Mathematics

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

AIMS Mathematics, 5(5): 5171–5183.
DOI:10.3934/math.2020332
Received: 13 February 2020
Accepted: 09 June 2020
Published: 15 June 2020

Research article

A special shift splitting iteration method for absolute value equation

ShiLiang Wu* **and CuiXia Li**

School of Mathematics, Yunnan Normal University, Kunming, Yunnan, 650500, P.R. China

* **Correspondence:** Email: wushiliang1999@126.com.

Abstract: In this paper, based on the shift splitting (SS) technique, we propose a special SS iteration method for solving the absolute value equation (AVE), which is obtained by reformulating equivalently the AVE as a two-by-two block nonlinear equation. Theoretical analysis shows that the special SS method is absolutely convergent under proper conditions. Numerical experiments are given to demonstrate the feasibility, robustness and effectiveness of the special SS method.

Keywords: absolute value equation; shift splitting method; convergence
Mathematics Subject Classification: 65F10, 90C05, 90C30

1. Introduction

Considering the absolute value equation (AVE)

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

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $|\cdot|$ denotes the absolute value. The AVE (1.1) is used as a very important tool and often arises in scientific and engineering computing, such as linear programming, bimatrix games, quadratic programming, the quasi-complementarity problems, and so on [1–5].

In recent years, to obtain the numerical solution of the AVE (1.1), a large number of efficient iteration methods have been proposed to solve the AVE (1.1), including the successive linearization method [2], the Picard and Picard-HSS methods [7, 9], the nonlinear HSS-like iteration method [15], the sign accord method [6] and the hybrid algorithm [12], the generalized Newton (GN) method [11]. Other forms of the iteration methods, one can see [8–10, 13] for more details.

Recently, Ke and Ma in [18] extended the classical SOR-like iteration method in [19] for solving the AVE (1.1) and proposed the following SOR-like iteration method

$$\begin{bmatrix} A & 0 \\ -\alpha\hat{D} & I \end{bmatrix}\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} = \begin{bmatrix} (1-\alpha)A & \alpha I \\ 0 & (1-\alpha)I \end{bmatrix}\begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix} + \begin{bmatrix} \alpha b \\ 0 \end{bmatrix}, \tag{1.2}$$

with $\alpha > 0$ and $\hat{D} = D(x) = diag(sign(x)), x \in \mathbb{R}^n$, where $D(x) = diag(sign(x))$ denotes a diagonal matrix corresponding to $sign(x)$, and $sign(x)$ denotes a vector with components equal to $1, 0, -1$ depending on whether the corresponding component of x is positive, zero or negative. The SOR-like iteration method (1.2) can be designed by using the idea of SOR-like method in [19] for reformulating the AVE (1.1) as the two-by-two block nonlinear equation

$$\begin{cases} Ax - y = b, \\ -|x| + y = 0, \end{cases}$$

or

$$\bar{A}z = \begin{bmatrix} A & -I \\ -\hat{D} & I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} = \bar{b}. \tag{1.3}$$

The convergence condition of the SOR-like iteration method in [18] was given in the light of assumptions imposed on the involved parameter. Further, in [20], some new convergence conditions were obtained from the involved iteration matrix of the SOR-like iteration method.

When $|x|$ is vanished in the AVE (1.1), the AVE (1.1) reduces to the linear system

$$Ax = b. \tag{1.4}$$

In [14], Bai et al. proposed the following shift splitting (SS) of the matrix A, that is,

$$A = \frac{1}{2}(\alpha I + A) - \frac{1}{2}(\alpha I - A), \alpha > 0,$$

and designed the shift splitting (SS) scheme

$$x^{(k+1)} = (\alpha I + A)^{-1}(\alpha I - A)x^{(k)} + 2(\alpha I + A)^{-1}b, \ k = 0, 1, 2, \ldots$$

for solving the non-Hermitian positive definite linear system (1.4). In this paper, we generalize this idea and propose the special SS iteration method for the two-by-two block nonlinear Eq. (1.3) to solve the AVE (1.1).

The remainder of the paper is organized as follows. In Section 2, the special SS iteration method is introduced to solve the AVE (1.1) and the convergence analysis of the special SS iteration method is studied in detail. In Section 3, numerical experiments are reported to show the effectiveness of the proposed method. In Section 4, some conclusions are given to end the paper.

