A cluster-based technique for solving the GAVE

Wenxiu Guo
School of Mathematical Sciences, Capital Normal University, 105 West Third Ring Road North, Haidian District, Beijing, 100048 China
E-mail: wxguo@cnu.edu.cn

Abstract. Generalized absolute value equation (abbreviated as GAVE) has wide applications in scientific computing and engineering. In this paper, we propose a cluster-based technique for solving the GAVE. First, the GAVE is shown to be equivalent to a linear equation which is related to the signs of the solution. In order to avoid the disadvantage of the existence of the sign index sets which is problem depended, by the framework of a fixed-point iteration, the analysis of the signs of the iteration vectors is given. Some necessary conditions of the nonzero and zero entries of the iteration vector are presented. Then, the signs of the solution of the GAVE can be partitioned by a clustering process. Numerical examples show that the proposed technique can work and significantly accelerate the computational efficiency comparing to some existing methods.

1. Introduction
Consider the generalized absolute value equation (GAVE): given \( A, B \in \mathbb{R}^{n \times n} \) and \( b \in \mathbb{R}^n \), find vector \( x \in \mathbb{R}^n \) from \( Ax + B | x \| = b \), where \(|x| = (|x_1|,|x_2|,\ldots,|x_n|)^T\). When \( B = 0 \), GAVE reduces to the absolute value equation (AVE).

The GAVE has various important applications in many fields, such as scientific computing and engineering [7-10]. On the other hand, the GAVE can be equivalent to some complementarity problems [4], which has wide applications in free boundary problems, bimatrix games, optimization and so on [8,10,14,15].

There had been many studies on the numerical algorithms for solving the GAVE and the AVE. In [2], a globally and quadratically convergent Newton's method was proposed by approximating the modulus item by a smooth function. A generalized Gauss-Seidel iteration method was established in [13]. Levenberg-Marquardt method was given in [6] by a different line search process. A generalized Newton method was proposed directly to solve the AVE in [11]. A hybrid algorithm consisting of solving a linear equations and a linear program was given in [12]. The iteration method for the AVE with symmetry system matrix coupled with the minimization technique was presented in [13].

Although the GAVE is linear, some efficient methods for linear equations like Krylov subspace methods cannot be used to solve the GAVE due to the existence of the \(|x|\). It is clear that, once the signs of the solution of the GAVE are known, the GAVE is essentially a linear equation. Then, the computational efficiency can be significantly improved. In this paper, based on some matrix splitting iteration, we introduce a technique to find the signs of the solution of the GAVE. Numerical examples show that the proposed technique can effectively improve the convergence rate of the iteration.

The rest of this paper is organized as follows. We propose the new technique in Section 2. In
Section 3 we present some numerical examples. A concluding remark is given in Section 4.

2. Main results
Let \( x^* \in \mathbb{R}^n \) be the solution of the GAVE. Denote three sign sets of \( x^* \) as below:
\[
\alpha = \{ i \mid x^*_i > 0, 1 \leq i \leq n \}, \beta = \{ i \mid x^*_i < 0, 1 \leq i \leq n \}, \gamma = \{ i \mid x^*_i = 0, 1 \leq i \leq n \}.
\]

Let \( u, v \) be two sets of indices. We denote the submatrix of \( A \) whose rows and columns are indexed by \( u, v \), respectively.

**Lemma 2.1** Let \( x^* \in \mathbb{R}^n \) be the solution of the GAVE. Then we have
\[
\begin{pmatrix}
A_{aa} - B_{aa} & A_{af} & A_{af} + B_{af} \\
A_{ba} - B_{ba} & A_{bb} & A_{bb} + B_{bb}
\end{pmatrix}
\begin{pmatrix}
x^*_a \\
x^*_b
\end{pmatrix}
= \begin{pmatrix}
b^*_a \\
b^*_b
\end{pmatrix}.
\]

**Proof.** It is easy to see that, there exists a permutation matrix \( P \) such that
\[
P A P^T = \begin{pmatrix}
A_{aa} & A_{af} & A_{af} \\
A_{ba} & A_{bb} & A_{bb} \\
A_{aa} & A_{af} & A_{af}
\end{pmatrix},
\]
\[
B P B^T = \begin{pmatrix}
B_{aa} & B_{af} & B_{af} \\
B_{ba} & B_{bb} & B_{bb} \\
B_{aa} & B_{af} & B_{af}
\end{pmatrix},
\]
\[
P x^* = \begin{pmatrix}
x^*_a \\
x^*_b
\end{pmatrix},
P b = \begin{pmatrix}
b^*_a \\
b^*_b
\end{pmatrix}.
\]

Then, by directly computation to \( Ax + B \mid x = b \), we have (1).

