b, A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, x denotes the vector of unknowns, and $|x|$ represents the absolute value of the vector x component-wise.

The generalized Jacobian of $f(x) = Ax - |x| - b$ at x is given by

$$\partial f(x) = A - H(x),$$

where $H(x) := \text{diag}(\text{sign}(x_1), \text{sign}(x_2), \ldots, \text{sign}(x_n)) \in \mathbb{R}^{n\times n}$ is a diagonal matrix. Note that $H(x)x = |x|$.

To solve AVE (1.1), several iterative approaches have been proposed. In particular, we focus on three representative and well-established methods, which provide a basis for identifying challenges and gaps that motivate our new approach.

The first method is the two-step iterative (TSI) method introduced by Feng and Liu [16], given as follows:

$$\begin{aligned} y^{(k)} &= x^{(k)} + (A - H(x^{(k)}))^{-1} f(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - (A - H(x^{(k)}))^{-1} (f(y^{(k)}) - f(x^{(k)})), \ k = 0, 1, 2, \ldots . \end{aligned} \tag{2.2}$$

The second method is the Newton technique (NT) introduced by Khan et al. [18], which is expressed as:

$$\begin{aligned} t^{(k)} &= x^{(k)} - (A - H(x^{(k)}))^{-1} b, \\ x^{(k+1)} &= x^{(k)} - 6\left(f'(x^{(k)}) + 4f'\left(\frac{x^{(k)} + t^{(k)}}{2} \right) + f'(t^{(k)}) \right)^{-1} f(x^{(k)}), \ k = 0, 1, 2, \ldots . \end{aligned} \tag{2.3}$$

The third method is the modified fixed-point iteration (MFPI) method proposed by Yu et al. [26]. This method begins by reformulating the AVE (1.1) as a nonlinear equation of blocks of two-by-two:

$$\begin{pmatrix} A & -Q \\ -H(x) & Q \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \tag{2.4}$$

where $Q \in \mathbb{R}^{n\times n}$ is invertible and $H(x) := \text{diag}(\text{sign}(x_1), \text{sign}(x_2), \ldots, \text{sign}(x_n)) \in \mathbb{R}^{n\times n}$ is a diagonal matrix. If $Q = I$, where I is an identity, then (2.4) is equivalent to the original AVE (1.1). When A and Q are nonsingular, the MFPI iteration formula is given by

$$\begin{aligned} x^{(k+1)} &= A^{-1}(Qy^{(k)} + b), \\ y^{(k+1)} &= (1 - \tau)y^{(k)} + \tau Q^{-1} \left| x^{(k+1)} \right|, \end{aligned} \tag{2.5}$$

where τ is a parameter. It should be noted that the inverse of Q is involved in each step of the MFPI method, increasing the computational cost of each iteration, especially when Q is dense or ill-conditioned. A common strategy is to choose Q as a diagonal matrix, particularly a scalar matrix. For example, in the numerical example provided by Yu et al. [26], Q is set to $Q = \frac{1}{q}I$ with $q = q_l + 100 > 0$, and I is the order of the identity matrix n. The conditions of setting the value τ reduce to

$$0 < v < \frac{1}{\alpha\sqrt{1+q^2}}$$

and

$$\frac{1-\sqrt{1-\frac{v^2}{q^2}}}{1-v} < \tau < \frac{1+\sqrt{1-\frac{v^2}{q^2}}}{1+v}.$$

The conditions provide sufficient guarantees that AVE (1.1) is uniquely solvable for any $b \in \mathbb{R}$, as described in the following theorem.

Theorem 2.1. *([29]) Let $A \in \mathbb{R}^{n\times n}$. If A satisfies any of the following conditions, then AVE (1.1) has a unique solution for any given $b \in \mathbb{R}^n$:*

(1) $\sigma_{\min}(A) > 1$ where $\sigma_{\min}(A)$ is the minimum singular value of A.

(2) $\left\|A^{-1}\right\| < 1$, provided A is non-singular.

(3) The matrix $A^TA - I$ is positive definite.

All three methods, the TSI method, the NT, and the MFPI method, have demonstrated that the sequences $\{x^{(k)}\}$ generated by Eqs (2.3) and (2.5) converge to the exact solution of AVE (1.1) when $\left\|A^{-1}\right\| < 1$.

Despite their effectiveness, these methods exhibit several limitations, mainly reflected in the computational effort per iteration. The TSI method typically requires multiple function evaluations (e.g., evaluating both $f(x^{(k)})$ and $f(y^{(k)})$) together with at least one linear-system solve involving $(A - H(x^{(k)}))$ in each iteration, which increases the overall computational burden. The Newton technique (NT) involves additional Jacobian-related operations and solves within each iteration (see (2.3)), which can be computationally intensive for large-scale problems. For the MFPI method (2.5), each iteration applies both A^{-1} and Q^{-1} (equivalently, it requires solves with A and with Q), so in general it involves two matrix factorizations (or two preconditioned solves) per iteration when A and Q are not specially structured; this can be costly when Q is dense or ill-conditioned. These considerations motivate the development of a new iterative method that retains the strengths of existing approaches while reducing the per-iteration computational cost.

3. Propose method

Mann introduced a classical approach for approximating a fixed point of a nonlinear mapping $x = T(x)$ by 1953 [28], defined as follows:

$$x^{k+1} = (1-\beta_k)x^k + \beta_k T(x^k), \quad k = 0, 1, 2, ...,$$

where $\{\beta_k\}$ is a sequence in $[0, 1]$. This process is commonly referred to as Mann iteration. By formulating the nonlinear Eq (2.1) in the form of a fixed-point equation, we can express it as

$$x = A^{-1}(|x| - b),$$

which allows us to obtain a new formulation for the corrector step.

Building upon this foundation, we extend this concept to the corrector step in formula (2.2). The new iterative method is defined as follows:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} f(x^{(k)}), \\ x^{(k+1)} &= (1-\beta_k)y^{(k)} + \beta_k(A^{-1}\left|y^{(k)}\right| + A^{-1}(b)),\ k = 0, 1, 2, ..., \end{aligned} \tag{3.1}$$

where the iteration parameter β_k lies in the interval $[0, 1]$.

In the iterative formula (3.1), the proposed method incorporates the NT technique as the predictor step, while the fixed-point approach is employed as the corrector step. The following algorithm outlines the proposed technique.

