Linear Algebra and its Applications 439 (2013) 856–878

Contents lists available at SciVerse ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Fast enclosure for solutions of Sylvester equations

Shinya Miyajima

Faculty of Engineering, Gifu University, Gifu-shi 501-1193, Japan

ARTICLE INFO

Article history:
Received 3 November 2011
Accepted 1 July 2012
Available online 4 August 2012

Submitted by F. Dopico

AMS classification:
65F05
65G20
65G50

Keywords:
Numerical enclosure
Sylvester equation
Lyapunov equation

ABSTRACT

Fast algorithms for enclosing solutions of Sylvester equations $AX + XB = C$, $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $X, C \in \mathbb{C}^{m\times n}$ are proposed. The results obtained by these algorithms are "verified" in the sense that all the possible rounding errors have been taken into account. For developing these algorithms, theories which *directly* supply error bounds for numerical solutions are established. The proposed algorithms require only $\mathcal{O}(m^3 + n^3)$ operations if A and B are diagonalizable. Techniques for accelerating the enclosure and obtaining smaller error bounds are introduced. Numerical results show the properties of the proposed algorithms.

1. Introduction

In this paper, we are concerned with the accuracy of numerically computed solutions in the Sylvester equation

$$AX + XB = C, \tag{1}$$

where $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ and $C \in \mathbb{C}^{m\times n}$ are given. The Sylvester equation (1) appears in many important problems in science and technology, e.g., control theory [1], model reduction [2,3], the numerical solution of Riccati equations [4], image processing [5] and so forth. Moreover (1) includes as special cases several important linear equation problems: linear system, multiple right-hand side linear system, matrix inversion, and eigenvector x corresponding to given eigenvalue b: $(A-bI_m)x = 0$, where I_p denotes the $p \times p$ identity matrix, and commuting matrices: $AX - XA = 0$. Especially block-diagonalization of a block triangular matrix is equivalent to solving (1) (see e.g., [6]). The case $B = A^T$ is called the Lyapunov equation which arises in several applications in control theory, e.g., in stability and robust stability, model reduction, internal balancing and determining the H_2-norm [1].

E-mail address: miyajima@gifu-u.ac.jp

0024-3795/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.laa.2012.07.001

It is well known (e.g., [6]) that (1) can be written as the following linear system:

$$\mathsf{P}\mathrm{vec}(X) = \mathrm{vec}(C), \quad \mathsf{P} := I_n \otimes A + B^T \otimes I_m, \tag{2}$$

where $\otimes$ is the Kronecker product (see e.g., [7]) and vec is the operation which stacks the columns of a matrix in order to obtain one long vector. Therefore P is an $mn \times mn$ complex matrix, and $\mathrm{vec}(X)$ and $\mathrm{vec}(C)$ are complex mn-vectors.

Let $M \in \mathbb{C}^{p\times p}$ and $\lambda(M)$ be the spectrum of M. If $\lambda(A) = \{\lambda_1(A), \ldots, \lambda_m(A)\}$ and $\lambda(B) = \{\lambda_1(B), \ldots, \lambda_n(B)\}$, it holds (see [7]) that

$$\lambda(\mathsf{P}) = \{\lambda_i(A) + \lambda_j(B) : i = 1, \ldots, m,\ j = 1, \ldots, n\},$$

including algebraic multiplicities in all three cases. Therefore (1) has a unique solution X^* if and only if A and $-B$ have no eigenvalues in common. In this paper, it is assumed that this condition is always satisfied.

Standard direct methods for solving (1) have been proposed in [8,9]. These methods are based on the Schur decomposition, by which the original equation is transformed into a form that is easily solved by a forward or backward substitution.

In this paper, we consider numerically enclosing the exact solution X^* in (1). There are several algorithms for enclosing solutions in (1), e.g., [10–14]. Most of the algorithms in [10] involve floating point operations whose number is proportional to the exponential of m and n. Another algorithm in [10] requires $\mathcal{O}(m^3n^2 + m^2n^3)$ operations. The algorithms in [11–13] involve $\mathcal{O}(m^3n^3)$ operations. As opposed to these algorithms, the algorithm in [14] requires only $\mathcal{O}(m^3 + n^3)$ operations if A and B are diagonalizable. Namely this algorithm is pioneering work for enclosing X^* with cubic complexity. The algorithm in [14] is based on numerical diagonalization and the Krawczyk operator [15], and supplies an interval matrix containing X^* by empirically creating a candidate interval matrix and examining whether the created matrix includes X^* or not.

The purpose of this paper is to propose algorithms for enclosing X^* which *directly* supply error bounds $X^\varepsilon \in \mathbb{R}^{m\times n}$ satisfying $|\tilde{X} - X^*| \le X^\varepsilon$, where $\tilde{X}$ and $|\tilde{X} - X^*|$ denote a numerical solution in (1) and the matrix with elements $|(\tilde{X} - X^*)_{ij}|$, respectively, and inequalities between matrices hold componentwise. These algorithms also require only $\mathcal{O}(m^3+n^3)$ operations if A and B are diagonalizable. We present theories for *directly* obtaining X^ε to construct the proposed algorithms, and introduce techniques for accelerating the enclosure and obtaining smaller error bounds. The proposed algorithms allow the presence of underflow in floating point arithmetic.

This paper is organized as follows: In Section 2, theories for computing the upper bound for $|\tilde{X}-X^*|$ are established. In Sections 3 and 4, techniques for accelerating the enclosure and obtaining smaller error bounds are introduced, respectively. In Section 5, numerical results are reported to show the property of the proposed algorithms. Finally Section 6 summarizes the results in this paper and highlights possible extensions and future work.

2. Enclosure theories

In this section, we establish theories for enclosing solutions in (1). For $M \in \mathbb{C}^{m\times n}$, M_{ij}, $M_{i:}$ and $M_{:j}$ denote the (i, j) element, the i-th row and the j-th column of M, respectively, $|M| := \{|M_{ij}|\}$, $M^T := \{M_{ji}\}$, $\|M\|_\infty := \max_i \sum_j |M_{ij}|$, $\|M\|_1 := \max_j \sum_i |M_{ij}|$ and $\|M\|_{\mathrm{M}} := \max_{i,j} |M_{ij}|$. For a complex vector v, v_i denotes the i-th element of v. For $M, N \in \mathbb{R}^{m\times n}$, $\min(M, N) := \{\min(M_{ij}, N_{ij})\}$. For $d_1, \ldots, d_n \in \mathbb{C}$, $\mathrm{diag}(d_1, \ldots, d_n)$ denotes a diagonal matrix whose diagonal elements are $d_1, \ldots, d_n$. Let $\boldsymbol{u}$, $\underline{\boldsymbol{u}}$, I_p, $\otimes$, ./ and $\mathrm{fl}(\cdot)$ be unit roundoff, underflow unit (especially $\boldsymbol{u} = 2^{-53}$ and $\underline{\boldsymbol{u}} = 2^{-1074}$ in IEEE 754 double precision), the $p \times p$ identity matrix, the Kronecker product, pointwise division, a result of floating point operations, where all inside parenthesis are executed by ordinary floating point arithmetic fulfilling rounding-to-nearest mode, respectively, $\gamma_p := p\boldsymbol{u}/(1 - p\boldsymbol{u})$, $\zeta_p := \sqrt{2}\gamma_4 + (1 + \sqrt{2}\gamma_4)\gamma_p$, $s^{(p)} := (1, \ldots, 1)^T \in \mathbb{R}^p$ and $E := (s^{(m)}, \ldots, s^{(m)}) \in \mathbb{R}^{m\times n}$. For $M \in \mathbb{C}^{m\times n}$ and $v \in \mathbb{C}^{mn}$, we define

$$\mathrm{vec}(M) := \begin{pmatrix} M_{:1} \\ \vdots \\ M_{:n} \end{pmatrix} \quad \text{and} \quad \mathrm{mat}(v) := \begin{pmatrix} v_1 & v_{m+1} & \cdots & v_{m(n-1)+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_m & v_{2m} & \cdots & v_{mn} \end{pmatrix},$$

respectively. Then we have $\mathrm{mat}(\mathrm{vec}(M)) = M$ and $\mathrm{vec}(\mathrm{mat}(v)) = v$. Assume $v_i \neq 0$ for all i and let $\mathsf{V} := \mathrm{diag}(v_1, \dots, v_{mn})$. Then it holds that $\mathsf{V}^{-1}\mathrm{vec}(M) = \mathrm{vec}(M./\mathrm{mat}(v))$. Let $\mathbb{FR}$ and $\mathbb{FC}$ be sets of all floating point real and complex numbers, respectively.

We cite Lemmas 1 and 2, and present Lemma 3 which are used in the proof of Theorems 1, 2, 3 or 4 shown below.

Lemma 1 (E.g., Horn et al. [7]). *For any complex matrices K, L, M and N with compatible sizes, it holds that*

$$(K \otimes L)(M \otimes N) = (KM \otimes LN)$$
$$\mathrm{vec}(LMN) = (N^T \otimes L)\mathrm{vec}(M).$$

Lemma 2 (E.g., Meyer [16]). *For $S \in \mathbb{C}^{m\times m}$ and $1 \le p \le \infty$, if $\|S\|_p < 1$, $I_m - S$ is nonsingular.*

Lemma 3 is a modification of [17, Theorem 3] suited for estimating upper bounds for matrices rather than vectors.

Lemma 3. *Let $S, G \in \mathbb{C}^{m\times m}$, $F \in \mathbb{C}^{m\times n}$ and $D_F := \mathrm{diag}(\|F_{:1}\|_\infty, \dots, \|F_{:n}\|_\infty)$. If $\|S\|_\infty < 1$, it holds that*

$$|(I_m - S)^{-1}F| \le |F| + \frac{1}{1 - \|S\|_\infty}|S|ED_F$$
$$|(I_m - S)^{-1}G|E \le \left(|G| + \frac{\|G\|_\infty}{1 - \|S\|_\infty}|S|\right)E. \tag{3}$$

Remark 1. From $\|S\|_\infty < 1$ and Lemma 2, $I_m - S$ is nonsingular.

Proof. The Neumann series (e.g., [16, Chapter 3]) gives

$$\begin{aligned} |(I_m - S)^{-1}F| &\le |(I_m - S)^{-1}||F| = |I_m + S + S^2 + \cdots||F| \\ &\le (I_m + |S| + |S|^2 + \cdots)|F| = |F| + (|S| + |S|^2 + \cdots)(|F|_{:1}, \dots, |F|_{:n}). \end{aligned} \tag{4}$$

For $i = 1, \dots, n$, it holds from $\|S\|_\infty < 1$ that

$$\begin{aligned} (|S| + |S|^2 + \cdots)|F|_{:i} &= |S||F|_{:i} + |S|(|S||F|_{:i}) + \cdots \\ &\le \||F|_{:i}\|_\infty |S|s^{(m)} + \||S||F|_{:i}\|_\infty |S|s^{(m)} + \cdots \\ &\le \|F_{:i}\|_\infty |S|s^{(m)} + \|S\|_\infty \|F_{:i}\|_\infty |S|s^{(m)} + \cdots \\ &= \|F_{:i}\|_\infty(1 + \|S\|_\infty + \|S\|_\infty^2 + \cdots)|S|s^{(m)} = \frac{\|F_{:i}\|_\infty}{1 - \|S\|_\infty}|S|s^{(m)}. \end{aligned} \tag{5}$$

This and (4) yield

$$\begin{aligned} |(I_m - S)^{-1}F| &\le |F| + \frac{1}{1 - \|S\|_\infty}(\|F_{:1}\|_\infty |S|s^{(m)}, \dots, \|F_{:n}\|_\infty |S|s^{(m)}) \\ &= |F| + \frac{1}{1 - \|S\|_\infty}(|S|s^{(m)}, \dots, |S|s^{(m)})D_F = |F| + \frac{1}{1 - \|S\|_\infty}|S|ED_F. \end{aligned}$$

From derivations similar to (4) and (5), we obtain

$$\begin{aligned}|(I_m - S)^{-1}G|s^{(m)} &\le (I_m + |S| + |S|^2 + \cdots)|G|s^{(m)} \\ &\le |G|s^{(m)} + (|S| + |S|^2 + \cdots)|G|s^{(m)} \\ &\le |G|s^{(m)} + \frac{\||G|s^{(m)}\|_\infty}{1 - \|S\|_\infty}|S|s^{(m)} = \left(|G| + \frac{\|G\|_\infty}{1 - \|S\|_\infty}|S|\right)s^{(m)},\end{aligned}$$

which shows (3). □

2.1. Theory based on spectral decomposition

Assume that A and B are diagonalizable, i.e., there exists diagonal $D_A \in \mathbb{C}^{m\times m}$ and $D_B \in \mathbb{C}^{n\times n}$, and nonsingular $V_A \in \mathbb{C}^{m\times m}$ and $V_B \in \mathbb{C}^{n\times n}$ such that

$$A = V_A D_A V_A^{-1} \quad \text{and} \quad B^T = V_B D_B V_B^{-1}.$$

In this subsection, we formulate and prove Theorems 1 and 2 for enclosing solutions in (1) based on the spectral decomposition.

Theorem 1. *Let $\tilde X \in \mathbb{C}^{m\times n}$, $\tilde D_A, \tilde V_A, W_A \in \mathbb{C}^{m\times m}$, $\tilde D_B, \tilde V_B, W_B \in \mathbb{C}^{n\times n}$, $\tilde D_A$ and $\tilde D_B$ be diagonal, $\tilde D_A$ and $-\tilde D_B$ have no diagonal elements in common, and*

$$\begin{aligned}&R_A := W_A(\tilde V_A \tilde D_A - A\tilde V_A),\ R_B := W_B(\tilde V_B \tilde D_B - B^T\tilde V_B), S_A := I_m - W_A\tilde V_A,\ S_B := I_n - W_B\tilde V_B, \\ &T_A := |R_A| + \frac{\|R_A\|_\infty}{1 - \|S_A\|_\infty}|S_A|,\ T_B := |R_B| + \frac{\|R_B\|_\infty}{1 - \|S_B\|_\infty}|S_B|, \\ &T := T_A E + E T_B^T,\ \tilde D := \tilde D_A E + E\tilde D_B,\ T_D := T\ ./\ |\tilde D|.\end{aligned}$$

If $\|S_A\|_\infty < 1$ and $\|S_B\|_\infty < 1$, $\tilde V_A$ and W_A, and $\tilde V_B$ and W_B are nonsingular, respectively. Additionally if $\|T_D\|_{\mathrm{M}} < 1$, (1) has a unique solution X^*.

