mathematics

Article

A New Efficient Method for Absolute Value Equations

Peng Guo [1], Javed Iqbal [2,*], Syed Muhammad Ghufran [2], Muhammad Arif [2], Reem K. Alhefthi [3] and Lei Shi [1]

[1] School of Mathematics and Statistics, Anyang Normal University, Anyan 455002, China; guop@aynu.edu.cn (P.G.); shimath@aynu.edu.cn (L.S.)
[2] Department of Mathematics, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan; smghufran@awkum.edu.pk (S.M.G.); marifmaths@awkum.edu.pk (M.A.)
[3] College of Sciences, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia; raseeri@ksu.edu.sa
* Correspondence: javedmath@awkum.edu.pk

Abstract: In this paper, the two-step method is considered with the generalized Newton method as a predictor step. The three-point Newton–Cotes formula is taken as a corrector step. The proposed method's convergence is discussed in detail. This method is very simple and therefore very effective for solving large systems. In numerical analysis, we consider a beam equation, transform it into a system of absolute value equations and then use the proposed method to solve it. Numerical experiments show that our method is very accurate and faster than already existing methods.

Keywords: absolute value equations; Newton–Cotes open formula; convergence analysis; numerical results; beam equation

MSC: 65F10; 65H10

check for updates

Citation: Guo, P.; Iqbal, J.; Ghufran, S.M.; Arif, M.; Alhefthi, R.K.; Shi, L. A New Efficient Method for Absolute Value Equations. *Mathematics* **2023**, *11*, 3356. https://doi.org/10.3390/math11153356

Academic Editors: Fajie Wang and Ji Lin

Received: 25 June 2023
Revised: 24 July 2023
Accepted: 28 July 2023
Published: 31 July 2023

1. Introduction

Suppose an AVE of the form

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

where $A \in \mathbb{R}^{n \times n}$, $x, b \in \mathbb{R}^n$ and $|\cdot|$ represents an absolute value. The AVE

$$Ax + B|x| = b \tag{2}$$

is the generalized form of Equation (1) for $B \in \mathbb{R}^{n \times n}$, which was first presented by Rohn [1]. The AVE Equation (1) has many applications in pure and applied sciences [2]. It is difficult to find the exact solution of Equation (1) because of the absolute values of x. For some works on this aspect, we refer to [3–5]. Many iterative methods were proposed to study the AVE Equation (1), for example [6–15].

Nowadays, the two-step techniques are very poplar for solving AVE Equation (1). Liu [16,17] presented two-step iterative methods to solve AVEs. Khan et al. [18] have suggested a new method based on a generalized Newton's technique and Simpson's rule for solving AVEs. Shi et al. [19] have developed a two-step Newton-type method with linear convergence for AVEs. Noor et al. [20] have suggested minimization techniques for AVEs and discussed the convergence of these techniques under some suitable conditions. In [21], the two-step Gauss quadrature method was suggested for solving AVEs. When the coefficient matrix A in the AVE Equation (1) has the Toeplitz structure, Gu et al. [22] suggested the nonlinear CSCS-like method and the Picard–CSCS method for solving this problem.

In this paper, the Newton–Cotes open method along with the generalized Newton technique [23] is suggested to solve Equation (1). This new method is straightforward and very effective. The proposed method's convergence is proved under the condition that $\|A^{-1}\| < \frac{1}{10}$ in Section 3. To prove the effectiveness, we consider several examples in Section 4. The main aim of this new method is to obtain the solution of (1) in a few iterations with good accuracy. This new method successfully solves large systems of AVEs. In most cases, this new method requires just one iteration to find the approximate solution of Equation (1) with accuracy up to 10^{-13}. The following notations are used. Let $sign(x)$ be a vector with entries $1, 0, -1$, based on the associated entries of x. The generalized Jacobian $\sigma|x|$ of $|x|$ based on a subgradient [24,25] of the entries of $|x|$ is the diagonal matrix D given by

$$D(x) = \sigma|x| = diag(sign(x)). \tag{3}$$

$svd(A)$ denotes the n singular values of A, $\|A\| = (\lambda)^{\frac{1}{2}}$ represents the 2-norm of A and λ is the maximum eigenvalue of $A^T A$ in absolute. $\|x\| = \sqrt{(x^T, x)}$ is the 2-norm of the vector x; for more detail, see [26].

