Journal of Mathematical Modeling
Vol. 13, No. 3, 2025, pp. 553-562. Research Article
JMM

An iterative method based on Simpson's $3/8$ rule to solve absolute value equations

Farzad Rahpeymaii[†][*], Majid Rostami[†]

[†]*Department of Mathematics, National University of Skills (NUS), Tehran, Iran*
Email(s): rahpeyma_83@yahoo.com, majid403rostami@yahoo.com

Abstract. Newton's method is one of the important algorithms for solving absolute value equations. In this paper, we introduce an efficient two-step iterative method to improve the Newton algorithm. The new method adopts the predictor-corrector technique in which the first step is generalized Newton method and the second step is based on Simpson's $3/8$ rule. Under some standard assumptions, the convergence of new method and its linear convergence rate are obtained. Numerical results show that the our method is efficient and robust to solve absolute value equations.

Keywords: Absolute value equation, Iteration scheme, Simpson rule, Convergence analysis.
AMS Subject Classification 2010: 65K10, 90C30, 65F10, 65B99.

1 Introduction

The generalized absolute value equation (GAVE) is of the form

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

in which $A, B \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $x \in \mathbb{R}^n$ is the unknown vector . Also, $|\cdot|$ denotes the absolute value function. For the first time, GAVE introduced by Rohn in [27] and then investigated by the other researchers, see [18, 25, 33]. For $B = -I$, the GAVE reduces to absolute value equation (AVE) [7, 13]

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

The AVE (2) has many applications in scientific computing and engineering. For example, linear complementarity problem (LCP), linear programming, convex quadratic programming and bimatrix games are equivalent to AVE [9, 26, 32]. Solving the AVE is an NP-hard problem [21] and the conditions of existence, non-existence and uniqueness the solution are investigated in [21, 29, 35]. So far, some numerical

[*]Corresponding author
Received: 7 November 2024 / Revised: 2 February 2025 / Accepted: 3 February 2025
DOI: 10.22124/jmm.2025.28919.2572

algorithms have been provided to obtain the approximate solution of AVE [3, 8, 14, 15, 17, 29–31]. The most important of these algorithms are non-smooth Newton method [22], mixed-type splitting technique [11], Levenberg-Marquardt technique [24], SOR-like method [16], Picard iteration method [37], Gauss-Seidel iteration method [10], a method based on interval matrix [28], complementarity and smoothing functions method [1], smoothing techniques for non-Lipschitz absolute value equations [36], alternating projections method [2], generalized Gauss-Seidel iteration method [4], iteration methods for solving absolute value equations for an M-matrix or strictly diagonally dominant matrix [5], fixed point method and the modified generalized Gauss-Seidel method [6] and the combination of Newton method and Simpson rule [19]. In this paper, we present a two-step iterative method for solving the absolute value equation (2). Our algorithm is based on predictor-corrector technique, in which the predictor step is Newton's algorithm and the corrector step is based on Simpson's $3/8$ rule.

The remainder of this paper is organized as follows. In Section 2, we introduce a new algorithm to solve AVE (2). Our iterative method has two steps which are based on Newton's method and Simpson's $3/8$ rule. The convergence of new method and linear convergence rate are established in Section 3. Numerical simulation of the new algorithm has been done for some standard examples in Section 4. Finally, some conclusions are given in Section 5.

2 Newton-$\frac{3}{8}$ Simpson method

Let

$$F(x) := Ax - |x| - b. \tag{3}$$

Solving the AVE is equivalent to solve the equation $F(x) = 0$. The generalized Jacobian of $|x|$ is defined by

$$D(x) := \partial|x| = \mathrm{diag}(\mathrm{sgn}(x)), \tag{4}$$

where $\mathrm{sgn}(x)$ is the signum function. Since $|x| = D(x)x$, we have

$$F(x) = \big(A - D(x)\big)x - b. \tag{5}$$

Hence, the generalized Jacobian of $F(x)$ is obtained by

$$\partial F(x) = A - D(x). \tag{6}$$

Recently, Alamgir Khan et al. [19] proposed an improved Newton-type technique for solving AVE. This algorithm has two steps where the first step is the generalized Newton technique (Predictor) and the second step is based on Simpson's rule (Corrector). This method is very effective to solve large systems.