Proof. Lemma 2, $\|S_A\|_\infty < 1$ and $\|S_B\|_\infty < 1$ give the nonsingularity of $I_m - S_A$ and $I_n - S_B$, which imply the nonsingularity of $\tilde V_A$, W_A, $\tilde V_B$ and W_B. Let P be as in (2). As discussed in Section 1, (1) can be written as (2). From Lemma 1, we have

$$\mathsf{P} = (\tilde V_B \otimes \tilde V_A)\mathsf{Q}(\tilde V_B^{-1} \otimes \tilde V_A^{-1}), \quad \mathsf{Q} := I_n \otimes (\tilde V_A^{-1} A \tilde V_A) + (\tilde V_B^{-1} B^T \tilde V_B) \otimes I_m. \tag{6}$$

Therefore if Q is nonsingular, P is also nonsingular, so that (1) has a unique solution X^*. Thus we prove the nonsingularity of Q. Let $Q_A := (I_m - S_A)^{-1}R_A$, $Q_B := (I_n - S_B)^{-1}R_B$, $\Delta := I_n \otimes \tilde D_A + \tilde D_B \otimes I_m$ and $\Omega := I_n \otimes Q_A + Q_B \otimes I_m$. Observe that Δ is nonsingular, since $\tilde D_A$ and $-\tilde D_B$ have no diagonal elements in common. It follows that

$$\begin{aligned}\mathsf{Q} &= I_n \otimes (\tilde D_A - \tilde D_A + \tilde V_A^{-1} A \tilde V_A) + (\tilde D_B - \tilde D_B + \tilde V_B^{-1} B^T \tilde V_B) \otimes I_m \\ &= I_n \otimes (\tilde D_A - \tilde V_A^{-1}(\tilde V_A \tilde D_A - A\tilde V_A)) + (\tilde D_B - \tilde V_B^{-1}(\tilde V_B \tilde D_A - B^T \tilde V_B)) \otimes I_m \\ &= I_n \otimes (\tilde D_A - \tilde V_A^{-1} W_A^{-1} W_A(\tilde V_A \tilde D_A - A\tilde V_A)) + (\tilde D_B - \tilde V_B^{-1} W_B^{-1} W_B(\tilde V_B \tilde D_A - B^T \tilde V_B)) \otimes I_m \\ &= I_n \otimes (\tilde D_A - (W_A\tilde V_A)^{-1}R_A) + (\tilde D_B - (W_B\tilde V_B)^{-1}R_B) \otimes I_m \\ &= I_n \otimes (\tilde D_A - (I_m - S_A)^{-1}R_A) + (\tilde D_B - (I_n - S_B)^{-1}R_B) \otimes I_m \\ &= I_n \otimes (\tilde D_A - Q_A) + (\tilde D_B - Q_B) \otimes I_m = I_n \otimes \tilde D_A - I_n \otimes Q_A + \tilde D_B \otimes I_m - Q_B \otimes I_m \\ &= \Delta - \Omega = \Delta(I_{mn} - \Delta^{-1}\Omega).\end{aligned} \tag{7}$$

Hence if $I_{mn} - \Delta^{-1}\Omega$ is nonsingular, Q is nonsingular. Lemmas 1 and 3 yield

$$\begin{aligned}
\|\Delta^{-1}\Omega\|_\infty &= \||\Delta^{-1}\Omega|s^{(mn)}\|_\infty = \||\Delta^{-1}||\Omega|s^{(mn)}\|_\infty = \||\Delta^{-1}||\Omega|\mathrm{vec}(E)\|_\infty \\
&\leq \||\Delta^{-1}|(I_n \otimes |Q_A| + |Q_B| \otimes I_m)\mathrm{vec}(E)\|_\infty \\
&= \||\Delta^{-1}|((I_n \otimes |Q_A|)\mathrm{vec}(E) + (|Q_B| \otimes I_m)\mathrm{vec}(E))\|_\infty \\
&= \||\Delta^{-1}|(\mathrm{vec}(|Q_A|E) + \mathrm{vec}(E|Q_B|^T))\|_\infty \\
&= \||\Delta^{-1}|\mathrm{vec}(|Q_A|E + E|Q_B|^T)\|_\infty \\
&= \||\Delta^{-1}|\mathrm{vec}(|Q_A|E + (|Q_B|E^T)^T)\|_\infty \leq \||\Delta^{-1}|\mathrm{vec}(T_AE + (T_BE^T)^T)\|_\infty \\
&= \||\Delta^{-1}|\mathrm{vec}(T)\|_\infty = \|\mathrm{vec}(T\ ./\ |\tilde{D}|)\|_\infty = \|\mathrm{vec}(T_D)\|_\infty = \|T_D\|_{\mathrm{M}}. \qquad (8)
\end{aligned}$$

This and $\|T_D\|_{\mathrm{M}} < 1$ give $\|\Delta^{-1}\Omega\|_\infty < 1$. From this inequality and Lemma 2, $I_{mn} - \Delta^{-1}\Omega$ is nonsingular, so that Q and P are also nonsingular. Consequently (1) has a unique solution X^*. □

Theorem 2. *Let $\tilde{X}$, W_A, W_B, S_A, S_B, $\tilde{D}$ and T_D be as in Theorem 1, and*

$$\begin{aligned}
&R := A\tilde{X} + \tilde{X}B - C,\ R_W := W_ARW_B^T, \\
&D_W^r := \mathrm{diag}(\|R_{W_{1:}}\|_1, \ldots, \|R_{W_{m:}}\|_1),\ R_W^{(1)} := |R_W| + \frac{1}{1 - \|S_B\|_\infty}D_W^rE|S_B|^T, \\
&D_W^{(1)} := \mathrm{diag}(\|R_{W:1}^{(1)}\|_\infty, \ldots, \|R_{W:n}^{(1)}\|_\infty),\ R_V^{(1)} := R_W^{(1)} + \frac{1}{1 - \|S_A\|_\infty}|S_A|ED_W^{(1)}, \\
&D_W^c := \mathrm{diag}(\|R_{W:1}\|_\infty, \ldots, \|R_{W:n}\|_\infty),\ R_W^{(2)} := |R_W| + \frac{1}{1 - \|S_A\|_\infty}|S_A|ED_W^c, \\
&D_W^{(2)} := \mathrm{diag}(\|R_{W_{1:}}^{(2)}\|_1, \ldots, \|R_{W_{m:}}^{(2)}\|_1),\ R_V^{(2)} := R_W^{(2)} + \frac{1}{1 - \|S_B\|_\infty}D_W^{(2)}E|S_B|^T, \\
&R_V := \min(R_V^{(1)}, R_V^{(2)}),\ R_D := R_V\ ./\ |\tilde{D}|,\ U := R_D + \frac{\|R_D\|_{\mathrm{M}}}{1 - \|T_D\|_{\mathrm{M}}}T_D.
\end{aligned}$$

If all the assumptions in Theorem 1 are satisfied, it holds that

$$|\tilde{X} - X^*| \leq X^\varepsilon,\ X^\varepsilon := |\tilde{V}_A|U|\tilde{V}_B|^T.$$

Proof. Let P be as in (2), $\tilde{V}_A$ and $\tilde{V}_B$ be as in Theorem 1, and Q, Δ and Ω be as in the proof of Theorem 1. It holds from (6) and (7) that

$$\begin{aligned}
\mathrm{vec}(\tilde{X}) - \mathrm{vec}(X^*) &= \mathrm{vec}(\tilde{X}) - \mathrm{P}^{-1}\mathrm{vec}(C) = \mathrm{P}^{-1}(\mathrm{P}\mathrm{vec}(\tilde{X}) - \mathrm{vec}(C)) \\
&= \mathrm{P}^{-1}((I_n \otimes A)\mathrm{vec}(\tilde{X}) + (B^T \otimes I_m)\mathrm{vec}(\tilde{X}) - \mathrm{vec}(C)) \\
&= \mathrm{P}^{-1}(\mathrm{vec}(A\tilde{X}) + \mathrm{vec}(\tilde{X}B) - \mathrm{vec}(C)) = \mathrm{P}^{-1}\mathrm{vec}(R) \\
&= (\tilde{V}_B^{-1} \otimes \tilde{V}_A^{-1})^{-1}\mathrm{Q}^{-1}(\tilde{V}_B \otimes \tilde{V}_A)^{-1}\mathrm{vec}(R) \\
&= (\tilde{V}_B \otimes \tilde{V}_A)\mathrm{Q}^{-1}(\tilde{V}_B^{-1} \otimes \tilde{V}_A^{-1})\mathrm{vec}(R) \\
&= (\tilde{V}_B \otimes \tilde{V}_A)(I_{mn} - \Delta^{-1}\Omega)^{-1}\Delta^{-1}\mathrm{vec}(\tilde{V}_A^{-1}R\tilde{V}_B^{-T}). \qquad (9)
\end{aligned}$$

From this, Lemma 3, and (8), we obtain

$$\begin{aligned}
&\mathrm{vec}(|\tilde{X} - X^*|) \\
&= |\mathrm{vec}(\tilde{X} - X^*)| = |\mathrm{vec}(\tilde{X}) - \mathrm{vec}(X^*)| \\
&= |(\tilde{V}_B \otimes \tilde{V}_A)(I_{mn} - \Delta^{-1}\Omega)^{-1}\Delta^{-1}\mathrm{vec}(\tilde{V}_A^{-1}R\tilde{V}_B^{-T})| \\
&\leq |\tilde{V}_B \otimes \tilde{V}_A||(I_{mn} - \Delta^{-1}\Omega)^{-1}\Delta^{-1}\mathrm{vec}(\tilde{V}_A^{-1}R\tilde{V}_B^{-T})|
\end{aligned}$$

$$
\begin{aligned}
&= |\tilde{V}_B \otimes \tilde{V}_A||(I_{mn} - \Delta^{-1}\Omega)^{-1}\mathrm{vec}((\tilde{V}_A^{-1} R \tilde{V}_B^{-T}) \,./\, \tilde{D})| \\
&\le |\tilde{V}_B \otimes \tilde{V}_A| \left(|\mathrm{vec}((\tilde{V}_A^{-1} R \tilde{V}_B^{-T}) \,./\, \tilde{D})| + \frac{\|\mathrm{vec}((\tilde{V}_A^{-1} R \tilde{V}_B^{-T}) \,./\, \tilde{D})\|_\infty}{1 - \|\Delta^{-1}\Omega\|_\infty} |\Delta^{-1}\Omega| s^{(mn)} \right) \\
&\le |\tilde{V}_B \otimes \tilde{V}_A| \left(|\mathrm{vec}((\tilde{V}_A^{-1} R \tilde{V}_B^{-T}) \,./\, \tilde{D})| + \frac{\|\mathrm{vec}((\tilde{V}_A^{-1} R \tilde{V}_B^{-T}) \,./\, \tilde{D})\|_\infty}{1 - \|T_D\|_{\mathrm{M}}} \mathrm{vec}(T_D) \right) \\
&= (|\tilde{V}_B| \otimes |\tilde{V}_A|) \left(\mathrm{vec}(|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}|) + \frac{\|\mathrm{vec}(|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}|)\|_\infty}{1 - \|T_D\|_{\mathrm{M}}} \mathrm{vec}(T_D) \right) \\
&= (|\tilde{V}_B| \otimes |\tilde{V}_A|) \mathrm{vec} \left(|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}| + \frac{\||\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}|\|_{\mathrm{M}}}{1 - \|T_D\|_{\mathrm{M}}} T_D \right) \\
&= \mathrm{vec} \left(|\tilde{V}_A| \left(|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}| + \frac{\||\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}|\|_{\mathrm{M}}}{1 - \|T_D\|_{\mathrm{M}}} T_D \right) |\tilde{V}_B|^T \right),
\end{aligned}
$$

that is,

$$
|\tilde{X} - X^*| \le |\tilde{V}_A| \left(|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}| + \frac{\||\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \,./\, |\tilde{D}|\|_{\mathrm{M}}}{1 - \|T_D\|_{\mathrm{M}}} T_D \right) |\tilde{V}_B|^T. \tag{10}
$$

We derive the upper bound for $|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}|$. Since

$$
\begin{aligned}
|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| &= |\tilde{V}_A^{-1} W_A^{-1} W_A R W_B^T W_B^{-T} \tilde{V}_B^{-T}| = |\tilde{V}_A^{-1} W_A^{-1} R_W W_B^{-T} \tilde{V}_B^{-T}| \\
&= |(W_A \tilde{V}_A)^{-1} R_W (W_B \tilde{V}_B)^{-T}| = |(I_m - S_A)^{-1} R_W (I_n - S_B)^{-T}|,
\end{aligned}
$$

we can derive the following two types of the upper bounds using Lemma 3:

$$
\begin{aligned}
|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| &\le |(I_m - S_A)^{-1}||R_W (I_n - S_B)^{-T}| = |(I_m - S_A)^{-1}||(I_n - S_B)^{-1} R_W^T|^T \\
&\le |(I_m - S_A)^{-1}| \left(|R_W^T| + \frac{1}{1 - \|S_B\|_\infty} |S_B| E^T \mathrm{diag}(\|R^T_{W:1}\|_\infty, \ldots, \|R^T_{W:m}\|_\infty) \right)^T \\
&= |(I_m - S_A)^{-1}| \left(|R_W|^T + \frac{1}{1 - \|S_B\|_\infty} |S_B| E^T D_W^r \right)^T = |(I_m - S_A)^{-1}| R_W^{(1)} \\
&\le R_W^{(1)} + \frac{1}{1 - \|S_A\|_\infty} |S_A| E D_W^{(1)} = R_V^{(1)}, \\
|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| &\le |(I_m - S_A)^{-1} R_W||(I_n - S_B)^{-1}|^T \\
&\le \left(|R_W| + \frac{1}{1 - \|S_A\|_\infty} |S_A| E D_W^c \right) |(I_n - S_B)^{-1}|^T \\
&= R_W^{(2)} |(I_n - S_B)^{-1}|^T = (|(I_n - S_B)^{-1}| R_W^{(2)T})^T \\
&\le \left(R_W^{(2)T} + \frac{1}{1 - \|S_B\|_\infty} |S_B| E^T \mathrm{diag}(\|R^{(2)T}_{W:1}\|_\infty, \ldots, \|R^{(2)T}_{W:m}\|_\infty) \right)^T \\
&= \left(R_W^{(2)T} + \frac{1}{1 - \|S_B\|_\infty} |S_B| E^T D_W^{(2)} \right)^T = R_W^{(2)} + \frac{1}{1 - \|S_B\|_\infty} D_W^{(2)} E |S_B|^T = R_V^{(2)}.
\end{aligned} \tag{11}
$$

These two upper bounds yield $|\tilde{V}_A^{-1} R \tilde{V}_B^{-T}| \le R_V$. It finally follows from this inequality and (10) that

$$|\tilde{X}-X^*| \le |\tilde{V}_A|\left(R_V\ ./\ |\tilde{D}| + \frac{\|R_V\ ./\ |\tilde{D}|\|_{\mathrm{M}}}{1-\|T_D\|_{\mathrm{M}}}T_D\right)|\tilde{V}_B|^T$$
$$= |\tilde{V}_A|\left(R_D + \frac{\|R_D\|_{\mathrm{M}}}{1-\|T_D\|_{\mathrm{M}}}T_D\right)|\tilde{V}_B|^T = |\tilde{V}_A|U|\tilde{V}_B|^T = X^{\varepsilon}. \quad \square$$

The proposed algorithms based on Theorems 1 and 2 compute X^{ε} considering rounding errors. For practically computing X^{ε}, we need to determine $\tilde{X}$, $\tilde{D}_A$, $\tilde{V}_A$, W_A, $\tilde{D}_B$, $\tilde{V}_B$ and W_B. If $\tilde{X}$ is close to X^*, the absolute value of each component of R becomes small. Hence we determine $\tilde{X}$ as a numerical result for X^*. We can expect that $\|S_A\|_\infty < 1$ and $\|S_B\|_\infty < 1$ hold if W_A and W_B are not far from $\tilde{V}_A^{-1}$ and $\tilde{V}_B^{-1}$, respectively. Adding with these conditions, if $\tilde{D}_A$, $\tilde{V}_A$, $\tilde{D}_B$ and $\tilde{V}_B$ are not far from D_A, V_A, D_B and V_B, respectively, we can expect that $\|T_D\|_{\mathrm{M}} < 1$ follows. Thus we determine $\tilde{D}_A$, $\tilde{V}_A$, W_A, $\tilde{D}_B$, $\tilde{V}_B$ and W_B as numerical results for D_A, V_A, $\tilde{V}_A^{-1}$, D_B, V_B and $\tilde{V}_B^{-1}$, respectively. Note that the overall computation involves only $\mathcal{O}(m^3+n^3)$ operations.