2. The special SS iteration method

In this section, we introduce the special SS iteration method to solve the AVE (1.1). To this end, based on the iteration methods studied in [14], a special shift splitting (SS) of the $\bar{A}$ in (1.3) can be constructed as follows

$$\bar{A} = \begin{bmatrix} A & -I \\ -\hat{D} & I \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & I + I \end{bmatrix} - \frac{1}{2}\begin{bmatrix} \alpha I - A & I \\ \hat{D} & I - I \end{bmatrix},$$

where α is a given positive constant and I is an identity matrix. This special splitting naturally leads to the special SS iteration method for solving the nonlinear Eq. (1.3) and describes below.

The special SS iteration method: Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular and $b \in \mathbb{R}^n$. Given initial vectors $x^{(0)} \in \mathbb{R}^m$ and $y^{(0)} \in \mathbb{R}^n$, for $k = 0, 1, 2, ...$ until the iteration sequence $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ is convergent, compute

$$\begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & 2I \end{bmatrix} \begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} = \begin{bmatrix} \alpha I - A & I \\ \hat{D} & 0 \end{bmatrix} \begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix} + 2 \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{2.1}$$

where α is a given positive constant.

Clearly, the iteration matrix M_α of the special SS method is

$$M_\alpha = \begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & 2I \end{bmatrix}^{-1} \begin{bmatrix} \alpha I - A & I \\ \hat{D} & 0 \end{bmatrix}.$$

Let $\rho(M_\alpha)$ denote the spectral radius of matrix M_α. Then the special SS iteration method is convergent if and only if $\rho(M_\alpha) < 1$. Let λ be an eigenvalue of matrix M_α and $[x, y]^T$ be the corresponding eigenvector. Then

$$\begin{bmatrix} \alpha I - A & I \\ \hat{D} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & 2I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

which is equal to

$$(\lambda - 1)\alpha x + (\lambda + 1)Ax - (\lambda + 1)y = 0, \tag{2.2}$$

$$(1 + \lambda)\hat{D}x - 2\lambda y = 0. \tag{2.3}$$

To study the convergence conditions of the special SS iteration method, we first give some lemmas.

Lemma 2.1. [16] *The AVE* (1.1) *has a unique solution for any* $b \in \mathbb{R}^n$ *if and only if matrix* $A - I + 2D$ *or* $A + I - 2D$ *is nonsingular for any diagonal matrix* $D = diag(d_i)$ *with* $0 \le d_i \le 1$.

Based on Lemma 2.1, it is easy to obtain when matrix A in the AVE (1.1) satisfies $\sigma_{\min}(A) > 1$, where $\sigma_{\min}(A)$ denotes the smallest singular value of the matrix A, then the AVE (1.1) has a unique solution for any $b \in \mathbb{R}^n$. One can see [16] for more details.

Lemma 2.2. [17] *Consider the real quadratic equation* $x^2 - bx + d = 0$*, where* b *and* d *are real numbers. Both roots of the equation are less than one in modulus if and only if* $|d| < 1$ *and* $|b| < 1 + d$.

Lemma 2.3. *Let* A *be a symmetric positive definite matrix and satisfy the conditions of Lemma 2.1. Then*

$$\bar{A} = \begin{bmatrix} A & -I \\ -\hat{D} & I \end{bmatrix}$$

is nonsingular.

Proof. By simple computations, we have

$$\bar{A} = \begin{bmatrix} I & 0 \\ -\hat{D}A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & I - \hat{D}A^{-1} \end{bmatrix} \begin{bmatrix} I & -A^{-1} \\ 0 & I \end{bmatrix}.$$

Clearly, we just need to prove that matrix $I - \hat{D}A^{-1}$ is nonsingular. If not, then for some nonzero $y \in \mathbb{R}^n$ we have that $(I - \hat{D}A^{-1})y = 0$, which will derive a contradiction. This implies that $y = \hat{D}A^{-1}y$. Let $A^{-1}y = z$. Then $Az = \hat{D}z$. Further, we have

$$z^T z < z^T A^T Az = z\hat{D}^T \hat{D}z < z^T z.$$

Obviously, this inequality is invalid. This completes the proof. □

Lemma 2.4. *Let A be a symmetric positive definite matrix and satisfy the conditions of Lemma 2.1. Then*

$$\hat{A} = \begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & 2I \end{bmatrix}$$

is nonsingular.