Algorithm 1: NTFPI technique:

1) Initialization: choose an initial vector $x^{(0)} \in \mathbb{R}^n$, $k = 0$, and set β_k within the interval $(0, 1]$.
2) Predictor step: compute an approximation $y^{(k)} = x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} f(x^{(k)})$.
3) Corrector step: using $y^{(k)}$, compute $x^{(k+1)} = (1-\beta_k)y^{(k)} + \beta_k(A^{-1}\left|y^{(k)}\right| + A^{-1}(b))$.
4) Convergence check: if the relative residual (RES),

$$\text{RES} = \frac{\|Ax^{(k)} - |x^{(k)}| - b\|}{\|b\|} < \epsilon,$$

is met, then it is terminated; otherwise, update $k = k + 1$ and repeat from step 2).

To establish the convergence of Algorithm 1, it is necessary to refer to some theoretical results. Therefore, we begin by citing the following two results from [4, 30], respectively.

Lemma 3.1. *The singular values of the matrix $A \in \mathbb{R}^{n\times n}$ exceed 1 if and only if the minimum eigenvalue of A^TA exceeds 1.*

Lemma 3.1 gives sufficient conditions for the Newton iteration in the predictor step to be well defined. The next lemma is the useful consequence of Lemma 3.1, which gives sufficient conditions so that the Newton iteration in the predictor step is well defined.

Lemma 3.2. *If the singular values of $A \in \mathbb{R}^{n\times n}$ exceed 1 for method (3.1), then $(A - H)^{-1}$ exists for any diagonal matrix H whose diagonal elements are ± 1 or 0.*

Therefore, the sequence of approximated solutions that you get from the predictor step is well defined, no matter how you choose the initial vector $x^{(0)}$.

In the following, we prove some results inspired by the results in [15, 30]. However, the formula in the corrector step will be different.

Note that if $y^{(k)}$ is the solution of AVE (1.1), we can simplify formula (3.1) to derive the following form of the linear search form for AVE (1.1):

Assume $y^{(k)}$ is the solution of AVE (1.1), i.e., $Ay^{(k)} - |y^{(k)}| = b$. Then, we have

$$y^{(k)} = A^{-1}(|y^{(k)}| + b).$$

Considering the second step of formula (3.1), we obtain

$$\begin{aligned} x^{(k+1)} &= (1-\beta_k)y^{(k)} + \beta_k(A^{-1}|y^{(k)}| + A^{-1}(b)) \\ &= (1-\beta_k)y^{(k)} + \beta_k y^{(k)} \\ &= y^{(k)} \\ &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} f(x^{(k)}) \\ &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1}(Ax^{(k)} - |x^{(k)}| - b). \end{aligned}$$

Thus,

$$x^{(k+1)} = x^{(k)} + d^{(k)}, \tag{3.2}$$

where

$$d^{(k)} = -\left(A - H(x^{(k)})\right)^{-1}(Ax^{(k)} - |x^{(k)}| - b).$$

It is clear that the NTFPI method is a Newton method with a specific linear search.

We now prove that the proposed Newton direction of (3.1) is a descent direction for the objective function $\|f(x)\|^2$.

Lemma 3.3. *If the singular values of $A \in \mathbb{R}^{n\times n}$ exceed 1, then the proposed Newton direction of (3.1) is a descent direction for the objective function*

$$F(x) = \frac{1}{2}\|f(x)\|^2.$$

Proof. Since $f(x) = Ax - |x| - b$, $\partial f(x) = A - H(x)$, and $(A - H(x))^{-1}$ exists for any diagonal matrix H whose diagonal elements are ± 1 or 0, in addition we have

$$d^k = -(\partial f(x))^{-1} f(x) = -\left(A - H(x^{(k)})\right)^{-1}(Ax^{(k)} - |x^{(k)}| - b)$$

and

$$\begin{aligned} y^k &= x^{(k)} - (A - H(x^{(k)}))^{-1}(Ax^{(k)} - |x^{(k)}| - b) \\ &= x^{(k)} + d^{(k)}. \end{aligned}$$

Moreover,

$$F(x) = \frac{1}{2}\|f(x)\|^2,$$

thus $\partial F x = (\partial f(x))^T f(x)$. So,

$$\begin{aligned} \left\langle \partial F(x), d^k \right\rangle &= \left\langle (\partial f(x))^T f(x), -(\partial f(x))^{-1} f(x) \right\rangle \\ &= -(f(x))^T (\partial f(x))(\partial f(x))^{-1} f(x) \\ &= -(f(x))^T f(x) \end{aligned}$$

$$= -\|f(x)\|^2 < 0.$$

Consequently, d^k is a descent direction for the objective function $F(x)$; that is, d^k being a descent direction ensures $F(x)$ decreases locally, and $y^{(k)} = x^{(k)} + d^{(k)}$ represents the candidate for the next point in the optimization process. Proper step size and choice of descent direction are critical for effective minimization. □

Although Lemma 3.3 shows that $d^{(k)}$ is a descent direction for the objective function

$$F(x) = \frac{1}{2}\|f(x)\|^2,$$

a descent direction alone does not imply that the residual $\|f(x^{(k)})\|$ converges to zero. To obtain a rigorous global decrease along the iterations generated by Algorithm 1, we impose the following standard sufficient-decrease condition (SD).

Assumption (SD). Let

$$F(x) = \frac{1}{2}\|f(x)\|^2.$$

We assume that the sequence $\{x^{(k)}\}$ generated by Algorithm 1 satisfies the following: There exists a constant $c \in (0, 1)$ such that for all $k \geq 0$,

$$F(x^{(k+1)}) \leq F(x^{(k)}) - c\,\|f(x^{(k)})\|^2. \tag{3.3}$$

Remark 3.4. Assumption (3.3) is commonly used in the global convergence analysis of Newton-type methods and can be enforced in practice by an Armijo-type line search together with an acceptance (or damping) rule for the corrector step.

Lemma 3.5. *Let $F(x) = \frac{1}{2}\|f(x)\|^2$. If* (3.3) *holds, then $\|f(x^{(k)})\| \to 0$ as $k \to \infty$.*

Proof. From (3.3),

$$c\,\|f(x^{(k)})\|^2 \leq F(x^{(k)}) - F(x^{(k+1)}).$$

Summing from $k = 0$ to N yields

$$c\sum_{k=0}^{N}\|f(x^{(k)})\|^2 \leq F(x^{(0)}) - F(x^{(N+1)}) \leq F(x^{(0)}),$$

since $F \geq 0$. Letting $N \to \infty$ gives

$$\sum_{k=0}^{\infty}\|f(x^{(k)})\|^2 < \infty,$$

hence $\|f(x^{(k)})\| \to 0$. □

Lemma 3.6. *Assume that the singular values of A exceed* 1 *and that Assumption* (3.3) *holds. Then, the sequence $\{x^{(k)}\}$ generated by the improved NTFPI method* (3.1) *is bounded.*

