TÜBİTAK

Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2018) 42: 1130 –
© TÜBİTAK
doi:10.3906/mat-1711-76

The united stable solution set of interval continuous-time algebraic Riccati equation and verified numerical computation of its outer estimation

Tayyebe HAQIRI[1,2,*], **Mahmoud MOHSENI MOGHADAM**[3], **Azim RIVAZ**[3]

[1]School of Mathematics and Computer Science, Damghan University, Damghan, Iran
[2]Member of Young Researchers Society of Shahid Bahonar University of Kerman, Kerman, Iran
[3]Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

Received: 19.11.2016 • **Accepted/Published Online:** 13.10.2017 • **Final Version:** 08.05.2018

Abstract: This paper introduces the interval continuous-time algebraic Riccati equation $\mathbf{A}^*X + X\mathbf{A} + \mathbf{Q} - X\mathbf{G}X = 0$, where $\mathbf{A}, \mathbf{G}$, and $\mathbf{Q}$ are known $n \times n$ complex interval matrices, $\mathbf{G}$ and $\mathbf{Q}$ are Hermitian, and X is an unknown matrix of the same size, and develops two approaches for enclosing the united stable solution set of this interval equation. We first discuss the united stable solution set and then derive a nonlinear programming method in order to find an enclosure for the united stable solution set. We also advance an efficient technique for enclosing the united stable solution set based on a variant of the Krawczyk method together with some modifications. These modifications enable us to reduce the computational complexity significantly. Various numerical experiments established upon a number of standard benchmark examples are also given to show the efficiency of this modified Krawczyk technique.

Key words: Interval continuous-time algebraic Riccati equation, united stable solution set, Krawczyk's method, verified computation, interval analysis, preconditioning

1. Introduction

The continuous-time algebraic Riccati equation ($CARE$)

$$A^*X + XA + Q - XGX = 0; \quad A, G, Q \in \mathbb{C}^{n\times n}, G^* = G, Q^* = Q, \tag{1.1}$$

plays a fundamental role in many areas such as control theory, filter design, model reduction, differential equations, and robust control [5, 6, 11, 19, 20, 31], where X is an unknown matrix and A^* denotes the conjugate transpose of the matrix A. Here we mention one of the relevant examples that arises in the theory of automatic control and linear filtering, i.e. *linear-quadratic optimal control problem* or LQ, in brief. This is a typical problem where CAREs are involved [5]:

Consider the differential equation

$$x'(t) = Ax(t) + Fu(t), \tag{1.2}$$

with $A \in \mathbb{C}^{n\times n}$ and $F \in \mathbb{C}^{n\times m}$, where $x(t) \in \mathbb{C}^n$ is the state vector and $u(t) \in \mathbb{C}^m$ is the control input vector. The differential equation (1.2) defines a continuous-time linear dynamical system and a classic problem is to

*Correspondence: haqiri@math.uk.ac.ir

2010 *AMS Mathematics Subject Classification:* 65G40, 65F30

compute an optimal feedback control

$$u(t) = Gx(t),$$

for $G \in \mathbb{C}^{m\times n}$. This optimal feedback control minimizes

$$J(u) = \int_0^{+\infty} |x^*(t)Cx(t) + u^*(t)Ru(t)|dt,$$

where $C \in \mathbb{C}^{n\times n}$ is Hermitian positive semidefinite and $R \in \mathbb{C}^{m\times m}$ is Hermitian positive definite. Under suitable assumptions on A, F, and C, the CARE

$$C + XA + A^*X - XFR^{-1}F^*X = 0$$

has a unique Hermitian positive semidefinite stabilizing solution and the desired optimal feedback control is $u(t) = -R^{-1}F^*Xx(t)$.

Recall that a Hermitian solution X_s (X_a) of CARE (1.1) is called *stabilizing* (resp. *anti-stabilizing*) if all the eigenvalues of the closed loop matrix $A - GX_s$ (resp. $A - GX_a$) have negative (resp. positive) real parts or $A - GX_s$ are *Hurwitz stable* [5, 19]. In practice only these two solutions are needed in engineering applications. Moreover, the verification of an anti-stabilizing solution is completely analogous to the verification of a stabilizing solution; it is sufficient to switch the sign of A, G, and Q and everything works. A complete discussion of the theoretical properties and numerical algorithms for CAREs are given in [19] and [25] and the references therein.

To further highlight the importance of matrix equation (1.1), consider two common matrix equations included, namely, the *Lyapunov matrix equation* $A^*X + XA + Q = 0$ and the *matrix square root* $X^2 + Q = 0$. The Lyapunov matrix equation has a vital role in many areas particularly in studying stability, controllability, and observability in dynamical systems or in solving PDEs on tensorized domains [8]. The matrix square root plays key roles in, for example, the matrix sign function [15, Chapter 5], the definite generalized eigenvalue problem [15, Chapter 2], the polar decomposition [15, Chapters 2 and 8], and the geometric mean [15, Chapter 2].

The problem of computing verified solutions to CARE (or some particular types of it) has been addressed before in the literature; see for instance [13, 22]. Moreover, except when we are computing, nearly all measurements, experiments, and models of real life or physical phenomena contain uncertainty. Indeed, the elements of A, G, and Q in equation (1.1) almost always contain doubt. Thus, they would be represented in interval form to guarantee bounds on the set of possible result values. Thus, the following **interval continuous-time algebraic Riccati equation** (**ICARE**) should be solved:

$$\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X + \mathbf{Q} = 0, \tag{1.3}$$

where boldface letters $\mathbf{A}, \mathbf{G}$, and $\mathbf{Q}$ are known interval matrices and $\mathbf{A}^*$ is a matrix whose center and radius are the conjugate transpose of center and radius of $\mathbf{A}$, respectively. We also assume that the interval elements in all interval matrices are mutually independent and the matrices $\mathbf{G}$ and $\mathbf{Q}$ are Hermitian interval matrices. A *Hermitian* interval matrix is a square interval matrix such that both its center and radius matrices are Hermitian. This definition is not taken directly from another paper; it is not particularly complicated or innovative, but we do not know of a reference where it appears in that exact form. Moreover, note that this definition does *not* imply that each point matrix in a Hermitian interval matrix is Hermitian. Meanwhile, we shall call the point matrix $\text{mid}(\mathbf{A} - \mathbf{G}X), X \in \mathbb{C}^{n\times n}$, a *closed loop matrix associated to ICARE* (1.3).

As we know, a few works concerning the interval form of CARE (1.1) (or some special cases of it) have already been done; see e.g. [9, 12, 14, 28–30]. Seif et al. [28] propose different techniques for approximating an outer estimation for the united solution set of the interval Sylvester matrix equation $\mathbf{A}X + X\mathbf{B} = \mathbf{C}$ in which $\mathbf{A}$ and $\mathbf{B}$ are real interval matrices of size m and n, respectively. Most of these techniques have an exponential complexity in m and n. In [29, 30], Shashikhin uses an interval linear system correspondence to the interval Sylvester matrix equation to find an interval enclosure for the united solution set. Hansen and Walster [12, Chapter 8] describe a procedure for computing the roots of the one-dimensional quadratic equation

$$\mathbf{a}x^2 + \mathbf{b}x + \mathbf{c} = 0, \tag{1.4}$$

where $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ are real intervals and x is an unknown real number. The interval roots of (1.4) are the set of real roots x of the quadratic equations $ax^2 + bx + c = 0$, for all $a \in \mathbf{a}, b \in \mathbf{b}$ and $c \in \mathbf{c}$. Hashemi and Dehghan [14] propose a major modification of the Krawczyk operator with a significant reduction in computational complexity of obtaining an outer estimation of the united solution set to the interval Lyapunov matrix equation $\mathbf{A}X + X\mathbf{A}^T = \mathbf{Q}$, which is applicable when the point matrix $\mathrm{mid}(\mathbf{A})$ is diagonalizable. Generalized solution sets of the interval generalized Sylvester matrix equation $\sum_{i=1}^{p} \mathbf{A}_i X_i + \sum_{j=1}^{q} Y_j \mathbf{B}_j = \mathbf{C}$ and several approaches for inner and outer estimations for some special cases of AE-solution sets are presented in [9].

We provide here a brief characterization of the united stable solution set of ICARE (1.3) and a nonlinear programming approach built on this characterization. Then we present a method that uses interval arithmetic to compute a certified enclosure for the united stable solution set of ICARE (1.3). This algorithm is based on the modified Krawczyk method used e.g. in [10, 13], and includes some improvements. In particular, we utilize two changes of bases:

- First change of basis is applied to original interval equation (1.3) expecting better performance due to a reduction in the spectral condition number of the eigenvector matrix of the midpoint closed loop matrix.
- The second improvement consists of a new change of basis that precedes again applying the Krawczyk method with the aim of reducing the wrapping effects. This technique was used before, e.g. in [10, 13].

There are also other improvements:

- An enclosure for the so-called *slope* matrix is needed where we are applying the Krawczyk method. The interval evaluation of the Jacobian of the function at hand is the typical choice to find this enclosure, but in order to result in a tighter interval matrix, we use a slightly different algebraic expression.
- To verify the stabilizing property of all solutions in the computed interval matrix, $\mathbf{X}$, we have used the method described in Algorithm 7 from [13] based on the method in [21] and in [22, Lemma 2.4]. In fact, this is an indirect strategy for proving uniqueness: if all the matrices inside the interval matrix $\mathbf{A} - \mathbf{GX}$ are Hurwitz stable then it is automatically verified that the interval matrix $\mathbf{X}$ contains the only one stabilizing solution X_s of each CARE in the united stable solution set [5, Corollary 3.10].

One can see then that the resulting modified Krawczyk method has several steps in common with the Krawczyk method described in [13]. This time, though, the improvements are used skillfully to establish a relation between "verified stabilizing solution of continuous-time algebraic Riccati equation" and "verified

outer estimation of the united *stable* solution set of *interval* continuous-time algebraic Riccati equation". Indeed, Haqiri and Poloni [13] develop algorithms that use interval arithmetic, combined those with the latest techniques on stabilization, resulting in a method to compute a verified enclosure for the stabilizing solution of a continuous-time algebraic Riccati equation that is competitive to the floating point ones.

This paper is structured as follows. After introducing some symbols and notation in Section 2, we define and partially characterize the united stable solution set to the interval continuous-time algebraic Riccati equation (1.3) in Section 3. In Sections 3.1 and 3.2, we develop two approaches for finding outer estimations of the united stable solution set. We test the performance of our Krawczyk-type algorithm on a number of standard benchmark examples in Section 4 and present the conclusion in Section 5.

2. Notation and preliminary concepts

We use $\mathbb{K}$ to denote either of the fields of real, $\mathbb{R}$, or complex numbers, $\mathbb{C}$. With the notations $\mathbb{K}^n, \mathbb{K}^{n\times n}, \mathbb{IK}^n$, and $\mathbb{IK}^{n\times n}$, we denote, respectively, the space of n-dimensional vectors, the space of $n\times n$ matrices, the set of all n-dimensional interval vectors, and the set of all $n\times n$ interval matrices, all over $\mathbb{K}$. In the present paper, starting from the introduction, all interval quantities will be typeset in boldface whereas lower case will imply scalar quantities or vectors and upper case will denote matrices. Underscores and overscores will show lower bounds and upper bounds of interval quantities, correspondingly.