2. Proposed Method

We develop a new two-step (NTS) method for AVE Equation (1) in this section. Let

$$J(x) = Ax - |x| - b. \tag{4}$$

Then, $J'(x)$ is given by:

$$J'(x) = \sigma(J(x)) = A - D(x). \tag{5}$$

Consider the predictor step as:

$$\gamma^k = \left(A - D\left(x^k\right)\right)^{-1} b. \tag{6}$$

Let v be the solution of Equation (1). To construct the corrector step, we proceed as follows:

$$\int_u^v J'(x)dx = J(v) - J(u) = -J(u). \tag{7}$$

Now, using the three-point Newton–Cotes formula, we have

$$\int_u^v J'(x)dx = \frac{1}{3}\left[2J'\left(\frac{3u+v}{4}\right) - J'\left(\frac{u+v}{2}\right) + 2J'\left(\frac{u+3v}{4}\right)\right](v-u). \tag{8}$$

From Equations (7) and (8), we have

$$-J(u) = \frac{1}{3}\left[2J'\left(\frac{3u+v}{4}\right) - J'\left(\frac{u+v}{2}\right) + 2J'\left(\frac{u+3v}{4}\right)\right](v-u). \tag{9}$$

Thus,

$$v = u - 3\left[2J'\left(\frac{3u+v}{4}\right) - J'\left(\frac{u+v}{2}\right) + 2J'\left(\frac{u+3v}{4}\right)\right]^{-1} J(u). \tag{10}$$

From Equation (10), the NTS method can be written as (Algorithm 1):

Algorithm 1: NTS Method

1: Choose $x^0 \in \mathbb{R}^n$.
2: For k, calculate $\gamma^k = \left(A - D\left(x^k\right)\right)^{-1} b$.
3: Using Step 2, calculate
$x^{k+1} = x^k - 3\left[2J'\left(\frac{3x^k+\gamma^k}{4}\right) - J'\left(\frac{x^k+\gamma^k}{2}\right) + 2J'\left(\frac{x^k+3\gamma^k}{4}\right)\right]^{-1} J\left(x^k\right)$.
4: If $\left\|x^{k+1} - x^k\right\| < tol$, then stop. Otherwise, go to step 2.

3. Convergence

Now, we examine the convergence of the NTS method. The predictor step

$$\gamma^k = \left(A - D\left(x^k\right)\right)^{-1} b \tag{11}$$

is well defined; see Lemma 2 [23]. To prove that

$$2J'\left(\frac{3x^k + \gamma^k}{4}\right) - J'\left(\frac{x^k + \gamma^k}{2}\right) + 2J'\left(\frac{x^k + 3\gamma^k}{4}\right) \tag{12}$$

is nonsingular, first we consider

$$\phi^k = \frac{3x^k + \gamma^k}{4}, \quad \delta^k = \frac{x^k + \gamma^k}{2}, \quad \tau^k = \frac{x^k + 3\gamma^k}{4}. \tag{13}$$

Now

$$\begin{aligned}
&2J'\left(\frac{3x^k + \gamma^k}{4}\right) - J'\left(\frac{x^k + \gamma^k}{2}\right) + 2J'\left(\frac{x^k + 3\gamma^k}{4}\right) \\
=&2A - 2D\left(\frac{3x^k + \gamma^k}{4}\right) - A + D\left(\frac{x^k + \gamma^k}{2}\right) + 2A - D\left(\frac{x^k + 3\gamma^k}{4}\right) \\
=&3A - 2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right),
\end{aligned}$$

where $D\left(\phi^k\right)$, $D\left(\delta^k\right)$ and $D\left(\tau^k\right)$ are diagonal matrices defined in Equation (3).

