The Newton-type splitting iterative method for a class of coupled Sylvester-like absolute value equation*

Changfeng Ma[1,2]†, Yuhong Wu[2], Yajun Xie[3]‡

(1. School of Data and Computer Science, Xiamen Institute of Technology, Xiamen 361021, P.R. China)

(2. School of Mathematics and Statistics, Fujian Normal University, Fuzhou 350117, P.R. China)

(3. School of Big data, Fuzhou University of International Studies and Trade, Fuzhou 350202, P.R. China)

Abstract In this paper, the Newton-type splitting iterative method for a class of coupled Sylvester-like absolute value equation is proposed. Some sufficient conditions for the existence of the unique solution of the coupled Sylvester-like absolute value equation are given and sufficient conditions for the nonexistence of solution is discussed. The Newton-base bimatrix splitting iteration method, the Newton-base generalized Gauss-Seidel bimatrix splitting iteration method and the inexact relaxed generalized Newton bimatrix splitting method are proposed to solve the coupled Sylvester-like absolute value equation. Numerical experiments confirm the conclusions proposed in this paper.

Keywords: coupled Sylvester-like absolute value equation; unique solution; kronecker product; bimatrix splitting.

1 Introduction

In this paper, we consider a class of coupled Sylvester-like absolute value equation:

$$\begin{cases} A_1XB_1 + C_1|Y|D_1 = E_1, \\ A_2YB_2 + C_2|X|D_2 = E_2, \end{cases} \tag{1.1}$$

where $A_1, A_2, C_1, C_2 \in \mathbb{R}^{m\times n}, B_1, B_2, D_1, D_2 \in \mathbb{R}^{p\times q}, E_1, E_2 \in \mathbb{R}^{m\times q}$ are known, $X, Y \in \mathbb{R}^{n\times p}$ are unknown. Here, $|X| = (\hat{x}_{ij}), \hat{x}_{ij} = |x_{ij}|, i = 1, \cdots, n, j = 1, \cdots, p$. When A_2, B_2, D_2 are identity matrix with appropriate sizes, $E_2 = \mathbf{0}_{m\times q}$ and $-C_2$ is a $n \times n$ identity matrix, then (1.1) reduces to Sylvester-like absolute value equation

$$AXB + C|X|D = E \tag{1.2}$$

*Supported by the National natural Science Foundation of China (12371378).

†Corresponding author. Email address: macf@fjnu.edu.cn (C.-F. Ma).

‡Co-corresponding author. Email address: xyj@fzfu.edu.cn (Y.-J. Xie).

in [18]. Here $A, C \in \mathbb{R}^{m\times n}, B, D \in \mathbb{R}^{p\times q}, E \in \mathbb{R}^{m\times q}, X \in \mathbb{R}^{n\times p}$.

When $p = q = 1, m = n, B = D = 1$, (1.2) reduces to the generalized absolute value equation (GAVE)

$$Ax + C|x| = e \tag{1.3}$$

in [1]. In particular, when $C = -I_n$ where I_n is a $n \times n$ identity matrix, (1.3) becomes absolute value equation (AVE)

$$Ax - |x| = e \tag{1.4}$$

in [2].

Recently, in [23], AVE (1.4) is expressed as the nonlinear equation

$$F(x) = Ax - |x| - e = 0, \tag{1.5}$$

and using the Newton iterative method $x^{(k+1)} = x^{(k)} - F^{'}(x^{(k)})^{-1}F(x^{(k)})$, then generalized Newton method (GN)

$$x^{(k+1)} = x^{(k)} - (A - D(x^{(k)}))^{-1}(Ax^{(k)} - |x^{(k)}| - e) \tag{1.6}$$

is obtained, where $F^{'}(x^{(k)})$ denote the Jacobin of F at $x^{(k)}$ and $D(x^{(k)}) = \text{diag}(\text{sign}(x^{(k)}))$. Also, the inexact version of the GN method

$$(A - D(x^{(k)}))x^{(k+1)} = e + r_k \text{ with } \|r_k\| \leqslant \theta\|Ax^{(k)} - |x^{(k)}| - e\| \tag{1.7}$$

is investigated for solving the AVE (1.4) in [26].

In the calculation, due to the change of matrix $A - D(x^{(k)})$ in the GN method, the computations of the generalized Newton method may be very expensive. To avoid changing the Jacobian, Wang, Cao and Chen utilize $A + \Omega$ as the approximation of $F^{'}(x^{(k)})$ and then get the modified Newton method (MN):

$$x^{(k+1)} = x^{(k)} - (A + \Omega)^{-1}(Ax^{(k)} - |x^{(k)}| - e), \tag{1.8}$$

Ω is positive semi-definite here.

Absolute value equation may arise in diverse fields, including complementarity problem, programming problem, and so on, see [3–9]. In recent years, many scholars have studied the properties of GAVE (1.3) including Mezzadri in [10], Propkeyev in [11], Rohn in [5,12,14], Wu and Li in [15], and Mangasarian and Meyer in [7]. The theoretical study of GAVE mainly focuses on its solvability and uniqueness, such as [13], [16] and [19]. See Lemma 2.1 in Section 2 for details. There are also many scholars studying some other forms of AVE, such as [1] and [18].

If we can convert (1.1) to (1.3), it is easy to obtain some sufficient conditions for the existence of the unique solution to the coupled Sylvester-like absolute value equation (1.1) and some theorems that (1.1) has no solution.

From observation, the theory and practice of (1.1) is interesting and challenging because it has three characteristics. (i)The most obvious feature is that compared with GAVE (1.3), the solution is no longer a vector but a matrix solution pair; (ii)There are non-differentiable terms $C_1|Y|D_1, C_2|X|D_2$ in (1.1); (iii)The coefficient matrices appear on both sides of the unknown X, Y and $|X|, |Y|$.

Since there are nonlinear terms $C_1|Y|D_1, C_2|X|D_2$ in (1.1), determining the existence of the solution of (1.1) is an NP-hard problem. In order to determine the existence of the solution, we use the properties of the Kronecker product and the appropriate assumptions to convert (1.1) to (1.3) and then we can give the existence of the solution of (1.1).

The rest of the paper is organized as follows: In section 2, we give some useful lemmas to help us obtain the results presented in this paper. In section 3, sufficient conditions for the existence of the unique solution of coupled Sylvester-like absolute value equation (1.1) are given. A sufficient condition that the solution does not exist is also discussed. In Section 4, we provide some methods for solving coupled Sylvester-like absolute value equation. In Section 5, the numerical results is used to verify the theorems presented. In Section 6, we summarize the work done in this paper.

At the end of this section, we present some notations which will be used throughout this paper. Let $\mathbb{R}^{m\times n}$ be the set of all $m \times n$ real matrices and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. The transposition of matrix A is denoted by A^T. $|\cdot|$ denotes the absolute value for real scalar. For $x \in \mathbb{R}^n$, $\|x\|$ denotes its 2-norm and $\mathrm{diag}(x)$ indicates a diagonal matrix with x_i as its diagonal entries for every $i = 1, 2, \cdots, n$. I_n be the identity matrix of order n, zero matrix of order n identified by $\mathbf{0}_n$, a matrix of order n with all entries α denoted by $\mathbf{1}_n(\alpha)$, lower triangular and upper triangular matrix of order n with all entries α denoted by $L_n(\alpha)$ and $U_n(\alpha)$ respectively. Also, tridiagonal matrix of order n with all elements on the main diagonal, first diagonal below, and the first diagonal above the main diagonal equal β, α and γ respectively, by $T_n(\alpha, \beta, \gamma)$. By default, $\|A\|$ denotes the spectral norm of A and is defined by the formula $\|A\| := \max\{\|Ax\| : x \in \mathbb{R}^n, \|x\| = 1\}$.

2 Preliminaries

In this section, we give some useful lemmas to help us reach the conclusions proposed in this paper.

First we review the definition of the vec operator and Kronecker product.

If $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, then $\mathrm{vec}(A) = (a_{11}, \cdots, a_{m1}, a_{12}, \cdots, a_{m2}, \cdots, a_{1n}, \cdots, a_{mn})^T$.

Let $A = (a_{ij}) \in \mathbb{R}^{m\times n}, B = (b_{ij}) \in \mathbb{R}^{p\times q}$, we call the block matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp\times nq}$$

the Kronecker product of A and B, and abbreviated as $A \otimes B = (a_{ij}B)$.

Lemma 2.1. *If any of the following conditions hold, the generalized absolute value equation (1.3)*

$$Ax + C|x| = e$$

has the unique solution for each right-hand side vector e.

(i) $\sigma_{\max}(C) < \sigma_{\min}(A)$*, where* $\sigma_{\max}(C)$ *denotes the largest singular value of C and* $\sigma_{\min}(A)$ *denotes the smallest singular value of A. [15]*

(ii) A and C are square matrices, A is nonsingular and $\sigma_{max}(A^{-1}C) < 1$*, where* $\sigma_{max}(A^{-1}C)$ *denotes the largest singular value of* $A^{-1}C$*. [16]*

(iii) The inequality $|Ax| \leqslant |C||x|$ *has only the trivial solution* $x = 0$ *where A and C are real square matrices. [5]*

Lemma 2.2. *For any real matrices A, B, C and D with appropriate sizes, the following conclusions are valid*

(i) $vec(ABC) = \left(C^T \otimes A\right) vec(B)$.

(ii) $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.

(iii) Let A and B be square matrices. The eigenvalues of $A \otimes B$ *consist of all pairwise products of the eigenvalues of A and B. In particular,* $\rho(A \otimes B) = \rho(A)\rho(B)$.

(iv) The singular values of $A \otimes B$ *consist of all pairwise products of the singular values of A and B. In particular,* $\sigma_{max}(A \otimes B) = \sigma_{max}(A)\sigma_{max}(B), \sigma_{min}(A \otimes B) = \sigma_{min}(A)\sigma_{min}(B)$.

(v) If square matrices A and B are nonsingular, then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.

(vi) $|A \otimes B| = |A| \otimes |B|$.

Lemma 2.3. *[17] Let* $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times n}$*. If* $|A| \leqslant B$*, then* $\rho(A) \leqslant \rho(B)$.

Proposition 2.1. *Let* $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times n}$*. If* $|A| \leqslant B$*, then* $\|A\|_2 \leqslant \|B\|_2$.

Proof. From $|A| \leqslant B$, it is following that $|A^TA| \leqslant |A^T||A| \leqslant B^TB$. By the Lemma 2.3, $\rho(A^TA) \leqslant \rho(B^TB)$. Therefore, $\|A\|_2 \leqslant \|B\|_2$. □

Lemma 2.4. *[25] Let λ be any root of the quadratic equation* $x^2 - bx + c = 0$ *where* $b, c \in \mathbb{R}$*. Then* $|\lambda| < 1$ *if and only if* $|c| < 1$ *and* $|b| < 1 + c$.

Lemma 2.5. *[20] Let* $x,\ y \in \mathbb{R}^n$*, then* $||x| - |y|| \leqslant |x - y|$.

3 Existence of solutions for the coupled Sylvester-like absolute value equation

Using the vec operator, (1.1) is equivalent to

$$\begin{cases} S_1 x + T_1|y| = e_1, \\ S_2 y + T_2|x| = e_2, \end{cases} \tag{3.1}$$

where $S_1 = B_1{}^T \otimes A_1, S_2 = B_2{}^T \otimes A_2, T_1 = D_1{}^T \otimes C_1, T_2 = D_2{}^T \otimes C_2, e_1 = \mathrm{vec}(E_1), e_2 = \mathrm{vec}(E_2), x = \mathrm{vec}(X), y = \mathrm{vec}(Y)$. Then (3.1) can be expressed as the following generalized absolute value equation form

$$\mathcal{A}z + \mathcal{B}|z| = \mathcal{E}, \tag{3.2}$$

where

$$\mathcal{A} = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}, \mathcal{B} = \begin{pmatrix} 0 & T_1 \\ T_2 & 0 \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, \mathcal{E} = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}. \tag{3.3}$$

In the following content, let λ_{max} represents the maximum eigenvalue of the corresponding matrix and λ_{min} represents the minimum eigenvalue of the corresponding matrix. σ_{max} represents the maximum singular value of the corresponding matrix and σ_{min} represents the minimum singular value of the corresponding matrix.