The *Kronecker product* of two matrices $A=(A_{ij})\in\mathbb{K}^{m\times n}$ and $B\in\mathbb{K}^{p\times q}$ denoted by $A\otimes B\in\mathbb{K}^{mp\times nq}$ is an $m\times n$ block matrix whose (i,j) block is the $p\times q$ matrix $[A_{ij}B]$. Moreover, $\overline{A}$ denotes the complex conjugate of A and when A is an invertible matrix, then $A^{-T}:=(A^T)^{-1}$ and $A^{-*}:=(A^*)^{-1}$. The *Hadamard division* of a matrix $A=(A_{ij})\in\mathbb{K}^{m\times n}$ by a matrix $B=(B_{ij})\in\mathbb{K}^{m\times n}$ denoted by $A./B$ results in an $m\times n$ matrix $C=(C_{ij})$ whose (i,j) element is given by $C_{ij}=A_{ij}/B_{ij}$ provided that $B_{ij}\neq 0$, for each $1\leq i\leq m$ and $1\leq j\leq n$. For a given diagonal matrix $D\in\mathbb{K}^{n\times n}$, $\mathrm{diag}(D)\in\mathbb{K}^n$ is the vector whose $(i,1)$ element is the $i-$th diagonal entry of D. Conversely, given a vector $d=(d_1,d_2,\ldots,d_n)^T\in\mathbb{K}^n$, $\mathrm{Diag}(d)\in\mathbb{K}^{n\times n}$ is the diagonal matrix whose (i,i) entry is d_i. Moreover, vec is the operator that stacks the columns of a matrix vertically from first to last. I_N is also the identity matrix of size N. Furthermore, most of these notions and operations are analogously defined for interval quantities [17, 23].

We use here the definition of complex intervals as discs: a complex interval $\mathbf{x}$ is a closed circular disc of radius $\mathrm{rad}(\mathbf{x})\in\mathbb{R}$ with $\mathrm{rad}(\mathbf{x})\geq 0$ and center $\mathrm{mid}(\mathbf{x})\in\mathbb{C}$. Indeed, it is defined as $\mathbf{x}:=\{z\in\mathbb{C}\colon |z-\mathrm{mid}(\mathbf{x})|\leq\mathrm{rad}(\mathbf{x})\}=\langle\mathrm{mid}(\mathbf{x}),\mathrm{rad}(\mathbf{x})\rangle$. The operations on the circular complex intervals, $\mathbb{IC}$, are introduced as the generalizations of operations on complex numbers [2]. For *all* the four basic arithmetic operations $\circ\in\{+,-,\cdot,/\}$ one has

$$\mathbf{x}\circ\mathbf{y}\supseteq\{x\circ y : x\in\mathbf{x}, y\in\mathbf{y}\}, \tag{2.1}$$

in which $\mathbf{x},\mathbf{y}\in\mathbb{IC}$ (in the case of division, we need to assume that $0\notin\mathbf{y}$ for the operation to be well defined).

The *magnitude* of $\mathbf{x}\in\mathbb{IC}$ is defined as $\mathrm{mag}(\mathbf{x}):=\max\{|x| : x\in\mathbf{x}\}$. The *interval hull* $\square(\mathbf{x},\mathbf{y})$ of two intervals in $\mathbb{IC}$ is the interval of smallest radius containing $\mathbf{x}$ and $\mathbf{y}$. Moreover, $\mathrm{int}(\mathbf{X})$ is the topological interior of $\mathbf{X}\in\mathbb{IC}^{m\times n}$.

An $m\times n$ interval matrix $\mathbf{A}:=(\mathbf{A}_{ij})\in\mathbb{IC}^{m\times n}$ is denoted by $\mathbf{A}=\langle\mathrm{mid}(\mathbf{A}),\mathrm{rad}(\mathbf{A})\rangle$ in which $\mathbf{A}_{ij}=\langle\mathrm{mid}(\mathbf{A}_{ij}),\mathrm{rad}(\mathbf{A}_{ij})\rangle\in\mathbb{IC}$, with $\mathrm{rad}(\mathbf{A}_{ij})\geq 0$; $1\leq i\leq m, 1\leq j\leq n$. For interval vectors and matrices, $\mathrm{mid},\mathrm{rad},\mathrm{mag}$, and $\square$ will be applied component-wise. Particularly, if Σ is a bounded set of $m\times n$

real matrices, then we have $\square\Sigma := [\inf \Sigma, \sup \Sigma]$ [24].

If $\mathbf{D} = \mathrm{Diag}\Big((\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N)^T\Big)$ is a diagonal interval matrix and $0 \notin \mathbf{d}_i$ for each $i = 1, 2, \ldots, N$, then we may define

$$\mathbf{D}^{-1} := \mathrm{Diag}\Big((\frac{1}{\mathbf{d}_1}, \frac{1}{\mathbf{d}_2}, \ldots, \frac{1}{\mathbf{d}_N})^T\Big),$$

otherwise, the definition of inverse of an interval matrix may be problematic in general.

Different parts of the following lemmas, which also appear e.g. in [10] or [16], include some basic properties of the Kronecker product and the vec operator.

Lemma 2.1 *Assume that $A = (A_{ij})$, $B = (B_{ij})$, $C = (C_{ij})$, and $D = (D_{ij})$ be complex matrices with compatible sizes. Then,*

1. $(A \otimes B)(C \otimes D) = AC \otimes BD$,
2. $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$,
3. $(A \otimes B)^* = A^* \otimes B^*$,
4. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, *if A and B are invertible,*
5. $\mathrm{vec}(ABC) = (C^T \otimes A)\,\mathrm{vec}(B)$,
6. $\Big(\mathrm{Diag}(\mathrm{vec}(A))\Big)^{-1} \mathrm{vec}(B) = \mathrm{vec}(B./A)$, *if $A_{ij} \neq 0$ for each (i, j).*

Lemma 2.2 *Let $\mathbf{A} = (\mathbf{A}_{ij})$, $\mathbf{B} = (\mathbf{B}_{ij})$, and $\mathbf{C} = (\mathbf{C}_{ij})$ be complex interval matrices of compatible sizes. Then,*

1. $\Big\{(C^T \otimes A)\,\mathrm{vec}(B) \colon A \in \mathbf{A}, B \in \mathbf{B}, C \in \mathbf{C}\Big\} \subseteq \begin{cases} \mathrm{vec}\Big(\mathbf{A}(\mathbf{BC})\Big), \\ \mathrm{vec}\Big((\mathbf{AB})\mathbf{C}\Big), \end{cases}$
2. $\Big(\mathrm{Diag}(\mathrm{vec}\,(\mathbf{A}))\Big)^{-1} \mathrm{vec}(\mathbf{B}) = \mathrm{vec}(\mathbf{B}./\mathbf{A})$, *if $0 \notin \mathbf{A}_{ij}$ for all (i, j).*

The next lemma contains some information about the main properties of $\mathrm{mid}, \mathrm{rad}$, and $\square$.

Lemma 2.3 *(see e.g. [2, 24]) Let $\mathbf{A}$, $\mathbf{B} \in \mathbb{IC}^{n \times n}$ and X be a matrix with complex elements and compatible size. Then,*

1. $\mathbf{A} \subseteq \mathbf{B} \Leftrightarrow |\,\mathrm{mid}(\mathbf{B}) - \mathrm{mid}(\mathbf{A})| \leq \mathrm{rad}(\mathbf{B}) - \mathrm{rad}(\mathbf{A})$,
2. $\mathrm{mid}(\mathbf{A} \pm \mathbf{B}) = \mathrm{mid}(\mathbf{A}) \pm \mathrm{mid}(\mathbf{B})$,
3. $\mathrm{rad}(\mathbf{A} \pm \mathbf{B}) = \mathrm{rad}(\mathbf{A}) + \mathrm{rad}(\mathbf{B})$,
4. $\mathrm{mid}(\mathbf{A}X) = \mathrm{mid}(\mathbf{A})X$,
5. $\mathrm{mid}(X\mathbf{A}) = X\,\mathrm{mid}(\mathbf{A})$,

6. $\operatorname{rad}(\mathbf{A}X) = \operatorname{rad}(\mathbf{A})|X|$,

7. $\operatorname{rad}(X\mathbf{A}) = |X|\operatorname{rad}(\mathbf{A})$.

3. The united stable solution set of ICARE: partial characterization and estimation

The most common approach to define a solution set of an interval matrix equation is the one that we describe more exactly for ICARE (1.3) in this section: the united solution set defined as

$$\Sigma(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) :=$$
$$\{X \in \mathbb{K}^{n\times n} : ((\exists A \in \mathbf{A})(\exists G \in \mathbf{G})(\exists Q \in \mathbf{Q})(A^*X + XA + Q = XGX))\}.$$

This solution set is formed by all possible solutions of all point CAREs, $A^*X + XA + Q = XGX$ with $A \in \mathbf{A}, G \in \mathbf{G}$, and $Q \in \mathbf{Q}$. The united solution set has abundant applications in the so-called scientific computation field such as computational optimization, numerical simulations, and verification in system engineering [1]. On the other hand, the stabilizing solution is the one of interest in almost all applications [5, 11] and so we adjust the above definition to the united *stable* solution set as

$$\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) :=$$
$$\{X \in \mathbb{K}^{n\times n} : ((\exists A \in \mathbf{A})(\exists G \in \mathbf{G})(\exists Q \in \mathbf{Q})$$
$$(A^*X + XA + Q = XGX) \text{ and } A - GX \text{ is Hurwitz stable})\}.$$

$\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ contains (at most) one solution per CARE since the stabilizing solution is *unique* (when it exists) [13, Theorem 3.9].

According to the description provided above, we focus on the united stable solution set to ICARE.

The following theorem about the united stable solution set of ICARE (1.3) is very similar to results for the interval Lyapunov matrix equation [12] and the interval generalized Sylvester matrix equation [9].

Theorem 3.1

$$\Sigma_s(\textit{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) \subseteq \{X \in \mathbb{K}^{n\times n} : (\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X) \cap (-\mathbf{Q}) \neq \emptyset\}, \tag{3.1}$$

and

$$\Sigma_s(\textit{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) \subseteq \{X \in \mathbb{K}^{n\times n} : 0 \in \mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X + \mathbf{Q}\}, \tag{3.2}$$

where $0 \in \mathbb{R}^{n\times n}$. *Moreover,*

$$X \in \Sigma_s(\textit{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) \Rightarrow |\operatorname{mid}(\mathbf{A})^*X + X\operatorname{mid}(\mathbf{A}) - X\operatorname{mid}(\mathbf{G})X + \operatorname{mid}(\mathbf{Q})| \tag{3.3}$$
$$\leq \operatorname{rad}(\mathbf{A})^*|X| + |X|\operatorname{rad}(\mathbf{A}) + |X|\operatorname{rad}(\mathbf{G})|X| + \operatorname{rad}(\mathbf{Q}).$$

Proof Since $\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) \subseteq \Sigma(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$, we only need to show that relations (3.1), (3.2), and (3.3) are valid for $\Sigma(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. Now let $X \in \Sigma(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. Thus, $A^*X + XA - XGX = -Q$ for some $A \in \mathbf{A}, G \in \mathbf{G}$, and $Q \in \mathbf{Q}$; hence $-Q \in (\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X) \cap (-\mathbf{Q})$. Since $\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X \cap (-\mathbf{Q}) \neq \emptyset$, $0 \in \mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X + \mathbf{Q}$ and the first part of the theorem follows. For the second part, note that by the first part if $X \in \Sigma(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ then

$$X \in \{X \in \mathbb{K}^{n\times n} : (\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X) \cap (-\mathbf{Q}) \neq \emptyset\}.$$

Thus, $|\mathrm{mid}(\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X) - \mathrm{mid}(-\mathbf{Q})| \leq \mathrm{rad}(\mathbf{A}^*X + X\mathbf{A} - X\mathbf{G}X) + \mathrm{rad}(-\mathbf{Q})$. Now, by means of Lemma 2.3, we conclude that

$$\begin{aligned} &|\,\mathrm{mid}(\mathbf{A})^*X + X\,\mathrm{mid}(\mathbf{A}) - X\,\mathrm{mid}(\mathbf{G})X + \mathrm{mid}(\mathbf{Q})| \\ &\leq \mathrm{rad}(\mathbf{A})^*|X| + |X|\,\mathrm{rad}(\mathbf{A}) + |X|\,\mathrm{rad}(\mathbf{G})|X| + \mathrm{rad}(\mathbf{Q}). \end{aligned}$$

□

Because $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ is generally *not* an interval matrix, it is usually impractical to try to use it. Instead, it is common practice to seek an interval matrix containing $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$, for example, the interval hull of $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. Besides, only some estimation of the rigorous solution set suffices for factual goals in real life conditions. From this point on, let us suppose that $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ and $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ are nonempty and bounded.