Lemma 1. *If $svd(A) > 1$, then $\left(3A - 2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)\right)^{-1}$ exists for any diagonal matrix D defined in Equation (3).*

Proof. If $3A - 2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)$ is singular, then

$$\left(3A - 2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)\right)u = 0 \tag{14}$$

for some $u \neq 0$. As $svd(A) > 1$, therefore, using Lemma 1 [23], we have

$$\begin{aligned}
u^T u < u^T A^T A u =& \frac{1}{9} u^T \left(\left(2D\left(\phi^k\right) - D\left(\delta^k\right) + 2D\left(\tau^k\right)\right)\left(2D\left(\phi^k\right) - D\left(\delta^k\right) + 2D\left(\tau^k\right)\right)\right)u \\
=& \frac{1}{9} u^T \left(4D\left(\phi^k\right)D\left(\phi^k\right) - 4D\left(\phi^k\right)D\left(\delta^k\right) + 8D\left(\phi^k\right)D\left(\tau^k\right) + D\left(\delta^k\right)D\left(\delta^k\right)\right. \\
&\left. - 4D\left(\delta^k\right)D\left(\tau^k\right) + 4D\left(\tau^k\right)D\left(\tau^k\right)\right)u \\
\leq& \frac{1}{9} u^T (9) u \\
=& u^T u,
\end{aligned}$$

which is a contradiction, hence $3A - 2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)$ is nonsingular. □

Lemma 2. *If $svd(A) > 1$, then the sequence of the NTS method is well defined and bounded with an accumulation point $\tilde{x}$ such that*

$$\tilde{x} = \tilde{x} - 3\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1} J(\tilde{x}), \tag{15}$$

or it is equivalent to

$$\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\widetilde{x} = \left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\widetilde{x} - 3J(\widetilde{x}). \tag{16}$$

Hence, there exists an accumulation point $\widetilde{x}$ with

$$\left(A - \widetilde{D}(\widetilde{x})\right)\widetilde{x} = b, \tag{17}$$

for some diagonal matrix $\widetilde{D}$ whose diagonal entries are 0 or ± 1 depending on whether the corresponding component of $\widetilde{x}$ is zero, positive, or negative as defined in (3).

Proof. The proof is the same as given in [23]. Thus, it is skipped. □

Theorem 1. *If $\left|\left|\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1}\right|\right| < \frac{1}{9}$, then the NTS method converges to a solution v of Equation (1).*

Proof. Consider

$$\begin{aligned} x^{k+1} - v &= x^k - 3\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1} J\left(x^k\right) - v \\ &= x^k - v - 3\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1} J\left(x^k\right). \end{aligned}$$

It is seen that

$$\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^{k+1} - v\right) = \left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^k - v\right) - 3J\left(x^k\right). \tag{18}$$

As the solution to Equation (1) is v, therefore

$$J(v) = Av - |v| - b = 0. \tag{19}$$

From Equations (18) and (19), we have

$$\begin{aligned} &\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^{k+1} - v\right) \\ =&\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^k - v\right) - 3J\left(x^k\right) + 3J(v) \\ =&\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^k - v\right) - 3\left(J\left(x^k\right) - J(v)\right) \\ =&\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)\left(x^k - v\right) - 3\left(Ax^k - \left|x^k\right| - Av + |v|\right) \\ =&\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right) - 3A\right)\left(x^k - v\right) + 3\left(\left|x^k\right| - |v|\right) \\ =&\left(-2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)\right)\left(x^k - v\right) + 3\left(\left|x^k\right| - |v|\right). \end{aligned}$$