Theorem 3.1. *Suppose C_i, D_i are invertible, $i = 1, 2$. (1.1) is uniquely solvable for any right-hand side matrices E_1, E_2 if any of the following is true:*

(i) $\max_i\{\sigma_{max}(D_i)\sigma_{max}(C_i)\} < \min_i\{\sigma_{min}(B_i)\sigma_{min}(A_i)\}, i = 1, 2.$

(ii) *A_1, A_2, B_1, B_2 are square nonsingular matrices,*

$$\begin{cases} \sigma_{max}(D_1 B_1{}^{-1})\sigma_{max}(A_1{}^{-1}C_1) < 1, \\ \sigma_{max}(D_2 B_2{}^{-1})\sigma_{max}(A_2{}^{-1}C_2) < 1. \end{cases}$$

(iii) *$A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2$ are real square matrice and the inequality*

$$\begin{cases} |A_1 X B_1| \leqslant |C_1||Y||D_1|, \\ |A_2 Y B_2| \leqslant |C_2||X||D_2|, \end{cases}$$

has only the trivial solution-pair $(X, Y) = (0, 0)$.

Proof. First, prove the first part. According to the definition of singular value, we can know that

$$\begin{aligned}
\sigma_{min}(\mathcal{A}) &= \sigma_{min}\begin{pmatrix} B_1{}^T \otimes A_1 & 0 \\ 0 & B_2{}^T \otimes A_2 \end{pmatrix} \\
&= \sqrt{\lambda_{min}\left(\begin{pmatrix} (B_1{}^T \otimes A_1)^T & 0 \\ 0 & (B_2{}^T \otimes A_2)^T \end{pmatrix}\begin{pmatrix} B_1{}^T \otimes A_1 & 0 \\ 0 & B_2{}^T \otimes A_2 \end{pmatrix}\right)} \\
&= \sqrt{\lambda_{min}\begin{pmatrix} (B_1{}^T \otimes A_1)^T(B_1{}^T \otimes A_1) & 0 \\ 0 & (B_2{}^T \otimes A_2)^T(B_2{}^T \otimes A_2) \end{pmatrix}}
\end{aligned}$$

It is not difficult to see that for any $A, B \in \mathbb{R}^{n\times n}$, the eigenvalues of the matrix $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$ are the eigenvalues of A and the eigenvalues of B. Hence

$$\begin{aligned} \sigma_{min}(\mathcal{A}) &= \sqrt{\min\{\lambda_{min}((B_1{}^T \otimes A_1)^T(B_1{}^T \otimes A_1)), \lambda_{min}((B_2{}^T \otimes A_2)^T(B_2{}^T \otimes A_2))\}} \\ &= \min\{\sigma_{min}(B_1{}^T \otimes A_1), \sigma_{min}(B_2{}^T \otimes A_2)\} \\ &= \min\{\sigma_{min}(B_1)\sigma_{min}(A_1), \sigma_{min}(B_2)\sigma_{min}(A_2)\}, \end{aligned} \tag{3.4}$$

where the last equation comes from part (iv) of Lemma 2.2.

Similarly, it can be obtained

$$\sigma_{max}(\mathcal{B}) = \sigma_{max}\begin{pmatrix} 0 & D_1{}^T \otimes C_1 \\ D_2{}^T \otimes C_2 & 0 \end{pmatrix} = \max\{\sigma_{max}(D_1)\sigma_{max}(C_1), \sigma_{max}(D_2)\sigma_{max}(C_2)\}. \tag{3.5}$$

When $\max\{\sigma_{max}(D_1)\sigma_{max}(C_1), \sigma_{max}(D_2)\sigma_{max}(C_2)\} < \min\{\sigma_{min}(B_1)\sigma_{min}(A_1), \sigma_{min}(B_2)\sigma_{min}(A_2)\}$, we can obtain

$$\sigma_{max}\begin{pmatrix} 0 & D_1{}^T \otimes C_1 \\ D_2{}^T \otimes C_2 & 0 \end{pmatrix} < \sigma_{min}\begin{pmatrix} B_1{}^T \otimes A_1 & 0 \\ 0 & B_2{}^T \otimes A_2 \end{pmatrix}.$$

According to part (i) of Lemma 2.1, (3.2) has the unique solution, that is, (1.1) has the unique solution-pair.

Next, similar to (3.5), we can see

$$\begin{aligned} \sigma_{max}(\mathcal{A}^{-1}\mathcal{B}) &= \sigma_{max}\left(\begin{pmatrix} B_1{}^T \otimes A_1 & 0 \\ 0 & B_2{}^T \otimes A_2 \end{pmatrix}^{-1}\begin{pmatrix} 0 & D_1{}^T \otimes C_1 \\ D_2{}^T \otimes C_2 & 0 \end{pmatrix}\right) \\ &= \max\{\sigma_{max}(B_2{}^{-T}D_2{}^T \otimes A_2^{-1}C_2), \sigma_{max}(B_1{}^{-T}D_1{}^T \otimes A_1^{-1}C_1)\} \\ &= \max\{\sigma_{max}(D_2B_2{}^{-1})\sigma_{max}(A_2^{-1}C_2), \sigma_{max}(D_1B_1{}^{-1})\sigma_{max}(A_1^{-1}C_1)\} < 1, \end{aligned}$$

where the last equation is derived from part (iv) of Lemma 2.2. Thus, applying part (ii) of Lemma 2.1 to the AVE (3.2) we know that (3.2) has a unique solution. Then (1.1) is uniquely solvable for any right-hand side matrices E_1, E_2.

Finally, according to equivalence of (1.1) and (3.2) and part (iii) of Lemma 2.1, if

$$\left|\begin{pmatrix} B_1{}^T \otimes A_1 & 0 \\ 0 & B_2{}^T \otimes A_2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}\right| \leqslant \left|\begin{pmatrix} 0 & D_1{}^T \otimes C_1 \\ D_2{}^T \otimes C_2 & 0 \end{pmatrix}\right|\left|\begin{pmatrix} x \\ y \end{pmatrix}\right| \tag{3.6}$$

has only the trivial solution pair, then (1.1) has a unique solution-pair. And it is easy to find (3.6) can be converted to

$$\begin{pmatrix} |(B_1{}^T \otimes A_1)x| \\ |(B_2{}^T \otimes A_2)y| \end{pmatrix} \leqslant \begin{pmatrix} |D_1{}^T \otimes C_1||y| \\ |D_2{}^T \otimes C_2||x| \end{pmatrix}, \tag{3.7}$$

i.e.,

$$\begin{cases} |\text{vec}(A_1XB_1)| \leqslant \text{vec}(|C_1||Y||D_1|), \\ |\text{vec}(A_2YB_2)| \leqslant \text{vec}(|C_2||X||D_2|). \end{cases}$$

That is $\begin{cases} |A_1XB_1| \leqslant |C_1||Y||D_1| \\ |A_2YB_2| \leqslant |C_2||X||D_2| \end{cases}$ has only the trivial solution-pair $(X, Y) = (0, 0)$, then (1.1) has the unique solution-pair. □

Remark 3.1. *Notice that for any nonsingular matrix A, we have $\sigma_{min}(A) \cdot \sigma_{max}(A^{-1}) = 1$. Hence, when C_1, C_2, D_1, D_2 are nonsingular matrices in (1.1),*

$$\begin{cases} \sigma_{min}(B_1{D_1}^{-1})\sigma_{min}({C_1}^{-1}A_1) > 1, \\ \sigma_{min}(B_2{D_2}^{-1})\sigma_{min}({C_2}^{-1}A_2) > 1, \end{cases}$$

can be substituted for the condition in (ii) of Theorem 3.2. By the simple computations, we get $\sigma_{min}\left(\mathcal{B}^{-1}\mathcal{A}\right) \geqslant \sigma_{min}\left(\mathcal{B}^{-1}\right)\sigma_{min}(\mathcal{A}) = \frac{\sigma_{min}(\mathcal{A})}{\sigma_{max}(\mathcal{B})}$. It can be seen that condition

$$\begin{cases} \sigma_{min}(B_1{D_1}^{-1})\sigma_{min}({C_1}^{-1}A_1) > 1, \\ \sigma_{min}(B_2{D_2}^{-1})\sigma_{min}({C_2}^{-1}A_2) > 1, \end{cases}$$

is slightly weaker than condition $\max\limits_{i=1,2}\{\sigma_{max}(D_i)\sigma_{max}(C_i)\} < \min\limits_{i=1,2}\{\sigma_{min}(B_i)\sigma_{min}(A_i)\}$.

The following theorem states that the regularity of the interval matrix can also be used to guarantee the unique solvability of (1.1) for any right-hand side matrices E_1, E_2.

First review the definition of interval matrix. Given two matrices $\underline{T} = (\underline{t}_{ij})$ and $\overline{T} = (\overline{t}_{ij})$, an interval matrix $[\underline{T}, \overline{T}]$ is defined by $[\underline{T}, \overline{T}] := \{T : \underline{T} \leqslant T \leqslant \overline{T}\}$, where for two matrices $X = (x_{ij})$ and $Y = (y_{ij})$, matrix inequality $X \leqslant Y$ refers to $x_{ij} \leqslant y_{ij}$ for any i, j.

Theorem 3.2. *For any $U_1 \in [-|T_1|, |T_1|], U_2 \in [-|T_2|, |T_2|]$, if $\begin{pmatrix} S_1 & U_1 \\ U_2 & S_2 \end{pmatrix}$ is nonsingular matrix, then (1.1) is uniquely solvable for any right-hand side matrices E_1, E_2.*

Proof. According to Theorem 2.2 in [19], the theorem can be obtained directly. □

Theorem 3.3. *Let C_1, C_2, D_1, D_2 be square nonsingular matrices and $0 \neq {C_1}^{-1}E_1{D_1}^{-1}, {C_2}^{-1}E_2{D_2}^{-1} \geqslant 0$,*

$$\prod_{i=1}^{2} \sigma_{max}(A_i)\sigma_{max}(B_i) < \prod_{i=1}^{2} \sigma_{min}(C_i)\sigma_{min}(D_i), \tag{3.8}$$

then

$$\begin{cases} A_1XB_1 - C_1|Y|D_1 = E_1, \\ A_2YB_2 - C_2|X|D_2 = E_2, \end{cases} \tag{3.9}$$

has no solution.