3.1. Outer estimation of $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ via a nonlinear programming approach (for real data only)

Continuous-time algebraic Riccati equations frequently arise with real coefficient matrices and so it is natural to investigate. Such equations are the topic of this section and, of course, the preceding theorems in Section 3 apply but more structure can be expected in the solution set of a "real" CARE. The stabilizing solution of the real CARE

$$A^TX + XA + Q = XGX; \quad A, G, Q \in \mathbb{R}^{n\times n}, G^T = G, Q^T = Q,$$

is real [6]. Thus, if $\mathbf{A}, \mathbf{G}$, and $\mathbf{Q}$ belong to $\mathbb{IR}^{n\times n}$ and $\mathbf{G}$ and $\mathbf{Q}$ are symmetric, then every $X \in \Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ is also real. A square real interval matrix $\mathbf{A} := [\mathrm{mid}(\mathbf{A}) - \mathrm{rad}(\mathbf{A}), \mathrm{mid}(\mathbf{A}) + \mathrm{rad}(\mathbf{A})]$ is called *symmetric* if both $\mathrm{mid}(\mathbf{A})$ and $\mathrm{rad}(\mathbf{A})$ are symmetric. Thus, a symmetric interval matrix may also contain nonsymmetric matrices.

Afterwards, the inequality (3.3) appearing in Theorem 3.1 can provide us with an approach to discover an outer estimation for $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ (in real case) that itself is an enclosure for $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$.

By definition, $\square\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ is the tightest interval matrix that encloses $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. Thereupon, it could be supposed as the sharpest outer estimation for $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. Since we have assumed that the solution set $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ is *nonempty* and *bounded*, we can define its *exact* interval hull as

$$\square\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q}) := [\underline{X}, \overline{X}],$$

where for $i, j = 1 : n$,

$$\begin{cases} \underline{X} = (\underline{X}_{ij}), & \underline{X}_{ij} = \inf\{X_{ij} : X = (X_{ij}) \in \Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})\}, \\ \overline{X} = (\overline{X}_{ij}), & \overline{X}_{ij} = \sup\{X_{ij} : X = (X_{ij}) \in \Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})\}. \end{cases}$$

Now inequality (3.3) appearing in the last part of Theorem 3.1 turns out to be

$$\begin{cases} \mathrm{mid}(\mathbf{A})^*X + X\,\mathrm{mid}(\mathbf{A}) - X\,\mathrm{mid}(\mathbf{G})X - \mathrm{rad}(\mathbf{A})^*|X| - |X|\,\mathrm{rad}(\mathbf{A}) - |X|\,\mathrm{rad}(\mathbf{G})|X| \leq -\underline{Q}, \\ \mathrm{mid}(\mathbf{A})^*X + X\,\mathrm{mid}(\mathbf{A}) - X\,\mathrm{mid}(\mathbf{G})X + \mathrm{rad}(\mathbf{A})^*|X| + |X|\,\mathrm{rad}(\mathbf{A}) + |X|\,\mathrm{rad}(\mathbf{G})|X| \geq -\overline{Q}, \end{cases}$$

in which X is an arbitrary member of $\Sigma(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. If $S = (S_{ij})$ denotes the *sign matrix* of X, then $|X| = S \odot X$, where $\odot$ denotes the so-called *Hadamard* or component-wise product. Now, in order to determine

the lower and upper bound for each element X_{ij} of $X = (X_{ij})$, the following nonlinear programming problems for all possible cases of S and fixed $(i,j), 1 \leq i,j \leq n$ should be solved:

$$\begin{array}{l} \min / \max \quad X_{ij} \\ \text{s.t.} \end{array}$$

$$\begin{cases} \operatorname{mid}(\mathbf{A})^* X + X \operatorname{mid}(\mathbf{A}) - X \operatorname{mid}(\mathbf{G}) X - \operatorname{rad}(\mathbf{A})^* (S \odot X) - (S \odot X) \operatorname{rad}(\mathbf{A}) - (S \odot X) \operatorname{rad}(\mathbf{G})(S \odot X) \leq -\underline{Q}, \\ \operatorname{mid}(\mathbf{A})^* X + X \operatorname{mid}(\mathbf{A}) - X \operatorname{mid}(\mathbf{G}) X + \operatorname{rad}(\mathbf{A})^* (S \odot X) + (S \odot X) \operatorname{rad}(\mathbf{A}) + (S \odot X) \operatorname{rad}(\mathbf{G})(S \odot X) \geq -\overline{Q}. \end{cases}$$

Therefore, we are required to solve $2n^2 \times 2^{n^2}$ nonlinear programming problems that may grow exponentially as the dimension n increases, making this algorithm cumbersome and time-consuming for matrices of high order. Fortunately, the next section exploits a method for obtaining an outer estimation for the united stable solution set with complex coefficients and without this computational issue.

3.2. Verified outer estimation of $\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$ via a modified Krawczyk algorithm

There exist several modifications of Krawczyk's algorithm; see e.g. [13, 15]. The main purpose of these modifications is to make the Krawczyk algorithm as efficient as possible via, for example, decreasing the wrapping effects or some heuristic attempts. The next sections include some of these approaches.

3.2.1. Preconditioning the original ICARE

Before using the modified Krawczyk operator for obtaining an outer estimation for $\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$, we want to propose a preconditioning technique provided that the matrix $\operatorname{mid}(\mathbf{A} - \mathbf{G}\check{X})$ is diagonalizable where $\check{X}$ is an accurate approximation to the stabilizing solution of the midpoint system associated to ICARE (1.3), i.e.

$$\operatorname{mid}(\mathbf{A})^* X + X \operatorname{mid}(\mathbf{A}) + \operatorname{mid}(\mathbf{Q}) - X \operatorname{mid}(\mathbf{G}) X = 0. \tag{3.4}$$

Thus, assume that there exist matrices V_1, W_1, and $\Lambda_1 \in \mathbb{C}^{n \times n}$ such that an approximate spectral decomposition of $\operatorname{mid}(\mathbf{A} - \mathbf{G}\check{X})$ is available as

$$\operatorname{mid}(\mathbf{A} - \mathbf{G}\check{X}) \approx V_1 \Lambda_1 W_1 \quad \text{with } V_1, W_1, \Lambda_1 \in \mathbb{C}^{n \times n}, \tag{3.5a}$$

$$\Lambda_1 = \operatorname{Diag}(\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1n}), \quad V_1 W_1 \approx I_n. \tag{3.5b}$$

Of course, $W_1 = V_1^{-1}$ in theory, but it will turn out to have this appended notation available due to the fact that the exact inverse of a matrix is often not attainable when we are computing in floating point arithmetic. A similar reason is valid for $\lambda_{1i}, i = 1, 2, \ldots, n$, computed numerically with a standard method such as MATLAB's `eig`.

Then the right preconditioner V_1 and the left preconditioner V_1^* will change the original equation (1.3) to the right-left preconditioned system

$$\mathbf{A}_c^* X_c + X_c \mathbf{A}_c + \mathbf{Q}_c = X_c \mathbf{G}_c X_c, \tag{3.6}$$

where

$$\mathbf{A}_c = V_1^{-1} \mathbf{A} V_1, \quad X_c = V_1^* X V_1, \quad \mathbf{Q}_c = V_1^* \mathbf{Q} V_1, \quad \mathbf{G}_c = V_1^{-1} \mathbf{G} V_1^{-*}, \tag{3.7}$$

assuming that V_1 and W_1 in (3.5) are nonsingular. Note that the subscript "c"does *not* mean the center matrix. Then the center and radius matrices of $\mathbf{G}_c$ and $\mathbf{Q}_c$ are easily seen to be Hermitian again

$$(\operatorname{mid}(\mathbf{G}_c))^* = (\operatorname{mid}(V_1^{-1}\mathbf{G}V_1^{-*}))^* = V_1^{-1}(\operatorname{mid}(\mathbf{G}))^*V_1^{-*}$$
$$V_1^{-1}\operatorname{mid}(\mathbf{G})V_1^{-*} = \operatorname{mid}(V_1^{-1}\mathbf{G}V_1^{-*}) = \operatorname{mid}(\mathbf{G}_c),$$

and

$$(\operatorname{rad}(\mathbf{G}_c))^* = (\operatorname{rad}(V_1^{-1}\mathbf{G}V_1^{-*}))^* = |V_1^{-1}|(\operatorname{rad}(\mathbf{G}))^*|V_1^{-*}|$$
$$|V_1^{-1}|\operatorname{rad}(\mathbf{G})|V_1^{-*}| = \operatorname{rad}(V_1^{-1}\mathbf{G}V_1^{-*}) = \operatorname{rad}(\mathbf{G}_c).$$

A similar argument reveals that $(\operatorname{mid}(\mathbf{Q}_c))^* = \operatorname{mid}(\mathbf{Q}_c)$ and $(\operatorname{rad}(\mathbf{Q}_c))^* = \operatorname{rad}(\mathbf{Q}_c)$.

Consequently, the matrix $\mathbf{A}_c - \mathbf{G}_c\check{X}_c$ has an approximate diagonal center matrix, since

$$\operatorname{mid}(\mathbf{A}_c - \mathbf{G}_c\check{X}_c) = \operatorname{mid}(\mathbf{A}_c) - \operatorname{mid}(\mathbf{G}_c\check{X}_c) =$$
$$\operatorname{mid}(V_1^{-1}\mathbf{A}V_1) - \operatorname{mid}(V_1^{-1}\mathbf{G}\check{X}V_1) = V_1^{-1}\operatorname{mid}(\mathbf{A} - \mathbf{G}\check{X})V_1 \approx \Lambda_1.$$

Hence, it is natural that we suppose that $\mathbf{A}_c - \mathbf{G}_c\check{X}_c$ is diagonalizable and

$$\operatorname{mid}(\mathbf{A}_c - \mathbf{G}_c\check{X}_c) \approx V_2\Lambda_2W_2 \quad \text{with } V_2, W_2, \Lambda_2 \in \mathbb{C}^{n\times n}, \tag{3.8a}$$
$$\Lambda_2 = \operatorname{Diag}(\lambda_{21}, \lambda_{22}, \ldots, \lambda_{2n}), \quad V_2W_2 \approx I_n. \tag{3.8b}$$

Afterwards, we can see that this fact is very useful in the simplification of the Krawczyk operator. However, V_1, W_1, and Λ_2 are computed numerically and therefore they fulfill (3.8) just approximately.

3.2.2. Applying the Krawczyk operator in a residual format

In this section, as well as in the next section, we actually deal with a variant of the Krawczyk method that is simply a mean value form evaluation of the modified Newton operator.

Let $H : \mathbb{C}^{N\times N} \longrightarrow \mathbb{C}^{N\times N}$ be a Fréchet differentiable matrix function. Recall that the *Fréchet derivative* of H at $X \in \mathbb{C}^{N\times N}$ [15] is the unique function $L_H(X,.)$ that is linear in its second argument and for all $E \in \mathbb{C}^{N\times N}$ satisfies

$$H(X+E) - H(X) - L_H(X,E) = O(\|E\|).$$

Associated with the Fréchet derivative, its *Kronecker form* is the unique matrix $K_H(X) \in \mathbb{C}^{N^2\times N^2}$ such that for any $E \in \mathbb{C}^{N\times N}$

$$\operatorname{vec}(L_H(X,E)) = K_H(X)\operatorname{vec}(E),$$

holds.