Proof. By Eq (3.1), we have

$$\begin{aligned} y^{(k)} &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} f(x^{(k)}) \\ &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} (Ax^{(k)} - \left|x^{(k)}\right| - b) \\ &= x^{(k)} - \left(A - H(x^{(k)})\right)^{-1} [(Ax^{(k)} - H(x^{(k)})x^{(k)} - b)] \\ &= x^{(k)} - x^{(k)} + \left(A - H(x^{(k)})\right)^{-1} b \\ &= \left(A - H(x^{(k)})\right)^{-1} b. \end{aligned}$$

Hence, $y^k = (A - H(x^k))^{-1} b$. Substituting into the second step of formula (3.1), i.e.,

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

we obtain

$$\begin{aligned} x^{(k+1)} &= (1 - \beta_k) y^{(k)} + \beta_k (A^{-1}|y^{(k)}| + A^{-1}b) \\ &= (1 - \beta_k) y^{(k)} + \beta_k A^{-1}(Ay^{(k)} - f(y^{(k)})) \\ &= y^{(k)} - \beta_k y^{(k)} + \beta_k y^{(k)} - \beta_k A^{-1}(f(y^{(k)})) \\ &= (A - H(x^k))^{-1} b - \beta_k A^{-1}(f(y^{(k)}), \\ \left\|x^{(k+1)}\right\| &= \left\|(A - H(x^k))^{-1} b - \beta_k A^{-1}(f(y^{(k)})\right\| \\ &\leq \left\|(A - H(x^k))^{-1}\right\| \|b\| + \|\beta_k\| \left\|A^{-1}\right\| \left\|(f(y^{(k)})\right\|. \end{aligned}$$

By Assumption (3.3) and Lemma 3.5, we have

$$\|f(y^{(k)})\| \to 0 \quad \text{as } k \to \infty.$$

Thus,

$$\left\|x^{(k+1)}\right\| \leq \left\|(A - H(x^k))^{-1}\right\| \|b\|.$$

Since $(A - H(x^k))^{-1}$exists and $H(x^k)$ have elements that are either ± 1or 0, then $(A - H(x^k))^{-1}$is bounded. We find that the sequence $\left\{x^{(k)}\right\}$ is bounded. The proof is complete. □

Theorem 3.7. *Let* $\{x^{(k)}\}$ *be generated by Algorithm 1 (NTFPI) for a solvable AVE*

$$Ax - |x| - b = 0.$$

Assume that $\|A^{-1}\| < 1$ *(equivalently,* $\sigma_{\min}(A) > 1$ *for the* 2*-norm) and that Assumption* (SD) *holds, i.e., for* $F(x) = \frac{1}{2}\|f(x)\|^2$ *there exists* $c \in (0, 1)$ *such that*

$$F(x^{(k+1)}) \leq F(x^{(k)}) - c\,\|f(x^{(k)})\|^2, \qquad k \geq 0.$$

Then, $\{x^{(k)}\}$ *converges from any starting point* $x^{(0)}$ *to a solution* x^* *of the AVE.*

Proof. First, we define a function $f(x)$ associated with AVE (1.1) as $f(x) = Ax - |x| - b$. Suppose that x^* is a solution of AVE (1.1) and let $x^{(k+1)}$ be the $(k+1)$-th iterate generated by Algorithm 1.

Observe that the predictor step can be algebraically expressed as computed by Algorithm 1:

$$y^{(k)} - x^{(k)} = \left(A - H(x^{(k)})\right)^{-1} f(x^{(k)}).$$

From the corrector step, we have

$$x^{(k+1)} = (1-\beta_k)y^k + \beta_k(A^{-1}\left|y^k\right| + A^{-1}(b)).$$

Then, we compute the difference between $Ax^{(k+1)}$ and Ax^*:

$$\begin{aligned} A(x^{(k+1)} - x^*) &= (1-\beta_k)A(y^{(k)} - x^*) + \beta_k\left(|y^{(k)}| + b - Ax^*\right) \\ &= (1-\beta_k)A(y^{(k)} - x^*) + \beta_k(|y^{(k)}| - |x^*|), \\ x^{(k+1)} - x^* &= (1-\beta_k)(y^{(k)} - x^*) + A^{-1}\beta_k(|y^{(k)}| - |x^*|). \end{aligned}$$

We know that for any vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,

$$\| |x| - |y| \| \leq \|x - y\|.$$

Applying this inequality, we obtain

$$\begin{aligned} \|x^{(k+1)} - x^*\| &\leq (1-\beta_k)\|y^{(k)} - x^*\| + \|A^{-1}\|\beta_k\||y^{(k)}| - |x^*|\| \\ &\leq \left[(1-\beta_k) + \|A^{-1}\|\beta_k\right]\|y^{(k)} - x^*\| \\ &= \left[1 - \beta_k(1 - \|A^{-1}\|)\right]\|y^{(k)} - x^*\|. \end{aligned}$$

Let $\alpha_k := 1 - \beta_k(1 - \|A^{-1}\|)$, where $\|A^{-1}\| = K < 1$ and $\beta_k \in (0, 1]$, hence $\alpha_k < 1$.

To estimate $\|x^{(k+1)} - x^*\|$, we write

$$\begin{aligned} \left\|x^{k+1} - x^*\right\| &= \alpha_k \left\|y^k - x^*\right\| \\ &\leq \alpha_k \left[\left\|y^k - x^k\right\| + \left\|x^k - x^*\right\|\right] \\ &\leq \alpha_k \left[\left\|\left(A - H(x^k)\right)^{-1} f(x^k)\right\| + \left\|x^k - x^*\right\|\right] \\ &\leq \alpha_k \left\|x^k - x^*\right\| + \alpha_k \left\|\left(A - H(x^k)\right)^{-1} f(x^k)\right\|. \end{aligned}$$

Let

$$\varepsilon_k = \alpha_k \left\|\left(A - H(x^k)\right)^{-1} f(x^k)\right\|.$$

By the assumption (SD) and Lemma 3.5, we have $\|f(x^{(k)})\| \to 0$. Moreover, since $H(x^{(k)})$ is diagonal with entries in $\{-1, 0, 1\}$, there are only finitely many possible matrices H. Under $\|A^{-1}\| < 1$ (equivalently $\sigma_{\min}(A) > 1$), $A - H$ is nonsingular for all such H; hence, $\{(A - H(x^{(k)}))^{-1}\}$ is uniformly bounded. Therefore, there exists $M > 0$ such that

$$\|(A - H(x^{(k)}))^{-1}\| \leq M \quad \text{for all } k.$$

Since $\alpha_k \leq q < 1$ for all k, we have

$$\varepsilon_k \leq q\, M \|f(x^{(k)})\| \to 0$$

and

$$\left\|x^{k+1} - x^*\right\| \leq q \left\|x^k - x^*\right\| + \varepsilon_k.$$

From this we deduce that

$$\left\|x^{k+1} - x^*\right\| \leq q^k \left\|x^1 - x^*\right\| + \sum_{j=0}^{k} q^{k-j}\varepsilon_j.$$

Let

$$S_k = \sum_{j=0}^{k} q^{k-j}\varepsilon_j,$$

and $\delta > 0$.