Proof. It is assumed that (3.9) has a non-zero solution. Since C_1, C_2, D_1, D_2 are square nonsingular matrices, we have

$$\begin{cases} {C_1}^{-1}A_1XB_1{D_1}^{-1} - |Y| = {C_1}^{-1}E_1{D_1}^{-1} \geqslant 0, \\ {C_2}^{-1}A_2YB_2{D_2}^{-1} - |X| = {C_2}^{-1}E_2{D_2}^{-1} \geqslant 0, \end{cases} \text{i.e.,} \begin{cases} {C_1}^{-1}A_1XB_1{D_1}^{-1} \geqslant |Y|, \\ {C_2}^{-1}A_2YB_2{D_2}^{-1} \geqslant |X|. \end{cases}$$

According to Proposition 2.1 and norm inequality, we get

$$\begin{aligned} \|Y\| &\leqslant \|{C_1}^{-1}A_1XB_1{D_1}^{-1}\| \\ &\leqslant \|{C_1}^{-1}\|\|A_1\|\|X\|\|B_1\|\|{D_1}^{-1}\|, \\ \|X\| &\leqslant \|{C_2}^{-1}A_2YB_2{D_2}^{-1}\| \\ &\leqslant \|{C_2}^{-1}\|\|A_2\|\|Y\|\|B_2\|\|{D_2}^{-1}\| \\ &\leqslant \|{C_2}^{-1}\|\|A_2\|\|{C_1}^{-1}\|\|A_1\|\|X\|\|B_1\|\|{D_1}^{-1}\|\|B_2\|\|{D_2}^{-1}\|. \end{aligned} \tag{3.10}$$

And (3.8) can be written as $\|A_1\|\|B_1\|\|A_2\|\|B_2\| < \frac{1}{\|{C_1}^{-1}\|\|{D_1}^{-1}\|\|{C_2}^{-1}\|\|{D_2}^{-1}\|}$. Combining (3.10),we get the contradiction $\|X\| < \|X\|$. □

Remark 3.2. *A special case of Theorem 3.3 concerns the coupled Sylvester-like AVE*

$$\begin{cases} A_1XB_1 - |Y| = E_1, \\ A_2YB_2 - |X| = E_2. \end{cases} \tag{3.11}$$

A sufficient condition for the non-solvability of this equation is that $0 \neq E_1, E_2 \geqslant 0$ *and*

$$\prod_{i=1}^{2} \sigma_{max}(A_i)\sigma_{max}(B_i) < 1. \tag{3.12}$$

4 Solve the coupled Sylvester-like absolute value equation

4.1 The Newton-base bimatrix splitting iteration method for solving the coupled Sylvester-like absolute value equation

In this section, the Newton-base bimatrix splitting iteration method is established to solve the (1.1). In the rest of this paper, we assume that $A_1, A_2, C_1, C_2 \in \mathbb{R}^{m\times m}, B_1, B_2, D_1, D_2 \in \mathbb{R}^{n\times n}, E_1, E_2 \in \mathbb{R}^{m\times n}$ and $X, Y \in \mathbb{R}^{m\times n}$.

From the above, (1.1) is equivalent to

$$\mathcal{F}(z) = 0, \text{ with } \mathcal{F}(z) = \mathcal{A}z + \mathcal{B}|z| - \mathcal{E}, \tag{4.1}$$

where

$$\mathcal{A} = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}, \mathcal{B} = \begin{pmatrix} 0 & T_1 \\ T_2 & 0 \end{pmatrix}, z = \begin{pmatrix} x \\ y \end{pmatrix}, \mathcal{E} = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}.$$

Thus, the Newton iteration method can be written as

$$z^{(k+1)} = z^{(k)} - \mathcal{F}'\left(z^{(k)}\right)^{-1} \mathcal{F}\left(z^{(k)}\right), \quad k = 0, 1, 2, \ldots. \tag{4.2}$$

Then generalized Newton method (GN) [23] can be expressed as

$$z^{(k+1)} = z^{(k)} - (\mathcal{A} + \mathcal{B}D(z^{(k)}))^{-1}(\mathcal{A}z^{(k)} + \mathcal{B}|z^{(k)}| - \mathcal{E}), \tag{4.3}$$

where $D(z^{(k)}) = \text{diag}(\text{sign}(z^{(k)}))$. The modified Newton method (MN) [24] is proposed accordingly to

$$z^{(k+1)} = z^{(k)} - (\mathcal{A} + \Omega)^{-1}(\mathcal{A}z^{(k)} + B|z^{(k)}| - \mathcal{E}), \tag{4.4}$$

Ω is positive semi-definite here. But if $\mathcal{A} + \Omega$ is ill-conditioned, the MN method may be expensive in practical calculations. Furthermore, in [6], the author proposes a Newton-based matrix splitting method (NM)

$$z^{(k+1)} = z^{(k)} - (\mathcal{M} + \Omega)^{-1}(\mathcal{A}z^{(k)} + \mathcal{B}|z^{(k)}| - \mathcal{E}), \tag{4.5}$$

where $\mathcal{A} = \mathcal{M} - \mathcal{N}$ and Ω is positive semi-definite.

In order to improve iteration efficiency, based on the MN method and the NM method, we propose the Newton-base bimatrix splitting iteration method combining the characteristics of the coupled Sylvester-like absolute value equation. The matrices A_1, A_2 split into $A_1 = m_1 - n_1, A_2 = m_2 - n_2$, where $\Omega_1, \Omega_2 \in \mathbb{R}^{n\times n}$ satisfy $m_1 + \Omega_1, m_2 + \Omega_2$ are invertible. Set $M_1 = m_1 + \Omega_1, M_2 = m_2 + \Omega_2, N_1 = n_1 + \Omega_1, N_2 = n_2 + \Omega_2$. The split of the matrices B_1, B_2 are $B_1 = P_1 - Q_1, B_2 = P_2 - Q_2$, respectively. Then we can get the Newton-base bimatrix splitting iteration method

$$\begin{cases} X = M_1^{-1}(E_1 - C_1|Y|D_1 + N_1XB_1 + M_1XQ_1)P_1^{-1}, \\ Y = M_2^{-1}(E_2 - C_2|X|D_2 + N_2YB_2 + M_2YQ_2)P_2^{-1}. \end{cases} \tag{4.6}$$

Algorithm 4.1. (*The Newton-base bimatrix splitting iteration method*)

Step 1 *Given initial point* $X^{(0)}, Y^{(0)} \in \mathbb{R}^{m\times n}$ *and the parameter* $\varepsilon > 0$. *Assume the split of the matrices* A_1, A_2, B_1, B_2 *are* $A_1 = m_1 - n_1, A_2 = m_2 - n_2, B_1 = P_1 - Q_1, B_2 = P_2 - Q_2$, *respectively. Given* $\Omega_1, \Omega_2 \in \mathbb{R}^{m\times m}$ *which satisfies* $M_1 = m_1 + \Omega_1, M_2 = m_2 + \Omega_2$ *are invertible and* $N_1 = n_1 + \Omega_1, N_2 = n_2 + \Omega_2$.

Step 2 $\sqrt{\|A_1X^{(k)}B_1 + C_1|Y^{(k)}|D_1 - E_1\|^2 + \|A_2Y^{(k)}B_2 + C_2|X^{(k)}|D_2 - E_2\|^2}/\sqrt{\|E_1\|^2 + \|E_2\|^2} < \varepsilon$, *stop.*

Step 3 *Compute* $x^{(k+1)}$ *and* $y^{(k+1)}$ *by*

$$\begin{cases} X^{(k+1)} = M_1^{-1}(E_1 - C_1|Y^{(k)}|D_1 + N_1X^{(k)}B_1 + M_1X^{(k)}Q_1)P_1^{-1}, \\ Y^{(k+1)} = M_2^{-1}(E_2 - C_2|X^{(k+1)}|D_2 + N_2Y^{(k)}B_2 + M_2Y^{(k)}Q_2)P_2^{-1}. \end{cases} \tag{4.7}$$

Step 4 *Set* $k := k + 1$ *and go to Step 2.*

Let (X^*, Y^*) are the solution-pair of the coupled Sylvester-like absolute value equation (1.1). The iteration errors $e_k^X = X^* - X^{(k)}, e_k^Y = Y^* - Y^{(k)}$ where $X^{(k)}, Y^{(k)}$ are generated by (4.7).

Theorem 4.1. *Let $A_1, A_2, C_1, C_2 \in \mathbb{R}^{m\times m}, B_1, B_2, D_1, D_2 \in \mathbb{R}^{n\times n}, E_1, E_2 \in \mathbb{R}^{m\times n}$. Given the split of the matrices A_1, A_2, B_1, B_2 are $A_1 = m_1 - n_1, A_2 = m_2 - n_2, B_1 = P_1 - Q_1, B_2 = P_2 - Q_2$, and $N_1 = n_1 + \Omega_1, N_2 = n_2 + \Omega_2 \in \mathbb{R}^{m\times m}, M_1 = m_1 + \Omega_1, M_2 = m_2 + \Omega_2 \in \mathbb{R}^{m\times m}$ are nonsingular. Denote $\|M_1^{-1}\|\|N_1\|\|B_1\|\|P_1^{-1}\| + \|M_1^{-1}\|\|M_1\|\|Q_1\|\|P_1^{-1}\| = s_1, \|M_2^{-1}\|\|N_2\|\|B_2\|\|P_2^{-1}\| + \|M_2^{-1}\|\|M_2\|\|Q_2\|\|P_2^{-1}\| = s_2, \|M_1^{-1}\|\|C_1\|\|D_1\|\|P_1^{-1}\| = t_1, \|M_2^{-1}\|\|C_2\|\|D_2\|\|P_2^{-1}\| = t_2$. If $s_1 + t_1t_2 + s_2 < 1$, then the Algorithm 4.1 is convergent.*

Proof. From (1.1), we have

$$\begin{cases} A_1X^*B_1 + C_1|Y^*|D_1 = E_1, \\ A_2Y^*B_2 + C_2|X^*|D_2 = E_2. \end{cases} \tag{4.8}$$

Then (4.8) is equivalent to

$$\begin{cases} X^* = M_1^{-1}(E_1 - C_1|Y^*|D_1 + N_1X^*B_1 + M_1X^*Q_1)P_1^{-1}, \\ Y^* = M_2^{-1}(E_2 - C_2|X^*|D_2 + N_2Y^*B_2 + M_2Y^*Q_2)P_2^{-1}. \end{cases} \tag{4.9}$$

Form (4.7), (4.9), we can get

$$\begin{aligned} \|e_{k+1}^X\| &= \|M_1^{-1}(C_1(|Y^*| - |Y^{(k)}|)D_1 + N_1(X^* - X^{(k)})B_1 + M_1(X^* - X^{(k)})Q_1)P_1^{-1}\| \\ &\leqslant \|M_1^{-1}\|(\|C_1\|\||Y^*| - |Y^{(k)}|\|\|D_1\| + \|N_1\|\|X^* - X^{(k)}\|\|B_1\| + \|M_1\|\|X^* - X^{(k)}\|\|Q_1\|)\|P_1^{-1}\| \\ &\leqslant \|M_1^{-1}\|(\|C_1\|\|e_k^Y\|\|D_1\| + \|N_1\|\|e_k^X\|\|B_1\| + \|M_1\|\|e_k^X\|\|Q_1\|)\|P_1^{-1}\| \\ &= (\|M_1^{-1}\|\|N_1\|\|B_1\|\|P_1^{-1}\| + \|M_1^{-1}\|\|M_1\|\|Q_1\|\|P_1^{-1}\|)\|e_k^X\| + \|M_1^{-1}\|\|C_1\|\|D_1\|\|P_1^{-1}\|\|e_k^Y\| \\ &= s_1\|e_k^X\| + t_1\|e_k^Y\|, \end{aligned} \tag{4.10}$$

and

$$\begin{aligned} \|e_{k+1}^Y\| &= \|M_2^{-1}(C_2(|X^*| - |X^{(k+1)}|)D_2 + N_2(Y^* - Y^{(k)})B_2 + M_2(Y^* - Y^{(k)})Q_2)P_2^{-1}\| \\ &\leqslant \|M_2^{-1}\|(\|C_2\|\||X^*| - |X^{(k+1)}|\|\|D_2\| + \|N_2\|\|Y^* - Y^{(k)}\|\|B_2\| + \|M_2\|\|Y^* - Y^{(k)}\|\|Q_2\|)\|P_2^{-1}\| \\ &\leqslant \|M_2^{-1}\|(\|C_2\|\|e_{k+1}^X\|\|D_2\| + \|N_2\|\|e_k^Y\|\|B_2\| + \|M_2\|\|e_k^Y\|\|Q_2\|)\|P_2^{-1}\| \\ &= (\|M_2^{-1}\|\|N_2\|\|B_2\|\|P_2^{-1}\| + \|M_2^{-1}\|\|M_2\|\|Q_2\|\|P_2^{-1}\|)\|e_k^Y\| + \|M_2^{-1}\|\|C_2\|\|D_2\|\|P_2^{-1}\|\|e_{k+1}^X\| \\ &= t_2\|e_{k+1}^X\| + s_2\|e_k^Y\| \\ &\leqslant t_2(s_1\|e_k^X\| + t_1\|e_k^Y\|) + s_2\|e_k^Y\| \\ &= t_2s_1\|e_k^X\| + (t_2t_1 + s_2)\|e_k^Y\|. \end{aligned} \tag{4.11}$$