Proof. Similar to Lemma 2.3, by simple computations, we have

$$\hat{A} = \begin{bmatrix} I & 0 \\ -\hat{D}(\alpha I + A)^{-1} & I \end{bmatrix} \begin{bmatrix} \alpha I + A & 0 \\ 0 & 2I - \hat{D}(\alpha I + A)^{-1} \end{bmatrix} \begin{bmatrix} I & -(\alpha I + A)^{-1} \\ 0 & I \end{bmatrix}.$$

Noting that A is a symmetric positive definite matrix. Clearly, when matrix $2I - \hat{D}(\alpha I + A)^{-1}$ is nonsingular, matrix $\hat{A}$ is also nonsingular. In fact, matrix $2I - \hat{D}(\alpha I + A)^{-1}$ is sure nonsingular. If not, then for some nonzero $y \in \mathbb{R}^n$ we have that $(2I - \hat{D}(\alpha I + A)^{-1})y = 0$, which will derive a contradiction. This implies that $2y = \hat{D}(\alpha I + A)^{-1}y$. Let $(\alpha I + A)^{-1}y = z$. Then $2(\alpha I + A)z = \hat{D}z$. Further, we have

$$2z^T(\alpha I + A)z = z\hat{D}z. \tag{2.4}$$

Let $\lambda = \frac{z^T Az}{z^T z}$ and $\mu = \frac{z^T Dz}{z^T z}$. Then, from (2.4) we can obtain

$$2\alpha + 2\lambda = \mu. \tag{2.5}$$

Since $\alpha > 0$, $\lambda > 1$ and $|\mu| \leq 1$, Eq. (2.5) is invalid. Therefore, Eq. (2.4) is also invalid. This completes the proof. □

Lemma 2.4 implies that the iteration matrix M_α is valid.

Lemma 2.5. *Let A be a symmetric positive definite matrix and satisfy the conditions of Lemma 2.1. If λ is an eigenvalue of the matrix M_α, then $\lambda \neq \pm 1$.*

Proof. If $\lambda = 1$. Then from (2.2) and (2.3) we have

$$\begin{cases} Ax - y = 0, \\ -|x| + y = 0. \end{cases}$$

Based on Lemma 2.3, it follows that $y = 0$ and $x = 0$, which contradicts the assumption that $[x, y]^T$ is an eigenvector of matrix M_α. Hence $\lambda \neq 1$.

Now, we prove that $\lambda \neq -1$. When $\lambda = -1$, from (2.2) and (2.3) we have

$$-2\alpha x = 0 \text{ and } 2\alpha y = 0.$$

Since $\alpha > 0$, we obtain $x = 0$ and $y = 0$, which also contradicts the assumption that $[x, y]^T$ is an eigenvector of matrix M_α. Hence $\lambda \neq -1$. □

Lemma 2.6. *Assume that A is a symmetric positive definite matrix and satisfies the conditions of Lemma 2.1. Let λ be an eigenvalue of M_α and $[x, y]^T$ be the corresponding eigenvector. Then $x \neq 0$. Moreover, if $y = 0$, then $|\lambda| < 1$.*

Proof. If $x = 0$, then from (2.2) we have $(\lambda + 1)y = 0$. By Lemma 2.5 we know that $\lambda \neq -1$. Therefore, $y = 0$, which contradicts the assumption that $[x, y]^T$ is an eigenvector of matrix M_α.

If $y = 0$, then from (2.2) we have

$$(\alpha I + A)^{-1}(\alpha I - A)x = \lambda x.$$

Since A is symmetric positive definite, then by [12, Lemma 2.1] we know that

$$|\lambda| \leq \|(\alpha I + A)^{-1}(\alpha I - A)\| < 1.$$

Thus, we complete the proof. □

Theorem 2.1. *Let A be a symmetric positive definite matrix and satisfy the conditions of Lemma 2.1, and α be a positive constant. Then*

$$\rho(M_\alpha) < 1, \forall \alpha > 0,$$

which implies that the special SS iteration method converges to the unique solution of the AVE (1.1).

Proof. If $\lambda = 0$, then $\rho(M_\alpha) < 1$. The results in Theorem 2.1 hold.