It follows that

$$\begin{aligned} x^{k+1} - v =& \left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1}\Big[\left(-2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)\right)\left(x^k - v\right) \\ &+3\left(\left|x^k\right| - |v|\right)\Big]. \end{aligned}$$

Thus, we know

$$\begin{aligned} \left|\left|x^{k+1} - v\right|\right| =& \left|\left|\left(2J'\left(\phi^k\right) - J'\left(\delta^k\right) + 2J'\left(\tau^k\right)\right)^{-1}\Big[\left(-2D\left(\phi^k\right) + D\left(\delta^k\right) - 2D\left(\tau^k\right)\right)\left(x^k - v\right) \right. \\ &\left. +3\left(\left|x^k\right| - |v|\right)\Big]\right|\right|. \end{aligned}$$

This leads to

$$\begin{aligned}
\left\|x^{k+1}-v\right\| \leq & \left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\| \left\|\left[\left(-2D\left(\phi^k\right)+D\left(\delta^k\right)-2D\left(\tau^k\right)\right)\left(x^k-v\right)\right.\right. \\
& \left.\left.+3\left(\left|x^k\right|-|v|\right)\right]\right\| \\
\leq & \left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\|\left[\left\|-2D\left(\phi^k\right)+D\left(\delta^k\right)-2D\left(\tau^k\right)\right\|\left\|x^k-v\right\|\right. \\
& \left.+3\left\|\left|x^k\right|-|v|\right\|\right].
\end{aligned} \tag{20}$$

Since $D\left(\phi^k\right)$, $D\left(\delta^k\right)$ and $D\left(\tau^k\right)$ are diagonal matrices, therefore

$$\left\|-2D\left(\phi^k\right)+D\left(\delta^k\right)-2D\left(\tau^k\right)\right\| \leq 3, \tag{21}$$

We also use the Lipchitz continuity (see Lemma 5 [23]), that is

$$\left\|\left|x^k\right|-|v|\right\| \leq 2\left\|x^k-v\right\|. \tag{22}$$

From Equations (20)–(22), we have

$$\begin{aligned}
\left\|x^{k+1}-v\right\| &\leq \left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\|\left[3\|x^k-v\|+6\|x^k-v\|\right] \\
&= 9\left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\|\left\|x^k-v\right\| \\
&< \left\|x^k-v\right\|.
\end{aligned} \tag{23}$$

In Equation (23), the supposition $\left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\| < \frac{1}{9}$ is used. Hence x^k converges linearly to the solution of Equation (1). □

Lemma 3. *Let $||A^{-1}|| < \frac{1}{10}$ and $D\left(\phi^k\right)$, $D\left(\delta^k\right)$, $D\left(\tau^k\right)$ be non-zeros. Then, for any b, the NTS method converges to the unique solution of Equation (1) for any initial guess $x^0 \in \mathbb{R}^n$.*

Proof. Since $||A^{-1}|| < \frac{1}{10}$, therefore, Equation (1) is uniquely solvable for any b see ([2], Proposition 4). Since A^{-1} exists, therefore, by Lemma 2.3.2 [26], we have

$$\begin{aligned}
& \left\|\left(2J'\left(\phi^k\right)-J'\left(\delta^k\right)+2J'\left(\tau^k\right)\right)^{-1}\right\| \\
= & \left\|3A-2D\left(\tau^k\right)+D\left(\delta^k\right)-2D\left(\tau^k\right)\right\| \\
\leq & \frac{\left\|(3A)^{-1}\right\|\left\|-2D\left(\phi^k\right)+D\left(\delta^k\right)-2D\left(\tau^k\right)\right\|}{1-\|(3A)^{-1}\|\|-2D(\phi^k)+D(\delta^k)-2D(\tau^k)\|} \\
\leq & \frac{\frac{1}{3}||A^{-1}||3}{1-\frac{1}{3}||A^{-1}||3} \\
< & \frac{\frac{1}{10}}{1-\frac{1}{10}} = \frac{1}{9}.
\end{aligned}$$

Hence, by Theorem 1, the NTS method converges to the unique solution of Equation (1). □

4. Numerical Result

In the section, several examples are taken to prove the efficiency of the suggested method. We use Matlab R2021a with a core (TM) i5@ 1.70 GHz. The CPU time in seconds, number of iterations and 2-norm of residuals are denoted by time, K and RES, respectively.