Algorithm 1 (NSM)**: (Newton and Simpson methods for solving** AVE**).**

(S0) Let $A \in \mathbb{R}^{n\times n}$ be a non-singular matrix, $b \in \mathbb{R}^n$, the initial vector $x^0 \in \mathbb{R}^n$ and parameter $\varepsilon > 0$.

(S1) If $\big\|Ax^k - |x^k| - b\big\| \le \varepsilon$, stop.

(S2) Compute

$$\begin{cases} y^k = \big(A - D(x^k)\big)^{-1} b, \\ x^{k+1} = x^k - 6\left(\partial F(x^k) + 4\partial F\Big(\dfrac{x^k + y^k}{2}\Big) + \partial F(y^k)\right)^{-1} F(x^k). \end{cases} \tag{7}$$

Then, return to (S1).

The new method uses Simpson's $3/8$ rule instead of Simpson's rule. These integration rules and the errors of them are as follows [12]. For the Simpson's rule we have

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right), \qquad E=\frac{1}{2880}(b-a)^5\left|f^{(4)}(\xi)\right|,$$

and for the Simpson's $3/8$ rule:

$$\int_a^b f(x)dx \approx \frac{b-a}{8}\left(f(a)+3f\left(\frac{2a+b}{3}\right)+3f\left(\frac{a+2b}{3}\right)+f(b)\right), \qquad E=\frac{1}{6480}(b-a)^5\left|f^{(4)}(\xi)\right|,$$

in which $\xi \in (a,b)$. Hence, the new algorithm has the following advantages:

- The error of Simpson's $3/8$ rule is smaller than Simpson's rule.
- The Simpson $3/8$ rule can be used with the number of segments are multiples of 3.
- If iterations be far from the solution of absolute value equations, then Newton's method is suitable. But if x^k be close to the solution, then Simpson's $3/8$ rule generates new iterations by combining x^k and y^k.

Algorithm 2 (N$\frac{3}{8}$SM)**: (Newton and** $3/8$ **Simpson methods for solving** AVE**).**

(S0) Let $A \in \mathbb{R}^{n\times n}$ be a non-singular matrix, $b \in \mathbb{R}^n$, the initial vector $x^0 \in \mathbb{R}^n$ and parameter $\varepsilon > 0$.

(S1) If $\left\|Ax^k - |x^k| - b\right\| \le \varepsilon$, stop.

(S2) Compute

$$\begin{cases} y^k = \left(A - D(x^k)\right)^{-1} b, \\ x^{k+1} = x^k - 8\left(\partial F(x^k) + 3\partial F\left(\frac{2x^k+y^k}{3}\right) + 3\partial F\left(\frac{x^k+2y^k}{3}\right) + \partial F(y^k)\right)^{-1} F(x^k). \end{cases} \tag{8}$$

Then, return to (S1).

3 Convergence analysis

In this section, we investigate the convergence of Algorithm 2 to solve AVE.

Lemma 1. *[22] Let $\Omega \subseteq \mathbb{R}^n$. Then, for all $x,y \in \Omega$, we have*

$$\big\||x| - |y|\big\| \le 2\|x-y\|.$$

Now, we explain some necessary and sufficient conditions for the existence or non-existence of a solution and the unique solvability of the absolute value equations.

Proposition 1. *[23] Let $A \in \mathbb{R}^{n\times n}$ be invertible. If $\|A^{-1}\| < 1$, then the* AVE (2) *has a unique solution for any $b \in \mathbb{R}^n$.*

Proposition 2. *[23] If the singular values of $A \in \mathbb{R}^{n\times n}$ exceed 1, the* AVE (2) *is uniquely solvable for any $b \in \mathbb{R}^n$.*

Proposition 3. *[23] Let $0 \neq b \geq 0$ and $\|A\| < 1$. Then, the* AVE (2) *has no solution.*

Proposition 4. *[23] Let $r \in \mathbb{R}^n$. The* AVE (2) *has no solution for $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ whenever*

$$-r \leq A^T r \leq r, \qquad \textit{and} \qquad b^T r > 0.$$

Proposition 5. *[34] The* AVE (2) *has a unique solution for any $b \in \mathbb{R}^n$ if and only if matrix $A - I + 2D$ be non-singular for any diagonal matrix $D = \operatorname{diag}(d_i)$ with $0 \leq d_i \leq 1$.*