If $\lambda \neq 0$. Then from (2.3) we have

$$y = \frac{1 + \lambda}{2\lambda}\hat{D}x. \tag{2.6}$$

Substituting (2.6) into (2.3) leads to

$$(\lambda - 1)\alpha x + (\lambda + 1)Ax - (\lambda + 1)\frac{1 + \lambda}{2\lambda}\hat{D}x = 0. \tag{2.7}$$

By Lemma 2.6, we know that $x \neq 0$. Multiplying $\frac{x^T}{x^T x}$ on both sides of Eq. (2.7), we obtain

$$(\lambda - 1)\alpha + (\lambda + 1)\frac{x^T Ax}{x^T x} - \frac{(\lambda + 1)^2}{2\lambda} \cdot \frac{x^T \hat{D}x}{x^T x} = 0. \tag{2.8}$$

Let

$$p = \frac{x^T Ax}{x^T x},\ q = \frac{x^T \hat{D}x}{x^T x}.$$

Then $p > 1$ and $|q| \leq 1$. From Eq. (2.8) we have

$$(\lambda - 1)\alpha + (\lambda + 1)p - \frac{(\lambda + 1)^2}{2\lambda}q = 0.$$

Further, we know that λ satisfies the following real quadratic equation

$$\lambda^2 + \frac{2p - 2\alpha - 2q}{2p + 2\alpha - q}\lambda - \frac{q}{2p + 2\alpha - q} = 0. \tag{2.9}$$

Based on Lemma 2.2, we know that a sufficient and necessary condition for the roots of the real quadratic Eq. (2.9) to satisfy $|\lambda| < 1$ if and only if

$$\left|\frac{q}{2p + 2\alpha - q}\right| < 1 \tag{2.10}$$

and

$$\left|\frac{2p - 2\alpha - 2q}{2p + 2\alpha - q}\right| < 1 - \frac{q}{2p + 2\alpha - q}. \tag{2.11}$$

It is easy to check that (2.10) and (2.11) hold for all $\alpha > 0$. Therefore, we obtain

$$\rho(M_\alpha) < 1, \forall \alpha > 0,$$

which implies that the special SS iteration method converges to the unique solution of the AVE (1.1). □

For the SOR-like iteration method (1.2), in [18], Ke and Ma given the following result.

Theorem 2.2. [18] *Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular and $b \in \mathbb{R}^n$. Denote*

$$\nu = \|A^{-1}\|, s = |1 - \alpha| \text{ and } t = \alpha^2\nu.$$

If

$$0 < \alpha < 2 \text{ and } s^4 - 3s^2 - 2st - 2t^2 + 1 > 0, \tag{2.12}$$

then

$$|||(e_{k+1}^x, e_{k+1}^y)||| < |||(e_k^x, e_k^y)|||, k = 0, 1, \ldots,$$

where

$$|||(e_k^x, e_k^y)||| = \sqrt{\|e_k^x\|^2 + \|e_k^y\|^2} \text{ with } e_k^x = x^* - x^{(k)}, e_k^y = y^* - y^{(k)}.$$

Comparing the convergence conditions of the special SS method and the SOR-like iteration method, see Theorem 2.1 and Theorem 2.2 [18], the convergence condition of the SOR-like iteration method not only is relatively strict, but also is difficult to compute. The main reason is that the condition (2.12) has to compute the inverse of the matrix A. This implies that the application of the SOR-like iteration method is limited. Base on this, the special SS method may be superior to the SOR-like iteration method under certain conditions.

Next, we turn to consider this situation where matrix A in (1.1) is the positive definite.

It is noted that Lemma 2.3, Lemma 2.5 and Lemma 2.6 are still valid under the condition of Lemma 2.1 when A in (1.1) is the positive definite.

Now, we need guarantee the invertibility of the first factor of M_α. To this end, we have Lemma 2.7.

Lemma 2.7. *Let positive definite matrix A satisfy the conditions of Lemma 2.1 and $\|(\alpha I + A)^{-1}\|_2 < 2$. Then*

$$\hat{A} = \begin{bmatrix} \alpha I + A & -I \\ -\hat{D} & 2I \end{bmatrix}$$

is nonsingular.

Proof. Based on the proof of Lemma 2.4, obviously, when $\|(\alpha I + A)^{-1}\|_2 < 2$, matrix $2I - \hat{D}(\alpha I + A)^{-1}$ is nonsingular. This implies that matrix $\hat{A}$ is nonsingular as well. □

For later use we define the set $\mathcal{S}$ as

$$\mathcal{S} = \{x \in \mathbb{C}^n : \mathbf{x} = [x; y] \text{ is an eigenvector of } M_\alpha \text{ with } \|x\|_2 = 1\}.$$

Obviously, the members of $\mathcal{S}$ are nonzero. Based on this, Theorem 2.3 can be obtained.