Further,

$$\begin{aligned} \begin{pmatrix} \|e_{k+1}^X\| \\ \|e_{k+1}^Y\| \end{pmatrix} &\leqslant \begin{pmatrix} s_1 & t_1 \\ t_2s_1 & t_2t_1 + s_2 \end{pmatrix} \begin{pmatrix} \|e_k^X\| \\ \|e_k^Y\| \end{pmatrix} \\ &\leqslant \begin{pmatrix} s_1 & t_1 \\ t_2s_1 & t_2t_1 + s_2 \end{pmatrix}^2 \begin{pmatrix} \|e_{k-1}^X\| \\ \|e_{k-1}^Y\| \end{pmatrix} \\ &\vdots \\ &\leqslant \begin{pmatrix} s_1 & t_1 \\ t_2s_1 & t_2t_1 + s_2 \end{pmatrix}^{k+1} \begin{pmatrix} \|e_0^X\| \\ \|e_0^Y\| \end{pmatrix}. \end{aligned} \tag{4.12}$$

Let $W = \begin{pmatrix} s_1 & t_1 \\ t_2 s_1 & t_2 t_1 + s_2 \end{pmatrix}$, we know that when $\rho(W) < 1$, $\lim_{k\to\infty} W^k = 0$. It is shown that $\lim_{k\to\infty} \|e_k^X\| = 0$, $\lim_{k\to\infty} \|e_k^Y\| = 0$. In other words, the Algorithm 4.1 converges to the unique solution-pair (X^*, Y^*).

Next, we need to prove $\rho(W) < 1$. Let λ be the eigenvalue of the matrix W. Then λ satisfies

$$\lambda^2 - (s_1 + t_2 t_1 + s_2)\lambda + (s_1(t_2 t_1 + s_2) - t_2 s_1 t_1) = 0.$$

After simple calculations, we have

$$\lambda^2 - (s_1 + t_2 t_1 + s_2)\lambda + s_1 s_2 = 0. \tag{4.13}$$

From $s_1 + t_1 t_2 + s_2 < 1$, we can get $t_1 t_2 + 2\sqrt{s_1 s_2} < s_1 + t_1 t_2 + s_2 < 1 < 1 + s_1 s_2$. Then $2\sqrt{s_1 s_2} < 1 - t_1 t_2 < 1$. It is obviously that $s_1 s_2 < (\frac{1}{2})^2 < 1$. According to Lemma 2.4, $\rho(W) < 1$. This completes the proof. $\square$

Corollary 4.1. *Let A_1 be positive definite and $A_1 = m_1 - n_1$ be its a splitting, where m_1 is positive definite. Assume that the matrix $\Omega_1 \in \mathbb{R}^{m\times m}$ is positive diagonal. If*

$$\|m_1^{-1}\| < \frac{1 - s_2}{\omega\|P_1^{-1}\| + (1 - s_1)\|\Omega_1\|}, \tag{4.14}$$

where $\omega = \|N_1\|\|B_1\| + \|M_1\|\|Q_1\| + \|C_1\|\|D_1\|\|M_2^{-1}\|\|C_2\|\|D_2\|\|P_2{}^{-1}\|$, then Algorithm 4.1 is convergent.

Proof. According to the hypothesis, it is see to that matrix $m_1 + \Omega_1$ is positive definite. Clearly, matrices $m_1 + \Omega_1$ and m_1 are invertible.

From the Banach perturbation lemma, we have

$$\begin{aligned} \|(m_1 + \Omega_1)^{-1}\| &\leqslant \tfrac{\|m_1^{-1}\|}{1 - \|m_1^{-1}\|\|\Omega_1\|} \\ &< \tfrac{\frac{1-s_2}{\omega\|P_1^{-1}\| + (1-s_1)\|\Omega_1\|}}{1 - \frac{1-s_2}{\omega\|P_1^{-1}\| + (1-s_1)\|\Omega_1\|}\|\Omega_1\|} \\ &= \tfrac{1-s_2}{\omega\|P_1^{-1}\|}. \end{aligned}$$

Therefore, Algorithm 4.1 is convergent under the condition (4.14). $\square$

Corollary 4.2. *Set $\Omega_1 = \omega_1 I$. Assume that m_1, P_1 are symmetric positive definite matrices. If $\frac{1}{(\lambda_{min}(m_1)+\omega_1)\lambda_{min}(P_1)} < \frac{1-S_2}{\omega}$ where $\lambda_{min}(m_1), \lambda_{min}(P_1)$ are the smallest eigenvalue of matrix m_1 and the smallest eigenvalue of matrix P_1, respectively, then the Algorithm 4.1 is convergent.*

Proof. Clearly, we know $\|(m_1 + \Omega_1)^{-1}\| = \frac{1}{\lambda_{min}(m_1)+\omega_1}$ and $\|P_1^{-1}\| = \frac{1}{\lambda_{min}(P_1)}$. Therefore, when $\frac{1}{(\lambda_{min}(m_1)+\omega_1)\lambda_{min}(P_1)} < \frac{1-s_2}{\omega}$, we have $s_1 + t_1 t_2 + s_2 < 1$. Thus, Algorithm 4.1 is convergent. $\square$

4.2 The Newton-base generalized Gauss-Seidel bimatrix splitting iteration method for solving the coupled Sylvester-like absolute value equation

Now we propose another method to obtain the solution of the coupled Sylvester-like absolute value equation.

Recalling that the coupled Sylvester-like absolute value equation has the following form,

$$\begin{cases} A_1XB_1 + C_1|Y|D_1 = E_1, \\ A_2YB_2 + C_2|X|D_2 = E_2. \end{cases}$$

Multiplying λ, then we have

$$\begin{cases} \lambda A_1XB_1 + \lambda C_1|Y|D_1 = \lambda E_1, \\ \lambda A_2YB_2 + \lambda C_2|X|D_2 = \lambda E_2. \end{cases} \tag{4.15}$$

Let

$$A_1 = D_{A_1} - L_{A_1} + \Omega - (U_{A_1} + \Omega), A_2 = D_{A_2} - U_{A_2} + \Omega - (L_{A_1} + \Omega), \tag{4.16}$$

where $D_{A_1} = \mathrm{diag}(A_1), D_{A_2} = \mathrm{diag}(A_2)$, U_{A_1}, L_{A_1} are strictly upper and lower triangular parts of A_1 and U_{A_2}, L_{A_2} are strictly upper and lower triangular parts of A_2, respectively. Ω satisfies $M_1 = D_{A_1} - L_{A_1} + \Omega,\ M_2 = D_{A_2} - U_{A_2} + \Omega$ are invertible. And assume the split of the matrices B_1 and B_2 are $B_1 = P_1 - Q_1, B_2 = P_2 - Q_2$, respectively. According to (4.15), (4.16) can be suggested as

$$\begin{cases} \lambda(D_{A_1} - L_{A_1} + \Omega - (U_{A_1} + \Omega))X(P_1 - Q_1) + \lambda C_1|Y|D_1 = \lambda E_1, \\ \lambda(D_{A_2} - U_{A_2} + \Omega - (L_{A_1} + \Omega))Y(P_2 - Q_2) + \lambda C_2|X|D_2 = \lambda E_2. \end{cases}$$

After simple calculations, the above formula is transformed into

$$\begin{cases} \lambda(D_{A_1} - L_{A_1} + \Omega)X + (U_{A_1} + \Omega)X = \lambda E_1P_1^{-1} - \lambda C_1|Y|D_1{P_1}^{-1} + \lambda A_1XQ_1{P_1}^{-1} + (\lambda+1)(U_{A_1} + \Omega)X, \\ \lambda(D_{A_2} - U_{A_2} + \Omega)Y + (L_{A_2} + \Omega)Y = \lambda E_2P_2^{-1} - \lambda C_2|X|D_2{P_2}^{-1} + \lambda A_2YQ_2{P_2}^{-1} + (\lambda+1)(L_{A_2} + \Omega)Y. \end{cases} \tag{4.17}$$

Using the iterative scheme, (4.17) can be written as

$$\begin{cases} X^{(k+1)} = (\lambda M_1)^{-1}(\lambda E_1P_1^{-1} - N_1X^{(k+1)} - \lambda C_1|Y^{(k)}|D_1{P_1}^{-1} + \lambda A_1X^{(k)}Q_1{P_1}^{-1} + (\lambda+1)N_1X^{(k)}), \\ Y^{(k+1)} = (\lambda M_2)^{-1}(\lambda E_2P_2^{-1} - N_2Y^{(k+1)} - \lambda C_2|X^{(k+1)}|D_2{P_2}^{-1} + \lambda A_2Y^{(k)}Q_2{P_2}^{-1} + (\lambda+1)N_2Y^{(k)}), \end{cases} \tag{4.18}$$

where $M_1 = D_{A_1} - L_{A_1} + \Omega, M_2 = D_{A_2} - U_{A_2} + \Omega, N_1 = U_{A_1} + \Omega, N_2 = L_{A_2} + \Omega$.

Based on this, we get the following algorithm.

Algorithm 4.2. (*The Newton-base generalized Gauss-Seidel bimatrix splitting iteration method I (NGGSBSI I)*)

Step 1 *Given initial point* $X^{(0)}, Y^{(0)} \in \mathbb{R}^{m\times n}$ *and the parameter* $\varepsilon, \lambda > 0$. *Assume the split of the matrices* A_1, A_2, B_1, B_2 *are* $A_1 = D_{A_1} - L_{A_1} + \Omega - (U_{A_1} + \Omega), A_2 = D_{A_2} - U_{A_2} + \Omega - (L_{A_2} + \Omega), B_1 = P_1 - Q_1, B_2 = P_2 - Q_2$, *respectively. Here,* Ω *satisfies* $M_1 = D_{A_1} - L_{A_1} + \Omega,\ M_2 = D_{A_2} - U_{A_2} + \Omega$ *are invertible and* $N_1 = U_{A_1} + \Omega,\ N_2 = L_{A_2} + \Omega$.

Step 2 *If* $\sqrt{\|A_1X^{(k)}B_1+C_1|Y^{(k)}|D_1-E_1\|^2+\|A_2Y^{(k)}B_2+C_2|X^{(k)}|D_2-E_2\|^2}/\sqrt{\|E_1\|^2+\|E_2\|^2}<\varepsilon$, *stop.*

Step 3 *Compute* $X^{(k+1)}$ *and* $Y^{(k+1)}$ *by*

$$\begin{cases} X^{(k+1)}=(\lambda M_1)^{-1}(\lambda E_1P_1^{-1}-N_1X^{(k+1)}-\lambda C_1|Y^{(k)}|D_1P_1{}^{-1}+\lambda A_1X^{(k)}Q_1P_1{}^{-1}+(\lambda+1)N_1X^{(k)}),\\ Y^{(k+1)}=(\lambda M_2)^{-1}(\lambda E_2P_2^{-1}-N_2Y^{(k+1)}-\lambda C_2|X^{(k+1)}|D_2P_2{}^{-1}+\lambda A_2Y^{(k)}Q_2P_2{}^{-1}+(\lambda+1)N_2Y^{(k)}). \end{cases} \tag{4.19}$$

Step 4 *Set* $k:=k+1$ *and go to Step 2.*

Taking a different split for A_1, i.e.,

$$A_1=D_{A_1}-U_{A_1}+\Omega-(L_{A_1}+\Omega), \tag{4.20}$$

the new iteration method is obtained.