Now suppose that

$$F(X_c) := A_c^*X_c + X_cA_c + Q_c - X_cG_cX_c, \tag{3.9}$$

where A_c, G_c, and Q_c are arbitrary point matrices chosen from $\mathbf{A}_c, \mathbf{G}_c$, and $\mathbf{Q}_c$ in (3.7), respectively. Note that the matrix function F in (3.9) does *not* necessarily meet the condition that the closed loop matrix $A_c - G_cX_c$

is diagonalizable for every $X_c \in \mathbb{C}^{n\times n}$. Now, for applying the modified Krawczyk algorithm, we consider the vector form of (3.9), $f : \mathbb{C}^{n^2} \longrightarrow \mathbb{C}^{n^2}$ written as

$$f(x_c) := \text{vec}(A_c^* X_c + X_c A_c + Q_c - X_c G_c X_c), \qquad x_c := \text{vec}\,(X_c). \tag{3.10}$$

The Frećhet derivative of the function in (3.9) at $X_c \in \mathbb{C}^{n\times n}$ with $X_c^* = X_c$ is $L_F(X_c, E) = E(A_c - G_c X_c) + (A_c - G_c X_c)^* E$. In addition, from Lemma 2.1 it turns out that the Kronecker form of $L_F(X_c, E)$ is

$$K_F(X_c) = I_n \otimes (A_c - G_c X_c)^* + (A_c - G_c X_c)^T \otimes I_n, \tag{3.11}$$

as long as X_c is Hermitian. We now want to state an important result in all the modified Krawczyk-type algorithms, but first we need a definition.

Definition 3.2 *(see e.g. [13]) Suppose $h : \Psi \subseteq \mathbb{C}^N \to \mathbb{C}^N$ and $x, y \in \mathbb{C}^N$. Then the* slope *$S(h; x, y) : \Psi \times \Psi \to \mathbb{C}^{N\times N}$ is defined to be that mapping such that*

$$h(y) - h(x) = S(h; x, y)(y - x).$$

Theorem 3.3 *(see e.g. [10]) Assume that $h : \Psi \subset \mathbb{C}^N \to \mathbb{C}^N$ is continuous. Let $R \in \mathbb{C}^{N\times N}, \check{x} \in \Psi$, and $\mathbf{z} \in \mathbb{IC}^N$ be such that $\check{x}+\mathbf{z} \subset \Psi$. Moreover, assume that $\mathcal{S} \subset \mathbb{C}^{N\times N}$ is a set of matrices such that $S(h; \check{x}, x') \in \mathcal{S}$ for every $x' \in \check{x} + \mathbf{z} =: \mathbf{x}$. If*

$$\mathcal{K}_h(\check{x}, R, \mathbf{z}, \mathcal{S}) := \{-Rh(\check{x}) + (I_N - RS)z : S \in \mathcal{S}, z \in \mathbf{z}\} \subseteq \text{int}(\mathbf{z}),$$

then the function h has a zero x_* in $\check{x} + \mathcal{K}_h(\check{x}, R, \mathbf{z}, \mathcal{S}) \subseteq \mathbf{x}$. Moreover, if $S(h; y, y') \in \mathcal{S}$ for each $y, y' \in \mathbf{x}$, then x_* is the only zero of h contained in $\mathbf{x}$.

The Krawczyk operator [18] associated to f in (3.10) is given by

$$\mathbf{k}_f(\check{x}_c, R, \mathbf{z}, \mathbf{S}) := -Rf(\check{x}_c) + (I_N - R\mathbf{S})\mathbf{z}, \quad x_c := \text{vec}(X_c),$$

where $\mathbf{S}$ is an interval matrix containing all slopes $S(f; y_c, y_c')$ for $y_c, y_c' \in \mathbf{x}_c := \check{x}_c + \mathbf{z}$. Thus, Krawczyk's algorithm has this advantage that it does not involve the inversion of interval matrices.

Let a mapping $h : \Omega \subseteq \mathbb{C}^N \longrightarrow \mathbb{C}^N$ be given by a mathematical expression $h(x)$. As a result of (2.1), we have the following *inclusion property*: if $\mathbf{h}$ is the interval evaluation of h, then

$$h(\mathbf{x}) := \{h(x) : x \in \mathbf{x}\} \subseteq \mathbf{h}(\mathbf{x}). \tag{3.12}$$

Recall that if one replaces the variable $x \in \mathbb{C}^N$ in $h(x)$ by the interval vector $\mathbf{x} \in \mathbb{IC}^N$ and also each arithmetic operation in the formula with the corresponding interval operation then the *interval (arithmetic) evaluation* of $h(x)$ over $\mathbf{x}$, $\mathbf{h}(\mathbf{x})$, will be obtained [3]. We will utilize the same approach to define the interval evaluation of a matrix function as well. On the other hand, different equivalent formulas for $h(x)$ could give different interval evaluations. Indeed, the process of turning the customary arithmetic into interval arithmetic is not free of pitfalls; issues such as *interval dependency* and the *wrapping phenomenon* have to be considered carefully. We refer the reader to the review article [26] for a thorough introduction.

The following result that appeared in [13] shows that a variant of the interval arithmetic evaluation of the Kronecker form of the Fréchet derivative of F in (3.9) can be used to obtain an enclosure for the slope(s) in the modified Krawczyk method. Indeed, Theorem 3.3 ensures that there is a unique answer in the computed interval matrix $\mathbf{X}$ when $\mathbf{S}$ contains all slopes $S(f;y,y')$ for all $y,y' \in \mathbf{x}$ while if $\mathbf{S}$ contains only the slopes $S(f;\check{x},y)$ for all $y \in \mathbf{x}$, as a result, there will be an answer in $\mathbf{X}$, not necessarily unique.

Theorem 3.4 *[13, Theorem 3.8] Let f be as in (3.10), $\mathbf{X} \in \mathbb{IC}^{n\times n}$ be an interval matrix, and $\check{X}_c \in \mathbf{X}$ be Hermitian. Then the interval matrix*

$$I_n \otimes (A_c - G_c\check{X}_c)^* + (A_c - G_c\mathbf{X})^T \otimes I_n$$

contains all slopes $S(f;\check{x}_c,y_c) = I_n \otimes (A_c - G_c\check{X}_c)^* + (A_c - G_cY_c)^T \otimes I_n$ for each $Y_c \in \mathbf{X}$, where $\check{x}_c = \mathrm{vec}(\check{X}_c)$ and $y_c = \mathrm{vec}(Y_c)$.

Note one subtle point: we can write $A_c^* - \check{X}_cG_c = (A_c - G_c\check{X}_c)^*$ because $\check{X}_c$ is Hermitian. One could obtain the equivalent expression $I_n \otimes (A_c - G_c\mathbf{X})^* + (A_c - G_c\check{X}_c)^T \otimes I_n$ with a similar proof, but this form would require this surplus hypothesis that $\mathbf{X}$ is a *Hermitian* interval matrix.

For applying the modified Krawczyk operator, we also need to clarify the preconditioner matrix R. Furthermore, we assume the nonsingularity of V_2 and W_2. It follows from (3.8) and (3.11) that $K_F(\check{X}_c)$ can be factorized as

$$\begin{aligned} K_F(\check{X}_c) &= I_n \otimes (A_c - G_c\check{X}_c)^* + (A_c - G_c\check{X}_c)^T \otimes I_n \\ &= (V_2^{-T} \otimes W_2^*)(I_n \otimes (W_2(A_c - G_c\check{X}_c)W_2^{-1})^* \\ &+ (V_2^{-1}(A_c - G_c\check{X}_c)V_2)^T \otimes I_n)(V_2^T \otimes W_2^{-*}). \end{aligned} \tag{3.13}$$

If we have an accurate computed solution for (3.4), then we can expect that

$$W_2(A_c - G_c\check{X}_c)W_2^{-1} \approx \Lambda_2, \quad \text{and} \quad V_2^{-1}(A_c - G_c\check{X}_c)V_2 \approx \Lambda_2.$$

Hence, we can choose R as

$$R = (V_2^{-T} \otimes W_2^*)\Delta^{-1}(V_2^T \otimes W_2^{-*}) \approx (K_F(\check{X}_c))^{-1}, \quad \Delta := I_n \otimes \Lambda_2^* + \Lambda_2^T \otimes I_n,$$

provided that Δ is also invertible.

Obviously, A_c and G_c in (3.9) are not necessarily equal to $\mathrm{mid}(\mathbf{A}_c)$ and $\mathrm{mid}(\mathbf{G}_c)$ nor are the matrices V_2 and W_2 equal to V_1 and W_1 in (3.5).

As a result of the inclusion property of circular arithmetic (3.12), we can now compute two enclosures for two terms in each member of $\mathcal{K}_f(\check{x}_c,R,\mathbf{z},\mathcal{S})$, namely $\mathrm{vec}(\mathbf{L})$ for $l := -Rf(\check{x}_c)$ and $\mathrm{vec}(\mathbf{U})$ for $u := (I_{n^2} - RS)z$. More details are presented via formulas below

$$\begin{aligned} l &:= -Rf(\check{x}_c) \\ &= -(I_n \otimes W_2^*V_2^*)[I_n \otimes \Lambda_2^* + \Lambda_2^T \otimes I_n]^{-1}(I_n \otimes W_2^{-1}V_2^{-1})f(\check{x}_c), \end{aligned}$$

and

$$\begin{aligned} u :=& (I_{n^2} - RS)z = (I_{n^2} - (V_2^{-T} \otimes W_2^*)\Delta^{-1}(V_2^T \otimes W_2^{-*}) \\ & (I_n \otimes (A_c - G_cY_c)^* + (A_c - G_cY_c^{'})^T \otimes I_n))z \\ & = ((V_2^{-T} \otimes W_2^*)\Delta^{-1} \\ & (\Delta - I_n \otimes (W_2(A_c - G_cY_c)W_2^{-1})^* - (V_2^{-1}(A_c - G_cY_c^{'})V_2)^T \otimes I_n) \\ & (V_2^T \otimes W_2^{-*}))z. \end{aligned}$$

3.2.3. Reducing wrapping effects

When solving equations by interval methods, the major difficulty is the wrapping effect. As said in [13], the wrapping effect that is intrinsic to interval computations is completely due to the fact that the image of an interval quantity (vector) under a map is not an interval quantity (vector), and so there is an overestimation in enclosing the image with an interval quantity (vector); see [26] for more details.

The key insight here is that for the purpose of reducing wrapping effects it will be useful to have a new change of basis via an affine transformation. We should start again from our crucial assumption, i.e. the existence of eigenvalue decomposition (3.8) and define

$$\hat{f}(\hat{x}) := (V_2^T \otimes W_2^{-*}) f((V_2^{-T} \otimes W_2^*)\hat{x}). \tag{3.14}$$

We continue to assume that an accurate approximation of the stabilizing solution of (3.4), i.e. $\check{X}$, is available. It is obvious from (3.14) that if $\check{x}_c = \mathrm{vec}(\check{X}_c)$ is an approximate solution to $f(x) = 0$, then $\hat{\check{x}}_c := (V_2^T \otimes W_2^{-*})\check{x}_c$ is an approximate solution to $\hat{f}(\hat{x}) = 0$. Thus, we avoided computing any transformation from $\check{x}_c$ to $\hat{\check{x}}_c$. This point is important since we have often the stabilizing solution of (3.9) instead of the stabilizing solution of (3.14).