Example 1 ([9]). *Consider*

$$A = tridiag(-1.5, 4, -0.5) \in \mathbb{R}^{s\times s}, \quad x \in \mathbb{R}^s \quad and \quad b = (1, 2, \cdots, s)^T. \tag{24}$$

A comparison of the NTS method with the MSOR-like method [9], generalized Newton method (GNM) [23] and RIM [11] is given in Table 1.

Table 1. NTS method verses MSOR-like method and RIM.

Method	s	1000	2000	3000	4000	5000	6000
RIM	K	24	25	25	25	25	25
	CPU	7.084206	54.430295	150.798374	321.604186	581.212038	912.840059
	RES	7.6844×10^{-7}	4.9891×10^{-7}	6.3532×10^{-7}	7.6121×10^{-7}	8.8041×10^{-7}	9.9454×10^{-7}
MSOR-Like	K	30	31	32	32	33	33
	CPU	0.0067390	0.0095621	0.0215634	0.0541456	0.0570134	0.0791257
	RES	5.5241×10^{-7}	7.0154×10^{-7}	5.8684×10^{-7}	9.0198×10^{-7}	5.6562×10^{-7}	7.4395×10^{-7}
GNM	K	5	5	5	5	5	5
	CPU	0.0059651	0.007333	0.0115038	0.0330345	0.0551818	0.0783684
	RES	3.1777×10^{-10}	7.8326×10^{-9}	2.6922×10^{-10}	3.7473×10^{-9}	8.3891×10^{-9}	5.8502×10^{-8}
NTS method	K	1	1	1	1	1	2
	CPU	0.001816	0.003410	0.018771	0.0326425	0.031539	0.069252
	RES	9.6317×10^{-12}	2.3697×10^{-11}	4.1777×10^{-11}	6.2756×10^{-11}	8.20814×10^{-11}	5.9998×10^{-11}

Table 1 shows that the NTS method finds the solution of Equation (1) very quickly. The RES of the NTS method shows that the new method is more accurate than all the methods stated in Table 1.

Example 2 ([16]). *Consider*

$$A = round(s \times (eye(s, s) - 0.02 \times (2 \times rand(s, s) - 1))). \tag{25}$$

Choose a random $x \in \mathbb{R}^s$ *and* $b = Ax - |x|$.

We compare the NTS method with INM [7], the GQ method [21] and TSI [16] in Table 2.

Table 2. Numerical results for Example 2.

Method	s	200	400	600	800	1000
TSI	K	3	3	3	4	4
	RES	7.6320×10^{-12}	9.0622×10^{-12}	1.9329×10^{-11}	4.0817×10^{-11}	7.1917×10^{-11}
	CPU	0.031619	0.120520	0.32591	0.83649	1.00485
INM	K	3	3	3	4	4
	RES	2.1320×10^{-12}	6.6512×10^{-12}	3.0321×10^{-11}	2.0629×10^{-11}	8.0150×10^{-11}
	CPU	0.012851	0.098124	0.156810	0.638421	1 0.982314
GQ method	K	2	2	2	2	2
	RES	2.1415×10^{-12}	4.4320×10^{-12}	1.0515×10^{-11}	1.9235×10^{-11}	2.8104×10^{-11}
	CPU	0.013145	0.038734	0.162439	0.204578	0.276701
NTS method	K	1	1	1	1	2
	RES	1.0637×10^{-12}	4.0165×10^{-12}	1.0430×10^{-11}	2.0644×10^{-11}	2.1660×10^{-11}
	CPU	0.012832	0.071124	0.153001	0.201356	0.274165

It is clear that the NTS method converges in one iteration in most cases. The other two methods require at least three iterations to find the solution of Equation (1) to achieve the given accuracy.