Algorithm 4.3. (*The Newton-base generalized Gauss-Seidel bimatrix splitting iteration method II (NGGSBSI II)*)

Step 1 *Given initial point* $X^{(0)},Y^{(0)}\in\mathbb{R}^{n\times p}$ *and the parameter* $\varepsilon,\lambda>0$. *Assume the split of the matrices* A_1,A_2,B_1,B_2 *are* $A_1=D_{A_1}-U_{A_1}+\Omega-(L_{A_1}+\Omega), A_2=D_{A_2}-U_{A_2}+\Omega-(L_{A_1}+\Omega), B_1=P_1-Q_1, B_2=P_2-Q_2$, *respectively. Here,* Ω *satisfies* $M_1=D_{A_1}-U_{A_1}+\Omega, M_2=D_{A_2}-U_{A_2}+\Omega$ *are invertible and* $N_1=L_{A_1}+\Omega, N_2=L_{A_2}+\Omega$.

Step 2 *If* $\sqrt{\|A_1X^{(k)}B_1+C_1|Y^{(k)}|D_1-E_1\|^2+\|A_2Y^{(k)}B_2+C_2|X^{(k)}|D_2-E_2\|^2}/\sqrt{\|E_1\|^2+\|E_2\|^2}<\varepsilon$, *stop.*

Step 3 *Compute* $X^{(k+1)}$ *and* $Y^{(k+1)}$ *by*

$$\begin{cases} X^{(k+1)}=(\lambda M_1)^{-1}(\lambda E_1P_1^{-1}-N_1X^{(k+1)}-\lambda C_1|Y^{(k)}|D_1P_1{}^{-1}+\lambda A_1X^{(k)}Q_1P_1{}^{-1}+(\lambda+1)N_1X^{(k)}),\\ Y^{(k+1)}=(\lambda M_2)^{-1}(\lambda E_2P_2^{-1}-N_2Y^{(k+1)}-\lambda C_2|X^{(k+1)}|D_2P_2{}^{-1}+\lambda A_2Y^{(k)}Q_2P_2{}^{-1}+(\lambda+1)N_2Y^{(k)}). \end{cases} \tag{4.21}$$

Step 4 *Set* $k:=k+1$ *and go to Step 2.*

Theorem 4.2. *Let* $A_1,A_2,C_1,C_2\in\mathbb{R}^{m\times m}, B_1,B_2,D_1,D_2\in\mathbb{R}^{n\times n}, E_1,E_2\in\mathbb{R}^{m\times n}$. *Denote* $\|(\lambda(D_{A_1}-L_{A_1}+\Omega))^{-1}\|\|U_{A_1}+\Omega\|=\phi_1, \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|L_{A_2}+\Omega\|=\phi_2, \|(D_{A_1}-L_{A_1}+\Omega)^{-1}\|\|C_1\|\|D_1P_1{}^{-1}\|=\psi_1, \|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|C_2\|\|D_2P_2{}^{-1}\|=\psi_2, \|(D_{A_1}-L_{A_1}+\Omega)^{-1}\|\|A_1\|\|Q_1P_1{}^{-1}\|=\alpha_1, \|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|A_2\|\|Q_2P_2{}^{-1}\|=\alpha_2, \|(\lambda(D_{A_1}-L_{A_1}+\Omega))^{-1}\|\|(\lambda+1)(U_{A_1}+\Omega)\|=\beta_1, \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|(\lambda+1)(L_{A_2}+\Omega)\|=\beta_2$. *If* $\frac{\alpha_1+\beta_1}{1-\phi_1}+\frac{\psi_1}{1-\phi_1}\frac{\psi_2}{1-\phi_2}+\frac{\alpha_2+\beta_2}{1-\phi_2}<1$, *then the Algorithm 4.2 is convergent.*

Proof. Similar to the proof of theorem 4.1, it can be seen that

$$\begin{aligned} \|e_{k+1}^X\| &= \|(\lambda(D_{A_1}-L_{A_1}+\Omega))^{-1}(-(U_{A_1}+\Omega)(X^{(k+1)}-X^*)-\lambda C_1(|Y^{(k)}|-|Y^*|)D_1P_1{}^{-1}\\ &\quad+\lambda A_1(X^{(k)}-X^*)Q_1P_1{}^{-1}+(\lambda+1)(U_{A_1}+\Omega)(X^{(k)}-X^*))\|\\ &\leqslant \|(\lambda(D_{A_1}-L_{A_1}+\Omega))^{-1}\|\|U_{A_1}+\Omega\|\|e_{k+1}^X\|+\|(D_{A_1}-L_{A_1}+\Omega)^{-1}\|\|C_1\|\|D_1P_1{}^{-1}\|\|e_k^Y\|\\ &\quad+\|(D_{A_1}-L_{A_1}+\Omega)^{-1}\|\|A_1\|\|Q_1P_1{}^{-1}\|\|e_k^X\|\\ &\quad+\|(\lambda(D_{A_1}-L_{A_1}+\Omega))^{-1}\|\|(\lambda+1)(U_{A_1}+\Omega)\|\|e_k^X\|\\ &= \phi_1\|e_{k+1}^X\|+(\alpha_1+\beta_1)\|e_k^X\|+\psi_1\|e_k^Y\|, \end{aligned} \tag{4.22}$$

$$
\begin{aligned}
\|e^Y_{k+1}\| &= \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}(-(L_{A_2}+\Omega)(Y^{(k+1)}-Y^*)-\lambda C_2(|X^{(k+1)}|-|X^*|)D_2{P_2}^{-1} \\
&\quad +\lambda A_2(Y^{(k)}-Y^*)Q_2{P_2}^{-1}+(\lambda+1)(L_{A_2}+\Omega)(Y^{(k)}-Y^*))\| \\
&\leqslant \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|L_{A_2}+\Omega\|\|e^Y_{k+1}\|+\|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|C_2\|\|D_2{P_2}^{-1}\|\|e^X_{k+1}\| \\
&\quad +\|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|A_2\|\|Q_2{P_2}^{-1}\|\|e^Y_k\| \\
&\quad +\|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|(\lambda+1)(L_{A_2}+\Omega)\|\|e^Y_k\| \\
&= \phi_2\|e^Y_{k+1}\|+(\alpha_2+\beta_2)\|e^Y_k\|+\psi_2\|e^X_{k+1}\|,
\end{aligned} \tag{4.23}
$$

Further,

$$
\begin{aligned}
\begin{pmatrix} \|e^X_{k+1}\| \\ \|e^Y_{k+1}\| \end{pmatrix} &\leqslant \begin{pmatrix} \frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_1}{1-\phi_1} \\ \frac{\psi_2}{1-\phi_2}\frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_2}{1-\phi_2}\frac{\psi_1}{1-\phi_1}+\frac{\alpha_2+\beta_2}{1-\phi_2} \end{pmatrix} \begin{pmatrix} \|e^X_k\| \\ \|e^Y_k\| \end{pmatrix} \\
&\leqslant \begin{pmatrix} \frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_1}{1-\phi_1} \\ \frac{\psi_2}{1-\phi_2}\frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_2}{1-\phi_2}\frac{\psi_1}{1-\phi_1}+\frac{\alpha_2+\beta_2}{1-\phi_2} \end{pmatrix}^2 \begin{pmatrix} \|e^X_{k-1}\| \\ \|e^Y_{k-1}\| \end{pmatrix} \\
&\ \vdots \\
&\leqslant \begin{pmatrix} \frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_1}{1-\phi_1} \\ \frac{\psi_2}{1-\phi_2}\frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_2}{1-\phi_2}\frac{\psi_1}{1-\phi_1}+\frac{\alpha_2+\beta_2}{1-\phi_2} \end{pmatrix}^{k+1} \begin{pmatrix} \|e^X_0\| \\ \|e^Y_0\| \end{pmatrix}.
\end{aligned} \tag{4.24}
$$

Let $W = \begin{pmatrix} \frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_1}{1-\phi_1} \\ \frac{\psi_2}{1-\phi_2}\frac{\alpha_1+\beta_1}{1-\phi_1} & \frac{\psi_2}{1-\phi_2}\frac{\psi_1}{1-\phi_1}+\frac{\alpha_2+\beta_2}{1-\phi_2} \end{pmatrix}$, we know that when $\rho(W)<1$, $\lim_{k\to\infty} W^k=0$. It is shown that $\lim_{k\to\infty}\|e^X_k\|=0$, $\lim_{k\to\infty}\|e^Y_k\|=0$. In other words, the Algorithm 4.2 converges to the unique solution-pair (X^*,Y^*).

According to the proof of Theorem 4.1, we know that when $\frac{\alpha_1+\beta_1}{1-\phi_1}+\frac{\psi_1}{1-\phi_1}\frac{\psi_2}{1-\phi_2}+\frac{\alpha_2+\beta_2}{1-\phi_2}<1$, $\rho(W)<1$. This completes the proof. $\square$

Theorem 4.3. *Let $A_1,A_2,C_1,C_2\in\mathbb{R}^{m\times m}, B_1,B_2,D_1,D_2\in\mathbb{R}^{n\times n}, E_1,E_2\in\mathbb{R}^{m\times n}$. Denote* $\|(\lambda(D_{A_1}-U_{A_1}+\Omega))^{-1}\|\|L_{A_1}+\Omega\|=\phi_1, \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|L_{A_2}+\Omega\|=\phi_2, \|(D_{A_1}-U_{A_1}+\Omega)^{-1}\|\|C_1\|\|D_1{P_1}^{-1}\|=\psi_1, \|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|C_2\|\|D_2{P_2}^{-1}\|=\psi_2, \|(D_{A_1}-U_{A_1}+\Omega)^{-1}\|\|A_1\|\|Q_1{P_1}^{-1}\|=\alpha_1, \|(D_{A_2}-U_{A_2}+\Omega)^{-1}\|\|A_2\|\|Q_2{P_2}^{-1}\|=\alpha_2, \|(\lambda(D_{A_1}-U_{A_1}+\Omega))^{-1}\|\|(\lambda+1)(L_{A_1}+\Omega)\|=\beta_1, \|(\lambda(D_{A_2}-U_{A_2}+\Omega))^{-1}\|\|(\lambda+1)(L_{A_2}+\Omega)\|=\beta_2$. *If* $\frac{\alpha_1+\beta_1}{1-\phi_1}+\frac{\psi_1}{1-\phi_1}\frac{\psi_2}{1-\phi_2}+\frac{\alpha_2+\beta_2}{1-\phi_2}<1$, *then the Algorithm 4.2 is convergent.*

4.3 The inexact relaxed generalized Newton bimatrix splitting method for solving the coupled Sylvester-like absolute value equation

In order to overcome the problem that the above two methods cannot solve, we propose an inexact method for solving the coupled Sylvester-like absolute value equation based on the equivalence of (1.1) and (3.2).

Algorithm 4.4. (*The inexact relaxed generalized Newton bimatrix splitting method*)

Step 1 *Given initial point $X^{(0)},Y^{(0)}\in\mathbb{R}^{m\times n}$ and the parameter $\varepsilon>0, 0\leqslant\theta<1$. Assume the split of the matrices A_1,A_2,B_1,B_2 are $A_1=m_1-n_1, A_2=m_2-n_2, B_1=P_1-Q_1, B_2=$*

$P_2 - Q_2$, respectively. Given $\Omega_1, \Omega_2 \in \mathbb{R}^{m\times m}$ which satisfy $M_1 = m_1 + \Omega_1, M_2 = m_2 + \Omega_2$ are invertible and $N_1 = n_1 + \Omega_1, N_2 = n_2 + \Omega_2$. $k = 0$.