Since $\varepsilon_j \to 0$, there is $N \in \mathbb{N}$ such that $\forall\, j > N$, $\varepsilon_j < \delta(1-q)$:

$$\begin{aligned} S_k &= \sum_{j=0}^{k} q^{k-j}\varepsilon_j \\ &= \sum_{j=0}^{N-1} q^{k-j}\varepsilon_j + \sum_{j=N}^{k} q^{k-j}\varepsilon_j \\ &\leq q^k \left(\sum_{j=0}^{N-1} q^{-j}\varepsilon_j\right) + \sum_{j=N}^{k} q^{k-j}\delta(1-q) \\ &= q^k C + \delta(1-q)\sum_{j=N}^{k} q^{k-j}, \end{aligned}$$

where

$$\begin{aligned} C &= \left(\sum_{j=0}^{N-1} q^{-j}\varepsilon_j\right) \\ &= q^k C + \delta\left(\frac{1-q}{1-q}\right) \\ &= q^k C + \delta. \end{aligned}$$

Since $q^k \to 0$ as $k \to \infty$ and δ is arbitrary, it implies that $S_k \to 0$ as $k \to \infty$. This shows that $\left\|x^{k+1} - x^*\right\| \to 0$. Consequently, $\left\{x^{k+1}\right\}$ converges to x^*. The proof is complete. □

4. Numerical examples

To evaluate the effectiveness of the proposed NTFPI method for solving AVEs, we present three benchmark examples. Among them, the first example is designed not only to test the general performance of the method, but also to investigate the impact of the parameter β on convergence

behavior and accuracy. Specifically, we study both fixed and variable β schemes while ensuring $\beta \in (0, 1]$, which is consistent with the theoretical convergence guarantees established earlier. In the remaining two examples, we focus exclusively on comparing the performance of the NTFPI method (3.1) with three established methods: the NT as shown in (2.3), the MFPI defined in (2.5), and the TSI method described in (2.2). Performance is assessed based on the iteration count (IT), the final residual (RES), and the execution time (T) in seconds. The execution time was measured using the built-in `tic()` and `toc()` functions, representing the total elapsed wall-clock time, while matrix inversions were computed using the `inv(A)` command. The residual error, serving as a stopping criterion, is defined as:

$$\text{RES} = \frac{\|Ax_k - |x_k| - b\|}{\|b\|}.$$

The iterative process was terminated if the convergence criterion RES $\leq 10^{-6}$ was satisfied or if the maximum number of iterations $k_{\max} = 1000$ was reached. For all subsequent tables, IT denotes the number of iterations, RES represents the final residual, and T signifies the execution time in seconds. All numerical simulations were conducted using Scilab 2025.1.0 on a machine equipped with an advanced micro devices (AMD) Ryzen 5 5600H processor and 8 GB RAM.

Example 4.1. [26] We consider AVE (1.1) where the matrix A is given as

$$A = \mu I_{m^2} + D,$$

with I being the identity matrix, and D a specific matrix defined as in [26]. The parameter μ affects the solvability of the AVE and must comply with the conditions stated in Theorem 1 to ensure convergence. For this experiment, the initial vector was set to zero, and we adopt the values of $\mu = 2.4$ and the vectors b and x of [26], and consider the dimension of the matrix $n = m^2 = 40^2$.

The main focus of this experiment is to investigate the impact of the parameter β_k on the NTFPI method. To do this, we consider two types of settings. In the first setting, β_k is kept constant throughout the iterations, with values taken from the set $\{0.1, 0.2, \ldots, 1.0\}$. In the second setting, we use a variable β_k defined by the formula

$$\beta_k = \frac{1}{(1 + k)^p},$$

where k denotes the iteration index and the exponent p is varied from 1 to 13. These configurations allow us to compare the performance of different strategies for controlling β_k in terms of convergence behavior and computational efficiency.

From Table 1, the numerical results indicate that the method converges for all tested choices of $\beta_k \in (0, 1]$, which is consistent with the theoretical requirement. Among the fixed values considered, $\beta_k = 0.3$ yields the smallest final residual, while the residual (RES) and CPU time vary only slightly for nearby values of β_k. A closer examination using finely tuned values ($\beta_k \in [0.31, 0.39]$) confirmed that the residual remains consistently low within this range, with the lowest RES at $\beta_k = 0.33$. In contrast, applying a variable scheme β of the form $\beta_k = \frac{1}{(1+k)^p}$ yielded even better accuracy. The residual gradually decreased as p increased and became stable for $p \geq 11$, achieving the smallest observed residual of 6.865×10^{-16}.

Table 1. Condensed performance summary of NTFPI under various β schemes ($n = 40^2$).

β_k type	Fixed β_k ($\beta_k \in (0, 1]$)
Optimal value/range	$\beta_k = 0.3$
Min RES	1.365×10^{-15}
Time range (s)	0.3407429 – 0.3973348
Observation	Low error; stable across $\beta \approx 0.3$
β_k type	Fine-tuned fixed β_k ($\beta_k \in [0.31, 0.39]$)
Optimal value/range	$\beta_k = 0.33$
Min RES	1.368×10^{-15}
Time range (s)	0.3433544 – 0.5130301
Observation	Slightly lower RES; minimal variation in this range
β_k type	Variable $\beta_k = \frac{1}{(1+k)^p}$
Optimal value/range	$p \geq 11$
Min RES	6.865×10^{-16}
Time range (s)	0.2698 – 0.4019
Observation	Lower residuals; stable performance for $p \geq 11$

To further evaluate the effectiveness of the proposed method, we compare the performance of NTFPI with that of the NT, MFPI, and TSI methods, and consider dimensions of the matrix $m = 40, 60, 80, 100$. The parameter settings are listed in Table 2, where the values of q_l and τ are adopted from [26]. The numerical results in Table 3 summarize the number of iterations, residual errors, and computational time for each method.

Table 2. Parameters for solving the AVE given in Example 4.1.

Parameter	$m = 40$	$m = 60$	$m = 80$	$m = 100$
q_l	0.4557	0.4571	0.4577	0.4579
τ	0.72	0.21	0.09	0.059
β_k at iteration k	$\frac{1}{(1+k)^{11}}$	$\frac{1}{(1+k)^{11}}$	$\frac{1}{(1+k)^{12}}$	$\frac{1}{(1+k)^{12}}$

Table 3. Comparison results of different methods for solving the AVE.

