

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

Farzad Rahpeymaii*, Majid Rostami
*Department of Mathematics,
Technical and Vocational University (TVU),
Tehran, Iran.
rahpeyma_83@yahoo.com*

February 22, 2024

Abstract

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

Keywords: Absolute value equation; Iteration scheme; Simpson rule; Convergence analysis.

Mathematics Subject Classification: 65K10, 90C30, 65F10, 65B99.

1 Introduction

The generalized absolute value equation (GAVE) is

$$Ax + B|x| = b, \quad (1.1)$$

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 was introduced by Rohn in [18] and then was investigated by the other researchers, see [10, 16, 21]. For $B = -I$, GAVE reduces to absolute value equation (AVE) [12]

$$Ax - |x| = b. \quad (1.2)$$

The AVE (1.2) 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 [4, 17, 22]. Solving the AVE is an NP-hard problem [12] and the conditions of existence, non-existence and uniqueness the solution of it are investigated in [12, 20, 23]. So far, some numerical algorithms have been provided to obtain the approximate solution of AVE [7, 8, 9, 20]. The most important of these algorithms are the non-smooth Newton method [13], the mixed-type splitting technique [6], the Levenberg-Marquardt technique [15], the SOR-like method [24], the Picard iteration method [26], the Gauss-Seidel iteration method [5], a method based on interval matrix [19], the complementarity and smoothing functions method [1], the smoothing techniques for non-Lipschitz absolute value equations [25], the alternating projections method [2], and the combination of Newton method and Simpson rule [11]. In this paper, we present a two-step iterative method for solving the absolute value equation (1.2). The first step of the new method is Newton's algorithm and the second 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 (1.2). Our iterative method has two steps which are based on Newton's method and Simpson's 3/8 rule. The convergence of new method is 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 New method

Let

$$F(x) := Ax - |x| - b. \quad (2.1)$$

Therefore, solving the AVE is equivalent to $F(x) = 0$. The generalized Jacobian of $|x|$ is defined by

$$D(x) := \partial|x| = \text{diag}(\text{sgn}(x)), \quad (2.2)$$

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

$$F(x) = (A - D(x))x - b. \quad (2.3)$$

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

$$\partial F(x) = A - D(x). \quad (2.4)$$

Recently, Alamgir Khan et. al. [11] proposed an improved Newton-type technique for solving AVE which has two-step. 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, which is described in Algorithm 2.1.

Algorithm 2.1 (An improved Newton method for 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 $\|Ax^k - |x^k| - b\| \leq \varepsilon$, stop.

(S2) Compute

$$\begin{cases} y^k = x^k - (A - D(x^k))^{-1}b, \\ x^{k+1} = x^k - 6 \left(\partial F(x^k) + 4\partial F\left(\frac{x^k + y^k}{2}\right) + \partial F(y^k) \right)^{-1} F(x^k). \end{cases} \quad (2.5)$$

Then, go back to (S1).

In (S2), Simpson's rule uses three points but Simpson's 3/8 rule uses four points. But, its convergence is stronger than Simpson's rule. Therefore, we use Simpson's 3/8 rule instead of Simpson's rule in the second step of Algorithm 2.1.

We now introduce the new two-step algorithm, below.

Algorithm 2.2 (A new improved Newton method for 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 $\|Ax^k - |x^k| - b\| \geq \varepsilon$, stop.

(S2) Compute

$$\begin{cases} y^k = x^k - (A - D(x^k))^{-1}b, \\ x^{k+1} = x^k - 8 \left(\partial F(x^k) + 3\partial F\left(\frac{2x^k + y^k}{3}\right) + 3\partial F\left(\frac{x^k + 2y^k}{3}\right) + \partial F(y^k) \right)^{-1} F(x^k). \end{cases} \quad (2.6)$$

Then, go back to (S1).

3 Convergence analysis

In this section, we investigate convergence for Algorithm 2.2 to solve AVE. For this goal, we assume the following assumptions:

(P1) Let $A \in \mathbb{R}^{n \times n}$. Then, all singular values of A are greater than 1.

(P2) Let $\Omega \subseteq \mathbb{R}^n$. Then, the Lipschitz continuity implies that [3]

$$\| |x| - |y| \| \leq 2\|x - y\|, \quad \forall x, y \in \Omega.$$

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. [14] Let $A \in \mathbb{R}^{n \times n}$ be invertible. If $\|A^{-1}\| < 1$, then the AVE (1.2) has a unique solution for any $b \in \mathbb{R}^n$.