Proposition 6. *[34] The* AVE (2) *has a unique solution for any $b \in \mathbb{R}^n$ if and only if matrix $A + I - 2D$ be non-singular for any diagonal matrix $D = \operatorname{diag}(d_i)$ with $0 \leq d_i \leq 1$.*

Lemma 2. *The second step in* $\mathrm{N}\frac{3}{8}\mathrm{SM}$ *is equivalent to*

$$x^{k+1} = x^k - 8M_k^{-1}F(x^k),$$

where

$$M_k = 8A - D(x^k) - 3D\Big(\frac{2x^k + y^k}{3}\Big) - 3D\Big(\frac{x^k + 2y^k}{3}\Big) - D(y^k). \tag{9}$$

Proof. From the second step of $\mathrm{N}\frac{3}{8}\mathrm{SM}$, we obtain

$$\begin{aligned}
M_k &= \partial F(x^k) + 3\partial F\left(\frac{2x^k + y^k}{3}\right) + 3\partial F\left(\frac{x^k + 2y^k}{3}\right) + \partial F(y^k) \\
&= A - D(x^k) + 3\left(A - D\Big(\frac{2x^k + y^k}{3}\Big)\right) + 3\left(A - D\Big(\frac{x^k + 2y^k}{3}\Big)\right) + \left(A - D(y^k)\right) \\
&= 8A - D(x^k) - 3D\Big(\frac{2x^k + y^k}{3}\Big) - 3D\Big(\frac{x^k + 2y^k}{3}\Big) - D(y^k).
\end{aligned}$$

□

Theorem 1. *Let $A \in \mathbb{R}^{n\times n}$ and all singular values of A be greater than 1. Then, the matrix M_k is invertible for all k .*

Proof. Using contradiction, let M_k be singular. Hence, $M_k z = 0$ for some $z \neq 0$. Now, (9) gives us

$$Az = \frac{1}{8}\left(D(x^k) + 3D\Big(\frac{2x^k + y^k}{3}\Big) + 3D\Big(\frac{x^k + 2y^k}{3}\Big) + D(y^k)\right)z.$$

Since all sigular values of A are greater than one, we obtain

$$\begin{aligned}
z^T z < z^T A^T A z &= \frac{1}{64} z^T \left[D(x^k) + 3D\Big(\frac{2x^k + y^k}{3}\Big) + 3D\Big(\frac{x^k + 2y^k}{3}\Big) + D(y^k)\right]^T \\
&\quad \times \left[D(x^k) + 3D\Big(\frac{2x^k + y^k}{3}\Big) + 3D\Big(\frac{x^k + 2y^k}{3}\Big) + D(y^k)\right] z \\
&= \frac{1}{64} z^T \left[D^2(x^k) + 6D(x^k)D\Big(\frac{2x^k + y^k}{3}\Big) + 6D(x^k)D\Big(\frac{x^k + 2y^k}{3}\Big)\right. \\
&\quad + 2D(x^k)D(y^k) + 9D^2\Big(\frac{2x^k + y^k}{3}\Big) + 18D\Big(\frac{2x^k + y^k}{3}\Big)D\Big(\frac{x^k + 2y^k}{3}\Big) \\
&\quad \left. + 6D(y^k)D\Big(\frac{2x^k + y^k}{3}\Big) + 9D^2\Big(\frac{x^k + 2y^k}{3}\Big) + 6D(y^k)D\Big(\frac{x^k + 2y^k}{3}\Big) + D^2(y^k)\right] z.
\end{aligned}$$

Furthermore, D is a diagonal matrix with diagonal elements of $\{-1,0,+1\}$. So

$$z^T z < \frac{1}{64}\left(64 z^T z\right) = z^T z,$$

which is contradiction and M_k is non-singular. □

Theorem 2. *Let $A \in \mathbb{R}^{n\times n}$ and all singular values of A be greater than 1. If $M = \sup_k \|M_k^{-1}\| < \frac{1}{24}$, then the generated sequence $\{x^k\}$ by* $\mathrm{N}\frac{3}{8}\mathrm{SM}$ *converges to a solution x^* of* AVE.