2.2. Theory based on block diagonalization

If A and/or B are not diagonalizable, or $\tilde{V}_A$ and/or $\tilde{V}_B$ are ill conditioned, we cannot verify $\|S_A\|_\infty < 1$, $\|S_B\|_\infty < 1$ or $\|T_D\|_{\mathrm{M}} < 1$, so that Theorems 1 and 2 are not applicable. In such situations, we can utilize block diagonalization

$$V_A^{b^{-1}}AV_A^b = D_A^b \quad \text{and} \quad V_B^{b^{-1}}B^TV_B^b = D_B^b,$$

where $V_A^b \in \mathbb{C}^{m\times m}$ and $V_B^b \in \mathbb{C}^{n\times n}$ are nonsingular, and D_A^b and D_B^b are *block* diagonal with each diagonal block being triangular. This block diagonalization has been introduced also in [14]. The algorithm in [18] [1] numerically computes these matrices. Although we can require either, upper or lower triangular form in principle, it is advantageous in the view of computational cost to assume that the both of D_A^b and D_B^b are upper or lower triangular, since $I_n \otimes D_A^b + D_B^b \otimes I_m$ becomes also triangular. This algorithm arrows to trade a better condition of $\tilde{V}_A^b$ for larger diagonal blocks in $\tilde{D}_A^b$.

In this subsection, we develop theory for enclosing solutions in (1) based on the block diagonalization. We present Lemma 4 and Corollary 1 for proving Theorems 3 and 4. Lemma 4 and Corollary 1 are modifications of [19, Lemma 7.3] and [19, Theorem 4.2], respectively, suited for estimating residuals for approximate inverses of complex sparse triangular matrices whose diagonal elements consist of sum of two numbers.

Lemma 4. *Let* $\mathsf{T} \in \mathbb{FC}^{mn\times mn}$ *be triangular,* $\mathsf{T}_{ii} = t_1^{(i)} + t_2^{(i)} \neq 0$, $i = 1,\dots,mn$, $d_{\mathsf{T}} := (\mathsf{T}_{11},\dots,\mathsf{T}_{mnmn})^T$, *the maximum number of nonzero elements in each row of* T *be bounded by* $\mu \le mn$, *additions* $t_1^{(i)} + t_2^{(i)}$ *be executed by floating point operation, and a linear system* $\mathsf{T}\mathsf{x} = \mathsf{b}$ *be solved via backward or forward substitution. Then including underflow, the computed solution* $\tilde{\mathsf{x}}$ *satisfies*

$$|\mathsf{b}-\mathsf{T}\tilde{\mathsf{x}}| \le \zeta_\mu|\mathsf{T}||\tilde{\mathsf{x}}| + \frac{2\sqrt{2}\underline{\boldsymbol{u}}}{1-\mu\boldsymbol{u}}(\mu s^{(mn)} + 2(1+\sqrt{2}\gamma_4)|d_{\mathsf{T}}|).$$

Proof. We defer a proof until Appendix, since techniques used in this proof are not needed in this section. $\square$

Corollary 1. *Let* $\tilde{D}_A^b \in \mathbb{FC}^{m\times m}$ *and* $\tilde{D}_B^b \in \mathbb{FC}^{n\times n}$ *be block diagonal with each diagonal block being triangular,* $\tilde{D}_A^b$ *and* $-\tilde{D}_B^b$ *have no diagonal elements in common,* α *and* β *be the maximum block size in* $\tilde{D}_A^b$ *and* $\tilde{D}_B^b$*, respectively,* $\sigma := \alpha+\beta-1$, $\Delta^b := I_n\otimes\tilde{D}_A^b + \tilde{D}_B^b\otimes I_m$, $d_\Delta := (\Delta_{11}^b,\dots,\Delta_{mnmn}^b)^T$, *additions in*

[1] For preserving real quantities, the algorithm in [18] actually computes only *quasi* triangular blocks, which contains real 2 × 2 diagonal subblocks for any pair of conjugate complex eigenvalues. For simplicity, we always reduce such subblocks to a truly triangular subblock containing the complex conjugate eigenvalues via a unitary similarity transformation.

Δ^b_{ii} be executed by floating point operations, $e^{(i)}$ be the i-th column of I_{mn}, F be the approximate inverse of Δ^b whose rows $e^{(j)T}\mathsf{F}$ are computed by substitution, in any order, of mn linear systems $\Delta^{bT}(\mathsf{F}^Te^{(i)}) = e^{(i)}$, and $\mathsf{S} := I_{mn} - \mathsf{F}\Delta^b$. Then including possible underflow,

$$|\mathsf{S}| \le \zeta_\sigma|\mathsf{F}||\Delta^b| + \frac{2\sqrt{2}\underline{\boldsymbol{u}}}{1-\sigma\boldsymbol{u}}s^{(mn)}(\sigma s^{(mn)T} + 2(1+\sqrt{2}\gamma_4)|d_\Delta|^T).$$

Remark 2. By exploiting Corollary 1, an upper bound for $|\mathsf{S}|s^{(mn)}$ can be computed without executing the matrix multiplication $\mathsf{F}\Delta^b$, which requires $\mathcal{O}(m^3n^3)$ operations. See (13) for detail.

Proof. The result follows by applying Lemma 4 for $\Delta^{bT}(\mathsf{F}^Te^{(i)}) = e^{(i)}$, $i = 1, \ldots, mn$. Observe that the maximum number of nonzero elements in each column of Δ^b is bounded by σ. □

We construct Theorems 3 and 4 for enclosing solutions in (1) based on the block diagonalization.

Theorem 3. *Let $\tilde{D}^b_A$, $\tilde{D}^b_B$, σ, d_Δ and F be as in Corollary 1, $\tilde{X} \in \mathbb{C}^{m\times n}$, $\tilde{V}^b_A$, $W^b_A \in \mathbb{C}^{m\times m}$, $\tilde{V}^b_B$, $W^b_B \in \mathbb{C}^{n\times n}$, S^b_A, S^b_B and T^b be similar to S_A, S_B and T in Theorem 1, respectively, for $\tilde{D}^b_A$, $\tilde{V}^b_A$, W^b_A, $\tilde{D}^b_B$, $\tilde{V}^b_B$ and W^b_B, and*

$$\mathsf{f}_T := |\mathsf{F}|\mathrm{vec}(T^b),\ \ \tilde{D}^b := |\tilde{D}^b_A|E + E|\tilde{D}^b_B|^T,$$
$$\mathsf{f}_D := \zeta_\sigma|\mathsf{F}|\mathrm{vec}(\tilde{D}^b) + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(mn\sigma + 2(1+\sqrt{2}\gamma_4)|d_\Delta|^Ts^{(mn)})}{1-\sigma\boldsymbol{u}}s^{(mn)},$$

$$T^b_D := \mathrm{mat}\left(\mathsf{f}_T + \frac{\|\mathsf{f}_T\|_\infty}{1-\|\mathsf{f}_D\|_\infty}\mathsf{f}_D\right).$$

If $\|S^b_A\|_\infty < 1$ and $\|S^b_B\|_\infty < 1$, then $\tilde{V}^b_A$ and W^b_A, and $\tilde{V}^b_B$ and W^b_B are nonsingular, respectively. Additionally if $\|\mathsf{f}_D\|_\infty < 1$ and $\|T^b_D\|_{\mathrm{M}} < 1$, (1) has a unique solution X^*.

Proof. Let P be as in (2), R be as in Theorem 2, Δ^b and S be as in Corollary 1, and Ω^b be defined similarly to Ω in the proof of Theorem 1 for $\tilde{D}^b_A$, $\tilde{D}^b_B$, $\tilde{V}^b_A$, $\tilde{V}^b_B$, W^b_A and W^b_B. Similarly to the proof of Theorem 1, $\tilde{V}^b_A$, W^b_A, $\tilde{V}^b_B$, W^b_B and Δ^b are nonsingular, and

$$\mathsf{P} = (\tilde{V}^b_B \otimes \tilde{V}^b_A)\mathsf{Q}^b({\tilde{V}^b_B}{}^{-1} \otimes {\tilde{V}^b_A}{}^{-1}),\quad \mathsf{Q}^b := \Delta^b(I_{mn} - {\Delta^b}^{-1}\Omega^b). \tag{12}$$

It holds from Lemma 1 and Corollary 1 that

$$\begin{aligned} |\mathsf{S}|s^{(mn)} &\le \zeta_\sigma|\mathsf{F}||\Delta^b|s^{(mn)} + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(mn\sigma + 2(1+\sqrt{2}\gamma_4)|d_\Delta|^Ts^{(mn)})}{1-\sigma\boldsymbol{u}}s^{(mn)} \\ &\le \zeta_\sigma|\mathsf{F}|(I_n \otimes |\tilde{D}^b_A| + |\tilde{D}^b_B| \otimes I_m)\mathrm{vec}(E) + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(mn\sigma + 2(1+\sqrt{2}\gamma_4)|d_\Delta|^Ts^{(mn)})}{1-\sigma\boldsymbol{u}}s^{(mn)} \\ &= \zeta_\sigma|\mathsf{F}|\mathrm{vec}(\tilde{D}^b) + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(mn\sigma + 2(1+\sqrt{2}\gamma_4)|d_\Delta|^Ts^{(mn)})}{1-\sigma\boldsymbol{u}}s^{(mn)} = \mathsf{f}_D. \end{aligned} \tag{13}$$

This, $\|\mathsf{f}_D\|_\infty < 1$ and Lemma 2 show the nonsingularity of $I_{mn} - \mathsf{S}$. It follows from this nonsingularity, $\|\mathsf{f}_D\|_\infty < 1$, Lemma 3, (8) and (13) that

$$\begin{aligned} |{\Delta^b}^{-1}\Omega^b|s^{(mn)} &\le |{\Delta^b}^{-1}||\Omega^b|\mathrm{vec}(E) \\ &\le |{\Delta^b}^{-1}|\mathrm{vec}(T^b) = |(I_{mn} - \mathsf{S})^{-1}\mathsf{F}|\mathrm{vec}(T^b) \\ &\le |(I_{mn} - \mathsf{S})^{-1}|\mathsf{f}_T \end{aligned}$$

$$\begin{aligned}
&\le \mathsf{f}_T + \frac{\|\mathsf{f}_T\|_\infty}{1-\|\mathsf{S}\|_\infty}|\mathsf{S}|s^{(mn)} \\
&\le \mathsf{f}_T + \frac{\|\mathsf{f}_T\|_\infty}{1-\|\mathsf{f}_D\|_\infty}\mathsf{f}_D = \mathrm{vec}(T_D^b).
\end{aligned} \tag{14}$$

This, $\|\mathrm{vec}(T_D^b)\|_\infty = \|T_D^b\|_{\mathsf{M}} < 1$, Lemma 2 and (12) give that P is nonsingular, i.e., (1) has a unique solution X^*. $\square$

Theorem 4. *Let F be as in Corollary 1, $\tilde{V}_A^b$, W_A^b, $\tilde{V}_B^b$, W_B^b, S_A^b, S_B^b, f_D and T_D^b be as in Theorem 3, R_V^b be similar to R_V in Theorem 2 for W_A^b, W_B^b, S_A^b and S_B^b,*

$$\mathsf{f}_R := |\mathsf{F}|\mathrm{vec}(R_V^b),\ \ R_D^b := \mathrm{mat}\left(\mathsf{f}_R + \frac{\|\mathsf{f}_R\|_\infty}{1-\|\mathsf{f}_D\|_\infty}\mathsf{f}_D\right),$$

and $X^{\varepsilon b}$ be defined similarly to X^{ε} in Theorem 2 for T_D^b, R_D^b, $\tilde{V}_A^b$ and $\tilde{V}_B^b$. If all the assumptions in Theorem 3 are satisfied, $|\tilde{X} - X^*| \le X^{\varepsilon b}$ holds.