Proposition 2. [14] If the singular values of $A \in \mathbb{R}^{n \times n}$ exceed 1, the AVE (1.2) is uniquely solvable for any $b \in \mathbb{R}^n$.

Proposition 3. [14] Let $0 \neq b \geq 0$ and $\|A\| < 1$. Then, the AVE (1.2) has no solution.

Proposition 4. [14] Let $r \in \mathbb{R}^n$. The AVE (1.2) has no solution for $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ whenever

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

Proposition 5. [1] The AVE (1.2) 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 $D = \text{diag}(d_i)$ with $0 \leq d_i \leq 1$.

Proposition 6. [1] The AVE (1.2) 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 $D = \text{diag}(d_i)$ with $0 \leq d_i \leq 1$.

Lemma 1. The second step in Algorithm 2.2 is equivalent to

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

where

$$M = 8A - D(x^k) - 3D\left(\frac{2x^k + y^k}{3}\right) - 3D\left(\frac{x^k + 2y^k}{3}\right) - D(y^k). \quad (3.1)$$

Proof. From the second step of Algorithm 2.2, we obtain

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

□

Theorem 1. Let $A \in \mathbb{R}^{n \times n}$ and (P1) holds. Then, the matrix M is invertible.

Proof. We do the proof by contradiction. Let M be singular. Hence, $Mz = 0$ for some $z \neq 0$. Now, (3.1) gives us

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

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

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

Furthermore, D is a diagonal matrix with diagonal elements of $\pm 1, 0$. So

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

which is contradiction and M is non-singular matrix. $\square$

Theorem 2. Suppose that (P1) and (P2) hold. If $\|M^{-1}\| < \frac{1}{24}$, then the generated sequence $\{x^k\}$ by Algorithm 2.2 converges to a solution x^* of AVE.

Proof. From Lemma 1, we obtain

$$M(x^{k+1} - x^*) = M(x^k - 8M^{-1}F(x^k) - x^*) = M(x^k - x^*) - 8F(x^k).$$

Now, $F(x^*) = 0$, (2.1) and (3.1) give us

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

so that

$$x^{k+1} - x^* = M^{-1} \left[8(|x^k| - |x^*|) - \widehat{M}(x^k - x^*) \right], \quad (3.2)$$

where

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

Furthermore

$$\|\widehat{M}\| \leq \|D(x^k)\| + 3\left\|D\left(\frac{2x^k + y^k}{3}\right)\right\| + 3\left\|D\left(\frac{x^k + 2y^k}{3}\right)\right\| + \|D(y^k)\| \leq 8. \quad (3.3)$$

We obtain from (P2) and (3.3)

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

$\square$

Remark 1. The inequality $\|x^{k+1} - x^*\| < \|x^k - x^*\|$ shows that the generated sequence $\{x^k\}$ by Algorithm 2.2 is linearly convergent to the solution of AVE.

4 Numerical experiments

In this section, we compare the numerical results of Algorithm 2.1 and Algorithm 2.2 to solve AVE. For all test problems the initial vector is $x^0 = (0, 0, \dots, 0)^T \in \mathbb{R}^n$. The stopping condition of all algorithms is to establish the following inequality

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

or the total number of iterates exceeds 500. For comparison of both algorithms, we use four test problems with different dimensions which the numerical results of them are reported in Tables 1-4. In these tables,

"Iter" and "Time" are the number of iterations and time in seconds, respectively. For both algorithms, we use the parameter $\varepsilon = 10^{-8}$ and generated vector $b \in \mathbb{R}^n$ by the following Matlab command

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

Here, $\text{eye}(n)$ is identity matrix and $\text{ones}(n, 1)$ is a vector with all components 1. 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 $A \in \mathbb{R}^{n \times n}$ which the components of matrix A are as

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

and other components are zero. Table 1 shows that Algorithm 2.1 is better than Algorithm 2.2 in Iter and Time for different dimensions of this problem.

Table 1: Numerical results for example 1

		Algorithm 2.1				Algorithm 2.2	
n	Iter	Time	RES	Iter	Time	RES	
100	17	0.039633	3.5818×10^{-9}	15	0.045765	2.5454×10^{-9}	
200	17	0.128073	5.1976×10^{-9}	15	0.110707	3.6899×10^{-9}	
300	17	0.257956	6.4187×10^{-9}	15	0.247175	4.5553×10^{-9}	
400	17	0.545315	7.4422×10^{-9}	15	0.512010	5.2808×10^{-9}	
500	17	0.997546	8.3410×10^{-9}	15	0.905140	5.9180×10^{-9}	
1000	18	7.353929	2.9628×10^{-9}	15	6.002890	8.4085×10^{-9}	
2000	18	33.955499	4.2005×10^{-9}	16	30.802567	2.3836×10^{-9}	
3000	18	96.600912	5.1488×10^{-9}	16	90.205021	2.9216×10^{-9}	

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

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

and other components are $a_{ij} = 0.5$. Numerical results of both algorithms are given in Table 2.