Proof. From Lemma 2, we obtain

$$M_k(x^{k+1}-x^*) = M_k\left(x^k - 8M_k^{-1}F(x^k) - x^*\right) = M_k(x^k - x^*) - 8F(x^k).$$

Now, $F(x^*) = 0$, (3) and (9) give us

$$\begin{aligned}
M_k(x^{k+1}-x^*) &= M_k(x^k - x^*) - 8\left(F(x^k) - F(x^*)\right)\\
&= M_k(x^k - x^*) - 8\left(Ax^k - |x^k| - Ax^* + |x^*|\right)\\
&= (M_k - 8A)(x^k - x^*) + 8\left(|x^k| - |x^*|\right)\\
&= -\left(D(x^k) + 3D\left(\frac{2x^k+y^k}{3}\right) + 3D\left(\frac{x^k+2y^k}{3}\right) + D(y^k)\right)(x^k - x^*) + 8\left(|x^k| - |x^*|\right).
\end{aligned}$$

So

$$x^{k+1} - x^* = M_k^{-1}\left[8\left(|x^k| - |x^*|\right) - \widehat{M}_k(x^k - x^*)\right], \tag{10}$$

where

$$\widehat{M}_k = D(x^k) + 3D\left(\frac{2x^k+y^k}{3}\right) + 3D\left(\frac{x^k+2y^k}{3}\right) + D(y^k).$$

Furthermore

$$\|\widehat{M}_k\| \le \|D(x^k)\| + 3\left\|D\left(\frac{2x^k+y^k}{3}\right)\right\| + 3\left\|D\left(\frac{x^k+2y^k}{3}\right)\right\| + \|D(y^k)\| \le 8. \tag{11}$$

Finally, Lemma 1, (10) and (11) imply

$$\begin{aligned}
\|x^{k+1} - x^*\| &\le \|M_k^{-1}\|\left[8\left\||x^k| - |x^*|\right\| + \|\widehat{M}_k\|\|x^k - x^*\|\right]\\
&\le \|M_k^{-1}\|\left[16\|x^k - x^*\| + 8\|x^k - x^*\|\right]\\
&= 24\|M_k^{-1}\|\|x^k - x^*\|.
\end{aligned}$$

So

$$\|x^{k+1} - x^*\| \le 24M\|x^k - x^*\|.$$

Finally

$$\|x^{k+1} - x^*\| \le \cdots \le (24M)^{k+1}\|x^0 - x^*\|.$$

Hence, the generated sequence $\{x^k\}$ by $\mathrm{N}\frac{3}{8}\mathrm{SM}$ is convergent to the solution of AVE. □

4 Numerical experiments

In this section, we compare the numerical results of N$\frac{3}{8}$SM with NSM and the Newton (N) method to solve AVE. The stopping criterion for all the algorithms is

$$\text{RES} := \left\|Ax^k - |x^k| - b\right\| < \varepsilon,$$

or the total number of iterates exceeds 500. For comparing the algorithms, we use four test problems with different dimensions. The numerical results are reported in Tables 1-4. In these tables, "Iter" and "Time" stand for the number of iterations and time in seconds, respectively. Also, we use the parameter $\varepsilon = 10^{-12}$ and the generated vector $b \in \mathbb{R}^n$ by the following MATLAB command

$$b = (A - \text{eye}(n)) * \text{ones}(n, 1).$$

Here, eye(n) is the identity matrix of order n and ones$(n, 1)$ is a vector with all components equal to 1 of order n. All algorithms are implemented in MATLAB 2017a programming environment on a 2.3Hz Intel core i3 processor Laptop and 4GB of RAM.

Example 1. Consider the AVE with $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ with

$$\begin{cases} a_{ii} = 4, & i = 1, 2, \ldots, n, \\ a_{i,(i+1)} = -1, & i = 1, \ldots, n-1, \\ a_{i,(i-1)} = -1, & i = 2, \ldots, n, \end{cases}$$

and the other components are zero. Also, the initial vector is set to be $x^0 = (0, 0, \ldots, 0)^T \in \mathbb{R}^n$. Table 1 shows that all the algorithms solve the test problems in different dimensions with the same number of iterations. Furthermore, Newton's algorithm is faster than the other two methods but NSM and N$\frac{3}{8}$SM solve the test problems almost at the same time. The residual error for NSM and N$\frac{3}{8}$SM is less than N.