Proof. Let R be as in Theorem 2, Δ^b and S be as in Corollary 1, Ω^b be as in the proof of Theorem 3, and U^b be defined similarly to U in Theorem 2 for T_D^b and R_D^b. Lemmas 1 and 3, (9), (11), (13) and (14) yield

$$\begin{aligned}
\mathrm{vec}(|\tilde{X} - X^*|) &= |(\tilde{V}_B^b \otimes \tilde{V}_A^b)(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}\Delta^{b^{-1}}\mathrm{vec}(\tilde{V}_A^{b^{-1}} R \tilde{V}_B^{b^{-T}})| \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}||\Delta^{b^{-1}}|\mathrm{vec}(|\tilde{V}_A^{b^{-1}} R \tilde{V}_B^{b^{-T}}|) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}||\Delta^{b^{-1}}|\mathrm{vec}(R_V^b) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}||(I_{mn} - \mathsf{S})^{-1}\mathsf{F}|\mathrm{vec}(R_V^b) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}||(I_{mn} - \mathsf{S})^{-1}|\mathsf{f}_R \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}|\left(\mathsf{f}_R + \frac{\|\mathsf{f}_R\|_\infty}{1-\|\mathsf{S}\|_\infty}|\mathsf{S}|s^{(mn)}\right) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}|\left(\mathsf{f}_R + \frac{\|\mathsf{f}_R\|_\infty}{1-\|\mathsf{f}_D\|_\infty}\mathsf{f}_D\right) \\
&= |\tilde{V}_B^b \otimes \tilde{V}_A^b||(I_{mn} - \Delta^{b^{-1}}\Omega^b)^{-1}|\mathrm{vec}(R_D^b) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b|\left(\mathrm{vec}(R_D^b) + \frac{\|\mathrm{vec}(R_D^b)\|_\infty}{1-\|\Delta^{b^{-1}}\Omega^b\|_\infty}|\Delta^{b^{-1}}\Omega^b|s^{(mn)}\right) \\
&\le |\tilde{V}_B^b \otimes \tilde{V}_A^b|\left(\mathrm{vec}(R_D^b) + \frac{\|\mathrm{vec}(R_D^b)\|_\infty}{1-\|\mathrm{vec}(T_D^b)\|_\infty}\mathrm{vec}(T_D^b)\right) \\
&= |\tilde{V}_B^b \otimes \tilde{V}_A^b|\mathrm{vec}\left(R_D^b + \frac{\|R_D^b\|_{\mathsf{M}}}{1-\|T_D^b\|_{\mathsf{M}}}T_D^b\right) = |\tilde{V}_B^b \otimes \tilde{V}_A^b|\mathrm{vec}(U^b) \\
&= (|\tilde{V}_B^b| \otimes |\tilde{V}_A^b|)\mathrm{vec}(U^b) = \mathrm{vec}(X^{\varepsilon b}),
\end{aligned}$$

proving $|\tilde{X} - X^*| \le X^{\varepsilon b}$. $\square$

The proposed algorithms based on Theorems 3 and 4 compute $X^{\varepsilon b}$ considering rounding errors. Similarly to Section 2.1, $\tilde{X}$, $\tilde{D}_A^b$, $\tilde{V}_A^b$, W_A^b, $\tilde{D}_B^b$, $\tilde{V}_B^b$ and W_B^b are determined as numerical results for X^*, D_A^b, V_A^b, $\tilde{V}_A^{b^{-1}}$, D_B^b, V_B^b and $\tilde{V}_B^{b^{-1}}$, respectively, in the practical execution. Note that the computation of $X^{\varepsilon b}$ requires $\mathcal{O}(\sigma m^2 n^2)$ operations, since the computationally most complex part is the calculation of F and this calculation involves $\mathcal{O}(\sigma m^2 n^2)$ operations.

3. Techniques for accelerating the enclosure

In this section, we introduce techniques for accelerating the computation in the proposed algorithms. Let $\tilde{V}_A$, W_A, S_A, S_B, T, $R_W^{(j)}$, $R_V^{(j)}$ for $j = 1, 2$ and X^{ε} be as in Section 2.1, and F, S_A^b, S_B^b and $X^{\varepsilon b}$ be as in Section 2.2.

As described in Section 2, the proposed algorithms compute X^{ε} or $X^{\varepsilon b}$ taking rounding errors into account. For computing X^{ε}, we need to compute T, $R_W^{(j)}$ and $R_V^{(j)}$ for $j = 1, 2$. If upper bounds for $|S_A|s^{(m)}$ and $|S_B|s^{(n)}$ have been obtained, we do not need to execute matrix multiplications $|S_A|E$ and $E|S_B|^T$, since

$$|S_A|E = (|S_A|s^{(m)}, \ldots, |S_A|s^{(m)}) \quad \text{and} \quad E|S_B|^T = (|S_B|s^{(n)}, \ldots, |S_B|s^{(n)})^T,$$

respectively. Analogously we do not need to execute matrix multiplications $|R_A|E$ and $E|R_B|^T$, if upper bounds for $|R_A|s^{(m)}$ and $|R_B|s^{(n)}$ have been obtained. Thus the computation of T requires only $\mathcal{O}(m^2 + n^2)$ operations if upper bounds for $|S_A|s^{(m)}$, $|S_B|s^{(n)}$, $|R_A|s^{(m)}$ and $|R_B|s^{(n)}$ have been obtained. Similarly the computation of $R_W^{(j)}$ and $R_V^{(j)}$ for $j = 1, 2$ requires only $\mathcal{O}(m^2 + n^2)$ operations if upper bounds for $|S_A|s^{(m)}$, $|S_B|s^{(n)}$ and $|R_W|$ have been obtained. Completely analogous discussion is possible also in the computation of $X^{\varepsilon b}$.

The following technique accelerates the computation of upper bounds for $|S_A|s^{(m)}$ and $|S_B|s^{(n)}$: From [20, Proof of Theorem 2] and Lemma 5, we have

$$\begin{aligned}|S_A|s^{(m)} &\le |\mathrm{fl}(S_A)|s^{(m)} + \varphi_{m-1}|W_A||\tilde{V}_A|s^{(m)} \\ &\quad + \boldsymbol{u}(|\mathrm{fl}(W_A\tilde{V}_A)|s^{(m)} + s^{(m)}) + 2\sqrt{2}m^2\underline{\boldsymbol{u}}(1 + \gamma_{m-1})s^{(m)},\end{aligned} \tag{15}$$

where $\varphi_p = (\sqrt{5}\boldsymbol{u} + (1 + \sqrt{5}\boldsymbol{u})\gamma_p)$, if $\boldsymbol{u} < 2^{-5}$. Note that (15) holds also in the presence of underflow. The analogous of (15) also follows for $|S_B|s^{(n)}$. From (15) and its analogous for $|S_B|s^{(n)}$, we need to execute matrix multiplications $W_A\tilde{V}_A$ and $W_B\tilde{V}_B$ only once in rounding-to-nearest mode for calculating the rigorous upper bounds for $|S_A|s^{(m)}$ and $|S_B|s^{(n)}$, respectively, so that the computations of T, $R_W^{(j)}$ and $R_V^{(j)}$ for $j = 1, 2$ can be accelerated.

Similar technique is applicable also in the computations of upper bounds for $|S_A^b|s^{(m)}$ and $|S_B^b|s^{(n)}$. In the computation of $X^{\varepsilon b}$, on the other hand, we cannot expect that this technique remarkably accelerates the overall computation. The reason is that the computation of F in Corollary 1 requires $\mathcal{O}(\sigma m^2 n^2)$ operations, while this technique reduces the frequency of $\mathcal{O}(m^3 + n^3)$ operations.

4. Techniques for obtaining smaller error bounds

In this section, we introduce techniques for obtaining smaller error bounds. Let P be as in (2), D_A, D_B, V_A, V_B, $\tilde{V}_A$, $\tilde{V}_B$, W_A, W_B, $\tilde{D}$, R and X^{ε} be as in Section 2.1, and D_A^b, D_B^b, Δ^b, F, σ, V_A^b, V_B^b, $\tilde{V}_A^b$, $\tilde{V}_B^b$ and $X^{\varepsilon b}$ be as in Section 2.2.

For reducing each component of X^{ε} and $X^{\varepsilon b}$, we need to reduce the absolute values of each component of R. For obtaining R whose components are small in the sense of absolute value, an accurate approximation $\tilde{X}$ to X^* is necessary. Such accurate $\tilde{X}$ can be obtained via iterative refinement. For $\mathsf{y} := \mathsf{P}^{-1}\mathrm{vec}(R)$, we have

$$\mathrm{vec}(\tilde{X}) - \mathsf{y} = \mathsf{P}^{-1}(\mathsf{P}\mathrm{vec}(\tilde{X}) - \mathrm{vec}(A\tilde{X} + \tilde{X}B) + \mathrm{vec}(C)) = \mathrm{vec}(X^*).$$

For producing accurate $\tilde{X}$, moreover, it is necessary to compute R with extended precision computation (see e.g., [21]). Hence iterative refinement for $\tilde{X}$ can be executed by repeating the following procedure:

1. Compute R with extended precision computation.
2. Solve linear systems $\mathsf{P}\mathsf{y} = \mathrm{vec}(R)$.
3. Update $\tilde{X}$ such that $\tilde{X} = \tilde{X} - \mathrm{mat}(\mathsf{y})$.

When we form P explicitly and solve $\mathsf{P}y = \mathrm{vec}(R)$ via direct methods, on the other hand, $\mathcal{O}(m^3n^3)$ operations are required, so that the iterative refinement is prohibitive in the view of computational cost. Thus we present the techniques for reducing the computational cost of this procedure to $\mathcal{O}(m^3 + n^3)$ operations.

4.1. Iterative refinement exploiting spectral decomposition

Consider the spectral decomposition discussed in Section 2.1. From the analogous derivation to the proof of Theorem 1, it holds that

$$\begin{aligned}\mathsf{P}^{-1}\mathrm{vec}(R) &= (V_B \otimes V_A)(I_n \otimes D_A + D_B \otimes I_m)^{-1}(V_B^{-1} \otimes V_A^{-1})\mathrm{vec}(R)\\ &= (V_B \otimes V_A)(I_n \otimes D_A + D_B \otimes I_m)^{-1}\mathrm{vec}(V_A^{-1}RV_B^{-T})\\ &= (V_B \otimes V_A)\mathrm{vec}((V_A^{-1}RV_B^{-T})\ ./\ D)\\ &= \mathrm{vec}(V_A((V_A^{-1}RV_B^{-T})\ ./\ D)V_B^T),\end{aligned}$$

where $D := D_AE + ED_B$. Therefore $\mathsf{P}y = \mathrm{vec}(R)$ can be numerically solved with $\mathcal{O}(m^3 + n^3)$ operations by exploiting $\tilde{V}_A$, $\tilde{V}_B$, W_A, W_B and $\tilde{D}$. This discussion modifies the above procedure as follows:

1. Compute R with extended precision computation.
2. Compute $Y := \tilde{V}_A((W_ARW_B^T)\ ./\ \tilde{D})\tilde{V}_B^T$.
3. Update $\tilde{X}$ such that $\tilde{X} = \tilde{X} - Y$.

Remark 3. The iterative refinement based on the standard direct methods in Section 1, i.e., that exploiting the Schur decomposition is also possible. Solving $\mathsf{P}y = \mathrm{vec}(R)$ with the Schur decompositions requires four matrix multiplications and the forward or backward substitution, if the decompositions are completed. This forward or backward substitution requires $\mathcal{O}(m^3 + n^3)$ operations. On the other hand, solving this equation with the spectral decompositions requires four matrix multiplications and the pointwise division, if $\tilde{V}_A$, $\tilde{V}_B$, W_A, W_B and $\tilde{D}$ have already been obtained. This pointwise division requires only $\mathcal{O}(m^2 + n^2)$ operations. Hence the iterative refinement based on the procedure given in this subsection is faster than that exploiting the Schur decomposition in the case when the algorithms based on Theorems 1 and 2 are executed. Analogous discussion is also possible for the iterative refinement in Section 4.2 if $\sigma = \mathcal{O}(1)$.

4.2. Iterative refinement exploiting block diagonalization

Consider the block diagonalization described in Section 2.2. It follows that

$$\begin{aligned}\mathsf{P}^{-1}\mathrm{vec}(R) &= (V_B^b \otimes V_A^b)(I_n \otimes D_A^b + D_B^b \otimes I_m)^{-1}(V_B^{b^{-1}} \otimes V_A^{b^{-1}})\mathrm{vec}(R)\\ &= (V_B^b \otimes V_A^b)(I_n \otimes D_A^b + D_B^b \otimes I_m)^{-1}\mathrm{vec}(V_A^{b^{-1}}RV_B^{b^{-T}})\\ &= (V_B^b \otimes V_A^b)\mathrm{vec}(\mathrm{mat}((I_n \otimes D_A^b + D_B^b \otimes I_m)^{-1}\mathrm{vec}(V_A^{b^{-1}}RV_B^{b^{-T}})))\\ &= \mathrm{vec}(V_A^b\mathrm{mat}((I_n \otimes D_A^b + D_B^b \otimes I_m)^{-1}\mathrm{vec}(V_A^{b^{-1}}RV_B^{b^{-T}}))V_B^{bT}).\end{aligned}$$

Hence we can exploit $\tilde{V}_A^b$, W_A^b, $\tilde{V}_B^b$, W_B^b and F. Then the above procedure can be modified as follows:

1. Compute R with extended precision computation.
2. Compute $Y^b := \tilde{V}_A^b\mathrm{mat}(\mathsf{F}\mathrm{vec}(W_A^bRW_B^{bT}))\tilde{V}_B^{bT}$.
3. Update $\tilde{X}$ such that $\tilde{X} = \tilde{X} - Y^b$.

However this procedure requires $\mathcal{O}(m^2n^2)$ operations, since $\mathsf{F} \in \mathbb{C}^{mn\times mn}$ is dense triangular. This procedure can be revised as follows, where $\tilde{D}_A^b$ and $\tilde{D}_B^b$ are assumed to be lower triangular without loss of generality:

1. Compute R with extended precision computation.
2. Solve $\Delta^b y = \mathrm{vec}(W_A^b R W_B^{bT})$ via forward substitution.
3. Compute $Y^b := \tilde{V}_A^b \mathrm{mat}(y) \tilde{V}_B^{bT}$.
4. Update $\tilde{X}$ such that $\tilde{X} = \tilde{X} - Y^b$.

This procedure requires $\mathcal{O}(m^3+n^3)$ operations. The reason is that $1 \le \sigma \le m+n-1$ and the second step requires $\mathcal{O}(\sigma mn)$ operations, since the maximum number of nonzero elements in each row of Δ^b is bounded by σ.

5. Numerical results

In this section, we report some numerical results to show the property of the proposed algorithms and performance of our implementation. Let X^*, $\tilde{X}$, $\tilde{D}_A$, $\tilde{D}_B$, $\tilde{V}_A$, $\tilde{V}_B$, W_A, W_B, S_A, R, T_D $\tilde{D}_A^b$, $\tilde{D}_B^b$, σ, $\tilde{V}_A^b$, $\tilde{V}_B^b$, W_A^b, W_B^b, f_D and T_D^b be as in Section 2.