Step 2 *If $\sqrt{\|A_1X^{(k)}B_1 + C_1|Y^{(k)}|D_1 - E_1\|^2 + \|A_2Y^{(k)}B_2 + C_2|X^{(k)}|D_2 - E_2\|^2}/\sqrt{\|E_1\|^2 + \|E_2\|^2} < \varepsilon$, stop;*

Step 3.1 *Set $i = 0, z^{(k)} = (vec(X^{(k)})^T, vec(Y^{(k)})^T)^T$. Given $X_k^{(0)}, Y_k^{(0)}$.*

Step 3.2 *Compute $X_k^{(i+1)}$ and $Y_k^{(i+1)}$ by*

$$\begin{cases} M_1X_k^{(i+1)} = (E_1 - C_1|Y_k^{(i)}|D_1 + N_1X^{(k)}B_1 + M_1X^{(k)}Q_1)P_1^{-1}, \\ M_2Y_k^{(i+1)} = (E_2 - C_2|X_k^{(i+1)}|D_2 + N_2Y^{(k)}B_2 + M_2Y^{(k)}Q_2)P_2^{-1}. \end{cases} \tag{4.25}$$

Step 3.3 *If $\|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))z^{(k+1)} - ((\mathcal{N}+\Omega)z^{(k)} + \mathcal{E})\| \leqslant \theta\|\mathcal{F}(z^{(k)})\|$, then $X^{(k+1)} = X_k^{(i+1)}, Y^{(k+1)} = Y_k^{(i+1)}$, $k := k+1$, go to Step 2. Here, $\mathcal{A} = (\mathcal{M}+\Omega) - (\mathcal{N}+\Omega)$, $\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}$ are given by (4.1) and $z^{(k+1)} = (vec(X_k^{(i+1)})^T, vec(Y_k^{(i+1)})^T)^T, D(z^{(k)}) = diag(sign(z^{(k)}))$.*

Step 4 *Set $i := i+1$ and go to Step 3.2.*

Theorem 4.4. *Suppose that $\mathcal{M}+\Omega+\mathcal{B}D(z)$ is invertible for each $z \in \mathbb{R}^{mn}$. Let $0 \leqslant \theta < 1$. Let z^* is the solution of (4.1), then for $z^{(k+1)} \in \mathbb{R}^{mn}$ generated by Algorithm 4.4 satisfying*

$$\begin{aligned} \|z^{(k+1)} - z^*\| \leqslant & \|(\mathcal{M}+\Omega+\mathcal{B}D(z^k))^{-1}\|(\theta(\|\mathcal{M}+\Omega+\mathcal{B}D(z^k)\| \\ & + 2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|) + 2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|)\|z^{(k)} - z^*\|. \end{aligned} \tag{4.26}$$

Then when

$$\|(\mathcal{M}+\Omega)^{-1}\| < \frac{1}{\theta\|\mathcal{M}+\Omega\| + (\theta+1)(3\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|)}, \tag{4.27}$$

Algorithm 4.4 is convergent.

Proof. In the light of $\mathcal{F}(z^*) = 0$ and the fact that

$$(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))z^{(k)} = \mathcal{F}(z^{(k)}) + (\mathcal{N}+\Omega)z^{(k)} + \mathcal{E}, \tag{4.28}$$

we obtain

$$\begin{aligned} z^{(k+1)} - z^* &= z^{(k+1)} - z^{(k)} + z^{(k)} - z^* \\ &= z^{(k+1)} - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}(\mathcal{F}(z^{(k)}) + (\mathcal{N}+\Omega)z^{(k)} + \mathcal{E}) + z^{(k)} - z^* \\ &= (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}((\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))z^{(k+1)} - ((\mathcal{N}+\Omega)z^{(k)} + \mathcal{E}) \\ &\quad + \mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) + (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^{(k)} - z^*)). \end{aligned} \tag{4.29}$$

Taking norms on both sides and utilizing the triangle inequality, one can obtain

$$\begin{aligned} \|z^{(k+1)} - z^*\| &\leqslant \|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}\|(\|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))z^{(k+1)} - ((\mathcal{N}+\Omega)z^{(k)} + \mathcal{E})\| \\ &\quad + \|\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) + (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^{(k)} - z^*)\|) \\ &\leqslant \|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}\|(\theta\|\mathcal{F}(z^{(k)})\| \\ &\quad + \|\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) + (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^{(k)} - z^*)\|). \end{aligned} \tag{4.30}$$

On the other hand,

$$\mathcal{F}(z^{(k)}) = (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^{(k)}-z^*) - (\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)})). \tag{4.31}$$

Similarly, if the norm is taken for both sides of the above equation, then

$$\begin{aligned} \|\mathcal{F}(z^{(k)})\| &= \|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^{(k)}-z^*) - (\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)}))\| \\ &\leqslant \|\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)})\|\|z^{(k)}-z^*\| + \|\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)})\|. \end{aligned} \tag{4.32}$$

Furthermore, by some calculations, it holds that

$$\begin{aligned} &\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)}) \\ &= (\mathcal{A}z^* + \mathcal{B}|z^*| - \mathcal{E}) - (\mathcal{A}z^{(k)} + \mathcal{B}|z^{(k)}| - \mathcal{E}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)}) \\ &= \mathcal{B}(|z^*| - |z^{(k)}|) - (\mathcal{N}+\Omega)(z^*-z^{(k)}) - \mathcal{B}D(z^{(k)})(z^*-z^{(k)}). \end{aligned} \tag{4.33}$$

Then

$$\|\mathcal{F}(z^*) - \mathcal{F}(z^{(k)}) - (\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))(z^*-z^{(k)})\| \leqslant (2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|)\|z^*-z^{(k)}\|. \tag{4.34}$$

Combining (4.32) and (4.34), we get

$$\|\mathcal{F}(z^{(k)})\| \leqslant (\|\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)})\| + 2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|)\|z^*-z^{(k)}\|. \tag{4.35}$$

Substitute (4.34) and (4.35) into (4.30), one obtains

$$\begin{aligned} \|z^{(k+1)}-z^*\| \leqslant \|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}\|(\theta(\|\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)})\| + 2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|) \\ + 2\|\mathcal{B}\| + \|\mathcal{N}+\Omega\|)\|z^{(k)}-z^*\|. \end{aligned} \tag{4.36}$$

It's clear that

$$(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1} = (I + (\mathcal{M}+\Omega)^{-1}\mathcal{B}D(z^{(k)}))^{-1}(\mathcal{M}+\Omega)^{-1}.$$

And $\|(\mathcal{M}+\Omega)^{-1}\mathcal{B}D(z^{(k)})\| \leqslant \|(\mathcal{M}+\Omega)^{-1}\|\|\mathcal{B}\| < 1$, based on the Banach perturbation, we have

$$\begin{aligned} \|(I + (\mathcal{M}+\Omega)^{-1}\mathcal{B}D(z^{(k)}))^{-1}\| &\leqslant \frac{1}{1 - \|(\mathcal{M}+\Omega)^{-1}\mathcal{B}D(z^{(k)})\|} \\ &\leqslant \frac{1}{1 - \|(\mathcal{M}+\Omega)^{-1}\|\|\mathcal{B}\|}. \end{aligned} \tag{4.37}$$

Then

$$\|(\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)}))^{-1}\| \leqslant \|(I + (\mathcal{M}+\Omega)^{-1}\mathcal{B}D(z^{(k)}))^{-1}\|\|(\mathcal{M}+\Omega)^{-1}\| \leqslant \frac{\|(\mathcal{M}+\Omega)^{-1}\|}{1 - \|(\mathcal{M}+\Omega)^{-1}\|\|\mathcal{B}\|}. \tag{4.38}$$

Substitute (4.38) into (4.36), there is

$$\begin{aligned}\|z^{(k+1)}-z^*\| &\leqslant \frac{\|(\mathcal{M}+\Omega)^{-1}\|}{1-\|(\mathcal{M}+\Omega)^{-1}\|\|\mathcal{B}\|}(\theta(\|\mathcal{M}+\Omega+\mathcal{B}D(z^{(k)})\|+2\|\mathcal{B}\|+\|\mathcal{N}+\Omega\|)\\ &\quad +2\|\mathcal{B}\|+\|\mathcal{N}+\Omega\|)\|z^{(k)}-z^*\|\\ &\leqslant \frac{\|(\mathcal{M}+\Omega)^{-1}\|}{1-\|(\mathcal{M}+\Omega)^{-1}\|\|\mathcal{B}\|}(\theta(\|\mathcal{M}+\Omega\|+3\|\mathcal{B}\|+\|\mathcal{N}+\Omega\|)\\ &\quad +2\|\mathcal{B}\|+\|\mathcal{N}+\Omega\|)\|z^{(k)}-z^*\|\end{aligned} \tag{4.39}$$

Thus, according to assumption, we have $\|z^{(k+1)}-z^*\| < \|z^{(k)}-z^*\|$. This completes the proof. □

5 Numerical results

We give the following three examples to verify the conclusions obtained in this paper and test the algorithms. And we intuitively analyze the effect of the algorithm from the iteration count (indicated as IT), the relative residual error (indicated as RES) and the elapsed CPU time (indicated as CPU) where RES is defined as

$$\mathrm{RES}=\frac{\|\mathcal{F}(z)\|}{\|\mathcal{E}\|}=\frac{\sqrt{\|A_1X^{(k)}B_1+C_1|Y^{(k)}|D_1-E_1\|^2+\|A_2Y^{(k)}B_2+C_2|X^{(k)}|D_2-E_2\|^2}}{\sqrt{\|E_1\|^2+\|E_2\|^2}},$$

where $\mathcal{F},\mathcal{E},z$ are given by (4.1). While RES $<10^{-6}$ or the prescribed iteration count $k_{max}=1000$ is surpassed, all iterations are terminated. The programming language used was MATLAB R2018a.

Example 5.1. In order to be more intuitive, we first consider a small problem for (1.1), though our conclusions can be used for much larger problems in practice. Let

$$A_1=\begin{bmatrix}2&-4&0\\0&2&2\\-2&1&0\end{bmatrix},\quad B_1=\begin{bmatrix}-5&2&9\\7&3&3\\-6&-12&0\end{bmatrix},\quad C_1=\begin{bmatrix}1&-1&0\\0&1&1\\-1&0&0\end{bmatrix},\quad D_1=\begin{bmatrix}1&-2&0\\0&1&1\\-1&1&0\end{bmatrix},$$

$$A_2=\begin{bmatrix}2&4&0\\0&2&2\\2&1&0\end{bmatrix},\quad B_2=\begin{bmatrix}-5&2&9\\7&3&2\\-6&6&0\end{bmatrix},\quad C_2=\begin{bmatrix}1&1&0\\0&1&-1\\1&0&0\end{bmatrix},\quad D_2=\begin{bmatrix}1&2&0\\0&1&-1\\1&1&0\end{bmatrix},$$

and

$$E_1=\begin{bmatrix}-154&56&-23\\30&-72&145\\101&-106&-68\end{bmatrix},\quad E_2=\begin{bmatrix}33&75&29\\-40&38&19\\28&40&27\end{bmatrix}.$$

By the simple computations, we have

$$\max\{\sigma_{max}(D_1)\sigma_{max}(C_1),\sigma_{max}(D_2)\sigma_{max}(C_2)\}=5.01<6.20=\min\{\sigma_{min}(B_1)\sigma_{min}(A_1),\sigma_{min}(B_2)\sigma_{min}(A_2)\}$$

and

$$\begin{cases}\sigma_{max}(D_1{B_1}^{-1})\sigma_{max}({A_1}^{-1}C_1)=0.11<1\\ \sigma_{max}(D_2{B_2}^{-1})\sigma_{max}({A_2}^{-1}C_2)=0.26<1\end{cases}$$

which satisfy part (i) and part (ii) of Theorem 3.1, respectively. Indeed, the unique solution is as follows:

$$X = \begin{bmatrix} 5 & 1 & 4 \\ 2 & 4 & 0 \\ 5 & -1 & -5 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 2 \end{bmatrix}.$$

In this example, with the help of the Theorem 3.1, we do not need to calculate 9×9 matrices $\mathcal{A}, \mathcal{B}$ and $\mathcal{A}^{-1}\mathcal{B}$, but only need to calculate some 3×3 matrices to judge the solution of the equation. Of course, it was not difficult to form $\mathcal{A}$ and $\mathcal{B}$ explicitly in the above example, but this cannot be done if the matrices are very large. For instance, if the size of double precision input matrix is about 200, then just storing S_1 may require more than 12 GB of memory! Then calculating $\mathcal{A}$ requires more memory.