Table 1: Comparison of N, NSM and N$\frac{3}{8}$SM in Iter, Times and RES for Example 1.

n		1000	2000	3000	4000	5000	avarage
	Iter	2	2	2	2	2	2
N	Time	0.4427	1.7865	4.6707	9.2925	18.6206	6.9626
	RES	2.0751×10^{-14}	2.9531×10^{-14}	3.6157×10^{-14}	4.1779×10^{-14}	4.6748×10^{-14}	3.4993×10^{-14}
	Iter	2	2	2	2	2	2
NSM	Time	0.8427	3.7682	9.6217	19.8775	40.7085	14.9637
	RES	1.3722×10^{-14}	1.9577×10^{-14}	2.4000×10^{-14}	2.7752×10^{-14}	3.1046×10^{-14}	2.3219×10^{-14}
	Iter	2	2	2	2	2	2
N$\frac{3}{8}$SM	Time	0.9009	3.8666	9.9900	21.0197	44.2504	16.0055
	RES	1.4424×10^{-14}	2.0659×10^{-14}	2.5318×10^{-14}	2.9272×10^{-14}	3.2812×10^{-14}	2.4497×10^{-14}

Example 2. Consider the AVE with $A \in \mathbb{R}^{n\times n}$ where the entries of the matrix A are as follows

$$\begin{cases} a_{ii} = 4n, & i = 1, 2, \ldots, n, \\ a_{i,(i+1)} = n, & i = 1, \ldots, n-1, \\ a_{i,(i-1)} = n, & i = 2, \ldots, n, \end{cases}$$

and other components are $a_{ij} = 0.5$. The initial vector is $x^0 = (0,0,\dots,0)^T \in \mathbb{R}^n$. Numerical results of all algorithms are given in Table 2 which shows that $\mathrm{N}\frac{3}{8}\mathrm{SM}$ is better than other algorithms in Iter, Times and RES.

Table 2: Comparison of N, NSM and $\mathrm{N}\frac{3}{8}\mathrm{SM}$ in Iter, Times and RES for Example 2.

n		200	500	1000	1500	2000	avarage
	Iter	500	500	500	500	500	500
N	Time	2.0794	12.9059	75.0321	224.0558	507.4120	164.2970
	RES	1.0363×10^{-10}	1.3586×10^{-9}	6.5838×10^{-9}	1.7790×10^{-8}	3.8796×10^{-8}	1.2926×10^{-8}
	Iter	500	9	9	7	19	108.8
NSM	Time	4.8604	0.5640	2.286	4.8864	33.2051	9.1605
	RES	1.4735×10^{-12}	6.4311×10^{-13}	0	0	0	4.2332×10^{-13}
	Iter	500	6	8	9	9	106.4
$\mathrm{N}\frac{3}{8}\mathrm{SM}$	Time	5.0982	0.3905	2.2208	6.5240	15.2313	5.8929
	RES	1.8887×10^{-12}	0	2.0949×10^{-13}	0	0	4.1964×10^{-13}

Example 3. Consider the AVE with $A \in \mathbb{R}^{n\times n}$ which is produced by the following command in MATLAB and all singular values are greater than 1:

$$A = \mathrm{round}(n * \mathrm{eye}(n)) - 0.02 * (2 * \mathrm{rand}(n,n) - 1).$$

Also, the initial vector is $x^0 = (0,0,\dots,0)^T \in \mathbb{R}^n$. The numerical results of this problem are reported in Table 3. This table shows that the Iter, Time and RES of $\mathrm{N}\frac{3}{8}\mathrm{SM}$ are better than those of the other algorithms.

Table 3: Comparison of N, NSM and $\mathrm{N}\frac{3}{8}\mathrm{SM}$ in Iter, Times and RES for Example 3.

n		1000	1500	2000	2500	3000	avarage
	Iter	500	500	500	500	500	500
N	Time	86.2749	220.5741	281.1413	322.0491	346.1719	251.2423
	RES	4.8234×10^{-11}	1.8125×10^{-10}	2.5741×10^{-10}	4.4910×10^{-10}	6.7114×10^{-10}	3.2143×10^{-10}
	Iter	3	4	4	4	5	4
NSM	Time	1.0293	3.7912	8.2153	15.6279	37.0463	13.142
	RES	8.4312×10^{-13}	5.5695×10^{-13}	4.5475×10^{-13}	9.0949×10^{-13}	4.5475×10^{-13}	6.4381×10^{-13}
	Iter	3	4	4	4	5	4
$\mathrm{N}\frac{3}{8}\mathrm{SM}$	Time	0.8692	7.0346	8.5758	15.7216	32.5731	12.9549
	RES	7.3677×10^{-13}	0	0	0	0	1.4735×10^{-13}