Method	Metric	$m = 40$	$m = 60$	$m = 80$	$m = 100$
NT	IT	2	2	2	2
	RES	1.0861×10^{-15}	1.348×10^{-15}	1.447×10^{-15}	1.509×10^{-15}
	T	0.7356788	12.534786	76.528237	824.29731
MFPI	IT	25	100	240	369
	RES	7×10^{-7}	9×10^{-7}	1×10^{-6}	1×10^{-6}
	T	0.3550316	6.4229568	49.309875	205.89427
TSI	IT	2	2	2	2
	RES	1.222×10^{-15}	1.080×10^{-15}	1.035×10^{-15}	1.010×10^{-15}
	T	0.3775448	6.0099733	42.704359	134.77372
NTFPI	IT	2	2	2	2
	RES	6.865×10^{-16}	6.676×10^{-16}	6.411×10^{-16}	6.728×10^{-16}
	T	0.3839002	6.0233779	42.558493	136.34545

As observed in Table 3, the NTFPI method achieves accurate solutions and performs competitively compared to MFPI, NT, and TSI. The results also indicate that introducing the relaxation parameter β_k

in the corrector step can improve the convergence behavior and that the method is reasonably stable over a range of values β_k.

4.1. Role of the relaxation parameter β_k.

In Algorithm 1, the corrector step can be written as a Mann-type relaxation

$$x^{(k+1)} = (1-\beta_k)\, y^{(k)} + \beta_k\, T(y^{(k)}), \qquad T(y) = A^{-1}(|y| + b),$$

so $\beta_k \in (0, 1]$ controls the balance between the Newton predictor $y^{(k)}$ and the fixed-point correction $T(y^{(k)})$. The convergence analysis only requires $\beta_k \in (0, 1]$ (and in our contraction estimate, $\beta_k \geq \underline{\beta} > 0$), but the numerical performance depends on the trade-off induced by β_k. Indeed, using $|||u|-|v||| \leq \|u-v\|$, one obtains the estimate

$$\|x^{(k+1)} - x^*\| \leq ((1-\beta_k) + \beta_k\|A^{-1}\|)\|y^{(k)} - x^*\| = (1-\beta_k(1-\|A^{-1}\|))\|y^{(k)} - x^*\|.$$

Hence, a larger β_k strengthens the contraction of the corrector when $y^{(k)}$ is already close to x^*. However, $y^{(k)}$ is produced by a Newton-type predictor involving $(A - H(x^{(k)}))^{-1}$, and when the sign pattern is still changing (especially near components close to zero), the predictor may exhibit oscillations due to sign switching. In such cases, overly large β_k can propagate this instability in the update, while overly small β_k reduces the benefit of the correction. This explains why intermediate values of β_k often provide the best practical performance in Example 4.1. For the larger-scale settings in Examples 4.2 and 4.3, we also include a short sensitivity check on a representative large-scale instance (e.g., $\beta \in \{0.1, 0.33, 0.9\}$) to illustrate the same stability–speed trade-off.

Example 4.2. (Linear complementarity problem [29]) We consider the transformation of a linear complementarity problem (LCP) into an equivalent AVE.

For $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$, the LCP is defined as the problem of finding $z \geq 0$ such that

$$w = Mz + q \geq 0, \text{and } z^T w = 0.$$

Let $w = |x| - x$ and $z = |x| + x$. Then, it follows that $w \geq 0$, $z \geq 0$, and $z^T w = 0$, satisfying the complementarity conditions. Substituting these expressions into the LCP yields the AVE:

$$Ax - |x| = b,$$

where $A = (I - M)^{-1}(I + M)$ and $b = -(I - M)^{-1}q$, provided that $(I - M)$ is invertible. If x solves this AVE, then $z = |x| + x \geq 0$ is a solution to the original LCP. We apply all four methods (NT, MFPI, TSI, NTFPI) to solve the resulting AVE for the large-scale case $n = 3000$.

In this experiment, the matrix $M = (a_{ij}) \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$ are defined as:

$$a_{ij} = \begin{cases} 0.6, & if\ i = j, \\ -0.01, & if\ |i-j| = 1, \\ 0, & otherwise, \end{cases}$$

and $q = -e$, where e is the vector of all ones.

Two different initial points are considered: $x_0^{(1)} = (0.001, ...0.001)^T$ and $x_0^{(2)} = (0.9, ...0.9)^T$. The exact solution is $x^* = (0.8477, 0.8618, 0.8621, ..., 0.8621, 0.8618, 0.8477)^T$ which corresponds to the LCP solution $z^* = (1.6954, 1.7237, 1.7241, ..., 1.7241, 1.7237, 1.6954)^T$.

In this experiment, we focus on solving the AVE reformulated from the LCP with parameters $q_l = 0.2757435$ and $\tau = 0.09$. To determine the value of the constant iteration parameter β_k as defined in the NTFPT method, we conducted preliminary numerical tests on a range of values, including $0.1, 0.33, 0.9, 1/(1+k)$, and $1/(1+k)^{11}$, where k denotes the iteration index at each step.

The empirical results indicated that the number of iterations required for convergence remained identical for all the values tested for β_k, and the resulting residuals were nearly equivalent. However, high-precision analysis revealed slight differences in the final precision: For the initial guess $x_0^{(1)} = (0.001, ...0.001)^T$, the value $\beta_k = 0.33$ yielded the minimum residual of 1.736×10^{-17}. In contrast, for $x_0^{(2)} = (0.9, ...0.9)^T$, the lowest residual of 1.868×10^{-16} was obtained at $\beta_k = 0.1$.

Table 4 shows that the proposed NTFPI method consistently delivers the most efficient performance at both initial points. It converges in a single iteration with very low residuals (e.g., 1.736×10^{-17}) and requires the shortest computation time, particularly 2.24 seconds for $x_0^{(1)}$ and only 2.22 seconds for $x_0^{(2)}$. In comparison, NT and TSI also converge quickly with similar accuracy but take more time. However, the MFPI method requires 184 iterations and results in significantly larger residuals, indicating slower and less accurate convergence.

Table 4. Comparison of numerical results solving the AVE reformulated from LCP.

Initial point	Method	IT	RES	T (s)
$x_0^{(1)}$	NT	1	1.873×10^{-16}	4.6624966
	MFPI	184	1×10^{-6}	8.2533287
	TSI	2	1.909×10^{-16}	4.5333532
	NTFPI	1	1.736×10^{-17}	2.2413566
$x_0^{(2)}$	NT	1	1.867×10^{-16}	4.6548811
	MFPI	184	1.0×10^{-6}	8.175917
	TSI	1	3.729×10^{-16}	2.2436887
	NTFPI	1	1.868×10^{-16}	2.2203792

These results suggest that NTFPI is accurate and efficient for this test problem and remains stable with respect to the two different initial guesses considered. Its balance of speed and precision demonstrates the practical effectiveness of the method for solving AVEs reformulated from LCPs.