We used a computer with Intel Xeon 2.66GHz Dual CPU, 4.00GB RAM and MATLAB 7.5 with Intel Math Kernel Library and IEEE 754 double precision. We applied the function `lyap` from the MATLAB Control System Toolbox for obtaining $\tilde{X}$. The function `bdschur` in MATLAB Control System Toolbox and the MATLAB function `schur` were executed for obtaining $\tilde{D}_A^b$ and $\tilde{V}_A^b$.[2] The function `bdschur` can take an upper bound for the condition number of $\tilde{V}_A^b$ as an input and adjusts the number of diagonal blocks in $\tilde{D}_A^b$ and their sizes accordingly. Moreover this function outputs the size of each block in $\tilde{D}_A^b$. Thus we can compute σ by exploiting this output. We asked for a condition number in $\tilde{V}_A^b$ of $1/\sqrt{\epsilon} \approx$ 6.7e+7, the default value in `bdschur`, where $\epsilon = 2^{-52}$ is the machine epsilon. The MATLAB functions `eig` and `inv` were called for obtaining $\tilde{D}_A$, $\tilde{D}_B$, $\tilde{V}_A$ and $\tilde{V}_B$, and W_A, W_B and W_A^b, respectively.

In some of the proposed algorithms, we computed X^ε or $X^{\varepsilon b}$ with the iterative refinements discussed in Sections 4.1 or 4.2, respectively. In order to compute matrix multiplications $A\tilde{X}$ and $\tilde{X}B$ within R with extended precision, we created and executed the modified version of the function `Acc_Mul`[3] in [22]. For $m \times n$ and $n \times p$ floating point matrices M and N, respectively, `Acc_Mul`(M, N, k, δ) outputs *one matrix* C which is an accurate approximation for MN. In `Acc_Mul`(M, N, k, δ), M and N are splited into unevaluated sum of floating point matrices such that $M = \sum_{i=1}^{k_M} M^{(i)}$ and $N = \sum_{j=1}^{k_N} N^{(j)}$, where $k_M \le k$ and $k_N \le k$, respectively. If the number of nonzero elements in $M^{(i)}$, $i = 1, \dots, k_M$ and/or $N^{(j)}$, $j = 1, \dots, k_N$ are smaller than δmn and/or δnp, then $M^{(i)}$ and/or $N^{(j)}$ are stored in sparse format, respectively. Namely the parameters k and δ are the maximum number of the matrix splitting and the criterion for using the sparse format, respectively (see [22] for detail). The modified version of `Acc_Mul` outputs *several matrices* $G^{(1)}, \dots, G^{(l)}$ where $\sum_{i=1}^{l} G^{(i)}$ is an accurate approximation of MN. The positive integer l is determined during the execution. Concretely, in the modified version, the last for-loop (sum) in `Acc_Mul` is deleted, and the output is changed from C (evaluated sum) to $G^{(1)}, \dots, G^{(l)}$ (unevaluated sum). We name the modified version as `Acc_Mul_noSum`.

For computing matrix additions and subtractions within R with extended precision, we created and executed the matrix version of the function `Sum2`[4] in [23]. For $m \times n$ floating point matrices $M^{(1)}, \dots, M^{(p)}$, the matrix version of `Sum2` computes the sum $\sum_{i=1}^{p} M^{(i)}$ in 2-fold working precision (see [23] for details). We name this matrix version as `Sum2_Mat`.

Exploiting `Acc_Mul_noSum` and `Sum2_Mat`, R can be computed with extended precision by the following code, where A_r and A_c are the real and imaginary parts of A, respectively, and B_r, B_c, C_r, C_c,

[2] We executed the algorithms based on Theorems 3 and 4 for the Lyapunov equation (16) only. Thus computations of $\tilde{D}_B^b$, $\tilde{V}_B^b$ and W_B^b were not needed.

[3] When we execute `Acc_Mul`, it is assumed that neither overflow nor underflow occurs during the execution. In the examples shown below, we verified that neither overflow nor underflow occurred during the execution.

[4] There are algorithms (e.g., [23–26]) which returns more accurate results for floating point summations than `Sum2`. However X^ε and $X^{\varepsilon b}$ were not improved even when we executed such algorithms in the examples below.

$\tilde{X}_r$ and $\tilde{X}_c$ are defined analogously:

```
$[Y_{rr}^{(1)}, \dots, Y_{rr}^{(l_1)}]$ = Acc_Mul_noSum($A_r, \tilde{X}_r$, 3, 0.1);
$[Y_{cc}^{(1)}, \dots, Y_{cc}^{(l_2)}]$ = Acc_Mul_noSum($A_c, \tilde{X}_c$, 3, 0.1);
$[Y_{rc}^{(1)}, \dots, Y_{rc}^{(l_3)}]$ = Acc_Mul_noSum($A_r, \tilde{X}_c$, 3, 0.1);
$[Y_{cr}^{(1)}, \dots, Y_{cr}^{(l_4)}]$ = Acc_Mul_noSum($A_c, \tilde{X}_r$, 3, 0.1);
$[Z_{rr}^{(1)}, \dots, Z_{rr}^{(l_5)}]$ = Acc_Mul_noSum($\tilde{X}_r, B_r$, 3, 0.1);
$[Z_{cc}^{(1)}, \dots, Z_{cc}^{(l_6)}]$ = Acc_Mul_noSum($\tilde{X}_c, B_c$, 3, 0.1);
$[Z_{rc}^{(1)}, \dots, Z_{rc}^{(l_7)}]$ = Acc_Mul_noSum($\tilde{X}_r, B_c$, 3, 0.1);
$[Z_{cr}^{(1)}, \dots, Z_{cr}^{(l_8)}]$ = Acc_Mul_noSum($\tilde{X}_c, B_r$, 3, 0.1);
$R_r$ = Sum2_Mat($Y_{rr}^{(1)}, \dots, Y_{rr}^{(l_1)}, -Y_{cc}^{(1)}, \dots, -Y_{cc}^{(l_2)}, Z_{rr}^{(1)}, \dots, Z_{rr}^{(l_5)}, -Z_{cc}^{(1)}, \dots, -Z_{cc}^{(l_6)}, -C_r$);
$R_c$ = Sum2_Mat($Y_{rc}^{(1)}, \dots, Y_{rc}^{(l_3)}, Y_{cr}^{(1)}, \dots, Y_{cr}^{(l_4)}, Z_{rc}^{(1)}, \dots, Z_{rc}^{(l_7)}, Z_{cr}^{(1)}, \dots, Z_{cr}^{(l_8)}, -C_c$);
$R = R_r$ + i*$R_c$;
```

We set the number of iteration as one,[5] and executed the iterative refinement before computing X^{ε} or $X^{\varepsilon b}$.

We need to enclose R for computing X^{ε} or $X^{\varepsilon b}$ considering rounding errors. In the enclosure of R in the proposed algorithms including iterative refinements, we also used the extended precision. For enclosing the result of matrix multiplications $A\tilde{X}$ and $\tilde{X}B$ within R with extended precision, we created and executed the modified version of Acc_Mul_noSum. For the above M, N and l, this modified version outputs floating point matrices $G^{(1)}, \dots, G^{(l_G)}, \underline{H}^{(1)}, \dots, \underline{H}^{(l_H)}, \overline{H}^{(1)}, \dots, \overline{H}^{(l_H)}$ satisfying $l_G + l_H = l, \underline{H}^{(j)} \le \overline{H}^{(j)}, j = 1, \dots, l_H, MN \in \left[\sum_{i=1}^{l_G} G^{(i)} + \sum_{j=1}^{l_H} \underline{H}^{(j)}, \sum_{i=1}^{l_G} G^{(i)} + \sum_{j=1}^{l_H} \overline{H}^{(j)}\right]$, where $[\underline{H}, \overline{H}]$ for $\underline{H} \le \overline{H}$ is a matrix interval whose infmum and supmum are $\underline{H}$ and $\overline{H}$, respectively. Concretely, in this modified version, the third for-loop (matrix multiplications including rounding error) in Acc_Mul_noSum are executed twice: once in rounding to $-\infty$ mode and once in rounding to ∞ mode. The matrices $\underline{H}^{(1)}, \dots, \underline{H}^{(l_H)}$ and $\overline{H}^{(1)}, \dots, \overline{H}^{(l_H)}$ are results of these rounded multiplications. The matrices $G^{(1)}, \dots, G^{(l_G)}$ are results of error-free matrix multiplications in the second for-loop in Acc_Mul_noSum. We name this modified version as Acc_Mul_noSum_Int.

For enclosing the result of matrix additions and subtractions within R with extended precision, we created and used the two modified versions of Sum2_Mat. For the above $M^{(1)}, \dots, M^{(p)}$, the first modification computes a rigorous lower bound for $\sum_{i=1}^{p} M^{(i)}$ in 2-fold working precision. Concretely, in the first modification, the last floating point summation in Sum2_Mat is executed in rounding to $-\infty$ mode. The second modification compute a rigorous upper bound of $\sum_{i=1}^{p} M^{(i)}$ whose mechanism is analogous to the first one. We name the first and the second modifications as Sum2_Mat_Inf and Sum2_Mat_Sup, respectively.

Exploiting Acc_Mul_noSum_Int, Sum2_Mat_Inf and Sum2_Mat_Sup, R can be enclosed with extended precision by the following code:

```
$[Y_{rr}^{(1)}, \dots, Y_{rr}^{(l_{Y1})}, \underline{P}_{rr}^{(1)}, \dots, \underline{P}_{rr}^{(l_{P1})}, \overline{P}_{rr}^{(1)}, \dots, \overline{P}_{rr}^{(l_{P1})}]$ = Acc_Mul_noSum_Int($A_r, \tilde{X}_r$, 3, 0.1);
$[Y_{cc}^{(1)}, \dots, Y_{cc}^{(l_{Y2})}, \underline{P}_{cc}^{(1)}, \dots, \underline{P}_{cc}^{(l_{P2})}, \overline{P}_{cc}^{(1)}, \dots, \overline{P}_{cc}^{(l_{P2})}]$= Acc_Mul_noSum_Int($A_c, \tilde{X}_c$, 3, 0.1);
$[Y_{rc}^{(1)}, \dots, Y_{rc}^{(l_{Y3})}, \underline{P}_{rc}^{(1)}, \dots, \underline{P}_{rc}^{(l_{P3})}, \overline{P}_{rc}^{(1)}, \dots, \overline{P}_{rc}^{(l_{P3})}]$= Acc_Mul_noSum_Int($A_r, \tilde{X}_c$, 3, 0.1);
$[Y_{cr}^{(1)}, \dots, Y_{cr}^{(l_{Y4})}, \underline{P}_{cr}^{(1)}, \dots, \underline{P}_{cr}^{(l_{P4})}, \overline{P}_{cr}^{(1)}, \dots, \overline{P}_{cr}^{(l_{P4})}]$ = Acc_Mul_noSum_Int($A_c, \tilde{X}_r$, 3, 0.1);
```

[5] In the examples below, X^{ε} and $X^{\varepsilon b}$ were not improved even when we set the number of iterations as two or more.

$$\begin{aligned}
&[Z_{rr}^{(1)}, \dots, Z_{rr}^{(l_{Z1})}, \underline{Q}_{rr}^{(1)}, \dots, \underline{Q}_{rr}^{(l_{Q1})}, \overline{Q}_{rr}^{(1)}, \dots, \overline{Q}_{rr}^{(l_{Q1})}] = \mathtt{Acc\_Mul\_noSum\_Int}(\tilde{X}_r, B_r, 3, 0.1);\\
&[Z_{cc}^{(1)}, \dots, Z_{cc}^{(l_{Z2})}, \underline{Q}_{cc}^{(1)}, \dots, \underline{Q}_{cc}^{(l_{Q2})}, \overline{Q}_{cc}^{(1)}, \dots, \overline{Q}_{cc}^{(l_{Q2})}] = \mathtt{Acc\_Mul\_noSum\_Int}(\tilde{X}_c, B_c, 3, 0.1);\\
&[Z_{rc}^{(1)}, \dots, Z_{rc}^{(l_{Z3})}, \underline{Q}_{rc}^{(1)}, \dots, \underline{Q}_{rc}^{(l_{Q3})}, \overline{Q}_{rc}^{(1)}, \dots, \overline{Q}_{rc}^{(l_{Q3})}] = \mathtt{Acc\_Mul\_noSum\_Int}(\tilde{X}_r, B_c, 3, 0.1);\\
&[Z_{cr}^{(1)}, \dots, Z_{cr}^{(l_{Z4})}, \underline{Q}_{cr}^{(1)}, \dots, \underline{Q}_{cr}^{(l_{Q4})}, \overline{Q}_{cr}^{(1)}, \dots, \overline{Q}_{cr}^{(l_{Q4})}] = \mathtt{Acc\_Mul\_noSum\_Int}(\tilde{X}_c, B_r, 3, 0.1);\\
&\underline{R}_r = \mathtt{Sum2\_Mat\_Inf}(Y_{rr}^{(1)}, \dots, Y_{rr}^{(l_{Y1})}, \underline{P}_{rr}^{(1)}, \dots, \underline{P}_{rr}^{(l_{P1})}, -Y_{cc}^{(1)}, \dots, -Y_{cc}^{(l_{Y2})},\\
&\qquad -\overline{P}_{cc}^{(1)}, \dots, -\overline{P}_{cc}^{(l_{P2})}, Z_{rr}^{(1)}, \dots, Z_{rr}^{(l_{Z1})}, \underline{Q}_{rr}^{(1)}, \dots, \underline{Q}_{rr}^{(l_{Q1})},\\
&\qquad -Z_{cc}^{(1)}, \dots, -Z_{cc}^{(l_{Z2})}, -\overline{Q}_{cc}^{(1)}, \dots, -\overline{Q}_{cc}^{(l_{Q2})}, -C_r);\\
&\underline{R}_c = \mathtt{Sum2\_Mat\_Inf}(Y_{rc}^{(1)}, \dots, Y_{rc}^{(l_{Y3})}, \underline{P}_{rc}^{(1)}, \dots, \underline{P}_{rc}^{(l_{P3})}, Y_{cr}^{(1)}, \dots, Y_{cr}^{(l_{Y4})},\\
&\qquad \underline{P}_{cr}^{(1)}, \dots, \underline{P}_{cr}^{(l_{P4})}, Z_{rc}^{(1)}, \dots, Z_{rc}^{(l_{Z3})}, \underline{Q}_{rc}^{(1)}, \dots, \underline{Q}_{rc}^{(l_{Q3})},\\
&\qquad Z_{cr}^{(1)}, \dots, Z_{cr}^{(l_{Z4})}, \underline{Q}_{cr}^{(1)}, \dots, \underline{Q}_{cr}^{(l_{Q4})}, -C_c);\\
&\overline{R}_r = \mathtt{Sum2\_Mat\_Sup}(Y_{rr}^{(1)}, \dots, Y_{rr}^{(l_{Y1})}, \overline{P}_{rr}^{(1)}, \dots, \overline{P}_{rr}^{(l_{P1})}, -Y_{cc}^{(1)}, \dots, -Y_{cc}^{(l_{Y2})},\\
&\qquad -\underline{P}_{cc}^{(1)}, \dots, -\underline{P}_{cc}^{(l_{P2})}, Z_{rr}^{(1)}, \dots, Z_{rr}^{(l_{Z1})}, \overline{Q}_{rr}^{(1)}, \dots, \overline{Q}_{rr}^{(l_{Q1})},\\
&\qquad -Z_{cc}^{(1)}, \dots, -Z_{cc}^{(l_{Z2})}, -\underline{Q}_{cc}^{(1)}, \dots, -\underline{Q}_{cc}^{(l_{Q2})}, -C_r);\\
&\overline{R}_c = \mathtt{Sum2\_Mat\_Sup}(Y_{rc}^{(1)}, \dots, Y_{rc}^{(l_{Y3})}, \overline{P}_{rc}^{(1)}, \dots, \overline{P}_{rc}^{(l_{P3})}, Y_{cr}^{(1)}, \dots, Y_{cr}^{(l_{Y4})},\\
&\qquad \overline{P}_{cr}^{(1)}, \dots, \overline{P}_{cr}^{(l_{P4})}, Z_{rc}^{(1)}, \dots, Z_{rc}^{(l_{Z3})}, \overline{Q}_{rc}^{(1)}, \dots, \overline{Q}_{rc}^{(l_{Q3})},\\
&\qquad Z_{cr}^{(1)}, \dots, Z_{cr}^{(l_{Z4})}, \overline{Q}_{cr}^{(1)}, \dots, \overline{Q}_{cr}^{(l_{Q4})}, -C_c);
\end{aligned}$$