As a consequence of this refinement, a set of slopes for $\hat{f}$ can be defined as

$$\hat{\mathcal{S}} := \{S(\hat{f}; \hat{x}_c, \hat{y}_c), \hat{x}_c, \hat{y}_c \in \hat{\mathbf{x}} := \hat{\check{x}}_c + \hat{\mathbf{z}}\}.$$

The members of $\hat{\mathcal{S}}$ can be computed by defining $x_c = (V_2^{-T} \otimes W_2^*)\hat{x}_c$, $y_c = (V_2^{-T} \otimes W_2^*)\hat{y}_c$ as

$$\begin{aligned} S(\hat{f}; \hat{y}_c, \hat{y}_c')(\hat{y}_c - \hat{y}_c') &= \hat{f}(\hat{y}_c) - \hat{f}(\hat{y}_c') \\ &= (V_2^T \otimes W_2^{-*})(f(y_c) - f(y_c')) \\ &= (V_2^T \otimes W_2^{-*})S(f; y_c, y_c')(y_c - y_c') \\ &= (V_2^T \otimes W_2^{-*})S(f; y_c, y_c')(V_2^{-T} \otimes W_2^*)(\hat{y}_c - \hat{y}_c'). \end{aligned}$$

Thence,

$$S(\hat{f}; \hat{x}_c, \hat{y}_c) = (V_2^T \otimes W_2^{-*})S(f; x_c, y_c)(V_2^{-T} \otimes W_2^*).$$

Now we are ready to compute the superset

$$\mathbf{k}_{\hat{f}}(\hat{\check{x}}_c, \hat{R}, \hat{\mathbf{z}}, \hat{\mathbf{S}}) = -\hat{R}\hat{f}(\hat{\check{x}}_c) + (I_{n^2} - \hat{R}\hat{\mathbf{S}})\hat{\mathbf{z}}$$

for

$$\mathcal{K}_{\hat{f}}(\hat{\check{x}}_c, \hat{R}, \hat{\mathbf{z}}, \hat{\mathcal{S}}) := \{-\hat{R}\hat{f}(\hat{\check{x}}_c) + (I_{n^2} - \hat{R}S)\hat{z} : S \in \hat{\mathcal{S}},\ \hat{z} \in \hat{\mathbf{z}}\}. \tag{3.15}$$

Regarding

$$(I_n \otimes (W_2(A_c - G_c\check{X}_c)W_2^{-1})^* + (V_2^{-1}(A_c - G_c\check{X}_c)V_2)^T \otimes I_n) \approx I_n \otimes \Lambda_2^* + \Lambda_2^T \otimes I_n,$$

a natural choice for $\hat{R}$ is the diagonal matrix

$$\hat{R} = \Delta^{-1}, \quad \Delta = I_n \otimes \Lambda_2^* + \Lambda_2^T \otimes I_n.$$

Moreover, $\mathrm{vec}(\hat{\mathbf{Z}}) := \hat{\mathbf{z}}$,

$$\hat{\mathcal{S}} = \{S(\hat{f}; \hat{y}_c, \hat{y}'_c), \hat{y}_c, \hat{y}'_c \in \hat{\mathbf{x}} := (V_2^T \otimes W_2^{-*})\check{x}_c + \hat{\mathbf{z}}\},$$

and

$$\hat{\mathbf{S}} = I_n \otimes (W_2(A_c - G_c\check{\mathbf{X}}_c)W_2^{-1})^* + (V_2^{-1}(A_c - G_c\check{\mathbf{X}}_c)V_2)^T \otimes I_n.$$

Now we can observe that $\hat{\mathcal{S}} \subseteq \hat{\mathbf{S}}$. A detailed computation of the enclosure

$$\begin{aligned}
\mathbf{k}_{\hat{f}}(\hat{\check{x}}_c, \hat{R}, \hat{\mathbf{z}}, \hat{\mathbf{S}}) &= -\hat{R}\hat{f}(\hat{\check{x}}_c) + (I_{n^2} - \hat{R}\hat{\mathbf{S}})\hat{\mathbf{z}} \\
&= -\Delta^{-1}((V_2^T \otimes W_2^{-*})f(\check{x}_c) \\
&\quad - (\Delta - I_n \otimes (W_2(A_c - G_c\check{\mathbf{X}}_c)W_2^{-1})^* \\
&\quad - (V_2^{-1}(A_c - G_c\check{\mathbf{X}}_c)V_2)^T \otimes I_n)\hat{\mathbf{z}}),
\end{aligned}$$

for $\mathcal{K}_{\hat{f}}(\hat{\check{x}}_c, \hat{R}, \hat{\mathbf{z}}, \hat{\mathcal{S}})$ is displayed in Algorithm 1. More precisely, we begin by recalling the extension of the enclosure property of interval circular arithmetic (3.12) to interval matrix operations and then Lemmas 2.1 and 2.2 provide us

$$-\hat{R}\hat{f}(\hat{\check{x}}_c) = -\Delta^{-1}(V_2^T \otimes W_2^{-*})f(\check{x}_c) \in -\mathrm{vec}((\mathbf{I}_{W_2}^*\hat{\mathbf{F}}V_2)./D) = \hat{\mathbf{l}} := \mathrm{vec}(\hat{\mathbf{L}}).$$

Thus,

$$\{-\hat{R}\hat{f}(\hat{\check{x}}_c) : A_c \in \mathbf{A}_c, G_c \in \mathbf{G}_c, Q_c \in \mathbf{Q}_c\} \subseteq \mathrm{vec}(\hat{\mathbf{L}}).$$

Similarly for the set of $(I_{n^2} - \hat{R}S)\hat{z}$ in (3.15), we can write

$$\begin{aligned}
&\{((\Lambda_2^* - W_2^{-*}(A_c - G_c\check{X}_c)^*W_2^*) \otimes I_n)\hat{z} : A_c \in \mathbf{A}_c, G_c \in \mathbf{G}_c, \hat{z} \in \hat{\mathbf{z}}\} \subseteq \\
&\mathrm{vec}((\Lambda_2^* - (\mathbf{I}_{W_2}^*(\mathbf{A}_c - \mathbf{G}_c\check{\mathbf{X}}_\mathbf{c})^*W_2^*))W_2^*\hat{\mathbf{Z}}\mathbf{I}_{V_2}),
\end{aligned} \tag{3.16}$$

and

$$\begin{aligned}
&\{(I_n \otimes (\Lambda_2 - V_2^{-1}(A_c - G_c(\check{X}_c + Z))V_2))\hat{z} : A_c \in \mathbf{A}_c, G_c \in \mathbf{G}_c, \hat{z} \in \hat{\mathbf{z}}\} \subseteq \\
&\mathrm{vec}(W_2^*\hat{\mathbf{Z}}\mathbf{I}_{V_2}(\Lambda_2 - (\mathbf{I}_{V_2}(\mathbf{A}_c - \mathbf{G}_c(\check{\mathbf{X}}_\mathbf{c} + W_2^*\hat{\mathbf{Z}}\mathbf{I}_{V_2}))V_2))).
\end{aligned} \tag{3.17}$$

Relations (3.16) and (3.17) assert that

$$\{(I_{n^2} - \hat{R}S)\hat{z} : S \in \hat{\mathcal{S}}, \hat{z} \in \hat{\mathbf{z}}\} = \{\Delta^{-1}(((\Lambda_2^* - W_2^{-*}(A_c - G_c\check{X}_c)^* W_2^*) \otimes I_n$$

$$+ I_n \otimes (\Lambda_2 - V_2^{-1}(A_c - G_c(\check{X}_c + Z))V_2)))\hat{z} : A_c \in \mathbf{A}_c, G_c \in \mathbf{G}_c, \hat{z} \in \hat{\mathbf{z}}\} \subseteq$$

$$\hat{\mathbf{u}} := \mathrm{vec}(\hat{\mathbf{U}}).$$

Hence,

$$\{-\hat{R}\hat{f}(\hat{\check{x}}_c) + (I_{n^2} - \hat{R}S)\hat{z} : S \in \hat{\mathcal{S}}, \hat{z} \in \hat{\mathbf{z}}\} \subseteq \mathrm{vec}(\hat{\mathbf{K}}).$$

Remarks are in order to clarify some points:

- As any inclusion method based on interval arithmetic tools, the Krawczyk method starts with an interval vector that contains a solution of a given (system of) equation(s) and improves this inclusion, iteratively. More often, an including interval vector is not known and one tries to compute an interval vector containing a solution by some operator such as ε*-inflation* process, which is commonly utilized today and introduced in e.g. [26]. In Algorithm 1, we start from the interval evaluation of $F(\check{X})$ as the residual matrix $\mathbf{Z}_0 := \mathbf{F}(\check{\mathbf{X}})$, and proceed enlarging this interval with the ε-inflation technique.
- There are slightly different versions of the iterative strategy in the literature to find a suitable interval matrix. For example, one of them involves intersecting the intervals obtained in different steps [10] while `verifynlss` in Intlab has a variant that does not update the derivatives, but it has an intersection in it. Here we will apply the simplest approach, following e.g. [13, 26], since we have tentatively found it to have better results with respect to the spent time.
- Point 2 of Lemma 2.2 has been used to transform the multiplication $\Gamma^{-1}\,\mathrm{vec}(\mathbf{B})$ into $\mathbf{B}./C$, where $\mathbf{B}$ is an $N \times N$ interval matrix, Γ is a diagonal matrix, and $C := (\bar{\Gamma}_{ii} + \Gamma_{jj})$. This point will appear in Algorithm 1 Lines 10 and 18.
- We will terminate Algorithm 1 with an error whenever one of the matrices V_1, V_2, W_1, W_2, or Δ cannot be inverted in a certified way in interval arithmetic.
- With respect to the lack of an associated law for multiplication of interval matrices, we omit parentheses everywhere regarding that evaluation is done from left to right.

Furthermore, when the stabilizability check succeeds, it is guaranteed that the interval matrix $\mathbf{X}$ contains exactly one solution per CARE in $\Sigma_s(\mathtt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$, which is the stabilizing one. Therefore, it is not possible that an interval solution, which is guaranteed to contain the unique stabilizing solutions of CAREs, actually also contains some nonstabilizing solutions. Besides, the cost for this verification is $\mathcal{O}(n^3)$ floating point operations [13].

One example of software that provides a fast implementation of a reliable interval arithmetic is the MATLAB toolbox INTLAB [27]; older versions of INTLAB are freely available for noncommercial use. The default arithmetic for both real and complex intervals in INTLAB is the midpoint-radius arithmetic [26].

The computational complexity analysis of Algorithm 1 yields the next result.

Theorem 3.5 *Algorithm 1 requires at most $\mathcal{O}(n^3 s)$ arithmetic operations in which s is the number of required iterations by the* `for` *loop.*

Proof Computing $\check{X}$ in Line 1, computing the eigendecompositions in Lines 2 and 4, and also computing the interval matrices $\mathbf{I}_{V_2}$ and $\mathbf{I}_{W_2}$ need $\mathcal{O}(n^3)$ operations. Furthermore, all the other operations that only involve $n \times n$ matrices have again cost $\mathcal{O}(n^3)$, at most. □

Algorithm 1 Efficient computation of an interval matrix $\mathbf{X}$ containing the united stable solution of ICARE (1.3) utilizing a modified Krawczyk algorithm.