Example 4.3. (Equilibrium problem of the hydrodynamic model [31]) We consider an equilibrium problem arising from a hydrodynamic model, described by the following non-differentiable algebraic equations:

$$Bx + \max(0, x) = c,$$

where $B \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$ are given. Using the identity

$$\max(a, b) = \frac{1}{2}(a - b + |a - b|)$$

in terms of equality, this hydrodynamic equation can be reformulated as an AVE. In particular, it

becomes

$$Bx + \frac{1}{2}(x + |x|) = c \Leftrightarrow Ax - |x| - b = 0,$$

where $A = -(2B + I)$ and $b = -2c$.

We now consider a specific case of the hydrodynamic model in which matrix $B \in \mathbb{R}^{n\times n}$ and vector c are given by

$$b_{ij} = \begin{cases} -25.5, & \text{if } i = j, \\ -2.5, & \text{if } |i - j| = 1, \\ 0, & \text{otherwise,} \end{cases}$$

and $c = (-27, -29.5, -29.5..., -29.5, -29.5, -27)^T$. In the experiments, two initial guesses were used: $x_0^{(1)} = (0.02, ..., 0.02)^T$ and $x_0^{(2)} = (0.9, ..., 0.9)^T$. We apply all four methods to solve the resulting AVE with problem size $n = 3000$. The computational results are summarized in Table 5. The exact solution to this example is $x^* = e$.

Table 5. Comparison of numerical results solving the equilibrium problem.

Initial point	Method	IT	RES	T (s)
$x_0^{(1)}$	NT	1	2.404×10^{-16}	3.5805525
	MFPI	6	1.0×10^{-6}	0.1913904
	TSI	2	1.008×10^{-17}	3.3092811
	NTFPI	1	9.329×10^{-18}	1.6562307
$x_0^{(2)}$	NT	1	0	3.7844467
	MFPI	6	1×10^{-6}	0.2004898
	TSI	1	0	1.6924733
	NTFPI	1	0	1.6333215

Similarly to Example 4.2, we conducted preliminary numerical tests for Example 4.3 to determine the optimal value of the iteration parameter β_k for the NTFPI method. We evaluated a range of candidate values, including $0.1, 0.33, 0.9, 1/(1 + k)$, and $1/(1 + k)^{11}$, where k denotes the iteration index at each step of the computational process. The empirical results indicated that the number of iterations (IT) required for convergence remained identical for all the values tested of β_k, with the resulting residuals nearly equivalent. However, high-precision analysis revealed slight differences in numerical accuracy. Specifically, for the initial guess $x_0^{(1)} = (0.02, ..., 0.02)^T$, the value $\beta_k = 0.1$ yielded the minimum residual of 9.329×10^{-18}. In particular, for $x_0^{(2)} = (0.9, ..., 0.9)^T$, the lowest residual achieved was 0 at $\beta_k = 0.1$. Based on these findings, which demonstrate superior numerical precision, we selected $\beta_k = 0.1$ as the constant iteration parameter for the performance comparisons in this example. For the MFPI method, the parameters were chosen as follows: $q_l = 0.250078$, and the relaxation parameter $\tau = 1.11$. These parameter values were selected on the basis of empirical tuning to ensure the stability and convergence of the proposed iterative methods for large-scale systems.

As shown in Table 5, all the methods discussed are capable of converging to the exact solution with high numerical precision. The NT, TSI, and NTFPI methods demonstrate extremely rapid convergence, requiring only 1 to 2 iterations to satisfy the stopping criteria. In contrast, while the MFPI method successfully reaches the defined threshold of 1.0×10^{-6} in 6 iterations, it does not

achieve the extreme high-precision depths (residuals 10^{-18} or 0) reached by the other techniques. Despite the higher iteration count, the MFPI method records the lowest total execution time (approximately 0.20 seconds), which validates the theoretical insight that its computational cost per iteration is the lowest among the four methods. This efficiency stems from its reliance on constant matrices A and Q, allowing for pre-computed factorizations rather than the repeated state-dependent matrix inversions required by the other schemes.

In summary, for large-scale hydrodynamic equilibrium problems ($n = 3000$), the proposed NTFPI method demonstrates superior performance by balancing convergence rate and computational efficiency. It achieves the highest numerical accuracy (minimum residual of 9.329×10^{-18}) while being more than twice as fast as the NT method in terms of execution time. For applications where extreme precision is not required, but speed is the priority, MFPI remains highly competitive due to its minimal computational overhead per iteration.

Example 4.4. (Stochastic matrices and global scaling) In response to the requirement for testing a wider class of matrices beyond structured tridiagonal forms, we evaluate our proposed method using **fully random matrices** with controlled rank. Following the methodology described in [32–34], we generate stochastic matrices $A \in \mathbb{R}^{n\times n}$ utilizing the **GenRand Algorithm 5.1**. This algorithm constructs a base matrix A_{base} by concatenating a normally distributed block $C \in \mathbb{R}^{\rho\times n}$, whose entries are generated from $N(0, 1)$, and a uniform block normalized by rows BC, where $B \in \mathbb{R}^{(n-\rho)\times\rho}$ contains entries from $U[-1, 1]$. This construction allows for the rigorous evaluation of the solver in both full-rank and rank-deficient stochastic systems. Moreover, the use of this GenRand-type test-matrix generation is consistent with recent studies on iterative solvers for matrix inversion and generalized inverses, and it may be useful for further development and comparison of this class of iterative methods [32, 34].

To guarantee the **unique solvability** of the AVE $Ax - |x| = b$, we adhere to the condition established of [26, Theorem 2.1], which requires the spectral norm of the inverse matrix to satisfy $\|A^{-1}\| < 1$. Because randomly generated matrices frequently exceed this bound, we implement a **global scaling** approach to maintain theoretical convergence. After generating A_{base}, we compute its inverse spectral norm $\nu_{base} = \|A_{base}^{-1}\|_2$. We then define a scaling factor $\lambda = \nu_{base}/0.5$ and set the final test matrix as $A = \lambda A_{base}$.

The experimental setup is completed by selecting a ground-truth solution vector x^* from a uniform distribution on $[-1, 1]$. This ensures a variety of signs in the solution components, which is essential for testing the non-linear absolute value term. The right-hand side is then computed as $b = Ax^* - |x^*|$, and all numerical comparisons are initiated from a shared initial guess $x_0 = \mathbf{0}$. For all dimensions of the problem tested, $n = 600, 1000, 2000$, the parameters q_l and τ for the MFPI method were selected to strictly satisfy the theoretical convergence conditions established by Yu et al. [26]. Furthermore, for the NTFPI method, the iteration parameter was defined as $\beta_k = 1/(k + 1)^{11}$, where k represents the iteration index.