Then $R \in [\underline{R}_r, \overline{R}_r] + i[\underline{R}_c, \overline{R}_c]$ follows. Note that the number of function calls in the above two codes can be reduced if A, B, C or $\tilde{X}$ are real.

We denote the compared algorithms as follows:

M1: The algorithm based on Theorems 1 and 2 with the techniques in Section 3.
M1i: M1 with the iterative refinement in Section 4.1.
M2: The algorithm based on Theorems 3 and 4 with the techniques in Section 3.
M2i: M2 with the iterative refinement in Section 4.2.
V: Versoft [13] function VERMATEQN.
FH1: The algorithm in [14] with usual double precision.
FH1i: The algorithm in [14] with improved precision [27].
FH1q: The algorithm in [14] with simulated quadruple precision [23].
FH2: The block diagonalization version of FH1.
FH2i: The block diagonalization version of FH1i.
FH2q: The block diagonalization version of FH1q.

We adopted the techniques in Section 3 also in M2 and M2i, since computing times were reduced slightly.

Let an interval matrix $\boldsymbol{X}$ include X^* and $\mathrm{rad}(\boldsymbol{X}_{ij})$ be the radius of $\boldsymbol{X}_{ij}$. Similarly to [14], in order to assess the quality of the enclosures, we define the relative radii

$$\xi_{ij} := \frac{\mathrm{rad}(\boldsymbol{X}_{ij})}{\max\{|X_{ij}| : X_{ij} \in \boldsymbol{X}_{ij}\}}, \quad i = 1, \dots, m, \ j = 1, \dots, n.$$

When X^* is enclosed by M1, M1i, M2 or M2i, we have

$$\xi_{ij} = \frac{|X_{ij}^{\varepsilon}|}{|\tilde{X}_{ij}| + |X_{ij}^{\varepsilon}|} \quad \text{or} \quad \xi_{ij} = \frac{|X_{ij}^{\varepsilon b}|}{|\tilde{X}_{ij}| + |X_{ij}^{\varepsilon b}|}.$$

We define maximum relative radius mrr and average relative radius arr as

$$\mathrm{mrr} := \max_{i,j} \xi_{ij} \quad \text{and} \quad \mathrm{arr} := \left(\prod_{i,j} \xi_{ij} \right)^{\frac{1}{mn}},$$

respectively. Let t_{enc}, $t_{\texttt{lyap}}$ and t_V be the computing time of an enclosing algorithm, `lyap` and the Versoft routine V, respectively. Define time ratios for `lyap` and V as

$$\text{time ratio for } \texttt{lyap} := \frac{t_{\mathrm{enc}}}{t_{\texttt{lyap}}} \quad \text{and} \quad \text{time ratio for V} := \frac{t_{\mathrm{enc}} + t_{\texttt{lyap}}}{t_V},$$

respectively. We report results for the numerical examples treated in [14]. The radii mrr and arr obtained by FH1, FH1i, FH1q, FH2, FH2i and FH2q are reproduced from [14]. We calculated the above time ratios of FH1, FH1i, FH1q, FH2, FH2i and FH2q from the computing times reported in [14]. Since FH1, FH1i, FH1q, FH2, FH2i and FH2q include the computation of $\tilde{X}$ by `lyap`, we calculated the above time ratios of these algorithms such that

$$\text{time ratio for } \texttt{lyap} = \frac{t_{\mathrm{enc}} - t_{\texttt{lyap}}}{t_{\texttt{lyap}}} \quad \text{and} \quad \text{time ratio for V} = \frac{t_{\mathrm{enc}}}{t_V}.$$

Note that the comparison of the time ratios of the proposed algorithms with those of the algorithms in [14] is not completely fair, since the results reported in [14] have been obtained using a different hardware.

5.1. Results for well conditioned problems

In this subsection, we report the obtained radii and time ratios for the parameterized test examples in [28]. Let a, b, s be real parameters, and

$$A = \mathrm{fl}(T_0^{-T} A_0 T_0^T), \quad B = \mathrm{fl}(T_0 B_0 T_0^{-1}) \quad \text{and} \quad C = \mathrm{fl}(T_0^{-T} C_0 T_0^{-1}),$$

where

$$\begin{aligned}
&A_0 := \mathrm{fl}(\mathrm{diag}(-1, -a, \ldots, -a^{n-1})), \quad B_0 := \mathrm{fl}(\mathrm{diag}(-1, -b, \ldots, -b^{n-1})),\\
&C_0 := \mathrm{diag}(1, \ldots, n), \quad T_0 := \mathrm{fl}(H_2 S_0 H_1), \quad S_0 := \mathrm{fl}(\mathrm{diag}(1, s, \ldots, s^{n-1})),\\
&H_1 := \mathrm{fl}(I_n - (2/n)ee^T), \quad e := (1, \ldots, 1)^T,\\
&H_2 := \mathrm{fl}(I_n - (2/n)ff^T), \quad f := (-1, 1, \ldots, (-1)^n)^T.
\end{aligned}$$

Similarly to [14], we took $a = 1.03$, $b = 1.008$ and $s = 1.001$. Table 1 displays mrr and arr given by M1, M1i, V, FH1, FH1i and FH1q for various n. Table 2 shows time ratios for `lyap` in a part of the examples in Table 1. In Tables 1–3, the notation OM means that V failed because of out of memory.

Table 1
The radii mrr and arr in the examples in [28].

n	M1	M1i	V	FH1	FH1i	FH1q
	mrr	mrr	mrr	mrr	mrr	mrr
	arr	arr	arr	arr	arr	arr
50	2.2e−10	2.1e−13	2.5e−12	6.0e−8	3.6e−9	2.2e−16
	1.2e−12	1.2e−15	1.8e−14	1.8e−11	1.2e−12	1.6e−16
100	6.9e−9	2.2e−12	OM	1.1e−6	1.5e−8	2.2e−16
	3.8e−12	1.9e−15	OM	1.9e−10	3.4e−12	1.6e−16
200	1.2e−7	1.1e−11	OM	6.3e−6	1.9e−8	2.2e−16
	9.8e−12	1.4e−15	OM	2.7e−9	1.3e−11	1.6e−16
300	1.8e−5	5.6e−11	OM	5.9e−3	1.3e−5	2.2e−16
	1.1e−10	1.1e−15	OM	3.0e−8	4.9e−11	1.6e−16
400	5.6e−4	1.3e−10	OM	1.3e−1	8.1e−5	7.3e−16
	1.2e−9	9.7e−16	OM	3.1e−7	1.3e−10	1.6e−16
500	4.5e−3	1.5e−10	OM	2.4e−1	8.3e−5	1.9e−15
	1.5e−8	9.7e−16	OM	4.5e−6	8.7e−10	1.6e−16

Table 2
The time ratios for `lyap` in the examples in [28].

n	M1	M1i	V	FH1	FH1i	FH1q
200	1.9	3.3	OM	5.0	7.0	12.0
300	2.0	3.8	OM	3.4	4.7	13.8
400	1.9	3.6	OM	2.9	3.9	12.4
500	1.8	3.5	OM	2.9	3.7	12.4

Table 3
The radii mrr and arr in the examples in [29].

Example	M1	M1i	V	FH1i	FH1q
number	mrr	mrr	mrr	mrr	mrr
in [29], m	arr	arr	arr	arr	arr
1.3, 4	9.8e–1	2.6e–1	1.0e+0	1.0e+0	1.0e+0
	5.2e–12	5.3e–14	1.2e–13	1.4e–12	1.3e–14
1.10, 8	9.8e–1	1.4e–1	1.0e+0	1.0e+0	1.0e+0
	1.5e–12	9.5e–15	8.7e–14	1.8e–13	4.6e–15
1.8, 9	6.3e–10	7.7e–14	8.2e–12	2.7e–13	2.2e–16
	1.8e–12	2.5e–15	7.0e–14	1.4e–14	1.6e–16
1.6, 30	1.0e+0	3.2e–1	1.0e+0	1.0e+0	1.0e+0
	6.5e–11	4.7e–14	9.8e–14	4.1e–13	1.3e–15
3.2, 40	7.8e–11	1.2e–13	4.1e–13	2.1e–12	2.2e–16
	9.4e–12	5.6e–15	1.7e–13	1.0e–13	1.5e–16
1.9, 55	1.0e+0	1.0e+0	1.0e+0	1.0e+0	1.0e+0
	2.8e–10	6.4e–12	1.6e–11	7.1e–11	1.7e–13
3.4, 421	1.0e+0	1.0e+0	OM	1.0e+0	6.4e–1
	7.9e–9	2.8e–11	OM	1.9e–9	1.7e–16

Table 4
The time ratios for V in the examples in [29].

m	M1	M1i	V	FH1i	FH1q
30	3.6e–3	5.2e–3	1	2.1e–2	2.1e–2
40	9.0e–4	1.3e–3	1	5.7e–3	5.7e–3
55	3.0e–4	4.4e–4	1	4.5e–4	4.5e–4

It can be seen from Table 1 that M1 and M1i gave smaller radii than FH1 and FH1i, respectively. When $n = 50$, M1 and M1i supplied larger and smaller radii than that by V, respectively. On the other hand, FH1q yielded the smallest radii of all. We can confirm from Table 2 that M1 was fastest of all and the required computing time was approximately twice as much as that of `lyap`. The algorithm M1i was slower than FH1 in many cases and faster than FH1i. The algorithm FH1q was slowest of all.

5.2. Results for the CTDSX benchmark collection

In this subsection, we report the obtained radii and time ratios for the examples in the CTDSX benchmark collection [29]. We took these examples which have the form

$$\dot{x}(t) = \mathcal{A}x(t) + \mathcal{B}u(t)$$

$$y(t) = \mathcal{C}x(t) + \mathcal{D}u(t),$$

where $\mathcal{A} \in \mathbb{R}^{m\times m}$ and $\mathcal{B} \in \mathbb{R}^{m\times n}$. From $\mathcal{A}$ and $\mathcal{B}$, we built the matrices $A = \mathcal{A}$, $B = A^T$ and $C = \mathrm{fl}(-\mathcal{B}\mathcal{B}^T)$ to formulate the Lyapunov equation

$$AX + XA^T = C. \tag{16}$$

Table 3 displays mrr and arr given by M1, M1i, V, FH1i and FH1q for various examples. The first column in Table 3 shows the example numbers in [29] and m. Table 4 shows time ratios for V in a part of the examples in Table 3.

We can confirm from Table 3 that mrr obtained by M1i were smaller than those by FH1q when example numbers were 1.3, 1.10 and 1.6. Moreover mrr by M1 were smaller than those by V when example numbers were 1.3 and 1.10. We see similar tendencies to Section 5.1 with respect to mrr and

Table 5
The radii mrr and arr for the example in [30].

Ex. no. in [30], *m*	M1	M1i	V	FH1i	FH1q
parameters	mrr	mrr	mrr	mrr	mrr
	arr	arr	arr	arr	arr
4.1, 10	2.8e−5	7.1e−10	3.1e−7	8.8e−8	2.2e−16
r =1.2, s =3.0	8.3e−6	3.0e−10	4.6e−8	2.6e−8	1.5e−16
4.1, 15	1.0e+0	1.5e−6	NaN	5.9e−4	1.8e−13
r =2.3, s =2.5	1.0e+0	5.2e−7	NaN	1.9e−4	5.9e−14
4.1, 50	9.6e−5	1.6e−11	3.0e−6	1.0e−8	2.2e−16
r =1.3, s =1.1	2.2e−8	8.6e−15	1.2e−9	2.3e−11	1.6e−16
4.2, 31	fail1	fail1	1.4e−9	1.0e+0	1.0e+0
s =1.2, λ =-1.1	fail1	fail1	2.2e−12	1.0e+0	1.0e+0
4.2, 25	fail2	fail2	5.1e−4	1.0e+0	1.0e+0
s =1.6, λ =-1.1	fail2	fail2	4.1e−5	1.0e+0	1.0e+0
4.2, 20	fail2	fail2	NaN	1.0e+0	9.3e−1
s =2.09, λ =-1.1	fail2	fail2	NaN	1.0e+0	1.7e−1

m	M2	M2i	FH2	FH2i	FH2q
	mrr	mrr	mrr	mrr	mrr
	arr	arr	arr	arr	arr
31	6.0e−8	2.5e−11	8.2e−6	1.3e−6	2.2e−16
	4.0e−10	1.5e−13	3.4e−8	5.3e−9	1.5e−16
25	5.3e−3	3.5e−8	1.0e+0	1.8e−1	1.3e−13
	5.0e−4	3.3e−9	9.1e−1	9.1e−3	5.6e−15
20	fail3	fail3	1.0e+0	1.0e+0	3.9e−7
	fail3	fail3	1.0e+0	1.0e+0	6.6e−9

Table 6
The time ratios for V in the examples in [30].

m	M1	M1i	V	FH1i	FH1q
50	3.3e−4	5.3e−4	1	7.1e−4	1.4e−3
31	fail1	fail1	1	1.9e−2	5.6e−2
25	fail2	fail2	1	5.3e−2	5.3e−2
20	fail2	fail2	1	9.1e−2	9.1e−2

m	M2	M2i	FH2	FH2i	FH2q
31	1.7e−1	1.8e−1	3.6e+0	3.6e+0	3.6e+0
25	2.3e−1	2.3e−1	5.5e+0	5.5e+0	5.6e+0
20	fail3	fail3	5.2e+0	5.2e+0	5.2e+0

arr except these results. Similarly to Section 5.1, Table 4 shows that M1 and M1i were faster than FH1i and FH1q. Although V did not cause out of memory when $m = 30$, 40 and 55, this algorithm was slowest of all.