Example 4. Consider the AVE with $A = M + \mu I$ $(\mu > 0)$ and

$$M = \mathrm{tridiag}(-I, \Sigma, -I) \in \mathbb{R}^{n^2\times n^2}, \qquad \Sigma = \mathrm{tridiag}(-1,4,-1) \in \mathbb{R}^{n\times n}.$$

Here, use $\mu = 0.5$. Numerical results have presented in Table 4. This table confirms that N is the fastest method for solving the absolute value equation of Example 4. Moreover, $\mathrm{N}\frac{3}{8}\mathrm{SM}$ is better than the other two methods from the number of iterations and RES.

Table 4: Comparison of N, NSM and $N\frac{3}{8}SM$ in Iter, Times and RES for Example 4.

n		200	400	600	800	1000	avarage
	Iter	3	3	3	3	3	3
N	Time	0.1130	0.5711	1.9594	5.0694	9.5802	3.4586
	RES	5.1341×10^{-14}	5.6977×10^{-14}	6.2203×10^{-14}	6.7387×10^{-14}	7.2512×10^{-14}	6.2084×10^{-14}
	Iter	2	2	2	2	2	2
NSM	Time	0.1784	0.7706	2.5832	6.5136	12.3523	4.4796
	RES	4.2960×10^{-14}	4.7888×10^{-14}	5.3234×10^{-14}	5.6500×10^{-14}	6.0910×10^{-14}	5.2298×10^{-14}
	Iter	2	2	2	2	2	2
$N\frac{3}{8}SM$	Time	0.1706	0.8193	2.6539	6.5616	12.0605	4.4532
	RES	4.1285×10^{-14}	4.3263×10^{-14}	5.0311×10^{-14}	5.6607×10^{-14}	6.1414×10^{-14}	5.0576×10^{-14}

5 Concluding remarks

In this paper, we introduced a two-step method to solve the absolute value equations. The first step is Newton method and the second step is based on Simpson's $3/8$ rule. The convergence of the new method and the linearly convergence rate are proven under some standard assumptions. Numerical examples have shown that the proposed method is effective in the number of iterations and CPU time. Hence, our method is better than some other algorithms for solving AVE.

References

[1] L. Abdallah, M. Haddou, T. Migot, *Solving absolute value equation using complementarity and smoothing functions*, J. Comput. Appl. Math. **327** (2018) 196–207.

[2] J.H. Alcantara, J.S. Chen, M.K. Tam, *Method of alternating projections for the general absolute value equation*, Fixed Point Theory Appl. **25** (2023) 1-38.

[3] R. Ali, A. Ali, I. Khan, A. Mohamed, *Two new generalized iteration methods for solving absolute value equations using M-matrix*, AIMS Mathematics **7** (2022) 8176–8187.

[4] R. Ali, K. Pan, *New generalized Gauss-Seidel iteration methods for solving absolute value equations*, Math. Methods Appl. Sci., https://doi.org/10.1002/mma.9062, 2023.

[5] R. Ali, K. Pan, *The new iteration methods for solving absolute value equations*, Appl. Math. **65** (2023) 109–122.

[6] R. Ali, Z. Zhang, *Exploring two new iterative methods for solving absolute value equations*, Appl. Math. Comput. **70** (2024) 6245–6258.

[7] C. Chen, D. Yu, D. Han, *Exact and inexact Douglas-Rachford splitting methods for solving large-scale sparse absolute value equations*, IMA J. Numer. Anal. **43** (2023) 1036–1060.

[8] C. Chen, B. Huang, D. Yu, D. Han, *O*ptimal parameter of the SOR-like iteration method for solving absolute value equations, Numer. Algorithms **96** (2024) 799–826.

[9] R.W. Cottle, J.S. Pang, R.E. Stone, *The linear complementarity problem*, SIAM, 1992.

[10] V. Edalatpour, D. Hezari, D.K. Salkuyeh, *A generalization of the Gauss-Seidel iteration method for solving absolute value equations*, Appl. Math. Comput. **293** (2017) 156-167.

[11] A.J. Fakharzadeh, N.N. Shams, *An efficient algorithm for solving absolute value equations*, J. Math. Ext. **15** (2021) 1-23.