Theorem 2.3. *Let the conditions of Lemma 2.7 be satisfied. For every $x \in \mathcal{S}$, let $s(x) = \Re(x^*Ax)$, $t(x) = \Im(x^*Ax)$ and $q = x^*\hat{D}x$. For each $x \in \mathcal{S}$, if*

$$(\alpha + s(x))^2(s(x) - q) + t(x)^2 s(x) > 0, \tag{2.13}$$

then

$$\rho(M_\alpha) < 1, \quad \forall \alpha > 0,$$

which implies that the special SS iteration method converges to the unique solution of the AVE (1.1).

Proof. Similar to the proof of Theorem 2.1, using x^* instead of x^T in (2.8) leads to

$$(\lambda - 1)\alpha + (\lambda + 1)x^*Ax - \frac{(\lambda + 1)^2}{2\lambda}x^*\hat{D}x = 0,$$

which is equal to

$$(\lambda - 1)\alpha + (\lambda + 1)(s(x) + t(x)i) - \frac{(\lambda + 1)^2}{2\lambda}q = 0, \tag{2.14}$$

For the sake simplicity in notations, we use s, t for $s(x)$, $t(x)$, respectively. From (2.14), we have

$$\lambda^2 + \phi\lambda + \psi = 0, \tag{2.15}$$

where

$$\phi = 2\frac{s + ti - \alpha - q}{2s + 2ti + 2\alpha - q} \text{ and } \psi = -\frac{q}{2s + 2ti + 2\alpha - q}.$$

By some calculations, we get

$$\begin{aligned}
\phi - \phi^*\psi &= 2\frac{s + ti - \alpha - q}{2s + 2ti + 2\alpha - q} + 2\frac{s - ti - \alpha - q}{2s - 2ti + 2\alpha - q} \cdot \frac{q}{2s + 2ti + 2\alpha - q} \\
&= 2\frac{s + ti - \alpha - q}{2s + 2ti + 2\alpha - q} \cdot \frac{2s - 2ti + 2\alpha - q}{2s - 2ti + 2\alpha - q} + 2\frac{(s - ti - \alpha - q)q}{(2s + 2\alpha - q)^2 + 4t^2} \\
&= 2\Big[\frac{(s + ti - \alpha - q)(2s - 2ti + 2\alpha - q)}{(2s + 2\alpha - q)^2 + 4t^2} + \frac{(s - ti - \alpha - q)q}{(2s + 2\alpha - q)^2 + 4t^2}\Big] \\
&= 2\frac{-(s + ti - \alpha - q)q + (s - ti - \alpha - q)q + (s + ti - \alpha - q)(2s - 2ti + 2\alpha)}{(2s + 2\alpha - q)^2 + 4t^2} \\
&= 4\frac{(s - \alpha - q)(s + \alpha) + t^2 + 2\alpha ti}{(2s + 2\alpha - q)^2 + 4t^2}.
\end{aligned}$$

Further, we have

$$\begin{aligned}
|\psi| &= \sqrt{\frac{q^2}{(2s + 2\alpha - q)^2 + 4t^2}} < 1, \\
|\phi - \phi^*\psi| &= \frac{4\sqrt{((s - \alpha - q)(s + \alpha) + t^2)^2 + 4\alpha^2 t^2}}{(2s + 2\alpha - q)^2 + 4t^2}.
\end{aligned} \tag{2.16}$$

thus, the necessary and sufficient condition for $|\lambda| < 1$ in (2.15) is

$$|\phi - \phi^*\psi| + |\psi|^2 < 1. \tag{2.17}$$

Substituting (2.16) into (2.17), we can obtain the condition (2.13), which completes the proof. □

3. Numerical experiments

In this section, two examples are given to demonstrate the performance of the special SS method (2.2) for solving the AVE (1.1). To this end, we compare the special SS method with the SOR-like method [18], the generalized Newton (GN) method [11] and the search direction (SD) method [10] from aspects of the iteration counts (denoted as 'IT') and the relative residual error (denoted as 'RES'), the elapsed CPU time in second (denoted as 'CPU'). In the implementations, we use the sparse LU factorization to calculate an inverse of the first factor of M_α. All runs are implemented in Matlab 7.0.

All initial vectors are chosen to zero vector and all iterations are terminated once the relative residual error satisfies

$$\text{RES} := \frac{||Ax^{(k)} - |x^{(k)}| - b||_2}{||b||_2} \leq 10^{-6}$$

or if the prescribed iteration number 500 is exceeded. In the following tables, '−' denotes RES $> 10^{-6}$ or the iteration counts larger than 500.