5.3. Results for the CTLEX benchmark collection

In this subsection, we report the obtained radii and time ratios for the examples in the CTLEX benchmark collection [30]. These examples directly give the matrices A and C in (16). We computed such A and C via floating point operations in rounding-to-nearest mode. Table 5 displays mrr and arr obtained by M1, M1i, V, FH1i, FH1q, M2, M2i, V, FH2, FH2i and FH2q for various examples. The first column in the above part of Table 5 shows the example numbers in [30], m and parameters. Table 6 shows time ratios for V in a part of the examples in Table 5. In Tables 5 or 6, the notations fail1, fail2 and fail3 mean that we could not prove $\|S_A\|_\infty < 1$, $\|T_D\|_{\mathrm{M}} < 1$, and $\|\mathrm{f}_D\|_\infty < 1$ and $\|T_D^b\|_{\mathrm{M}} < 1$, respectively, so that M1, M1i, M2 or M2i failed.

When $m = 15$, although M1 completed the execution, the result made no sense, since arr = 1.0. The algorithm V failed in this example. When $m = 31$ and 25, the results by FH1i and FH1q made no sense, and M1 and M1i failed. When $m = 20$, the result by FH1i made no sense, and M1, M1i and V failed. As described in [14], the reason why many algorithms did not succeed when $m = 31$, 25 and 20 is that

$\tilde{V}_A$ is very ill-conditioned. Since we asked for a condition number in $\tilde{V}_A^b$ of $1/\sqrt{\epsilon}$, M2, M2i, FH2, FH2i and FH2q succeeded even when $m = 31$ and 25. In these cases, M2 and M2i gave smaller radii than those by FH2 and FH2i. On the other hand, FH2q gave the smallest radii of all. When $m = 20$, FH2q succeeded, although M2 and M2i failed and FH2 and FH2i did not give meaningful results. From this it can be seen that FH2q is robustest of all. From the case when $m = 50$ in Table 6, we can confirm similar tendencies to Section 5.2 regarding to the computing times of M1, M1i, V, FH1i and FH1q. When $m =$ 31 and 25, M2 and M2i were much faster than FH2, FH2i and FH2q. In these cases, the computing times of M2 and M2i were approximately equal. The reason is that computation of F requires the largest cost in both of M2 and M2i, and the other parts, including the iterative refinement and the enclosure of R with the extended precision, did not remarkably influence the overall computing time.

6. Conclusion

In this paper, we proposed algorithms for enclosing solutions in (1). For developing these algorithms, we presented Theorems 1–4 and introduced techniques for accelerating the enclosure and obtaining smaller error bounds. Some numerical results were reported to show the properties of these algorithms. As long as we compare the results obtained by M1 and M2 with FH1 and FH2 in Tables 1 and 5, respectively, it can be seen that the gains in quality of the enclosure come from the fact that interval arithmetic, as used in [14], produces worse error bounds than those obtained by the proposed algorithms. These comparisons are fair since these four algorithms do not use the extended precision computation. By modifying these algorithms slightly, enclosing X^* where A, B or C are interval matrices is also possible. Our future work will be to clarify the reason why X^{ε} and $X^{\varepsilon b}$ were not improved even when we set the numbers of iterations in the iterative refinements as two or more.

Acknowledgements

The author sincerely thanks Dr. Katsuhisa Ozaki in Shibaura Institute of Technology for fruitful discussions, and acknowledges the referees for valuable comments. This research was partially supported by Grant-in-Aid for Scientific Research (C) (23560066, 2011–2015) from the Ministry of Education, Science, Sports and Culture of Japan.

Appendix

In what follows we give the proof of Lemma 4. Lemmas 5 and 6 are firstly presented and referred, respectively. Lemma 7 is then established using Lemmas 5 and 6. The proof of Lemma 4 is finally given utilizing Lemma 7. Lemma 5 is a modification of [6, Lemma 3.5] such that it covers underflow.

Lemma 5. *For $x, y \in \mathbb{FC}$, floating point addition, subtraction, multiplication and division according to IEEE 754 satisfy*

$$\begin{aligned}
\mathrm{fl}(x \pm y) &= (x \pm y)(1 + \delta_1^{\pm}) = \frac{x \pm y}{1 + \delta_2^{\pm}}, \quad |\delta_1^{\pm}|, |\delta_2^{\pm}| \le \boldsymbol{u}, \\
\mathrm{fl}(xy) &= xy(1 + \delta^{\times}) + \eta^{\times}, \quad |\delta^{\times}| \le \sqrt{2}\gamma_2,\ |\eta^{\times}| \le 2\sqrt{2}\underline{\boldsymbol{u}}, \\
\mathrm{fl}(x\,/\,y) &= \frac{x}{y(1 + \delta^{/})} + \eta^{/}, \quad |\delta^{/}| \le \sqrt{2}\gamma_4,\ |\eta^{/}| \le 4\sqrt{2}\underline{\boldsymbol{u}},
\end{aligned}$$

also in the presence of underflow.

Remark 4. In [20, Lemma 3], $|\delta^{\times}|$ and $|\eta^{\times}|$ are bounded such that $|\delta^{\times}| \le \sqrt{5}\boldsymbol{u}$ and $|\eta^{\times}| \le 4\underline{\boldsymbol{u}}$, respectively. Thus Lemma 5 overestimates and improves the upper bounds for $|\delta^{\times}|$ and $|\eta^{\times}|$, respectively. In the proof of Lemma 7, on the other hand, there is no advantage even when we utilize $|\delta^{\times}| \le \sqrt{5}\boldsymbol{u}$ instead of $|\delta^{\times}| \le \sqrt{2}\gamma_2$. However the upper bound for $|S_A|s^{(m)}$, where S_A is defined as in Theorem 1, can be improved by utilizing $|\eta^{\times}| \le 2\sqrt{2}\underline{\boldsymbol{u}}$ instead of $|\eta^{\times}| \le 4\underline{\boldsymbol{u}}$. See Section 3 for details, where the improved bound is utilized.

Proof. As written in [19], for $a, b \in \mathbb{FR}$, the above operations satisfy

$$\mathrm{fl}(a \circ b) = (1+\delta)(a \circ b) + \eta, \quad \circ \in \{+, -, \times, /\}, \ |\delta| \le \boldsymbol{u}, \ |\eta| \le \underline{\boldsymbol{u}},$$

also in the presence of underflow. Especially $\eta = 0$ holds if $\circ \in \{+, -\}$. These facts and [6, Proof of Lemma 3.5] give $\mathrm{fl}(x \pm y) = (x \pm y)(1 + \delta_1^{\pm})$, $|\delta_1^{\pm}| \le \boldsymbol{u}$ including possible underflow. From [6, Proof of Lemma 3.5], moreover, we have

$$|\mathrm{fl}(x+y) - (x+y)| \le (|x+y|\boldsymbol{u})^2 \le \left(\frac{|x+y|\boldsymbol{u}}{1-\boldsymbol{u}}\right)^2,$$

which yields

$$\mathrm{fl}(x \pm y) = (x \pm y)\left(1 + \frac{\hat{\delta}_2^{\pm}}{1 - \hat{\delta}_2^{\pm}}\right) = \frac{x+y}{1-\hat{\delta}_2^{\pm}}, \quad |\hat{\delta}_2^{\pm}| \le \boldsymbol{u}.$$

The result $\mathrm{fl}(x \pm y) = (x \pm y)/(1 + \delta_2^{\pm})$, $|\delta_2^{\pm}| \le \boldsymbol{u}$ follows by putting $\delta_2^{\pm} = -\hat{\delta}_2^{\pm}$.

Let $x = a + bi$ and $y = c + di$, where $a, b, c, d \in \mathbb{FR}$ and $i = \sqrt{-1}$. For multiplication, analogously to [6, Proof of Lemma 3.5], we obtain

$$\mathrm{fl}(xy) = ac(1+\theta_2) - bd(1+\theta_2') + \eta_2 + i(ad(1+\theta_2'') + bc(1+\theta_2''') + \eta_2'),$$

where $|\theta_2|, |\theta_2'|, |\theta_2''|, |\theta_2'''| \le \gamma_2$ and $|\eta_2|, |\eta_2'| \le 2\underline{\boldsymbol{u}}$. Hence $\mathrm{fl}(xy) = xy + e^{\times}$, where

$$\begin{aligned} |e^{\times}|^2 &\le (\gamma_2(|ac| + |bd|) + 2\underline{\boldsymbol{u}})^2 + (\gamma_2(|ad| + |bc|) + 2\underline{\boldsymbol{u}})^2 \\ &= \gamma_2^2((|ac| + |bd|)^2 + (|ad| + |bc|)^2) + 4\gamma_2\underline{\boldsymbol{u}}(|ac| + |bd| + |ad| + |bc|) + 8\underline{\boldsymbol{u}}^2. \end{aligned} \tag{17}$$

It holds that

$$\begin{aligned} (|ac| + |bd|)^2 + (|ad| + |bc|)^2 &= a^2c^2 + b^2d^2 + a^2d^2 + b^2c^2 + 4|abcd| \\ &= (a^2 + b^2)(c^2 + d^2) + 4\sqrt{a^2b^2}\sqrt{c^2d^2} \\ &\le 2(a^2 + b^2)(c^2 + d^2), \end{aligned} \tag{18}$$

$$\begin{aligned} &|ac| + |bd| + |ad| + |bc| \\ &= \sqrt{(|ac| + |bd| + |ad| + |bc|)^2} \\ &= \sqrt{(a^2 + b^2)(c^2 + d^2) + 4|abcd| + 2((a^2 + b^2)|cd| + (c^2 + d^2)|ab|)} \\ &= \sqrt{(a^2 + b^2)(c^2 + d^2) + 4\sqrt{a^2b^2}\sqrt{c^2d^2} + 2((a^2 + b^2)\sqrt{c^2d^2} + (c^2 + d^2)\sqrt{a^2b^2})} \\ &\le 2\sqrt{(a^2 + b^2)(c^2 + d^2)}. \end{aligned} \tag{19}$$

The inequalities (17), (18) and (19) give

$$\begin{aligned} |e^{\times}|^2 &\le 2\gamma_2^2(a^2 + b^2)(c^2 + d^2) + 8\gamma_2\underline{\boldsymbol{u}}\sqrt{(a^2 + b^2)(c^2 + d^2)} + 8\underline{\boldsymbol{u}}^2 \\ &= 2\gamma_2^2|xy|^2 + 8\gamma_2\underline{\boldsymbol{u}}|xy| + 8\underline{\boldsymbol{u}}^2 = 2(\gamma_2|xy| + 2\underline{\boldsymbol{u}})^2, \end{aligned}$$

which shows the result with respect to the multiplication.

For division, analogously to [6, Proof of Lemma 3.5], $\mathrm{fl}(\mathrm{Re}\, x/y) = \mathrm{Re}\, x/y + e_r^{/}$ and $\mathrm{fl}(\mathrm{Im}\, x/y) = \mathrm{Im}\, x/y + e_c^{/}$, where

$$|e_r^{/}| \le \frac{\gamma_4(|ac| + |bd|)}{c^2 + d^2} + 4\underline{\boldsymbol{u}} \quad \text{and} \quad |e_c^{/}| \le \frac{\gamma_4(|bc| + |ad|)}{c^2 + d^2} + 4\underline{\boldsymbol{u}},$$

respectively. From (18), (19) and these inequalities, we have

$$\begin{aligned} |\mathrm{fl}(x/y) - x/y|^2 &\le \left(\frac{\gamma_4(|ac| + |bd|)}{c^2 + d^2} + 4\underline{\boldsymbol{u}}\right)^2 + \left(\frac{\gamma_4(|bc| + |ad|)}{c^2 + d^2} + 4\underline{\boldsymbol{u}}\right)^2 \\ &= \frac{\gamma_4^2((|ac| + |bd|)^2 + (|bc| + |ad|)^2)}{(c^2 + d^2)^2} + \frac{8\gamma_4\underline{\boldsymbol{u}}(|ac|+|bd| + |ad| + |bc|)}{c^2 + d^2} + 32\underline{\boldsymbol{u}}^2 \\ &\le \frac{2\gamma_4^2(a^2 + b^2)(c^2 + d^2)}{(c^2 + d^2)^2} + \frac{16\gamma_4\underline{\boldsymbol{u}}\sqrt{(a^2 + b^2)(c^2 + d^2)}}{c^2 + d^2} + 32\underline{\boldsymbol{u}}^2 \\ &= 2\gamma_4^2|x/y|^2 + 16\gamma_4\underline{\boldsymbol{u}}|x/y| + 32\underline{\boldsymbol{u}}^2 = 2(\gamma_4|x/y| + 4\underline{\boldsymbol{u}})^2, \end{aligned}$$

which gives

$$|\mathrm{fl}(x/y) - x/y| \le \sqrt{2}\gamma_4|x/y| + 4\sqrt{2}\underline{\boldsymbol{u}} \le \frac{\sqrt{2}\gamma_4|x/y|}{1 - \sqrt{2}\gamma_4} + 4\sqrt{2}\underline{\boldsymbol{u}}.$$

Thus we obtain

$$\mathrm{fl}(x/y) = (x/y)\left(1 + \frac{\hat{\delta}^{/}}{1 - \hat{\delta}^{/}}\right) + \eta^{/} = \frac{x}{y(1 - \hat{\delta}^{/})} + \eta^{/}, \quad |\hat{\delta}^{/}| \le \sqrt{2}\gamma_4, \ |\eta^{/}| \le 4\sqrt{2}\underline{\boldsymbol{u}}.$$