Example 3 ([7]). *Let*

$$A = tridiag(-1,8,-1) \in \mathbb{R}^{s\times s}, \quad b = Ae - |e| \quad for \quad e = (-1,1,-1,\cdots,)^T \in \mathbb{R}^s, \tag{26}$$

where the initial vector is taken from [7].

We compare the NTS method with GGS [8], MGS [6] and Method II [7].

As seen in Table 3, the suggested method approximates the solution of Equation (1) in just one iteration. The residual shows that the NTS method is very accurate.

Table 3. Comparison of NTS method with GGS, MGS and Method II.

Methods	s	1000	2000	3000	4000	5000
GGS	K	11	11	11	11	11
	RES	2.4156×10^{-9}	2.7231×10^{-9}	3.1872×10^{-9}	3.2167×10^{-9}	3.4538×10^{-9}
	CPU	0.514656	1.045221	1.153442	1.843198	5.652411
MGS	K	7	8	8	8	8
	RES	6.7056×10^{-9}	7.30285×10^{-10}	7.6382×10^{-10}	9.57640×10^{-10}	8.52425×10^{-10}
	CPU	0.215240	0.912429	0.916788	1.503518	4.514201
Method II	K	6	6	6	6	6
	RES	3.6218×10^{-8}	5.1286×10^{-8}	6.2720×10^{-8}	7.2409×10^{-8}	8.0154×10^{-8}
	CPU	0.238352	0.541264	0.961534	1.453189	2.109724
NTS method	K	1	1	1	1	1
	RES	4.9774×10^{-15}	7.0304×10^{-15}	8.6069×10^{-15}	9.9363×10^{-15}	1.1107×10^{-14}
	CPU	0.204974	0.321184	0.462869	0.819503	1.721235

Example 4. *Consider the beam equation of the form*

$$\frac{d^2x}{dr^2} - |x| = \frac{Sx}{EM} - \frac{qr(r-L)}{2EM}, \tag{27}$$

with boundary conditions

$$x(0) = 0, \quad x(L) = 0, \tag{28}$$

where $L = 120$ in. is the length of the beam, the modulus of elasticity $E = 3 \times 10^7$ lb/in.^{2}, the intensity of uniform load $q = 100$ lb/ft, the stress at the ends is 1000 lb and the central moment of inertia $M = 625$ in.^{4}.

We use FDM to discretize this equation. A comparison of the NTS method with the solution by Maple is illustrated in Figure 1.

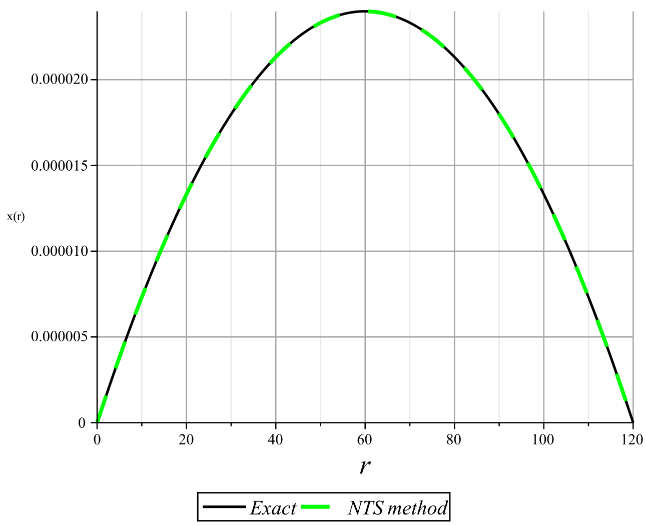

Figure 1. Deflection of beam for $h = 2$ (step size).