By Lemma 2.1, if \( \alpha \) and \( \beta \) are known, solving the GAVE is equivalent to solving a linear equation, where some efficient methods can be used to accelerate the convergence rate. However, \( \alpha \) and \( \beta \) are problem depended. Instead, we can try to find some information of the sign sets during a convergent iteration process. Let \( x^{(k)} \) be the \( k \)-th iteration vector in an iteration process. Note that
\[
\lim_{k \to \infty} x^{(k)}_\alpha > 0, \lim_{k \to \infty} x^{(k)}_\beta < 0, \lim_{k \to \infty} x^{(k)}_\gamma = 0.
\]

Since the iteration is convergent, when \( k \) is large enough and \( x^{(k)}_\gamma \approx 0 \), we can expect that
\[
\alpha = \{ i \mid x^{(k)}_i > 0, 1 \leq i \leq n \}, \beta = \{ i \mid x^{(k)}_i < 0, 1 \leq i \leq n \}.
\]

When these happen, we have
\[
\begin{align*}
\{ i \mid x^{(k)}_i > 0, 1 \leq i \leq n \} &= \{ i \mid x^{(k-1)}_i > 0, 1 \leq i \leq n \}, \\
\{ i \mid x^{(k)}_i < 0, 1 \leq i \leq n \} &= \{ i \mid x^{(k-1)}_i < 0, 1 \leq i \leq n \}.
\end{align*}
\]

Therefore, (2) can be a necessary condition to get \( \alpha \) and \( \beta \).

On the other hand, to get \( \gamma \), we turn to the fact that \( x^{(k)}_\gamma \approx 0 \). It is clear that we have
\[
\frac{|x^{(k)}_j - x^{(k-1)}_j|}{|x^{(k-1)}_j|} \ll \frac{|x^{(k)}_j - x^{(k-1)}_j|}{|x^{(k-1)}_j|} \ll_{\gamma}.
\]

Thus, (3) is a necessary condition to obtain \( \gamma \). Then, by (3), we consider introducing a clustering technique to identify \( \gamma \) from \( \alpha \) and \( \beta \). Moreover, we can apply the \( K \)-means clustering method to partition the entries of \( x^{(k)} \) into 2 clusters.
Summarizing the discussion above, we propose the next method:

**Method 2.2** New method with cluster-based technique

1) **Given** \(A, B, b, k_{\text{max}}, x^{(0)}\), **initialize**

2) **Run** a convergent iteration method two steps to obtain \(x^{(1)}, x^{(2)}\). **Initialize** \(k = 2\).

3) **When** \(k < k_{\text{max}}\),

4) **Apply** \(K\)-means cluster method to partition \(\{x^{(k)}_i - x^{(k-1)}_j \} / |x^{(k-1)}_j| \) into 2 clusters.

5) **Let** \(\gamma\) and \(s\) be the index sets with smaller and larger cluster center, respectively.

6) **If** \(\text{sign}(x^{(k)}_s) = \text{sign}(x^{(k-1)}_s)\),

7) \(\alpha = \{i \mid x^{*}_i > 0, i \in s\}, \beta = \{i \mid x^{*}_i < 0, i \in s\}\).

8) **Solve** \((1)\) for \(x^{(k+1)}_{\alpha}, x^{(k+1)}_{\beta}\) and let \(x^{(k+1)}_{\gamma} = 0\).

9) **If** the residual isn’t superlinear decreasing,

10) **Get** \(x^{(k+1)}\) by the original iteration.

11) **End** if

12) **Else**

13) **Get** \(x^{(k+1)}\) by the original iteration.

14) **End** if

15) **If** the iteration converges, break, otherwise, \(k = k + 1, \text{ back to 3}\).

Method 2.2 is a framework of cluster-based technique. For more details, we have the next remarks.

- In Step 2), one can use some efficient iteration methods based on matrix splitting, for example,

\[
Mx^{(k+1)} = Nx^{(k)} - B|x^{(k)}| + b,
\]

where \(A = M - N\) is a splitting of \(A\).