Example 1. ([18]) Consider the AVE (1.1) with

$$A = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{n\times n}, x^* = (-1, 1, -1, 1, \ldots, -1, 1)^T \in \mathbb{R}^n,$$

and $b = Ax^* - |x^*|$.

	α	0.7	0.9	1.1	1.2	1.5	2	2.5
SS	IT	59	46	38	35	27	20	16
	CPU	5.342	4.172	3.452	3.183	2.468	1.843	1.484
	RES	9.628e-7	8.901e-7	7.370e-7	6.682e-7	9.998e-7	9.605e-7	7.591e-7
SOR	IT	24	15	21	49	–	–	–
	CPU	0.016	0.016	0.016	0.016	–	–	–
	RES	9.676e-7	5.367e-7	9.431e-7	9.097e-7	–	–	–

Table 1. Numerical comparison of Example 1 with $n = 3000$.

	α	0.7	0.9	1.1	1.2	1.5	2	2.5
SS	IT	59	46	38	35	27	20	16
	CPU	9.748	7.655	6.280	5.67	4.342	3.187	2.515
	RES	9.629e-7	8.902e-7	7.372e-7	6.683e-7	9.999e-7	9.607e-7	7.593e-7
SOR	IT	24	15	21	49	–	–	–
	CPU	0.016	0.016	0.031	0.016	–	–	–
	RES	9.677e-7	5.368e-7	9.432e-7	9.099e-7	–	–	–

Table 2. Numerical comparison of Example 1 with $n = 4000$.

	n	1000	2000	3000	4000
SS	IT	14	14	14	14
	CPU	0.212	0.396	1.312	2.162
	RES	8.913e-7	8.924e-7	8.928e-7	8.930e-7
SOR	IT	11	11	11	11
	CPU	0.005	0.006	0.004	0.007
	RES	7.377e-7	7.382e-7	7.384e-7	7.384e-7
GN	IT	2	2	2	2
	CPU	0.232	1.589	4.988	11.337
	RES	1.107e-16	1.122e-16	1.117e-16	1.121e-16
SD	IT	13	13	13	13
	CPU	1.988	7.289	3.487	6.088
	RES	3.567e-7	3.575e-7	3.577e-7	3.579e-7

Table 3. Numerical comparison of SS, SOR, GN and SD for Example 1.

To compare the special SS method with the SOR-like method, we take the same parameter α. In this case, Tables 1 and 2 list some numerical results of the special SS method and the SOR-like method for Example 1 with different dimensions n and different parameters α. From Tables 1 and 2, we can see that when both iteration methods are convergent, the CPU times of the special SS method are more than that of the SOR-like method. It is noted that the SOR-like method occasionally fail to converge in 500 iterations, the special SS method is always convergent, which confirm the results in Theorem 2.1. From the view of this point, the special SS method is robust, compared with the SOR-like method.

Table 3 list the numerical results of the special SS method with the SOR-like method, the generalized Newton method, and search direction method. Here, the iteration parameter of the special SS method is set to be 3, the iteration parameter of the SOR-like method is set to be 1. From Table 3, the iteration counts of the special SS method are more than any other three methods, but it requires the less CPU times than the generalized Newton method and the search direction method. Among these methods, the SOR-like method compared to any other three methods requires the least iteration steps and CPU time, but it has some risks in the implementations.

On the above discussion, we can draw a conclusion that the special SS method is feasible, compared with the SOR-like method, the generalized Newton method, and search direction method.

Example 2 Let m be a prescribed positive integer and $n = m^2$. Consider the AVE in (1.1) with $A = \bar{A} + \mu I$

$$\bar{A} = \text{tridiag}(-I, S, -I) = \begin{bmatrix} S & -I & 0 & \cdots & 0 & 0 \\ -I & S & -I & \cdots & 0 & 0 \\ 0 & -I & S & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & S & -I \\ 0 & 0 & 0 & \cdots & -I & S \end{bmatrix} \in \mathbb{R}^{n\times n}$$

with

$$S = \text{tridiag}(-1, 4, -1) = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 4 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 4 \end{bmatrix} \in \mathbb{R}^{m\times m},$$

and $b = Ax^* - |x^*|$, where $x^* = (-1, 1, -1, 1, \cdots, -1, 1)$.