Figure 1 shows the effectiveness and accuracy of the NTS method. Clearly, the deflection of the beam is maximum at the center.

Example 5 ([20])**.** *Consider an AVE of the form*

$$a_{ii} = 4s, \quad a_{i,i+1} = a_{i+1,i} = s, \quad a_{ij} = 0.5, \quad i = 1, 2, \cdots, s. \tag{29}$$

Choose a constant vector b, and the initial guess is taken from [20]. A comparison of the NTS method with MM [20] and MMSGP [1] is presented in Table 4.

Table 4. The numerical results for Example 5.

s	K	MMSGP CPU	RES	K	MM CPU	RES	K	NTS Method CPU	RES
2	24	0.005129	5.6800×10^{-7}	2	0.029965	1.2079×10^{-12}	1	0.0032321	1.7763×10^{-15}
4	37	0.008701	9.7485×10^{-7}	4	0.027864	5.5011×10^{-8}	1	0.008621	3.5527×10^{-15}
8	45	0.009217	5.5254×10^{-7}	6	0.045387	6.9779×10^{-8}	1	0.004120	2.9296×10^{-14}
16	66	0.012458	5.8865×10^{-7}	7	0.356930	2.0736×10^{-8}	1	0.006156	8.4072×10^{-14}
32	55	0.031597	8.2514×10^{-7}	8	0.033277	4.9218×10^{-8}	1	0.005108	2.1645×10^{-13}
64	86	0.085621	7.6463×10^{-7}	9	0.185753	9.0520×10^{-9}	1	0.008120	1.0088×10^{-12}
128	90	0.521056	6.3326×10^{-7}	9	0.452394	1.7912×10^{-8}	1	0.362162	2.2822×10^{-12}

We observe that the NTS method is very successful for solving Equation (1). Furthermore, the NTS method is very consistent when n increases (large systems), whereas the other two methods need more iterations.

5. Conclusions

In this paper, we have used a two-step method for AVEs. In this new method, a three-point Newton–Cotes open formula is taken as a corrector step, while a generalized Newton method is taken as the predictor. The local convergence of the NTS method is proved in Section 2. Theorem 1 proves the linear convergence of the proposed method. A comparison shows that this method is very accurate and converges in just one iteration in most cases. This idea can be used to solve generalized AVEs and also to find all solutions of AVEs in the future.

Author Contributions: The idea of the present paper came from J.I. and S.M.G.; P.G., J.I., S.M.G., M.A. and R.K.A. wrote and completed the calculations; L.S. checked all the results. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Key Scientific Research Projects of Universities in Henan Province under Grant 22A110005.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to extend their sincere appreciation to Researchers Supporting Project number RSPD2023R802 KSU, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Rohn, J. A theorem of the alternatives for the equation $Ax + B|x| = b$. *Linear Multilinear Algebra* **2004**, *52*, 421–426. [CrossRef]
2. Mangasarian, O.L.; Meyer, R.R. Absolute value equations. *Linear Algebra Its Appl.* **2006**, *419*, 359–367. [CrossRef]
3. Mansoori, A.; Eshaghnezhad, M.; Effati, S. An efficient neural network model for solving the absolute value equations. *IEEE Tran. Circ. Syst. II Express Briefs* **2017**, *65*, 391–395. [CrossRef]
4. Chen, C.; Yang, Y.; Yu, D.; Han, D. An inverse-free dynamical system for solving the absolute value equations. *Appl. Numer. Math.* **2021**, *168*, 170–181. [CrossRef]
5. Cairong, C.; Yu, D.; Han, D. Exact and inexact Douglas-Rachford splitting methods for solving large-scale sparse absolute value equations. *IMA J. Numer. Anal.* **2023**, *43*, 1036–1060.
6. Ali, R. Numerical solution of the absolute value equation using modified iteration methods. *Comput. Math. Methods* **2022**, *2022*, 2828457. [CrossRef]