- The motivation of Step 4) is from \((3)\). Since the computational cost of \(K\)-means is no greater than \(O(n^2)\), we can expect the cluster process would not increase the CPU time.

- Steps 6)-8) come from \((2)\). Furthermore, since \((2)\) holds when \(k\) is large enough, Steps 9)-10) are necessary.

- Generally, even if the cluster-based technique does not work, Method 2.2 reduces to the original iteration method.

### 3. Numerical examples

In this section, we first present a test example to show the details of the advantage of the proposed technique. Moreover, numerical experiment is applied to a GAVE arising from a linear complementarity problem in free boundary value problem.

**Example 3.3** Let

\[
A = \begin{pmatrix}
8 & -1 & -1 & 0 \\
-1 & 8 & 0 & -1 \\
-1 & 0 & 8 & -1 \\
0 & -1 & -1 & 8
\end{pmatrix}, B = \begin{pmatrix}
0 & -1 & -1 & 0 \\
-1 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 \\
0 & -1 & -1 & 0
\end{pmatrix}, \quad b = Ax^* + B|x^*|,
\]

where \(x^* = (-1,1,0,0)^T\).
Use the Gauss-Seidel splitting in (4), where

\[ M = D - L, \quad N = U, \]

\( D, L \) and \( U \) are the diagonal, strictly lower triangular and strictly upper triangular parts of \( A \). Let

\[ x^{(0)} = (1,1,1,1)^T \]

and the stopping criterion be that the residual is less than \( 10^{-7} \). The results are listed in Table 1.

**Table 1. Iteration vectors of Example 3.3**

| \( x^{(0)} \) | \( x^{(1)} \) | \( x^{(2)} \) | \( x^{(3)} \) | \( x^{(4)} \) | \( x^{(5)} \) | \( x^{(6)} \) | \( x^{(7)} \) | \( x^{(8)} \) | \( x^{(9)} \) | \( x^{(10)} \) |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1.0000         | -0.7500        | -0.8594        | -0.9824        | -0.9978        | -0.9997        | -1.0000        | -1.0000        | -1.0000        | -1.0000        | -1.0000        |
| 1.0000         | 1.2813         | 1.0352         | 1.0044         | 1.0005         | 1.0001         | 1.0000         | 1.0000         | 1.0000         | 1.0000         | 1.0000         |
| 1.0000         | 0.2813         | 0.0352         | 0.0044         | 0.0005         | 0.0001         | 0.0000         | 0.0000         | 0.0000         | 0.0000         | 0.0000         |
| 1.0000         | 0.1953         | 0.0791         | 0.0099         | 0.0012         | 0.0002         | 0.0000         | 0.0000         | 0.0000         | 0.0000         | 0.0000         |

By Table 1, we can see that the Gauss-Seidel iteration converges in the 10th step. In view of the entries of \( x^{(k)} \), the condition

\[ \text{sign}(x_{\alpha,\beta}^{(k)}) = \text{sign}(x_{\alpha,\beta}^*) \]

has already been satisfied in the second iteration. This fact implies that the iteration can be ended earlier by some technique. If we introduce the cluster technique into \( p^{(k)} \) in this iteration, where

\[ p^{(k)}_i = \frac{|x_i^{(k)} - x_i^{(k-1)}|}{|x_i^{(k-1)}|}, i = 1, 2, 3, 4, \]

the results of the \( K \)-means cluster is listed in Table 2.

**Table 2. \( K \)-means cluster results of \( p^{(k)}_i \) in Example 1**

| cluster indices | \( p^{(1)} \) | \( p^{(2)} \) | \( p^{(3)} \) | \( p^{(4)} \) | \( p^{(5)} \) | \( p^{(6)} \) | \( p^{(7)} \) | \( p^{(8)} \) | \( p^{(9)} \) | \( p^{(10)} \) |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              |
| 1               | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              | 1              |
| 2               | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              |
| 2               | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              | 2              |
| centroid        | 0.7617         | 0.7350         | 0.8750         | 0.8750         | 0.8750         | 0.8750         | 0.8750         | 0.8750         | 0.8750         | 0.8750         |
| locations       | 0.2656         | 0.1690         | 0.0864         | 0.0097         | 0.0012         | 0.0002         | 0.0000         | 0.0000         | 0.0000         | 0.0000         |

By Table 2, the partitioning result satisfies \( \text{sign}(x^{(k)}) = \text{sign}(x^*) \) when the cluster technique is applied first in the second iteration. On the other hand, with the iteration step increasing, the distance between the two centroid locations of the cluster becomes large. This means that the proposed cluster technique works.