The result regarding to the division holds by setting $\delta^{/} = -\hat{\delta}^{/}$. □

Lemma 6 (E.g., Higham [6]). *If $|\delta_i| \le \boldsymbol{u}$ and $\rho_i = \pm 1$ for $i = 1, \dots, n$, and $n\boldsymbol{u} < 1$, then*

$$\prod_{i=1}^{n}(1 + \delta_i)^{\rho_i} = 1 + \theta_n, \text{ where } |\theta_n| \le \gamma_n.$$

Lemmas 5 and 6 yield the following lemma:

Lemma 7. *Let $a_1, \dots, a_{k-1}, b_1, \dots, b_{k-1}, c, d_1, d_2 \in \mathbb{F}\mathbb{C}$, $y = (c - \sum_{i=1}^{k-1} a_i b_i)/(d_1 + d_2)$ be evaluated in floating point arithmetic, and $\tilde{y}$ be the computed result. Then including possible underflow,*

$$\begin{aligned} &\left| c - \sum_{i=1}^{k-1} a_i b_i - (d_1 + d_2)\tilde{y} \right| \\ &\le \zeta_k \left(\sum_{i=1}^{k-1} |a_i b_i| + |(d_1 + d_2)\tilde{y}| \right) + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(k + 2(1 + \sqrt{2}\gamma_4)|d_1 + d_2|)}{1 - k\boldsymbol{u}}. \end{aligned}$$

Proof. We proceed as in [6] by first fixing the order of evaluation. Consider the following code:

```
$s = c$
for $i = 1 : k - 1$
  $s = s - a_i b_i$
end
$y = s/(d_1 + d_2)$
```

For the special case $k = 4$, repeated application of Lemma 5 yields for the computed value $\tilde{s}$

$$\begin{aligned}\tilde{s} = & (((c - a_1b_1(1+\delta_1^{\times}) - \eta_1^{\times})(1+\delta_1^{-}) - a_2b_2(1+\delta_2^{\times}) - \eta_2^{\times})(1+\delta_2^{-}) \\ & -a_3b_3(1+\delta_3^{\times}) - \eta_3^{\times})(1+\delta_3^{-}),\end{aligned}$$

where $|\delta_i^{-}| \le \boldsymbol{u}$, $|\delta_i^{\times}| \le \sqrt{2}\gamma_2$ and $|\eta_i^{\times}| \le 2\sqrt{2}\underline{\boldsymbol{u}}$ for $i = 1, 2, 3$. For general k, it follows that

$$\tilde{s} = c\prod_{i=1}^{k-1}(1+\delta_i^{-}) - \sum_{i=1}^{k-1}\left((a_ib_i(1+\delta_i^{\times}) + \eta_i^{\times})\prod_{j=i}^{k-1}(1+\delta_j^{-})\right). \tag{20}$$

From Lemma 5, we have

$$\tilde{y} = \mathrm{fl}\left(\frac{\tilde{s}}{d_1+d_2}\right) = \frac{\tilde{s}(1+\delta^{+})}{(d_1+d_2)(1+\delta^{/})} + \eta^{/},$$

where $|\delta^{+}| \le \boldsymbol{u}$, $|\delta^{/}| \le \sqrt{2}\gamma_4$ and $|\eta^{/}| \le 4\sqrt{2}\underline{\boldsymbol{u}}$, so that

$$\frac{(d_1+d_2)(1+\delta^{/})\tilde{y}}{1+\delta^{+}} = \tilde{s} + \frac{(d_1+d_2)(1+\delta^{/})\eta^{/}}{1+\delta^{+}}.$$

This and (20) give

$$\begin{aligned}\frac{(d_1+d_2)(1+\delta^{/})\tilde{y}}{1+\delta^{+}} = & \, c\prod_{i=1}^{k-1}(1+\delta_i^{-}) - \sum_{i=1}^{k-1}\left((a_ib_i(1+\delta_i^{\times}) + \eta_i^{\times})\prod_{j=i}^{k-1}(1+\delta_j^{-})\right) \\ & + \frac{(d_1+d_2)(1+\delta^{/})\eta^{/}}{1+\delta^{+}},\end{aligned}$$

that is,

$$\frac{(1+\delta^{/})(d_1+d_2)\tilde{y}}{(1+\delta^{+})\prod_{j=1}^{k-1}(1+\delta_j^{-})} = c - \sum_{i=1}^{k-1}\left(\frac{a_ib_i(1+\delta_i^{\times}) + \eta_i^{\times}}{\prod_{j=1}^{i-1}(1+\delta_j^{-})}\right) + \frac{(1+\delta^{/})(d_1+d_2)\eta^{/}}{(1+\delta^{+})\prod_{j=1}^{k-1}(1+\delta_j^{-})}.$$

From this and Lemma 6, we obtain

$$\begin{aligned}(1+\delta^{/})(1+\theta_k)(d_1+d_2)\tilde{y} = & \, c - \sum_{i=1}^{k-1}(a_ib_i(1+\delta_i^{\times}) + \eta_i^{\times})(1+\theta_{i-1}) \\ & + (1+\delta^{/})(1+\theta_k)(d_1+d_2)\eta^{/},\end{aligned}$$

where $\theta_0 = 0$ and $|\theta_j| \le \gamma_j$ for $j = 1, \dots, k$. This yields

$$\begin{aligned}& c - \sum_{i=1}^{k-1}a_ib_i - (d_1+d_2)\tilde{y} \\ & = \sum_{i=1}^{k-1}\left(a_ib_i(\delta_i^{\times} + \theta_{i-1} + \delta_i^{\times}\theta_{i-1}) + (1+\theta_{i-1})\eta_i^{\times}\right) \\ & \quad + (\delta_i^{/} + \theta_k + \delta_i^{/}\theta_k)(d_1+d_2)\tilde{y} - (1+\delta^{/})(1+\theta_k)(d_1+d_2)\eta^{/}.\end{aligned}$$

It follows from this and Lemma 5 that

$$
\begin{aligned}
&\left|c - \sum_{i=1}^{k-1} a_i b_i - (d_1 + d_2)\tilde{y}\right| \\
&\le \sum_{i=1}^{k-1}(|a_ib_i|(|\delta_i^\times| + |\theta_{i-1}| + |\delta_i^\times||\theta_{i-1}|) + (1 + |\theta_{i-1}|)|\eta_i^\times|) \\
&\quad +(|\delta_i^/| + |\theta_k| + |\delta_i^/||\theta_k|)|(d_1 + d_2)\tilde{y}| + (1 + |\delta^/|)(1 + |\theta_k|)|d_1 + d_2||\eta^/| \\
&\le \sum_{i=1}^{k-1}(|a_ib_i|(\sqrt{2}\gamma_2 + \gamma_{i-1} + \sqrt{2}\gamma_2\gamma_{i-1}) + 2\sqrt{2}\underline{\boldsymbol{u}}(1 + \gamma_{i-1})) \\
&\quad +(\sqrt{2}\gamma_4 + \gamma_k + \sqrt{2}\gamma_4\gamma_k)|(d_1 + d_2)\tilde{y}| + 4\sqrt{2}\underline{\boldsymbol{u}}(1 + \sqrt{2}\gamma_4)(1 + \gamma_k)|d_1 + d_2| \\
&\le \zeta_k \left(\sum_{i=1}^{k-1} |a_ib_i| + |(d_1 + d_2)\tilde{y}|\right) + 2\sqrt{2}\underline{\boldsymbol{u}}(1 + \gamma_k)(k + 2(1 + \sqrt{2}\gamma_4)|d_1 + d_2|).
\end{aligned}
$$

This and $(1 + \gamma_k) = 1/(1 - k\boldsymbol{u})$ prove the lemma. □

We give the proof of Lemma 4 utilizing Lemma 7.

Proof of Lemma 4. Assume without loss of generality that T is lower triangular. Let $\mu^{(k)}$ be the number of nonzero elements in $\mathsf{T}_{k:}$, and $\mathsf{T}_{k\omega_1}, \ldots, \mathsf{T}_{k\omega_{\mu^{(k)}-1}}$, $\mathsf{T}_{kk} = t_1^{(k)} + t_2^{(k)}$ be these nonzero elements. Then the k-th step of forward substitution is

$$
\tilde{\mathsf{x}}_k = \mathrm{fl}\left(\frac{1}{t_1^{(k)} + t_2^{(k)}}\left(\mathsf{b}_k - \sum_{i=1}^{\mu^{(k)}-1} \mathsf{T}_{k\omega_i}\tilde{\mathsf{x}}_i\right)\right).
$$

This and Lemma 7 give

$$
\begin{aligned}
\left|\mathsf{b}_k - \sum_{i=1}^{\mu^{(k)}-1} \mathsf{T}_{k\omega_i}\tilde{\mathsf{x}}_i - (t_1^{(k)} + t_2^{(k)})\tilde{\mathsf{x}}_k\right| &\le \zeta_{\mu^{(k)}}\left(\sum_{i=1}^{\mu^{(k)}-1} |\mathsf{T}_{k\omega_i}\tilde{\mathsf{x}}_i| + |(t_1^{(k)} + t_2^{(k)})\tilde{\mathsf{x}}_k|\right) \\
&\quad + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(\mu^{(k)} + 2(1 + \sqrt{2}\gamma_4)|t_1^{(k)} + t_2^{(k)}|)}{1 - \mu^{(k)}\boldsymbol{u}},
\end{aligned}
$$

that is,

$$
\begin{aligned}
|\mathsf{b}_k - (\mathsf{T}\tilde{\mathsf{x}})_k| &\le \zeta_{\mu^{(k)}}(|\mathsf{T}||\mathsf{x}|)_k + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(\mu^{(k)} + 2(1 + \sqrt{2}\gamma_4)|\mathsf{T}_{kk}|)}{1 - \mu^{(k)}\boldsymbol{u}}, \\
&\le \zeta_{\mu}(|\mathsf{T}||\mathsf{x}|)_k + \frac{2\sqrt{2}\underline{\boldsymbol{u}}(\mu + 2(1 + \sqrt{2}\gamma_4)|\mathsf{T}_{kk}|)}{1 - \mu\boldsymbol{u}},
\end{aligned}
$$

which proves the result. □

References

[1] B. Datta, Numerical Methods for Linear Control Systems, Elsevier Academic Press, Amsterdam, 2004.

[2] A. Antoulas, Approximation of Large-Scale Dynamical Systems, Advances in Design and Control, SIAM, Philadelphia, 2005.

[3] D. Sorensen, A. Antoulas, The Sylvester equation and approximate balanced reduction, Linear Algebra Appl. 351/352 (2002) 671–700.

[4] C. Choi, A. Laub, Efficient matrix-valued algorithm for solving stiff Riccati differential equations, IEEE Trans. Automat. Control 35 (1990) 770–776.

[5] D. Calvetti, L. Reichel, Application of ADI iterative methods to the restoration of noisy images, SIAM J. Matrix Anal. Appl. 17 (1996) 165–186.

[6] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM Publications, Philadelphia, 2002.

[7] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1994.
[8] R. Bartels, G. Stewart, Solution of the matrix equation $AX + XB = C$, Comm. ACM 15 (1972) 820–826.
[9] G. Golub, S. Nash, C. Van Loan, Hessenberg–Schur method for the problem $AX + XB = C$, IEEE Trans. Automat. Control AC-24 (1979) 909–913.
[10] N. Seif, S. Hussein, A. Deif, The interval Sylvester matrix equation, Computing 52 (1994) 233–244.
[11] V. Shashikhin, Robust assignment of poles in large-scale interval systems, Autom. Remote Control 63 (2002) 200–208.
[12] V. Shashikhin, Robust stabilization of linear interval systems, J. Appl. Math. Mech. 66 (2002) 393–400.
[13] J. Rohn, VERSOFT: Verification Software in MATLAB/INTLAB. Available from: <http://uivtx.cs.cas.cz/∼rohn/matlab>.
[14] A. Frommer, B. Hashemi, Verified error bounds for solutions of Sylvester matrix equations, Linear Algebra Appl. 436 (2012) 405–420.
[15] R. Krawczyk, Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, Computing 4 (1969) 187–201.
[16] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM Publications, Philadelphia, 2000.
[17] T. Yamamoto, Error bounds for approximate solutions of systems of equations, Japan J. Appl. Math. 1 (1984) 157–171.
[18] A. Bavely, G. Stewart, An algorithm for computing reducing subspaces by block diagonalization, SIAM J. Numer. Anal. 16 (1979) 359–367.
[19] S. Oishi, S.M. Rump, Fast verification of solutions of matrix equations, Numer. Math. 90 (4) (2002) 755–773.
[20] S. Miyajima, Fast enclosure for all eigenvalues in generalized eigenvalue problems, J. Comp. Appl. Math. 233 (2010) 2994–3004.
[21] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore and London, 1996.
[22] K. Ozaki, T. Ogita, S. Oishi, S.M. Rump, Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications, Numer. Algorithms 59 (2012) 95–118.
[23] T. Ogita, S.M. Rump, S. Oishi, Accurate sum and dot product, SIAM J. Sci. Comput. 26 (2005) 1955–1988.
[24] S.M. Rump, T. Ogita, S. Oishi, Accurate floating-point summation part I: faithful rounding, SIAM J. Sci. Comput. 31 (1) (2008) 189–224.
[25] S.M. Rump, T. Ogita, S. Oishi, Accurate floating-point summation part II: sign, K-fold faithful and rounding to nearest, SIAM J. Sci. Comput. 31 (2) (2008) 1269–1302.
[26] S.M. Rump, Ultimately fast accurate summation, SIAM J. Sci. Comput. 31 (5) (2009) 3466–3502.
[27] K. Ozaki, T. Ogita, S. Oishi, Tight and efficient enclosure of matrix multiplication by using optimized BLAS, Numer. Linear Algebra Appl. 18 (2011) 237–248.
[28] P. Benner, V. Sima, M. Slowik, Evaluation of the linear matrix equation solvers in SLICOT, J. Numer. Anal. Indust. Appl. Math. 2 (2007) 11–34.
[29] D. Kressner, V. Mehrmann, T. Penzl, CTDSX – a collection of benchmark examples for state-space realizations of continuous time dynamical systems, Tech. Rep. SLICOT Working Note 1998-9, 1998. Available from: <http://www.slicot.org/REPORTS/SLWN1998-9.ps.gz>.
[30] D. Kressner, V.Mehrmann, T. Penzl, CTLEX – a collection of benchmark examples for continuous-time Lyapunov equations, Tech. Rep. SLICOT Working Note 1999-6, 1999. Available from: <http://www.slicot.org/REPORTS/SLWN1999-6.ps.gz>.