The numerical results presented in Table 6 demonstrate that all the methods tested successfully converged for large-scale random matrices. In particular, the NTFPI and TSI accelerated schemes exhibited superior precision, achieving residual errors (RES) as low as 10^{-15} for $n = 1000$ within only two iterations, highlighting the effectiveness of high-order convergence structures. Although the **MFPI** method required more iterations for higher dimensions, it remained highly competitive in terms of efficiency, producing the fastest execution time of 0.3846 seconds for $n = 2000$. This

performance is in alignment with its design for computational simplicity and efficiency under the condition $\|A^{-1}\| < 1$ [26]. Overall, the NTFPI method significantly outperformed the standard Newton (NT) method as the problem dimension increased.

Table 6. Comparison of numerical results solving the random matrices.

Dimensuin	Method	IT	RES	T (s)
$n = 600$	NT	1	2×10^{-7}	0.0286376
	MFPI	1	6×10^{-7}	0.0011916
	TSI	1	4.199×10^{-9}	0.0100349
	NTFPI	1	4.142×10^{-9}	0.0121051
$n = 1000$	NT	2	1.512×10^{-8}	0.2126719
	MFPI	31	9×10^{-7}	0.1020308
	TSI	2	1.599×10^{-15}	0.0984531
	NTFPI	2	1.566×10^{-15}	0.0918848
$n = 2000$	NT	2	9.062×10^{-10}	2.1151617
	MFPI	25	9×10^{-7}	0.3846135
	TSI	1	6×10^{-7}	0.4335239
	NTFPI	1	6×10^{-7}	0.4483368

4.2. Computational cost analysis

To compare the efficiency of the considered methods, we examine the computational cost per iteration in terms of Jacobian evaluations, linear system solutions, and matrix characteristics.

Table 7 summarizes the computational effort per iteration for the considered methods based on their respective mathematical formulations. The TSI method maintains a relatively low per-iteration cost, requiring only one Jacobian evaluation ($H(x^k)$ and a single linear system solution involving the state-dependent matrix ($A - H(x^k)$. In contrast, the NT method is the most computationally intensive, necessitating three evaluations of the Jacobian-like function f' and two linear solutions, including the inversion of a complex weighted matrix sum in every step. The MFPI method is uniquely efficient for large-scale problems as it involves zero Jacobian evaluations; by utilizing constant matrices A and Q, its factorizations can be pre-computed, reducing each iteration to relatively inexpensive forward/backward substitutions. The NTFPI method adopts a mixed strategy to balance convergence and cost, combining one state-dependent Jacobian evaluation with two linear system solves, one with the variable matrix ($A - H(x^k)$ and the other with the constant matrix A. Execution times measured via Scilab's `tic` and `toc` functions confirm that while methods with higher per-iteration complexity may reduce total iterations, the lower operational overhead of the MFPI and NTFPI schemes often results in superior overall efficiency for high-dimensional equilibrium problems.

Table 7. Comparison of computational cost per iteration.

Method	Evaluations	Linear solves	Matrix type
TSI	1	1	State-dependent ($A - H(x^{(k)})$)
NT	3	2	State-dependent (weighted)
MFPI	0	2	Constant (A, Q)
NTFPI	1	2	Mixed

The scripts used to generate the numerical experiments are publicly available at: https://github.com/nifatamah-psu/numerical-AVE-experiments.git.

5. Conclusions

This study introduced the NTFPI method, a novel approach to solving AVEs that significantly enhances convergence speed and computational stability. Numerical experiments showed that NTFPI performs competitively compared to NT, MFPI, and TSI, often reducing the iteration count while maintaining high accuracy. The observed CPU time depends on the problem structure and implementation details, and NTFPI is particularly effective in several large-scale or challenging instances reported in this paper. Its efficient correction step makes it a superior alternative, especially for large-scale or challenging AVE problems where other iterative methods fall short. The proposed NTFPI method suggests several concrete avenues for future research. (i) *Adaptive selection of* β_k: In the current paper, β_k is chosen as a fixed constant in each example to highlight the stability–speed trade-off. A natural next step is to design adaptive rules that update β_k based on inexpensive indicators, such as the relative decrease of the objective function $F(x) = \frac{1}{2}\|f(x)\|^2$, the ratio $\|f(x^{(k+1)})\|/\|f(x^{(k)})\|$, or the switch frequency of the sign pattern. For example, one may increase β_k when the sign pattern stabilizes and decrease β_k when oscillations occur. (ii) *Refined convergence theory*: While our analysis guarantees convergence under standard assumptions, it is worthwhile to establish stronger results such as local superlinear rates under stabilized sign patterns, global convergence with weaker or more verifiable conditions than those currently used, and convergence guarantees under nonsmooth frameworks using Clarke-type generalized derivatives. (iii) *Large-scale sparse AVEs*: Many applications lead to sparse and high-dimensional matrices A. Thus, developing sparse-aware implementations is important, including reusing factorizations, exploiting the diagonal structure of $H(x^{(k)})$, and employing preconditioned Krylov methods to solve systems involving $A - H(x^{(k)})$ efficiently.

Author contributions

Aniruth Phon-On: conceptualization, methodology, formal analysis, Theoretical proof of the main results, Original writing draft, and Investigation. Nifatamah Makaje: conceptualization, methodology, validation, investigation, selection and refinement of numerical examples, development and implementation of the four computational methods, writing–original draft, writing–review, and editing, supervision. Asma Yafad: searching and proposing numerical examples, verification of solvability conditions, preliminary coding and computational testing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors thank anonymous referees for their valuable suggestions in this paper. This research was supported by the Faculty of Science and Technology, Prince of Songkla University, Pattani Campus, Thailand.

Conflict of interest

The authors declare no conflicts of interest.