Then, we give the following experiment to compare the convergence effect of the Newton-base bimatrix splitting iteration method and the Newton-base generalized Gauss-Seidel bimatrix splitting iteration method.

Initially, Algorithm 4.1 produces different iterative forms for different ways of splitting.

(1)When $M_1 = A_1, M_2 = A_2, N_1 = \mathbf{0}_m, N_2 = \mathbf{0}_m, P_1 = B_1, P_2 = B_2, Q_1 = \mathbf{0}_n, Q_2 = \mathbf{0}_n$, the algorithm is simply iterative method without matrix splitting, i.e.,

$$\begin{cases} X^{(k+1)} & = {A_1}^{-1}(E_1 - C_1|Y^{(k)}|D_1){B_1}^{-1}, \\ Y^{(k+1)} & = {A_2}^{-1}(E_2 - C_2|X^{(k+1)}|D_2){B_2}^{-1}, \end{cases} \tag{5.1}$$

which can be called a simply iterative method (SI).

(2)When $M_1 = D_{A_1} - L_{A_1} + \Omega_1, M_2 = D_{A_2} - U_{A_2} + \Omega_2, N_1 = U_{A_2} + \Omega_1, N_2 = L_{A_2} + \Omega_2, P_1 = D_{B_1} - L_{B_1}, Q_1 = U_{B_1}, P_2 = D_{B_2} - U_{B_2}, Q_2 = L_{B_2}$, where $D_{A_i} = \text{diag}(A_i), D_{B_i} = \text{diag}(B_i)$, $-L_{A_i}, -U_{A_i}, -L_{B_i}, -U_{B_i}$ represent the strictly lower-triangular and upper-triangular part of A_i and $B_i, i = 1, 2$, respectively, Algorithm 4.1 will be expressed as

$$\begin{cases} X^{(k+1)} & = (D_{A_1} - L_{A_1} + \Omega_1)^{-1}(E_1 - C_1|Y^{(k)}|D_1 + (U_{A_2} + \Omega_1)X^{(k)}B_1 \\ & \quad + (D_{A_1} - L_{A_1} + \Omega_1)X^{(k)}U_{B_1})(D_{B_1} - L_{B_1})^{-1}, \\ Y^{(k+1)} & = (D_{A_2} - U_{A_2} + \Omega_2)^{-1}(E_2 - C_2|X^{(k+1)}|D_2 + (L_{A_2} + \Omega_2)Y^{(k)}B_2 \\ & \quad + (D_{A_2} - U_{A_2} + \Omega_2)Y^{(k)}L_{B_2})(D_{B_1} - U_{B_1})^{-1}, \end{cases} \tag{5.2}$$

which can be called a Newton-base Gauss-Seidel bimatrix splitting iteration method I (NGS-BSI I).

(3)When $M_1 = D_{A_1} - U_{A_1} + \Omega_1, M_2 = D_{A_2} - U_{A_2} + \Omega_2, N_1 = L_{A_1} + \Omega_1, N_2 = L_{A_2} + \Omega_2, P_1 = D_{B_1} - L_{B_1}, Q_1 = U_{B_1}, P_2 = D_{B_2} - U_{B_2}, Q_2 = L_{B_2}$, where $D_{A_i} = \text{diag}(A_i), D_{B_i} = \text{diag}(B_i)$, $-L_{A_i}, -U_{A_i}, -L_{B_i}, -U_{B_i}$ represent the strictly lower-triangular and upper-triangular part of $A_i$and$B_i, i = 1, 2$, respectively, Algorithm 4.1 will be expressed as

$$\begin{cases} X^{(k+1)} & = (D_{A_1} - U_{A_1} + \Omega_1)^{-1}(E_1 - C_1|Y^{(k)}|D_1 + (L_{A_1} + \Omega_1)X^{(k)}B_1 \\ & \quad + (D_{A_1} - U_{A_1} + \Omega_1)X^{(k)}U_{B_1})(D_{B_1} - L_{B_1})^{-1}, \\ Y^{(k+1)} & = (D_{A_2} - U_{A_2} + \Omega_2)^{-1}(E_2 - C_2|X^{(k+1)}|D_2 + (L_{A_2} + \Omega_2)Y^{(k)}B_2 \\ & \quad + (D_{A_2} - U_{A_2} + \Omega_2)Y^{(k)}L_{B_2})(D_{B_1} - U_{B_1})^{-1}, \end{cases} \tag{5.3}$$

which can be called a Newton-base Gauss-Seidel bimatrix splitting iteration method II (NGS-BSI II).

(4)When $M_1 = \frac{1}{2}(A_1 + A_1{}^T) + \Omega_1, M_2 = \frac{1}{2}(A_2 + A_2{}^T) + \Omega_2, N_1 = -\frac{1}{2}(A_1 - A_1{}^T) + \Omega_1, N_2 = -\frac{1}{2}(A_2 - A_2{}^T) + \Omega_2, P_1 = \frac{1}{2}(B_1 + B_1^T), Q_1 = -\frac{1}{2}(B_1 - B_1{}^T), P_2 = \frac{1}{2}(B_2 + B_2{}^T), Q_2 = -\frac{1}{2}(B_2 - B_2{}^T)$, Algorithm 4.1 will be expressed as

$$\begin{cases} X^{(k+1)} & = (\frac{1}{2}(A_1 + A_1{}^T) + \Omega_1)^{-1}(E_1 - C_1|Y^{(k)}|D_1 + (-\frac{1}{2}(A_1 - A_1{}^T) + \Omega_1)X^{(k)}B_1 \\ & \quad + (\frac{1}{2}(A_1 + A_1{}^T) + \Omega_1)X^{(k)}(-\frac{1}{2}(B_1 - B_1{}^T)))(\frac{1}{2}(B_1 + B_1^T))^{-1}, \\ Y^{(k+1)} & = (\frac{1}{2}(A_2 + A_2{}^T) + \Omega_2)^{-1}(E_2 - C_2|X^{(k+1)}|D_2 + (-\frac{1}{2}(A_1 - A_1{}^T) + \Omega_2)Y^{(k)}B_2 \\ & \quad + (\frac{1}{2}(A_2 + A_2{}^T) + \Omega_2)Y^{(k)}(-\frac{1}{2}(B_2 - B_2{}^T)))(\frac{1}{2}(B_2 + B_2{}^T))^{-1}, \end{cases} \tag{5.4}$$

which can be called a Newton-base Hermitian and Skew-Hermitian bimatrix splitting iteration method (NHSBSI).

Example 5.2. Let

$$S = \begin{pmatrix} 6 & 1 & 0 & 0 & 0 \\ 0 & 6 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 6 & 1 \\ 0 & 0 & 0 & 0 & 6 \end{pmatrix} \in \mathbb{R}^{m\times m}, M_1 = \begin{pmatrix} S & I_0 & 0 & 0 & 0 \\ -I_0 & S & I_0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & -I_0 & S & I_0 \\ 0 & 0 & 0 & -I_0 & S \end{pmatrix} \in \mathbb{R}^{n\times n},$$

where $I_0 \in \mathbb{R}^{m\times m}$ is an identity matrix. Set $I \in \mathbb{R}^{n\times n}$ is an identity matrices with n dimensions, $n = m^2$. $M = M_1 + \mu I$.

Consider the coupled Sylvester-like absolute value equation

$$\begin{cases} A_1XB_1 + C_1|Y|D_1 = E_1, \\ A_2YB_2 + C_2|X|D_2 = E_2, \end{cases}$$

where $A_1 = M + I, B_1 = I, C_1 = 0.5I, D_1 = U_n(-0.5), A_2 = M - I, B_2 = 0.5I, C_2 = \mathbf{1}_n(0.1), D_2 = U_n(3)$, $E_1 = A_1X^*B_1 + C_1|Y^*|D_1, E_2 = A_2Y^*B_2 + C_2|Y^*|D_2$. Here, $X^* = \mathbf{1}_n(1.2), Y^* = \mathbf{1}_n(-0.8)$.

For Example 5.2, to improve the convergence speed of all the tested methods, the choice of $\Omega_1, \Omega_2, \Omega$ are $\Omega_1 = \Omega_2 = \Omega = M_1$. we take the parameter $\mu = 2$. The initial iteration points $X^{(0)}, Y^{(0)}$ are $X^{(0)} = Y^{(0)} = \mathbf{0}_n$.

According to the numerical results given in Table 1 and Figure 1, the SI method, the NGSBSI I method, the NGSBSI II method, the NHSBSI method, the NGGSBSI I method and the NGGSBSI II method can converge to the solution pair (X^*, Y^*) quickly for different problem sizes. Moreover, the performance of the NGSBSI II method and the NGGSBSI II method in Example 5.2 are relatively stable. It can be seen intuitively from Table 1 that CPU time of the NGSBSI II method are obviously better than the other methods in higher dimensions and the SI method is preformed well in lower dimensions.

Table 1. Numerical comparisons about the mentioned algorithms for Example 5.2

Algorithm	n	16	25	36	49	64
	IT	**8**	**13**	21	34	54
SI	RES	4.6088e-07	4.6371e-07	6.3739e-07	5.7206e-07	3.6369e-07
	CPU	**0.012429**	**0.014717**	0.023083	0.038384	0.071760
	IT	18	15	**15**	**16**	28
NGSBSI I	RES	4.5384e-07	7.2077e-07	7.7814e-07	4.3937e-07	8.3616e-07
	CPU	0.015742	0.018567	0.022242	0.030712	0.055818
	IT	19	18	17	18	**18**
NGSBSI II	RES	6.4256e-07	5.8163e-07	7.4857e-07	5.2393e-07	7.2404e-07
	CPU	0.014980	0.017993	**0.021840**	**0.027290**	**0.035258**
	IT	34	50	75	117	178
NHSBSI	RES	7.6282e-07	7.0507e-07	7.5756e-07	7.5040e-07	8.2014e-07
	CPU	0.015344	0.025279	0.044035	0.093088	0.192427
	IT	22	20	20	19	28
NGGSBSI I	RES	7.9421e-07	7.5295e-07	5.8553e-07	8.4232e-07	8.2923e-07
($\lambda = 5$)	CPU	0.019237	0.023603	0.027311	0.036835	0.060738
	IT	24	21	20	20	20
NGGSBSI II	RES	7.1782e-07	6.4080e-07	9.8325e-07	8.5439e-07	7.8258e-07
($\lambda = 5$)	CPU	0.018224	0.022863	0.026513	0.034405	0.046650

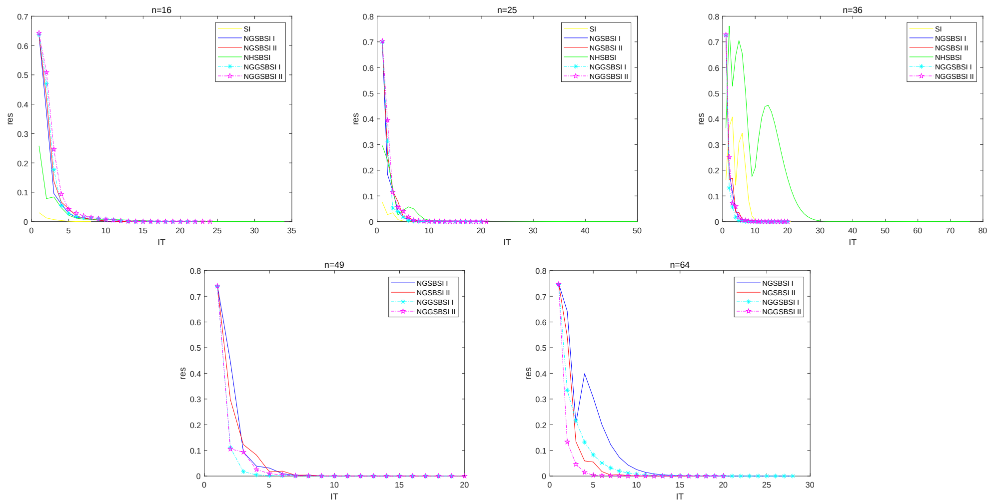

Figure 1. Convergence effect for Example 5.2. When n=49 and n=64, the iteration counts of the SI method and the NHSBSI method are so high that they are not reflected in the figure.