1: Compute an approximate stabilizing solution $\check{X}$ of CARE (3.4) {For instance, using ordered Schur method followed by one step of Newton refinement in simulated quadruple precision}
2: Compute approximations V_1, W_1, and Λ_1 for the eigendecomposition of $\operatorname{mid}(\mathbf{A} - \mathbf{G}\check{X})$ in floating point {For instance, using the MATLAB command `eig`}
3: Compute $\mathbf{A}_c, \mathbf{G}_c$, and $\mathbf{Q}_c$ satisfying (3.7)
4: Compute approximations V_2, W_2, and Λ_2 for the eigendecomposition of $\operatorname{mid}(\mathbf{A}_c - \mathbf{G}_c\check{X}_c)$ in floating point {For instance, using the MATLAB command `eig`}
5: Compute with floating point arithmetic $D := (D_{ij})$ such that $D_{ij} \approx (\bar{\Lambda}_2)_{ii} + (\Lambda_2)_{jj}$
6: Compute interval matrices $\mathbf{I}_{V_2}$ and $\mathbf{I}_{W_2}$ containing V_2^{-1} and W_2^{-1}, resp. {For instance, using `verifylss.m` from INTLAB.} If this fails, or if D has any zero elements, return **failure**
7: $\check{\mathbf{X}}_\mathbf{c} = \langle \check{X}_c, 0 \rangle$ {To ensure that all operations involving $\check{X}_c$ are *verified* ones}
8: $\mathbf{F} = \mathbf{Q}_c + \check{\mathbf{X}}_\mathbf{c}\mathbf{A}_c + (\mathbf{A}_c^* - \check{\mathbf{X}}_\mathbf{c}\mathbf{G}_\mathbf{c})\check{\mathbf{X}}_\mathbf{c}$ {Gathering $\check{\mathbf{X}}_\mathbf{c}$ in order to reduce the wrapping effects}
9: $\hat{\mathbf{F}} = \mathbf{I}_{W_2}^* \mathbf{F} V_2$
10: $\hat{\mathbf{L}} = -\hat{\mathbf{F}}./D$
11: $\hat{\mathbf{Z}} = \hat{\mathbf{L}}$
12: **for** $k = 1, 2, \ldots, k_{max}$ **do**
13: Set $\hat{\mathbf{Z}} = \square(0, \hat{\mathbf{Z}} \cdot \langle 1, 0.1 \rangle + \langle 0, \texttt{realmin} \rangle)$ {ε-inflation technique}
14: $\hat{\mathbf{M}} = W_2^* \hat{\mathbf{Z}} \mathbf{I}_{V_2}$
15: $\hat{\mathbf{N}} = \mathbf{I}_{W_2}^* (\mathbf{A}_c - \mathbf{G}_c\check{\mathbf{X}}_\mathbf{c})^* W_2^*$ {Subject to Theorem 3.4}
16: $\hat{\mathbf{O}} = \mathbf{I}_{V_2}(\mathbf{A}_c - \mathbf{G}_c(\check{\mathbf{X}}_\mathbf{c} + \hat{\mathbf{M}}))V_2$
17: $\hat{\mathbf{P}} = (\Lambda_2^* - \hat{\mathbf{N}})\hat{\mathbf{M}} + \hat{\mathbf{M}}(\Lambda_2 - \hat{\mathbf{O}})$
18: $\hat{\mathbf{U}} = \hat{\mathbf{P}}./D$
19: $\hat{\mathbf{K}} = \hat{\mathbf{L}} + \hat{\mathbf{U}}$
20: **if** $\hat{\mathbf{K}} \subset \operatorname{int}(\hat{\mathbf{Z}})$ {Successful inclusion} **then**
21: Return $\mathbf{X} = \mathbf{I}_{V_1}^*(\check{X}_c + W_2^* \hat{\mathbf{K}} \mathbf{I}_{V_2})\mathbf{I}_{V_1}$ {Back transformations due to (3.7) and (3.14)}
22: **end if**
23: $\hat{\mathbf{Z}} = \hat{\mathbf{K}}$
24: **end for**
25: Return **failure** {Maximum number of iterations reached}

4. Computational experiments

In this part, we test Algorithm 1 on a set of generated interval matrices. These interval matrices are built on a large set of standard benchmark problems for Riccati equations [4, 7] designed to be challenging for nonverified CARE solvers in machine arithmetic. The results are reported in Tables 1–9. Moreover, the `Test number` follows the order used in [7].

Table 1. Comparison among various perturbation parameters α in the fixed perturbation approach before and after preconditioning.

`Test` `number in [7]`	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time	itr	time	itr	time	itr	time	itr
		mr	stab	mr	stab	mr	stab	mr	stab
2	2	5.21e-02	1	6.76e-02	2	$\star$	$\star$	$\star$	$\star$
		3.98e-05	1	4.48e-03	1	NaN	$\star$	NaN	$\star$
		7.62e-02	1	6.93e-02	1	6.90e-02	1	1.98e-01	2
		5.33e-07	1	5.33e-05	1	5.33e-03	1	5.71e-01	-1
3	4	8.52e-02	1	8.52e-02	1	8.51e-02	1	$\star$	$\star$
		1.21e-06	1	1.21e-04	1	1.22e-02	1	NaN	$\star$
		1.09e-01	1	1.07e-01	1	1.07e-01	1	1.29e-01	2
		3.39e-07	1	3.39e-05	1	3.40e-03	1	3.68e-01	1
4	8	5.51e-02	1	5.51e-02	1	5.41e-02	1	1.14e-01	5
		5.20e-07	1	5.20e-05	1	5.22e-03	1	1.11e+00	1
		7.23e-02	1	7.29e-02	1	7.21e-02	1	1.02e-01	3
		3.15e-07	1	3.15e-05	1	3.16e-03	1	4.88e-01	1
5	9	5.55e-02	1	5.56e-02	1	5.55e-02	1	7.13e-02	2
		3.20e-07	1	3.20e-05	1	3.20e-03	1	3.38e-01	1
		7.38e-02	1	7.38e-02	1	7.29e-02	1	2.14e-01	2
		4.13e-06	1	4.13e-04	1	4.13e-02	1	4.75e+00	-1
6	30	2.24e-01	2	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
		1.13e-01	-1	NaN	$\star$	NaN	$\star$	NaN	$\star$
		2.90e-01	1	3.30e-01	2	$\star$	$\star$	$\star$	$\star$
		3.43e-02	-1	3.49e+00	-1	NaN	$\star$	NaN	$\star$
7	2	5.26e-02	1	5.24e-02	1	5.23e-02	1	5.20e-02	1
		5.64e-09	1	5.64e-07	1	5.64e-05	1	5.67e-03	1
		3.40e-02	1	3.37e-02	1	3.38e-02	1	3.44e-02	1
		9.93e-09	1	9.93e-07	1	9.93e-05	1	1.00e-02	1

To get interval coefficients, we randomly perturb the set above. Indeed, we have imagined those situations in which one wishes to find the stabilizing solution of a CARE while required data have been interrupted for some reason(s). To achieve this goal, we have considered two kinds of perturbation in the numerical experiments: *fixed* and *proportional*. In the "fixed" form, the perturbation radius matrix is a nonnegative matrix whose

Table 2. Comparison among various perturbation parameters α in the fixed perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time mr	itr stab	time mr	itr stab	time mr	itr stab	time mr	itr stab
9	2	5.22e-02	1	5.24e-02	1	$\star$	$\star$	$\star$	$\star$
		3.36e-02	1	3.40e+00	1	NaN	$\star$	NaN	$\star$
		3.44e-02	1	3.45e-02	1	9.79e-02	2	$\star$	$\star$
		2.83e-02	1	2.83e+00	1	3.09e+02	-1	NaN	$\star$
10	2	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
		NaN	$\star$	NaN	$\star$	NaN	$\star$	NaN	$\star$
		3.69e-02	1	3.67e-02	1	4.48e-02	2	$\star$	$\star$
		3.54e-02	-1	3.48e+00	-1	4.01e+02	-1	NaN	$\star$
11	2	7.45e-02	1	8.34e-02	1	8.34e-02	1	9.37e-02	2
		6.82e-08	1	6.82e-06	1	6.84e-04	1	8.74e-02	1
		4.75e-02	1	5.24e-02	1	5.29e-02	1	5.23e-02	1
		6.82e-08	1	6.82e-06	1	6.83e-04	1	7.27e-02	1
12	2	8.34e-02	1	8.35e-02	1	8.32e-02	1	8.47e-02	1
		5.93e-06	1	5.93e-04	1	5.93e-02	1	6.02e+00	-1
		4.76e-02	1	4.80e-02	1	4.74e-02	1	4.82e-02	1
		2.30e-06	1	2.30e-04	1	2.30e-02	1	2.30e+00	-1
14	2	5.33e-02	1	5.24e-02	1	5.20e-02	1	5.19e-02	1
		2.77e-08	1	2.77e-06	1	2.77e-04	1	2.88e-02	1
		3.42e-02	1	3.38e-02	1	3.37e-02	1	3.39e-02	1
		1.30e-08	1	1.30e-06	1	1.30e-04	1	1.31e-02	1
17	2	8.28e-02	1	8.30e-02	1	8.40e-02	1	8.33e-02	1
		1.28e-08	1	1.28e-06	1	1.28e-04	1	1.36e-02	1
		4.81e-02	1	4.74e-02	1	4.72e-02	1	5.22e-02	1
		1.27e-08	1	1.27e-06	1	1.27e-04	1	1.30e-02	1

entries are positive multiples of random matrix entries. The results of this approach are shown in Tables 1–4 below. For the "proportional" one, the perturbation radius matrix is a positive multiple of the corresponding mag. The results of the alternative approach are presented in Tables 5–9.

The suggested algorithm was tested in MATLAB 2013a with INTLAB v6 and run on a laptop with 2.00 GHz CPU and 1 GB of RAM. In addition, `realmin` in Algorithm 1 refers to the smallest positive normalized floating point number.

In all tables, we report the results of all experiments before and after preconditioning in two successive rows, respectively. We also use four values for the parameter α listed in the first row of all tables, in order to perturb matrices.

For simplicity sake, the center matrices are just those given in the benchmark [7] and their radius matrices for the fixed mode are achieved by the MATLAB commands

Table 3. Comparison among various perturbation parameters α in the fixed perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time mr	itr stab	time mr	itr stab	time mr	itr stab	time mr	itr stab
19	3	2.62e-02 5.29e-08	1 1	2.62e-02 5.29e-06	1 1	2.58e-02 5.30e-04	1 1	3.35e-02 6.26e-02	2 1
		3.45e-02 2.84e-08	1 1	3.41e-02 2.84e-06	1 1	3.43e-02 2.84e-04	1 1	3.42e-02 2.89e-02	1 1
20	3	3.35e-02 5.42e+10	2 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		4.15e-02 1.94e+10	2 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
21	4	4.29e-02 7.82e-05	1 1	4.29e-02 7.87e-03	1 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		5.45e-02 1.94e-05	1 1	5.40e-02 1.94e-03	1 1	5.70e-02 1.99e-01	1 -1	$\star$ NaN	$\star$ $\star$
22	4	4.31e-02 2.06e-04	1 -1	4.33e-02 2.79e-03	1 -1	6.35e-02 3.15e-01	2 -1	$\star$ NaN	$\star$ $\star$
		5.41e-02 3.43e-05	1 -1	5.41e-02 3.43e-03	1 -1	5.43e-02 3.55e-01	1 -1	$\star$ NaN	$\star$ $\star$
23	4	5.12e-02 5.44e-06	1 1	5.08e-02 5.47e-04	1 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		5.37e-02 3.65e-06	1 1	5.35e-02 3.65e-04	1 1	5.66e-02 3.67e-02	1 -1	$\star$ NaN	$\star$ $\star$
25	77	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		1.18e+00 1.40e-03	1 1	1.27e+00 1.51e-01	2 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$

```
For i=1:33
[A, G, Q] = ChuLM07Carex(i);n = length(Q);
reset(RandStream.getGlobalStream());RND = rand(n,n);
IA = midrad(A,alpha*RND);IG = midrad(G,alpha*RND);
IQ = midrad(Q,alpha*RND);end,
```

in which IA, IG, and IQ are equal to the interval matrices $\mathbf{A}, \mathbf{G}$, and $\mathbf{Q}$, respectively. In the proportional approach, we employ the mag of the center matrix instead of the random matrix RND.

The approximate solution of CARE associated with the left half plane for the midpoint system (3.4) (required in Line 1 of Algorithm 1) is obtained using the method described in [22] (ordered Schur method followed by one step of Newton refinement in simulated quadruple precision). You may change it easily to have the antistabilizing solutions instead of the stabilizing ones.