Table 2: Numerical results for example 2

		Algorithm 2.1				Algorithm 2.2	
n	Iter	Time	RES	Iter	Time	RES	
100	4	0.025570	2.6446×10^{-9}	4	0.024256	1.1754×10^{-9}	
200	4	0.041648	5.4416×10^{-10}	4	0.045107	1.0992×10^{-10}	
300	4	0.070536	1.9336×10^{-10}	4	0.083782	7.9216×10^{-11}	
400	4	0.154069	4.0532×10^{-10}	4	0.136943	6.9532×10^{-10}	
500	4	0.239064	1.2556×10^{-9}	4	0.232173	6.4584×10^{-10}	
1000	4	1.021319	6.6934×10^{-9}	4	1.026784	3.8928×10^{-9}	
2000	4	5.432641	2.9179×10^{-9}	4	5.136426	2.1603×10^{-10}	
3000	4	17.1648	1.7666×10^{-9}	4	16.117428	4.9557×10^{-10}	

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.

$$A = \text{rand}(n, n) * \text{rand}(n, n) + n * \text{eye}(n).$$

The numerical results of this problem are presented in Table 3 and Algorithm 2.2 is better than Algorithm 2.1 in Iter and Time.

Table 3: Numerical results for example 3

		Algorithm 2.1				Algorithm 2.2	
n	Iter	Time	RES	Iter	Time	RES	
100	5	0.026977	6.1101×10^{-11}	4	0.021311	9.2828×10^{-9}	
200	4	0.033023	2.3538×10^{-9}	4	0.035352	1.0271×10^{-9}	
300	4	0.059152	7.3592×10^{-10}	4	0.064947	4.2866×10^{-10}	
400	4	0.130032	7.5316×10^{-10}	4	0.126369	6.5955×10^{-10}	
500	4	0.221922	1.0972×10^{-9}	4	0.217078	1.0774×10^{-9}	
1000	4	1.170458	8.8623×10^{-9}	4	1.162338	9.1114×10^{-9}	

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

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

Here tridiag is Tri-diagonal matrix. We use the parameter $\mu = 0.5$. From Table 4, we conclude that Algorithm 2.2 is much better than Algorithm 2.1 in Iter and Time.

Table 4: Numerical results for example 4

		Algorithm 2.1				Algorithm 2.2	
n	Iter	Time	RES	Iter	Time	RES	
150	87	0.299008	9.9268×10^{-9}	74	0.310416	7.6338×10^{-9}	
300	91	1.224120	9.2491×10^{-9}	77	1.029168	7.6158×10^{-9}	
600	93	6.171972	9.6931×10^{-9}	78	5.702457	9.4731×10^{-9}	
900	94	26.312212	9.8650×10^{-9}	79	32.549920	9.0372×10^{-9}	
1200	95	40.276716	9.2478×10^{-9}	80	31.429867	7.9353×10^{-9}	
1500	96	68.809292	8.3219×10^{-9}	80	59.119007	8.9847×10^{-9}	
2100	96	200.992376	9.9827×10^{-9}	81	174.023022	8.0249×10^{-9}	
3000	97	719.865172	9.5840×10^{-9}	81	799.661881	9.6916×10^{-9}	

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 two-step method is 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 the two-step method based on Simpson's rule for solving AVE.

Declarations

Ethical Approval

Not applicable.

Competing interests

The authors declare that there are no competing interests.

Authors' contributions

Both the authors contributed equally to the work.

Funding

Not applicable.

Availability of data and materials

All the data and details of the results in this paper can be found within the paper.