	α	0.7	0.9	1.1	1.3	1.5	1.7	1.9
SS	IT	64	50	41	34	30	26	24
	CPU	29.033	23.423	19.377	15.915	13.744	11.895	11.258
	RES	9.965e-7	9.470e-7	8.172e-7	7.564e-7	8.272e-7	9.692e-7	6.752e-7
SOR	IT	34	21	42	–	–	–	–
	CPU	0.625	0.372	0.914	–	–	–	–
	RES	7.088e-7	6.828e-7	8.191e-7	–	–	–	–

Table 4. Numerical comparison of Example 2 with $n = 4900$ and $\mu = 1$.

	α	0.7	0.9	1.1	1.3	1.5	1.7	1.9
SS	IT	64	50	40	34	30	26	24
	CPU	48.405	38.641	32.807	25.840	22.918	20.234	18.07
	RES	9.352e-7	8.890e-7	9.993e-7	9.695e-7	7.760e-7	9.093e-7	6.351e-7
SOR	IT	34	21	42	–	–	–	–
	CPU	0.818	0.519	1.285	–	–	–	–
	RES	7.186e-7	6.927e-7	8.398e-7	–	–	–	–

Table 5. Numerical comparison of Example 2 with $n = 6400$ and $\mu = 1$.

Similarly, for Example 2, we still take the same parameter α for the special SS method with the SOR-like method when both are used. In this case, Tables 4, 5, 7 and 8 list the computing results for Example 2 with $\mu = 1$ and $\mu = 4$.

From Tables 4, 5, 7 and 8, these numerical results further confirm the observations from Tables 1 and 2. That is to say, when both are convergent, the computing efficiency of the SOR-like method advantages over the special SS method. Whereas, in the implementations, indeed, the SOR-like method has some risks; for the special SS method, it is free from such worries. This further verifies that the special SS method is steady.

Similar to Table 3, Tables 6 and 9 still enumerate the numerical results of the special SS method with the SOR-like method, the generalized Newton method, and search direction method. In Tables 6 and 9, the iteration parameter of the SOR-like method is set to be 1; for the special SS method, its

	n	3600	4900	6400	8100
SS	IT	22	22	22	22
	CPU	5.613	10.033	16.831	26.671
	RES	9.969e-7	9.920e-7	9.885e-7	9.857e-7
SOR	IT	17	17	17	17
	CPU	0.224	0.293	0.413	0.547
	RES	6.121e-7	6.260e-7	6.365e-7	6.447e-7
GN	IT	2	2	3	2
	CPU	8.316	20.43	66.477	134.068
	RES	2.046e-16	2.204e-16	2.192e-16	2.206e-16
SD	IT	33	33	33	33
	CPU	17.961	33.104	55.315	88.038
	RES	9.072e-7	9.351e-7	9.560e-7	9.723e-7

Table 6. Numerical comparison of SS, SOR, GN and SD for Example 2 with $\mu = 1$.

	α	0.9	1.1	1.3	1.5	2	3	4
SS	IT	67	55	47	41	30	20	15
	CPU	30.358	24.932	21.307	18.652	13.671	9.117	6.852
	RES	9.987e-7	9.486e-7	8.348e-7	7.605e-7	9.462e-7	8.044e-7	6.463e-7
SOR	IT	12	14	54	–	–	–	–
	CPU	0.203	0.235	1.328	–	–	–	–
	RES	3.256e-7	9.044e-7	8.133e-7	–	–	–	–

Table 7. Numerical comparison of Example 2 with $n = 4900$ and $\mu = 4$.

	α	0.9	1.1	1.3	1.5	2	3	4
SS	IT	67	55	47	40	30	20	15
	CPU	50.910	41.940	35.710	30.477	22.823	15.512	11.435
	RES	9.409e-7	8.934e-7	7.858e-7	9.452e-7	8.905e-7	7.561e-7	6.069e-7
SOR	IT	12	14	54	–	–	–	–
	CPU	0.331	0.375	1.794	–	–	–	–
	RES	3.266e-7	9.080e-7	8.202e-7	–	–	–	–

Table 8. Numerical comparison of Example 2 with $n = 6400$ and $\mu = 4$.

iteration parameter are two cases: $\alpha = 2.1$ in Table 6 and $\alpha = 5.5$ in Table 9. From Tables 6 and 9, the special SS method requires the less CPU times than the generalized Newton method and search direction method. Among these methods, the SOR-like method compared to any other three methods costs the least iteration steps and CPU time, but it has some risks in the implementations.