[12] F.B. Hildebrand, *Introduction to Numerical Analysis*, 2nd Edition, McGraw-Hill, New York, 1994.

[13] M. Hladik, H. Moosaei, F. Hashemi, S. Ketabchi, P.M. Pardalos, *An overview of absolute value equations: from theory to solution methods and challenges*, arXiv:2404.06319, 2024.

[14] S.L. Hu, Z.H. Huang, *A note on absolute value equations*, Optim. Lett. **4** (2010) 417–424.

[15] X.Q. Jiang, Y. Zhang, *A smoothing-type algorithm for absolute value equations*, J. Ind. Manag. Optim. **9** (2013) 789–798.

[16] Y.F. Ke, C.F. Ma, *SOR-like iteration method for solving absolute value equations*, Appl. Math. Comput. **311** (2017) 195–202.

[17] S. Ketabchi, H. Moosaei, *Minimum norm solution to the absolute value equation in the convex case*, J. Optim. Theory Appl. **154** (2012) 1080–1087.

[18] S. Ketabchi, H. Moosaei, *An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side*, Comput. Math. Appl. **64** (2012) 1882–1885.

[19] A. Khan, J. Iqbal, A. Akgul, R. Ali, Y. Du, A. Hussain, K.S. Nisar, V. Vijayakumar, *A Newton-type technique for solving absolute value equations*, Alex. Eng. J. **64** (2023) 291–296.

[20] S. Kumar, D.M. Hladik, H. Moosaei, *Characterization of unique solvability of absolute value equations: an overview, extensions, and future directions*, Optim. Lett. **18** (2024) 889–907.

[21] O.L. Mangasarian, *Absolute value programming*, Comput. Optim. Appl. **36** (2007) 43–53.

[22] O.L. Mangasarian, *A generalized Newton method for absolute value equations*, Optim. Lett. **3** (2009) 101–108.

[23] O.L. Mangasarian, P.R. Meyer, *Absolute value equations*, Linear Algebra Appl. **419** (2006) 359–367.

[24] X.H. Miao, K. Yao, C.Y. Yang, J.S. Chen, *Levenberg-Marquardt method for absolute value equation associated with second-order cone*, Nume. Algebra Control Optim. **12(1)** (2022) 47–61.

[25] M.A. Noor, J. Iqbal, K.I. Noor, E. Al-Said, *On an iterative method for solving absolute value equations*, Optim. Lett. **6** (2012) 1027–1033.

[26] H. Ren, X. Wang, X.B. Tang, T. Wang, *The general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems*, Comput. Math. Appl. **77** (2019) 1071–1081.

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

[28] J. Rohn, *An algorithm for solving the absolute value equation*, Electron. J. Linear Algebra **18** (2009) 589–599.

[29] J. Rohn, V. Hooshyarbakhsh, R. Farhadsefat, *An iterative method for solving absolute value equations and sufficient conditions for unique solvability*, Optim. Lett. **8** (2014) 35–44.

[30] D.K. Salkuyeh, *The Picard-HSS iteration method for absolute value equations*, Optim. Lett. **8** (2014) 2191–2202.

[31] J. Xie, H. Qi, D. Han, *Randomized iterative methods for generalized absolute value equations: Solvability and error bounds*, https://doi.org/10.48550/arXiv.2405.04091, 2024.

[32] X. Wang, X. Li, L.H. Zhang, R.C. Li, *An efficient numerical method for the symmetric positive definite second-order cone linear complementarity problem*, J. Sci. Comput. **79** (2019) 1608–1629.

[33] H. Wang, H. Liu, S. Cao, *A verification method for enclosing solutions of absolute value equations*, Collect. Math. **64** (2013) 17–18.

[34] S.L Wu, C.X. Li, *The unique solution of the absolute value equations*, Appl. Math. Lett. **76** (2018) 195–200.

[35] S.L. Wu, C.X. Li, *A note on unique solvability of the absolute value equation*, Optim. Lett. **14** (2020) 1957–1960.

[36] N. Yilmaz, A. Sahiner, *Smoothing techniques in solving non-Lipschitz absolute value equations*, Int. J. Comput. Math. **100** (2023) 867–879.

[37] L. Zheng, *The Picard-HSS-SOR iteration method for absolute value equations*, J. Inequal. Appl. 2020 (2020) 258.