Table 4. Comparison among various perturbation parameters α in the fixed perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time mr	itr stab	time mr	itr stab	time mr	itr stab	time mr	itr stab
27	397	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		8.78e+01 8.51e-02	4 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
28	8	2.79e-02 2.33e-08	1 1	2.70e-02 2.33e-06	1 1	2.69e-02 2.34e-04	1 1	$\star$ NaN	$\star$ $\star$
		3.75e-02 1.70e-08	1 1	3.64e-02 3.27e-06	1 1	5.30e-02 7.08e-04	1 1	3.57e-02 2.67e-02	1 1
29	64	1.04e-01 1.40e-06	1 1	1.21e-01 1.64e-04	2 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		1.72e-01 1.84e-06	1 1	1.72e-01 1.89e-04	1 1	4.82e-01 2.62e-02	2 -1	$\star$ NaN	$\star$ $\star$
32	100	4.78e-01 2.39e-08	1 1	4.33e-01 2.40e-06	1 1	4.42e-01 2.87e-04	3 1	$\star$ NaN	$\star$ $\star$
		7.21e-01 2.39e-08	1 1	5.96e-01 2.39e-06	1 1	5.80e-01 2.43e-04	1 1	$\star$ NaN	$\star$ $\star$
33	60	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		4.93e-01 7.32e-03	1 1	5.95e-01 1.10e+00	3 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$

To show the quality of the obtained enclosure $\mathbf{X}$, we display the maximum radius `mr` of the entries of $\mathbf{X}$, that is

$$\texttt{mr} := \max \operatorname{rad}(\mathbf{X}_{ij}), \quad \mathbf{X} = (\mathbf{X}_{ij}).$$

Note that when enclosing $\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$, we want an enclosure as tight as possible, hence a small `mr`. This gives us more restricting information on the solution set $\Sigma_s(\texttt{CARE}; \mathbf{A}, \mathbf{G}, \mathbf{Q})$. If we get a large set, even if it is all composed of stabilizing matrices, it is just less specific.

As one can see, in most of the cases, the algorithm with preconditioning obtains smaller values of `mr`. However, in some cases this is not true. That is almost always the case with preconditioning: sometimes the problem becomes better, sometimes it becomes worse, and it is difficult to tell exactly why beforehand. As far as we know there is *not* much theory to deal with this.

In order to show the efficiency of our algorithm, when it is successful, the number of iterations executed in Algorithm 1 for the Krawczyk loop, itr, is also presented. In any problem, if the algorithm breaks down or does not converge within the maximum number of steps, i.e. $\text{itr}_{\max} = 10$, then we report NaN for `mr` and $\star$ for time and itr. When a test fails for all the alpha values, the entire row is removed. The size of any test (value of n) and the total time (in seconds) are reported too.

Table 5. Comparison among various perturbation parameters α in the proportional perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time mr	itr stab	time mr	itr stab	time mr	itr stab	time mr	itr stab
2	2	5.83e-02	1	6.70e-02	2	$\star$	$\star$	$\star$	$\star$
		7.54e-05	1	9.64e-03	1	NaN	$\star$	NaN	$\star$
		7.47e-02	1	6.93e-02	1	2.03e-01	1	5.17e-02	3
		2.12e-06	1	2.12e-04	1	2.13e-02	-1	2.75e+00	-1
3	4	8.55e-02	1	8.49e-02	1	8.53e-02	1	$\star$	$\star$
		7.50e-07	1	7.50e-05	1	7.55e-03	1	NaN	$\star$
		5.36e-02	1	5.36e-02	1	5.37e-02	1	5.38e-02	1
		3.50e-07	1	3.50e-05	1	3.50e-03	1	3.70e-01	1
4	8	5.46e-02	1	5.45e-02	1	5.43e-02	1	5.38e-02	1
		1.05e-07	1	1.05e-05	1	1.05e-03	1	1.10e-01	1
		3.60e-02	1	3.65e-02	1	3.60e-02	1	3.60e-02	1
		7.89e-08	1	7.89e-06	1	7.89e-04	1	8.26e-02	1
5	9	5.57e-02	1	5.56e-02	1	5.53e-02	1	2.28e-01	2
		9.11e-07	1	9.11e-05	1	9.12e-03	1	1.01e+00	-1
		3.67e-02	1	3.70e-02	1	3.68e-02	1	1.36e-01	2
		2.61e-06	1	2.61e-04	1	2.61e-02	1	2.81e+00	-1
6	30	2.30e-01	2	3.42e-01	4	$\star$	$\star$	$\star$	$\star$
		3.61e-03	-1	4.11e-01	-1	NaN	$\star$	NaN	$\star$
		2.54e-01	2	2.49e-01	2	4.51e-01	4	$\star$	$\star$
		1.06e-03	-1	1.06e-01	-1	1.13e+01	-1	NaN	$\star$
7	2	2.66e-02	1	2.60e-02	1	2.59e-02	1	2.64e-02	1
		5.37e-09	1	5.37e-07	1	5.37e-05	1	5.40e-03	1
		3.43e-02	1	3.39e-02	1	3.45e-02	1	3.46e-02	1
		8.28e-09	1	8.28e-07	1	8.28e-05	1	8.32e-03	1

When the algorithm is successful, we check afterwards whether $\mathbf{A} - \mathbf{GX}$ is Hurwitz stable using Algorithm 7 in [13]. A number 1 in the corresponding stab column confirms the stability property of all solutions contained in the result enclosure $\mathbf{X}$, number -1 means failure to verify the stabilizing property, and a star means that the algorithm had already failed to compute an inclusion interval. As one can see, there is an acceptable number of cases in which the stabilization procedure fails.

Although the algorithm is iterative, actually one step in most of the examples is sufficient to attain convergence. Thus, in practice the cost of the algorithm is cubic. Moreover, in all cases, the number of required iterations after preconditioning is not greater than the number of iterations needed before preconditioning. Meanwhile, there is no case where the original CARE (1.3) has a solution, but the preconditioned system (3.6) does not.

In addition to all these, the norm-2 condition numbers of V_1 and V_2 for $\alpha = 1e-3$ in the fixed and proportional cases are compared. As shown in Figures 1 and 2, most of the points lie below the axes bisector

Table 6. Comparison among various perturbation parameters α in the proportional perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time	itr	time	itr	time	itr	time	itr
		mr	stab	mr	stab	mr	stab	mr	stab
8	2	2.58e-02	1	2.56e-02	1	2.57e-02	1	2.54e-02	1
		4.89e+03	1	4.01e+05	1	4.00e+07	1	4.02e+09	1
		3.39e-02	1	3.38e-02	1	3.38e-02	1	3.38e-02	1
		7.66e+04	1	4.73e+05	1	4.01e+07	1	4.03e+09	1
9	2	2.63e-02	1	2.60e-02	1	2.59e-02	1	$\star$	$\star$
		3.20e-05	1	3.20e-03	1	3.21e-01	1	NaN	$\star$
		3.45e-02	1	3.44e-02	1	3.44e-02	1	3.47e-02	1
		3.04e-05	1	3.04e-03	1	3.04e-01	1	3.07e+01	1
11	2	8.35e-02	1	8.32e-02	1	8.31e-02	1	8.34e-02	1
		3.22e-08	1	3.22e-06	1	3.22e-04	1	3.52e-02	1
		4.86e-02	1	4.78e-02	1	4.78e-02	1	5.26e-02	1
		3.22e-08	1	3.22e-06	1	3.22e-04	1	13.31e-02	1
12	2	8.29e-02	1	8.34e-02	1	8.38e-02	1	8.62e-02	1
		1.41e-05	1	1.41e-03	1	1.41e-01	1	1.46e+01	-1
		4.82e-02	1	4.76e-02	1	4.78e-02	1	4.91e-02	1
		4.83e-06	1	4.83e-04	1	4.83e-02	1	4.86e+00	-1
13	2	5.16e-02	1	1.66e-01	1	6.17e-02	1	6.12e-02	1
		6.02e-03	1	4.02e-01	-1	4.00e+01	-1	4.07e+03	-1
		3.43e-02	1	9.11e-02	1	4.21e-02	1	4.70e-02	1
		4.14e-03	1	4.00e-01	-1	4.00e+01	-1	4.06e+03	-1
14	2	5.19e-02	1	5.15e-02	1	5.14e-02	1	5.18e-02	1
		3.60e-08	1	3.60e-06	1	3.60e-04	1	3.79e-02	1
		4.05e-02	1	4.04e-02	1	4.06e-02	1	4.07e-02	1
		1.89e-08	1	1.89e-06	1	1.89e-04	1	1.91e-02	1

(drawn in red), which means that V_2 has generally a lower condition number than V_1. There are only one experiment in the proportional case (Test 28) and two experiments in the fixed case (Tests 28 and 29) for which $\text{cond}(V_2) > \text{cond}(V_1)$.

Considering the fact that there is no other algorithm that can be compared with the existing algorithm (as we know), the reader may be confused about the conservativeness of the enclosure obtained by Algorithm 1. On the other hand, the nonlinear programming approach mentioned in Section 3.1 is *not* a *verified* computation. In addition, it has been only devoted to real data. Therefore, comparing the results of these two methods is not allowed.

Table 7. Comparison among various perturbation parameters α in the proportional perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time	itr	time	itr	time	itr	time	itr
		mr	stab	mr	stab	mr	stab	mr	stab
15	2	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
		NaN	$\star$	NaN	$\star$	NaN	$\star$	NaN	$\star$
		4.04e-02	1	4.05e-02	1	$\star$	$\star$	$\star$	$\star$
		9.71e-09	1	9.82e-07	1	NaN	$\star$	NaN	$\star$
16	2	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
		NaN	$\star$	NaN	$\star$	NaN	$\star$	NaN	$\star$
		4.05e-02	1	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
		9.82e-09	1	NaN	$\star$	NaN	$\star$	NaN	$\star$
17	2	8.23e-02	1	8.26e-02	1	8.22e-02	1	1.04e-01	2
		3.24e-08	1	3.24e-06	1	3.25e-04	1	3.88e-02	1
		6.29e-02	1	5.72e-02	1	6.26e-02	1	5.67e-02	1
		3.51e-08	1	3.51e-06	1	3.51e-04	1	3.70e-02	1
19	3	5.20e-02	1	5.17e-02	1	5.21e-02	1	6.68e-02	2
		4.54e-08	1	4.54e-06	1	4.55e-04	1	5.22e-02	1
		3.45e-02	1	3.43e-02	1	3.44e-02	1	3.44e-02	1
		2.56e-08	1	2.56e-06	1	2.56e-04	1	2.60e-02	1
20	3	5.20e-02	1	5.19e-02	1	6.68e-02	2	5.47e-02	2
		4.26e+07	1	4.73e+07	1	5.71e+08	1	6.49e+10	1
		3.49e-02	1	4.19e-02	2	3.43e-02	1	3.42e-02	1
		4.85e+06	1	7.46e+06	1	2.68e+08	1	2.68e+10	1
21	4	4.30e-02	1	4.28e-02	1	4.29e-02	1	$\star$	$\star$
		2.00e-06	1	2.00e-04	1	2.04e-02	1	NaN	$\star$
		5.42e-02	1	5.40e-02	1	4.94e-02	1	6.46e-02	2
		1.11e-06	1	1.11e-04	1	1.11e-02	1	1.29e+00	-1

One thing we could do in order to resolve this ambiguity is choosing random matrices in each interval matrix, solving the resulting CARE in nonverified floating point arithmetics, repeat, for example, 10,000 times, and check what is the `mr` of the *interval hull* of all the 10,000 solutions we found. This test was carried out on those examples in which $n < 10$ for $\alpha = 1e-9$ or $\alpha = 1e-3$ and only when the results of verification of the stabilizing property are positive. The results of this attempt are presented in Figure 3. As expected, in all experiments, all points lie above or on the axes bisector and this means that at least for small dimensions and not-so-large radius of uncertainties, in general, the maximum radius of $\mathbf{X}$ obtained by Algorithm 1, `mr`, is greater than or equal to the maximum radii of the enclosure obtained by the random approach, $\texttt{mr}_r$. Actually, the gap between these two values is negligible.