7. Ali, R.; Khan, I.; Ali, A.; Mohamed, A. Two new generalized iteration methods for solving absolute value equations using M-matrix. *AIMS Math.* **2022**, *7*, 8176–8187. [CrossRef]
8. Edalatpour, V.; Hezari, D.; Salkuyeh, D.K. A generalization of the Gauss-Seidel iteration method for solving absolute value equations. *Appl. Math. Comput.* **2017**, *293*, 156–167.
9. Haung, B.; Li, W. A modified SOR-like method for absolute value equations associated with second order cones. *J. Comput. Appl. Math.* **2022**, *400*, 113745. [CrossRef]
10. Mansoori, A.; Erfanian, M. A dynamic model to solve the absolute value equations. *J. Comput. Appl. Math.* **2018**, *333*, 28–35. [CrossRef]
11. Noor, M.A.; Iqbal, J.; Al-Said, E. Residual Iterative Method for Solving Absolute Value Equations. *Abstr. Appl. Anal.* **2012**, *2012*, 406232. [CrossRef]
12. Salkuyeh, D.K. The Picard-HSS iteration method for absolute value equations. *Optim. Lett.* **2014**, *8*, 2191–2202. [CrossRef]
13. Abdallah, L.; Haddou, M.; Migot, T. Solving absolute value equation using complementarity and smoothing functions. *J. Comput. Appl. Math.* **2018**, *327*, 196–207. [CrossRef]
14. Yu, Z.; Li, L.; Yuan, Y. A modified multivariate spectral gradient algorithm for solving absolute value equations. *Appl. Math. Lett.* **2021**, *21*, 107461. [CrossRef]
15. Zhang, Y.; Yu, D.; Yuan, Y. On the Alternative SOR-like Iteration Method for Solving Absolute Value Equations. *Symmetry* **2023**, *15*, 589. [CrossRef]
16. Feng, J.; Liu, S. An improved generalized Newton method for absolute value equations. *SpringerPlus* **2016**, *5*, 1042. [CrossRef]
17. Feng, J.; Liu, S. A new two-step iterative method for solving absolute value equations. *J. Inequal. Appl.* **2019**, *2019*, 39. [CrossRef]
18. Khan, A.; Iqbal, J.; Akgul, A.; Ali, R.; Du, Y.; Hussain, A.; Nisar, K.S.; Vijayakumar, V. A Newton-type technique for solving absolute value equations. *Alex. Eng. J.* **2023**, *64*, 291–296. [CrossRef]
19. Shi, L.; Iqbal, J.; Arif, M.; Khan, A. A two-step Newton-type method for solving system of absolute value equations. *Math. Prob. Eng.* **2020**, *2020*, 2798080. [CrossRef]
20. Noor, M.A.; Iqbal, J.; Khattri, S.; Al-Said, E. A new iterative method for solving absolute value equations. *Int. J. Phys. Sci.* **2011**, *6*, 1793–1797.
21. Shi, L.; Iqbal, J.; Raiz, F.; Arif, M. Gauss quadrature method for absolute value equations. *Mathematics* **2023**, *11*, 2069. [CrossRef]
22. Gu, X.-M.; Huang, T.-Z.; Li, H.-B.; Wang, S.-F.; Li, L. Two CSCS-based iteration methods for solving absolute value equations. *J. Appl. Anal. Comput.* **2017**, *7*, 1336–1356.
23. Mangasarian, O.L. A generalized Newton method for absolute value equations. *Optim. Lett.* **2009**, *3*, 101–108. [CrossRef]
24. Polyak, B.T. *Introduction to Optimization*; Optimization Software Inc., Publications Division: New York, NY, USA, 1987.
25. Rockafellar, R.T. *Convex Analysis*; Princeton University Press: Princeton, NJ, USA, 1970.
26. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA; London, UK, 1970.