References

- [1] Abdallah, L., Haddou, M., and Migot, T., 2018. "Solving absolute value equation using complementarity and smoothing functions", *Journal of Computational and Applied Mathematics*, **327**, pp. 196-207.
- [2] Alcantara, J. H., Chen, J. S., and Tam, M. K., 2023. "Method of alternating projections for the general absolute value equation", *Journal of Fixed Point Theory and Applications*, **25(39)**, pp. 1-38.
- [3] Ali, R., Ali, A., Khan, I., and Mohamed, A., 2022. "Two new generalized iteration methods for solving absolute value equations using M-matrix", *AIMS Mathematics*, **7(5)**, pp. 8176-8187.
- [4] Cottle, R. W., Pang, J. S., and Stone, R. E., 1992. "The Linear Complementarity Problem", SIAM.
- [5] Edalatpour, V., Hezari, D., and Salkuyeh, D. K., 2017. "A generalization of the Gauss-Seidel iteration method for solving absolute value equations", *Applied Mathematics and Computation*, **293**, pp. 156-167.
- [6] Fakharzadeh, A. J., and Shams, N. N., 2021. "An efficient algorithm for solving absolute value equations", *Journal of Mathematical Extension*, **15**, pp. 1-23.
- [7] Hu, S. L., and Huang, Z. H., 2010. "A note on absolute value equations", *Optimization Letters*, **4**, pp. 417-424.
- [8] Jiang, X. Q., and Zhang, Y., 2013. "A smoothing-type algorithm for absolute value equations", *Journal of Industrial and Management Optimization*, **9**, pp. 789-798.
- [9] Ketabchi, S., and Moosaei, H., 2012. "Minimum norm solution to the absolute value equation in the convex case", *Journal of Optimization Theory and Applications*, **154**, pp. 1080-1087.
- [10] Ketabchi, S., and Moosaei, H., 2012. "An efficient method for optimal correcting of absolute value equations by minimal changes in the right hand side", *Computers & Mathematics with Applications*, **64**, pp. 1882-1885.
- [11] Khan, A., Iqbal, J., Akgul, A., Ali, R., Du, Y., Hussain, A., Nisar, K. S., and Vijayakumar, V., 2023. "A Newton-type technique for solving absolute value equations", *Alexandria Engineering Journal*, **64**, pp. 291-296.
- [12] Mangasarian, O. L., 2007. "Absolute value programming", *Computational Optimization and Applications*, **36**, pp. 43-53.
- [13] Mangasarian, O. L., 2009. "A generalized Newton method for absolute value equations", *Optimization Letters*, **3**, pp. 101-108.

- [14] Mangasarian, O. L., and Meyer, R. R., 2006. "Absolute value equations", *Linear Algebra and its Applications*, **419**, pp. 359-367.
- [15] Miao, X. H., Yao, k., Yang, C. Y., and Chen, J. S., 2022. "Levenberg-Marquardt method for absolute value equation associated with second-order cone", *Numerical Algebra, Control and Optimization*, **12(1)**, pp. 47-61.
- [16] Noor, M. A., Iqbal, J., Noor, K. I., and Al-Said, E., 2012. "On an iterative method for solving absolute value equations", *Optimization Letters*, **6**, pp. 1027-1033.
- [17] Ren, H., Wang, X., Tang, X. B., and Wang, T., 2019. "The general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems", *Computers & Mathematics with Applications*, **77**, pp. 1071-1081.
- [18] Rohn, J., 2004. "A theorem of the alternatives for the equation $Ax + B|x| = b$ ", *Linear and Multilinear Algebra*, **52(6)**, pp. 421-426.
- [19] Rohn, J., 2009. "An algorithm for solving the absolute value equation", *Electronic Journal of Linear Algebra*, **18(5)**, pp. 589-599.
- [20] Rohn, J., Hooshyrbakhsh, V., and Farhadsefat, R., 2014. "An iterative method for solving absolute value equations and sufficient conditions for unique solvability", *Optimization Letters*, **8**, pp. 35-44.
- [21] Wang, H., Liu, H., and Cao, S., 2013. "A verification method for enclosing solutions of absolute value equations", *Collectanea Mathematica*, **64**, pp. 17-18.
- [22] Wang, X., Li, X., Zhang, L. H., and Li, R. C., 2019. "An efficient numerical method for the symmetric positive definite second-order cone linear complementarity problem", *Journal of Scientific Computing*, **79**, pp. 1608-1629.
- [23] Wu, S. L., and Li, C. X., 2020. "A note on unique solvability of the absolute value equation", *Optimization Letters*, **14**, pp. 1957-1960.
- [24] Yi, F. K., and Ma, C. F., 2017. "SOR-like iteration method for solving absolute value equations", *Applied Mathematics and Computation*, **311**, pp. 195-202.
- [25] Yilmaz, N., and Sahiner, A., 2023. "Smoothing techniques in solving non-Lipschitz absolute value equations", *International Journal of Computer Mathematics*, **100(4)**, pp. 867-879.
- [26] Zheng, L., 2020. "The Picard-HSS-SOR iteration method for absolute value equations", *Journal of Inequalities and Applications*, **258**, <https://doi.org/10.1186/s13660-020-02525-3>.