From the numerical results from Example 2, we can still come to a conclusion that the special SS

	n	3600	4900	6400	8100
SS	IT	11	11	11	11
	CPU	2.812	5.014	8.389	13.309
	RES	7.057e-7	6.944e-7	6.857e-7	6.788e-7
SOR	IT	8	8	8	8
	CPU	0.109	0.141	0.203	0.25
	RES	9.188e-7	9.238e-7	9.275e-7	9.304e-7
GN	IT	2	2	2	2
	CPU	8.452	20.277	44.084	87.995
	RES	2.794e-16	2.901e-16	2.952e-16	2.968e-16
SD	IT	12	12	12	12
	CPU	6.53	12.044	20.198	31.867
	RES	5.212e-7	5.282e-7	5.335e-7	5.375e-7

Table 9. Numerical comparison of SS, SOR, GN and SD for Example 2 with $\mu = 4$.

method is feasible, compared with the SOR-like method, the generalized Newton method, and search direction method.

4. Conclusions

In this paper, based on the shift splitting (SS) of the coefficient matrix of a kind of the equivalent two-by-two block nonlinear equation of the AVE (1.1), a special shift splitting (SS) method is introduced. Some convergence conditions of special SS method are obtained. Numerical experiments show that the special SS method is feasible. However, the parameter α is chosen by the experimentally found optimal choice. So, how to choose the optimal parameter in the special SS method for the AVE is left as a further research work.

Acknowledgments

The author would like to thank the anonymous referee for providing helpful suggestions, which greatly improved the paper. This research was supported by National Natural Science Foundation of China (No.11961082).

Conflict of interest

The authors declare no conflict of interest in this paper.

References

1. J. Rohn, *A theorem of the alternatives for the equation $Ax + B|x| = b$*, Linear Multilinear A., **52** (2004), 421–426.
2. O. L. Mangasarian, *Absolute value programming*, Comput. Optim. Appl., **36** (2007), 43–53.

3. O. L. Mangasarian, R. R. Meyer, *Absolute value equations*, Linear Algebra Appl., **419** (2006), 359–367.
4. S. L. Wu, P. Guo, *Modulus-based matrix splitting algorithms for the quasi-complementarity problems*, Appl. Numer. Math., **132** (2018), 127–137.
5. R. W. Cottle, J. S. Pang, R. E. Stone, *The Linear Complementarity Problem*, Academic, San Diego, 1992.
6. J. Rohn, *An algorithm for solving the absolute value equations*, Electron. J. Linear Al., **18** (2009), 589–599.
7. 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.
8. M. A. Noor, J. Iqbal, E. Al-Said, *Residual iterative method for solving absolute value equations*, Abstr. Appl. Anal., **2012** (2012),1–9.
9. D. K. Salkuyeh, *The Picard-HSS iteration method for absolute value equations*, Optim. Lett., **8** (2014), 2191–2202.
10. M. A. Noor, J. Iqbal, K. I. Noor, et al. *On an iterative method for solving absolute value equations*, Optim. Lett., **6** (2012), 1027–1033.
11. O. L. Mangasarian, *A generalized Newton method for absolute value equations*, Optim. Lett., **3** (2009), 101–108.
12. O. L. Mangasarian, *A hybrid algorithm for solving the absolute value equation*, Optim. Lett., **9** (2015), 1469–1474.
13. A. Wang, Y. Cao, J. X. Chen, *Modified Newton-type iteration methods for generalized absolute value equations*, J. Optimiz. Theory App., **181** (2019), 216–230.
14. Z. Z. Bai, J. F. Yin, Y. F. Su, A *shift-splitting preconditioner for non-Hermitian positive definite matrices*, J. Comput. Math., **24** (2006), 539–552.
15. M. Z. Zhu, G. F. Zhang, Z. Z. Liang, *The nonlinear HSS-like iteration method for absolute value equations*, arXiv.org:1403.7013v2.
16. S. L. Wu, C. X. Li, *The unique solution of the absolute value equations*, Appl. Math. Lett., **76** (2018), 195–200.
17. S. L. Wu, T. Z. Huang, X. L. Zhao, *A modified SSOR iterative method for augmented systems*, J. Comput. Appl. Math., **228** (2009), 424–433.
18. Y. F. Ke, C. F. Ma, *SOR-like iteration method for solving absolute value equations*, Appl. Math. Comput., **311** (2017), 195–202.
19. G. H. Golub, X. Wu, J. Y. Yuan, *SOR-like methods for augmented systems*, BIT., **41** (2001), 71–85.
20. P. Guo, S. L. Wu, C. X. Li, *On the SOR-like iteration method for solving absolute value equations*, Appl. Math. Lett., **97** (2019), 107–113.