**Example 3.4** [1] Consider the linear complementarity problem from five-point finite difference discretization of a free boundary value problem about the flow of water through a porous dam. By a modulus-based method, it can be equivalently transformed to a GAVE, where

\[ A = \tilde{A} + 4I, \quad B = \tilde{A} - 4I, \]
0 0 0
0 0 0
\hat{0}
0 0 0
0 0 0
S I
I S I
I S
A

\begin{pmatrix}
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{pmatrix} \in \mathbb{R}^{n \times n},

n = m^2, S = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m \times m}, I \in \mathbb{R}^{m \times m} \text{ is the identity matrix and } 0 \in \mathbb{R}^{m \times m} \text{ is the zero matrix.}

Let the tolerance be $10^{-7}$. In Step 8) of Method 2.2, we use the CG method [5] to solve (1). Comparing to the Gauss-Seidel method, the numerical results of Example 3.4 are presented in Table 3, where “IT” and “CPU” denote the iteration step and CPU time in seconds.

| m       | IT | Gauss-Seidel method | Method 2.2 |
|---------|----|---------------------|-------------|
| 200     | 18 | 9                   |
|         | CPU| 0.1453              | 0.0774      |
| 400     | 15 | 9                   |
|         | CPU| 0.4686              | 0.2650      |
| 600     | 18 | 9                   |
|         | CPU| 1.0018              | 0.6749      |
| 1200    | 15 | 9                   |
|         | CPU| 2.6726              | 1.8127      |

As Table 3 shown, Method 2.2 converges faster than Gauss-Seidel method, which also implies that the partition technique works.

4. Conclusion
In this paper, by the analysis of the entries of the iteration vector, a partition technique based on the $K$-means cluster is introduced to find the signs of the solution of the GAVE. This technique is shown to be significant for identifying the signs of the iteration vector, which play an important role in constructing an effective method for solving the GAVE. Numerical examples confirm the efficiency of the proposed technique.

References
[1] Baiocchi C, Comincioli V, Guerri L and Volpi G 1973 Free boundary problems in the theory of fluid flow through porous media: A numerical approach Calcolo, 10 (1) 1-85
[2] Caccetta L, Qu B and Zhou G 2011 A globally and quadratically convergent method for absolute value equations Comput. Optim. Appl. 48 (1) 45-58
[3] Edalatpour V, Hezari D and Salkuyeh D K 2015 Generalized SOR iterative method for a class of complex symmetric linear system of equations Appl. Math. Comput. 92 802-805
[4] Ferris M C and Pang J S. 1997 Complementarity and Variational Problems: State of the Arts (Philadelphia: SIAM Publisher)
[5] Hestenes M R and Steifel E 1952 Method of conjugate gradient for solving linear equations J. Res. Nati. Bur. Stand. 49 409-436
[6] Javed I, Asif I and Muhammad A 2015 Levenberg-Marquardt method for solving systems of absolute value equations J. Comput. Appl. Math. 282 134-138
[7] Mangasarian O L and Pang J S 1995 The Extended Linear Complementarity Problem SIAM J.
Matrix Anal. Appl. 16 359-368

[8] Mangasarian O L and Meyer R R 2006 Absolute value equations Linear Algebra Appl. 419 (2-3) 359-367

[9] Mangasarian O L 2007 Absolute value equation solution via concave minimization Optim. Lett. 1 (1) 3-5

[10] Mangasarian O L 2007 Absolute value programming Comput. Optim. Appl. 36 (1) 43-53

[11] Mangasarian O L 2009 A generalized Newton method for absolute value equations Optim. Lett. 3 (1) 101-108

[12] Mangasarian O L 2015 A hybrid algorithm for solving the absolute value equation Optim. Lett. 9 (7) 1469-1474

[13] Noor M A, Iqbal J, Noor K I and Al-Said E 2012 On an iterative method for solving absolute value equations Optim. Lett. 6 (5) 1027-1033

[14] Prokopyev O 2009 On equivalent reformulations for absolute value equations Comput. Optim. Appl. 44 (3) 363-372

[15] Rohn J 2004 A theorem of the alternatives for the equation Ax+B|x|=b Linear Multilinear Algebra 52 (6) 421-426