Last example shows a comparison between the inexact relaxed generalized Newton bimatrix splitting method presented in this paper and the IGN method in [26] for solving absolute value equations.

We know Algorithm 4.4 produces different iterative forms for different ways of splitting.

(1)When $\mathcal{M} = \mathcal{A}$, $\mathcal{N} = \mathbf{0}_{2mn}$, the Algorithm 4.4 can be called the IRGN method.

(2)When $\mathcal{M} = \mathcal{D} - \mathcal{L} + \Omega$, $\mathcal{N} = \mathcal{U} + \Omega$, which $\mathcal{D}$ is the diagonal of $\mathcal{A}$, $\mathcal{U}$ and $\mathcal{L}$ are strictly upper and lower triangular parts of $\mathcal{A}$, then the Algorithm 4.4 can be called the IRGNS method.

Example 5.3. Let

$$S = \begin{pmatrix} 20 & 1 & 0 & 0 & 0 \\ -1 & 20 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & -1 & 20 & 1 \\ 0 & 0 & 0 & -1 & 20 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

and $M = S + \mu I_0$, where $I_0 \in \mathbb{R}^{n\times n}$ is an identity matrix.

Consider the coupled Sylvester-like absolute value equation

$$\begin{cases} A_1XB_1 + C_1|Y|D_1 = E_1, \\ A_2YB_2 + C_2|X|D_2 = E_2, \end{cases}$$

where $A_1 = M + I_0, B_1 = I_0, C_1 = 0.5I_0, D_1 = U_n(-0.1), A_2 = \frac{M-I_0}{3}, B_2 = 0.5I_0, C_2 = \mathbf{1}_n(0.1), D_2 = U_n(0.3),\ E_1 = A_1X^*B_1 + C_1|Y^*|D_1, E_2 = A_2Y^*B_2 + C_2|Y^*|D_2$. Here, $X^* = \mathbf{1}_n(1.2), Y^* = \mathbf{1}_n(-0.8)$.

For Example 5.3, to improve the convergence speed of all the tested methods, the choice of Ω_1, Ω_2 are $\Omega_1 = \Omega_2 = 0.1I_0, \Omega = 2.8I$, which $I \in \mathbb{R}^{2n^2\times 2n^2}$. we take the parameter $\mu = 1.5$. The initial iteration points $X^{(0)}, Y^{(0)}$ are $X^{(0)} = Y^{(0)} = \mathbf{0}_n$.

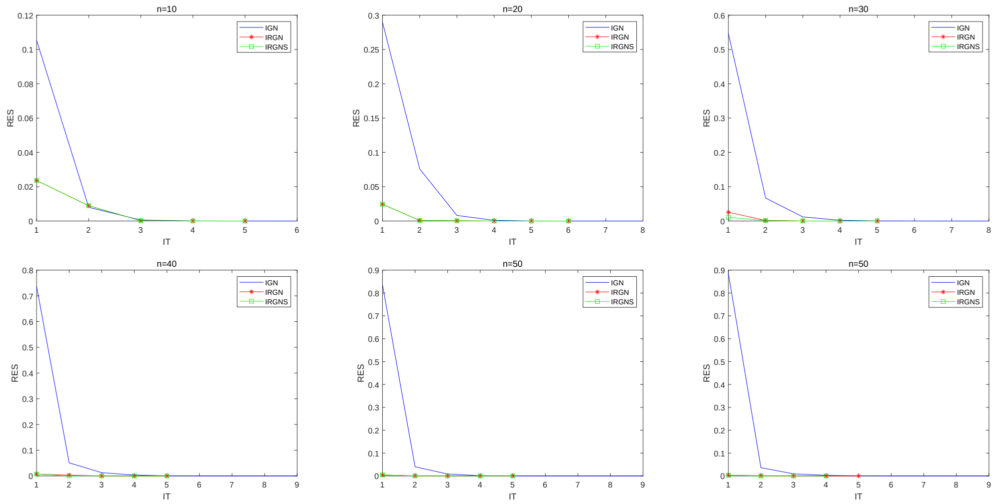

Figure 2. Convergence effect for Example 5.3

According to the numerical results given in Table 2 and Figure 2, the IGN method, the IRGN method and the IRGNS method can converge to the solution-pair (X^*, Y^*) quickly for different problem sizes. Moreover, the performance of the IGN method, the IRGN method and

Table 2. Numerical comparisons about the mentioned algorithms for Example 5.2

Algorithm	n	10	20	30	40	50	60	70
	IT	6	8	8	9	45	9	11
IGN	θ	0.6	0.6	0.6	0.6	0.6	0.6	0.6
	RES	9.2052e-07	2.0986e-07	6.6419e-07	3.4388e-07	2.0956e-07	9.3097e-07	7.2030e-07
	CPU	0.028328	0.441325	2.691583	12.977646	45.788947	116.424993	322.777565
	IT	5	6	5	5	5	5	5
IRGN	θ	0.6	0.6	0.6	0.6	0.6	0.6	0.6
	RES	4.2194e-07	5.1432e-08	6.1042e-07	2.1069e-07	7.3521e-08	5.7298e-07	2.3224e-07
	CPU	0.026210	0.197793	**1.205960**	**9.118140**	**28.179983**	**72.085520**	**170.321264**
	IT	**5**	**4**	**5**	**5**	**5**	**4**	**4**
IRGNS	θ	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	RES	4.2194e-07	2.3082e-07	6.0263e-07	2.1751e-07	8.5426e-08	7.8050e-07	5.7053e-07
	CPU	**0.023765**	**0.195116**	2.539173	12.601875	38.056623	92.537365	204.321974

the IRGNS method in Example 5.3 are relatively stable. However, when the dimension of the problem is relatively large, the CPU time of the three methods is relatively high. It can be seen intuitively from Table 2 that the iteration counts of the IRGNS method are less than the other two methods. The CPU time of the IRGN method performs slightly better than the other two methods in higher dimensions and the IRGNS method performs slightly better than the other two methods in lower dimensions.

6 Conclusions

In this paper, sufficient conditions for the existence of the unique solution of the coupled Sylvester-like absolute value equation (1.1) are given. Moreover, we discuss the sufficient condition that the solution does not exist. Numerical experiments confirm these conclusions. In addition, we propose the Newton-base bimatrix splitting iteration method, the Newton-base generalized Gauss-Seidel bimatrix splitting iteration method and the inexact relaxed generalized Newton bimatrix splitting method to solve the coupled Sylvester-like absolute value equation. These methods avoid the problem of converting the coupled Sylvester-like absolute value equation into the generalized absolute value equation, which leads to huge computation. Convergence properties of the new iteration schemes are analyzed in detail. Numerical experiments are reported to demonstrate the efficiency of these new iteration methods.

Declarations

Data availability statement Data will be made available on reasonable request.
Conflict of interest The author declares that they have no conflict of interest.

Ethical approval This manuscript does not contain any studies with human participants or animals performed by any of the authors.

References

[1] S.L. Wu, The unique solution of a class of the new generalized absolute value equation, Applied Mathematics Letters, 116 (2021) 107029.

[2] O.L. Mangasarian, R.R. Meyer, Absolute value equations, Linear Algebra and its Applications, 419 (2006) 359-367.

[3] B.H. Huang, C.F. Ma, Convergent conditions of the generalized Newton method for absolute value equation over second order cones, Applied Mathematics Letters, 92 (2019) 1-6.

[4] A. Mansoori, M. Erfanian, A dynamic model to solve the absolute value equations, Journal of Computational and Applied Mathematics, 333 (2018) 28-35.

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

[6] O.L. Mangasarian, Absolute value programming, Computational Optimization and Applications, 36 (2007) 43-53.

[7] O.L. Mangasarian, R.R. Meyer, Absolute value equations, Linear Algebra and Its Applications, 419 (2006) 359-367.

[8] R.W. Cottle, J.S. Pang, R.E. Stone, The Linear Complementarity Problem, Academic, San Diego, (1992).

[9] L. Abdallah, M. Haddou, T. Migot, Solving absolute value equation using complementarity and smoothing functions, Journal of Computational and Applied Mathematics, 327 (2018) 196-207.

[10] J. Rohn, Systems of linear interval equations, Linear Algebra and its Applications, 126 (1989) 39-78.

[11] F. Mezzadri, On the solution of general absolute value equations, Applied Mathematics Letters, 107 (2020) 7.

[12] O. Prokopyev, On equivalent reformulations for absolute value equations, Computational Optimization and Applications, 44 (2009) 363-372.

[13] J. Rohn, On unique solvability of the absolute value equation, Optimization Letters, 3 (2009) 603-606.

[14] J. Rohn, An algorithm for solving the absolute value equation, The Electronic Journal of Linear Algebra, 18 (2009) 589-599.

[15] S.L. Wu, C.X. Li, A note on unique solvability of the absolute value equation, Optimization Letters, 14 (2020) 1957-1960.

[16] S.L. Wu, S.Q. Shen, On the unique solution of the generalized absolute value equation, Optimization Letters, 6 (2020) 1-8.

[17] S.F. Xu, Matrix Calculation in Cybernetics, Higher Education Press, Beijing, (2011).

[18] B. Hashemi, Sufficient conditions for the solvability of a Sylvester-like absolute value matrix equation, Applied Mathematics Letters, 112 (2021) 3.

[19] J.Y. Tang, J.C. Zhou, A quadratically convergent descent method for the absolute value equation $Ax + B|x| = b$, Operations Research Letters, 47 (2019) 229-234.

[20] Y.F. Ke, The new iteration algorithm for absolute value equation, Applied Mathematics Letters, 99 (2020) 105990.

[21] C.F. Ma, Y.F. Ke, J. Tang, B.G.Chen, Numerical Linear Algebra and Algorithms, National Defence Industry Press, Beijing, (2017).

[22] Y.F. Ke, C.F. Ma, An alternating direction method for nonnegative solutions of the matrix equation $AX + YB = C$, Computational and Applied Mathematics, 36 (2017) 5.

[23] O.L. Mangasarian, A generalized Newton method for absolute value equations, Optimization Letters, 3 (2009) 101-108.

[24] A. Wang, Y. Cao, J.X. Chen, et al., Modified Newton-Type iteration methods for generalized absolute value equations, Journal of Optimization Theory and Applications, 181 (2019) 216-230.

[25] S.L. Wu, T.Z. Huang, X.L.Zhao, A modified SSOR iterative method for augmented systems, Computational and Applied Mathematics, 228 (2009) 424-433.

[26] J.Y.B. Cruz, O.P. Ferreira, L.F. Prudente, On the global convergence of the inexact semi-smooth Newton method for absolute value equation, Computational Optimization and Applications, 65 (2016) 93-108.