References

1. J. Rohn, A theorem of the alternatives for the equation $Ax + B|x| = b$, *Linear Multilinear Algebra*, **52** (2004), 421–426. https://doi.org/10.1080/0308108042000220686
2. X. H. Shao, W. C. Zhao, Relaxed modified Newton-based iteration method for generalized absolute value equations, *AIMS Math.*, **8** (2023), 4714–4725. https://doi.org/10.3934/math.2023233
3. S. L. Wu, S. Q. Shen, On the unique solution of the generalized absolute value equation, *Optim. Lett.*, **15** (2021), 2017–2024. https://doi.org/10.1007/s11590-020-01672-2
4. O. L. Mangasarian, R. R. Meyer, Absolute value equation, *Linear Algebra Appl.*, **419** (2006), 359–367.
5. R. W. Cottle, G. Dantzig, Complementary pivot theory of mathematical programming, *Linear Algebra Appl.*, **1** (1968), 103–125.
6. O. L. Mangasarian, Absolute value equation solution via concave minimization, *Optim. Lett.*, **1** (2007), 3–8. https://doi.org/10.1007/s11590-006-0005-6
7. F. K. Haghani, On generalized Traub's method for absolute value equations, *J. Optim. Theory Appl.*, **166** (2015), 619–625. https://doi.org/10.1007/s10957-015-0712-1
8. O. A. Prokopyev, On equivalent reformulations for absolute value equations, *Comput. Optim. Appl.*, **44** (2009), 363–372. https://doi.org/10.1007/s10589-007-9158-1
9. C. X. Li, A modified generalized Newton method for absolute value equations, *J. Optim. Theory Appl.*, **170** (2016), 1055–1059. https://doi.org/10.1007/s10957-016-0956-4
10. C. X. Li, A preconditioned AOR iterative method for the absolute value equations, *Int. J. Comput. Methods*, **14** (2017), 1750016. https://doi.org/10.1142/S0219876217500165
11. L. Abdallah, M. Haddou, T. Migot, Solving absolute value equation using complementarity and smoothing functions, *J. Comput. Appl. Math.*, **327** (2018), 196–207.
12. A. J. Fakharzadeh, N. N. Shams, An efficient algorithm for solving absolute value equations, *J. Math. Extension*, **15** (2021), 1–23.

13. R. Ali, Z. Zhang, F. A. Awwad, The study of new fixed-point iteration schemes for solving absolute value equations, *Heliyon*, **10** (2024), e34505. https://doi.org/10.1016/j.heliyon.2024.e34505

14. Y. F. Ke, C. F. Ma, SOR-like iteration method for solving absolute value equations, *Appl. Math. Comput.*, **311** (2017), 195–202. https://doi.org/10.1016/j.amc.2017.05.035

15. J. Feng, S. Liu, An improved generalized Newton method for absolute value equations, *SpringerPlus*, **5** (2016), 1042. https://doi.org/10.1186/s40064-016-2720-5

16. J. Feng, S. Liu, A new two-step iterative method for solving absolute value equations, *J. Inequal. Appl.*, **2019** (2019), 39. https://doi.org/10.1186/s13660-019-1969-y

17. M. Dehghan, A. Shirilord, Matrix multisplitting Picard-iterative method for solving generalized absolute value matrix equation, *Appl. Numer. Math.*, **158** (2020), 425–438. https://doi.org/10.1016/j.apnum.2020.08.001

18. A. Khan, J. Iqbal, A. Akgül, R. Ali, Y. Du, A. Hussain, et al., A Newton-type technique for solving absolute value equations, *Alex. Eng. J.*, **64** (2023), 291–296. https://doi.org/10.1016/j.aej.2022.08.052

19. R. Ali, A. Ali, M. M. Alam, A. Mohamed, Numerical solution of the absolute value equations using two matrix splitting fixed point iteration methods, *J. Funct. Spaces*, **2022** (2022), 7934796. https://doi.org/10.1155/2022/7934796

20. R. Ali, F. A. Awwad, E. A. A. Ismail, The development of new efficient iterative methods for the solution of absolute value equations, *AIMS Math.*, **9** (2024), 22565–22577. https://doi.org/10.3934/math.20241098

21. R. Ali, K. Pan, A. Ali, Two new iteration methods with optimal parameters for solving absolute value equations, *Int. J. Appl. Comput. Math.*, **8** (2022), 123. https://doi.org/10.1007/s40819-022-01324-2

22. R. Ali, K. Pan, Two new fixed point iterative schemes for absolute value equations, *Jpn. J. Ind. Appl. Math.*, **40** (2023), 303–314. https://doi.org/10.1007/s13160-022-00526-x

23. M. Guo, Q. Wu, Two effective inexact iteration methods for solving the generalized absolute value equations, *AIMS Math.*, **7** (2022), 18675–18689. https://doi.org/10.3934/math.20221027

24. S. X. Miao, X. T. Xiong, J. Wen, On Picard-SHSS iteration method for absolute value equation, *AIMS Math.*, **6** (2021), 1743–1753. https://doi.org/10.3934/math.2021104

25. H. Zhou, S. L. Wu, On the unique solution of a class of absolute value equations $Ax - B|Cx| = d$, *AIMS Math.*, **6** (2021), 8912–8919. https://doi.org/10.3934/math.2021517

26. D. Yu, C. Chen, D. Han, A modified fixed point iteration method for solving the system of absolute value equations, *Optimization*, **71** (2020), 449–461. https://doi.org/10.1080/02331934.2020.1804568

27. A. Yafad, N. Makaje, A. Phon-On, Numerical solution of the absolute value equations using two step iterative method, *Proceedings 50th International Congress on Science, Technology and Technology-based Innovation (STT 50)*, 2024.

28. W. R. Mann, Mean value methods in iteration, *Proc. Amer. Math. Soc.*, **4** (1953), 506–510. https://doi.org/10.2307/2032162

29. M. Achache, N. Anane, On unique solvability and Picard's iterative method for absolute value equations, *Bull. Transilv. Univ. Brasov Ser. III Math. Comput. Sci.*, **13** (2021), 13–26. https://doi.org/10.31926/but.mif.2021.1.63.1.2

30. O. L. Mangasarian, A generalized Newton method for absolute value equations, *Optim. Lett.*, **3** (2009), 101–108. https://doi.org/10.1007/s11590-008-0094-5

31. N. Anane, M. Achache, Preconditioned conjugate gradient methods for absolute value equations, *J. Numer. Anal. Approx. Theory*, **49** (2020), 3–14. https://doi.org/10.33993/jnaat491-1197

32. M. Kostadinov, M. Krstić, K. Rajković, M. D. Petković, Adaptive coefficients iterative method for computing matrix inverse, *Linear Algebra Appl.*, **731** (2026), 277–305. https://doi.org/10.1016/j.laa.2025.11.016

33. M. D. Petković, M. S. Petković, Hyper-power methods for the computation of outer inverses, *J. Comput. Appl. Math.*, **278** (2015), 110–118. https://doi.org/10.1016/j.cam.2014.09.024

34. M. D. Petković, M. A. Krstić, K. P. Rajković, Rapid generalized Schultz iterative methods for the computation of outer inverses, *J. Comput. Appl. Math.*, **344** (2018), 572–584. https://doi.org/10.1016/j.cam.2018.05.048

AIMS Press