Table 8. Comparison among various perturbation parameters α in the proportional perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time mr	itr stab	time mr	itr stab	time mr	itr stab	time mr	itr stab
22	4	4.35e-02 1.81e-04	1 -1	4.35e-02 3.35e-04	1 -1	4.26e-02 1.60e-02	1 -1	$\star$ NaN	$\star$ $\star$
		6.17e-02 2.83e-06	1 -1	6.50e-02 2.83e-04	1 -1	6.53e-02 2.85e-02	1 -1	$\star$ NaN	$\star$ $\star$
23	4	4.27e-02 8.41e-06	1 1	4.26e-02 8.49e-04	1 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		6.44e-02 5.24e-06	1 1	6.44e-02 5.24e-04	1 1	6.75e-02 5.29e-02	1 -1	$\star$ NaN	$\star$ $\star$
25	77	7.28e-01 4.35e-05	1 1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		1.27e+00 1.63e-05	2 1	1.21e+00 1.63e-03	2 1	1.37e+00 1.73e-01	3 -1	$\star$ NaN	$\star$ $\star$
26	237	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		1.94e+01 6.72e-04	2 1	2.10e+01 7.71e-02	3 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
27	397	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$
		7.60e+01 1.16e-04	2 1	7.72e+01 1.17e-02	2 -1	$\star$ NaN	$\star$ $\star$	$\star$ NaN	$\star$ $\star$

5. Conclusion

This paper focused on the united stable solution set to the interval continuous-time algebraic Riccati equation. After providing a weak characterization of the united stable solution set, we proposed a nonlinear programming technique and then a modified Krawczyk method on the preconditioned interval equation to efficiently compute an outer estimation for the united stable solution set. This modified Krawczyk method keeps the computational complexity down to cubic. Moreover, numerical experiments show the performance of Algorithm 1 on different terms.

Our modification to the Krawczyk method is based on the diagonalization of a closed loop matrix. If this decomposition does not exist, the problem of obtaining the outer estimation for the united stable solution set is not solved. This problem is still a topic for future research. One open problem is also to develop some estimations for different AE-solution sets to the interval algebraic Riccati equations. Another future work will be to develop a fast and more efficient verification algorithm for solutions of interval discrete-time algebraic Riccati equations. Moreover, the problem of how to compute an inner estimation for the united stable solution set of interval continuous-time algebraic Riccati equation is currently under consideration. We also wish to present, in future, an approach for finding the algebraic solutions of the interval algebraic Riccati equations.

Table 9. Comparison among various perturbation parameters α in the proportional perturbation approach before and after preconditioning.

Test number in [7]	size	$\alpha = 1e-9$		$\alpha = 1e-7$		$\alpha = 1e-5$		$\alpha = 1e-3$	
		time	itr	time	itr	time	itr	time	itr
		mr	stab	mr	stab	mr	stab	mr	stab
28	8	2.82e-02	1	2.78e-02	1	2.68e-02	1	4.18e-02	3
		1.39e-08	1	1.39e-06	1	1.40e-04	1	1.95e-02	1
		5.55e-02	1	3.62e-02	1	3.60e-02	1	3.56e-02	1
		1.95e-08	1	1.71e-06	1	1.73e-04	1	2.01e-02	1
29	64	1.03e-01	1	1.02e-01	1	$\star$	$\star$	$\star$	$\star$
		1.09e-07	1	1.10e-05	1	NaN	$\star$	NaN	$\star$
		1.70e-01	1	1.68e-01	1	1.69e-01	1	$\star$	$\star$
		1.51e-07	1	1.33e-05	1	1.40e-03	1	NaN	$\star$
32	100	3.53e-01	1	3.47e-01	1	3.95e-01	2	$\star$	$\star$
		5.71e-10	1	5.71e-08	1	6.51e-06	1	NaN	$\star$
		6.65e-01	2	6.20e-01	2	6.64e-01	3	$\star$	$\star$
		4.69e-10	1	4.70e-08	1	5.79e-06	1	NaN	$\star$
33	60	6.30e-01	1	5.30e-01	4	$\star$	$\star$	$\star$	$\star$
		1.46e-04	1	1.89e-02	-1	NaN	$\star$	NaN	$\star$
		4.85e-01	1	5.43e-01	1	9.19e-01	2	$\star$	$\star$
		1.23e-05	1	1.23e-03	1	1.26e-01	-1	NaN	$\star$

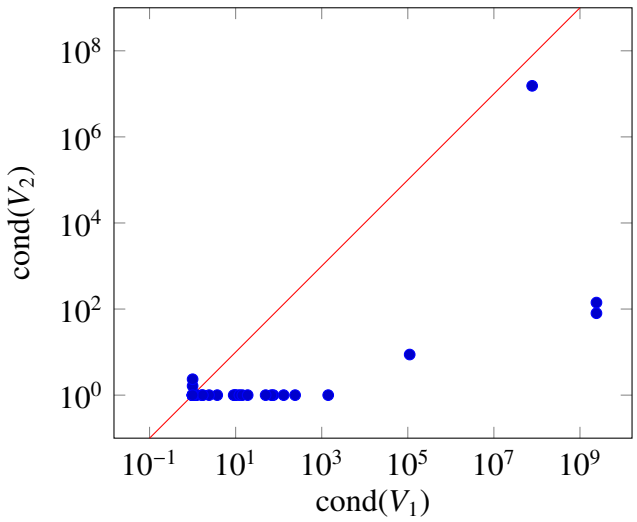

Figure 1. cond(V_2) vs. cond(V_1) when $\alpha = 1e-3$ in the fixed perturbation approach.

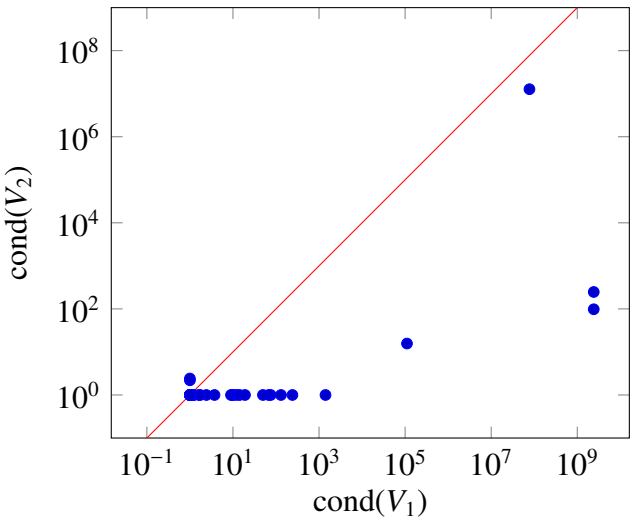

Figure 2. cond(V_2) vs. cond(V_1) when $\alpha = 1e-3$ in the proportional perturbation approach.

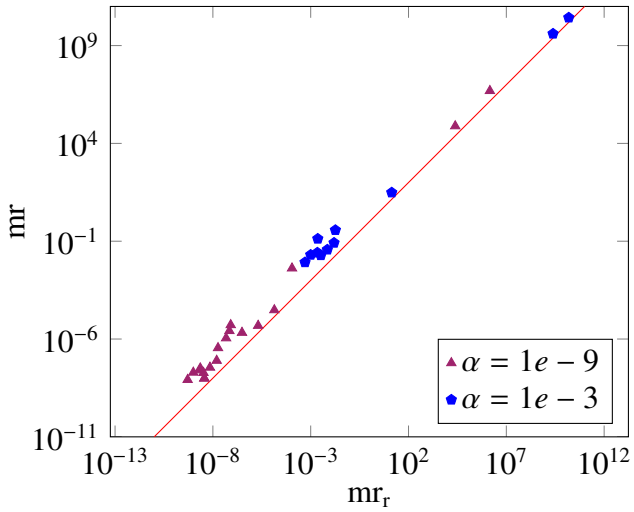

Figure 3. mr vs. mr_r when $\alpha = 1e-9$ and $\alpha = 1e-3$ in the proportional perturbation approach.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers. Reviewers' insightful comments led us to an improvement of the work. They also wish to thank Dr Milan Hladik and Dr Federico Poloni for their helpful suggestions concerning this paper.

References

[1] Adams E, Kulisch U. Scientific Computing with Automatic Result Verification. San Diego, CA, USA: Academic Press, 1992.

[2] Alefeld G, Herzberger J. Introduction to Interval Computations. New York, NY, USA: Academic Press, 1983.

[3] Alefeld G, Mayer G. Interval analysis: theory and applications. J Comput Appl Math 2000; 121: 421-464.

[4] Benner P, Li JR, Penzl T. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. Numer Linear Algebra Appl 2008; 15: 755-777.

[5] Bini DA, Iannazzo B, Meini B. Numerical Solution of Algebraic Riccati Equations. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2012.

[6] Bittanti S, Laub AJ, Willems JC. The Riccati Equation. Heidelberg, Germany: Springer, 2012.

[7] Chu D, Liu X, Mehrmann V. A numerical method for computing the Hamiltonian Schur form. Numer Math 2007; 105: 375-412.

[8] Datta BN. Numerical Methods for Linear Control Systems. San Diego, CA, USA: Elsevier Academic Press, 2004.

[9] Dehghani-Madiseh M, Dehghan M. Generalized solution sets of the interval generalized Sylvester matrix equation $\sum_{i=1}^{p} \mathbf{A}_i X_i + \sum_{j=1}^{q} Y_j \mathbf{B}_j = \mathbf{C}$ and some approaches for inner and outer estimations. Comput Math Appl 2014; 68: 1758-1774.

[10] Frommer A, Hashemi B. Verified computation of square roots of a matrix. SIAM J Matrix Anal Appl 2009; 31: 1279-1302.

[11] Gajic Z, Lim MT, Skataric D, Su WC, Kecman V. Optimal Control: Weakly Coupled Systems and Applications. Boca Raton, FL, USA: CRC Press, 2008.

[12] Hansen E, Walster GW. Global Optimization Using Interval Analysis: Revised and Expanded. Boca Raton, FL, USA: CRC Press, 2003.

[13] Haqiri T, Poloni F. Methods for verified solutions to continuous-time algebraic Riccati equations. J Comput Appl Math 2017; 313: 515-535.

[14] Hashemi B, Dehghan M. The interval Lyapunov matrix equation: analytical results and an efficient numerical technique for outer estimation of the united solution set. Math Comput Model 2012; 55: 622-633.

[15] Higham NJ. Functions of Matrices. Theory and Computation. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.

[16] Horn RA, Johnson CR. Topics in Matrix Analysis. Cambridge, UK: Cambridge University Press, 1994.

[17] Kearfott RB. Interval computations: introduction, uses, and resources. Euromath Bull 1996; 2: 95-112.

[18] Krawczyk R. Newton-algorithms for evaluation of roots with error bounds. Computing 1969; 4: 187-201.

[19] Lancaster P, Rodman L. Algebraic Riccati equations. New York, NY, USA: Oxford University Press, 1995.

[20] Mehrmann V. The autonomous linear quadratic control problem: theory and numerical solution. Heidelberg, Germany: Springer, 1991.

[21] Miyajima S. Fast enclosure for all eigenvalues and invariant subspaces in generalized eigenvalue problems. SIAM J Matrix Anal Appl 2014; 353: 1205-1225.

[22] Miyajima S. Fast verified computation for solutions of continuous-time algebraic Riccati equations. Jpn J Ind Appl Math 2015; 32: 529-544.

[23] Moore RE, Kearfott RB, Cloud MJ. Introduction to Interval Analysis. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[24] Neumaier A. Interval Methods for Systems of Equations. Cambridge, UK: Cambridge University Press, 1990.

[25] Poloni F. Algorithms for quadratic matrix and vector equations. PhD, Scuola Normale Superiore, Pisa, Italy, 2010.

[26] Rump SM. Verification methods: rigorous results using floating-point arithmetic. Acta Numer 2010; 19: 287-449.

[27] Rump SM: INTLAB-INTerval LABoratory. In: Tibor C, editor. Developments in Reliable Computing 1999, Dordrecht, Netherlands: Kluwer Academic Publishers, 1999, pp. 77-104.

[28] Seif NP, Hussein SA, Deif AS. The interval Sylvester equation. Computing 1994; 52: 233-244.

[29] Shashikhin VN. Robust assignment of poles in large-scale interval systems. Autom Remote Control 2002; 63: 200-208.

[30] Shashikhin VN. Robust stabilization of linear interval systems. J Appl Math Mech 2002; 66: 393-400.

[31] Zhou K, Doyle JC, Glover K. Robust and Optimal Control. Upper Saddle River, NJ, USA: Prentice Hall, 1996.