Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA

# EFFICIENT COMPUTATION OF ENCLOSURES FOR THE EXACT SOLVENTS OF A QUADRATIC MATRIX EQUATION*

BEHNAM HASHEMI† AND MEHDI DEHGHAN†

**Abstract.** None of the usual floating point numerical techniques available for solving the quadratic matrix equation $AX^2 + BX + C = 0$ with square matrices $A, B, C$ and $X$, can provide an exact solution; they can just obtain approximations to an exact solution. We use interval arithmetic to compute an interval matrix which contains an exact solution to this quadratic matrix equation, where we aim at obtaining narrow intervals for each entry. We propose a residual version of a modified Krawczyk operator which has a cubic computational complexity, provided that $A$ is nonsingular and $X$ and $X + A^{-1}B$ are diagonalizable. For the case that $A$ is singular or nearly singular, but $B$ is nonsingular we provide an enclosure method analogous to a functional iteration method. Numerical examples have also been given.

**Key words.** Quadratic matrix equation, Matrix square root, Interval analysis, Krawczyk operator, Automatic result verification.

**AMS subject classifications.** 65G20, 65F30.

## 1. Introduction.

Many applications such as multivariate rational expectations models [3], noisy Wiener-Hopf problems for Markov chains [11], quasi-birth death process [14], and the quadratic eigenvalue problem

$$Q(\lambda)\nu = (\lambda^2 A + \lambda B + C)\nu = 0, \quad \lambda \in \mathbb{C}, \quad \nu \in \mathbb{C}^n, \tag{1.1}$$

which comes from the analysis of damped structural systems, vibration problems [14, 15], and gyroscopic systems [12, 27] require the solution of the quadratic matrix equation

$$Q(X) = AX^2 + BX + C = 0. \tag{1.2}$$

In (1.2) the known real matrices $A, B, C$ and the unknown matrix $X$ are of dimension $n \times n$. A matrix $S$ satisfying $Q(S) = 0$ is called a solvent of $Q(X)$. The matrix square root problem is a special case of the quadratic matrix equation (1.2). More precisely for $A = I$, $B = 0$ and $C$ replaced by $-C$ we have $F(X) = X^2 - C = 0$ and every square root $S$ of the matrix $C$ satisfies the equation $F(S) = 0$.

*Received by the editors January 21, 2010. Accepted for publication August 16, 2010. Handling Editor: Daniel B. Szyld.

†Department of Applied Mathematics, Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, No.424 Hafez Avenue, Tehran 15914, Iran (hashemi_am@aut.ac.ir, mdehghan@aut.ac.ir).

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



The quadratic matrix equation (1.2) can have no solvents, a finite positive number, or infinitely many, as follows immediately from the theory of matrix square roots [19]. Suppose that $A$ is nonsingular in (1.2). Then $Q(\lambda)$ in (1.1) has $2n$ eigenvalues, all finite and can be ordered by their absolute values as

$$|\lambda_1| \geq |\lambda_2| \geq .... \geq |\lambda_{2n}|. \tag{1.3}$$

A solvent $S_1$ of $Q(X)$ is called a *dominant solvent* if $\lambda(S_1) = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ and $|\lambda_n| > |\lambda_{n+1}|$, where the eigenvalues $\lambda_i$ of $Q(\lambda)$ are ordered as in (1.3). A solvent $S_2$ of $Q(X)$ is called a *minimal solvent* if $\lambda(S_2) = \{\lambda_{n+1}, \lambda_{n+2}, ..., \lambda_{2n}\}$ and $|\lambda_n| > |\lambda_{n+1}|$.

THEOREM 1.1. [19] *Assume that the eigenvalues of $Q(\lambda)$, ordered according to (1.3) satisfy $|\lambda_n| > |\lambda_{n+1}|$ and that corresponding to $\{\lambda_i\}_{i=1}^{n}$ and $\{\lambda_i\}_{i=n+1}^{2n}$ there are two sets of linearly independent eigenvectors*

$$\{\nu_1, \nu_2, ..., \nu_n\}, \{\nu_{n+1}, \nu_{n+2}, ..., \nu_{2n}\}.$$

*Then there exists a dominant solvent and a minimal solvent of $Q(X)$. If, further, the eigenvectors of $Q(\lambda)$ are distinct then the dominant and minimal solvents are unique.*

Research on the quadratic matrix equation (1.2) goes back at-least to the works by Sylvester in the 1800s [19]. The problem of reducing an algebraic Riccati equation to (1.2) has been analyzed in [4]. Numerical methods for solving (1.2) has been considered including two linearly convergent algorithms for computing a dominant solvent [8, 10]. Davis [6, 7] used Newton's method for solving (1.2). Kim [19, 20] and Higham and Kim [14, 15] investigated theoretical and numerical results for solving the quadratic matrix equation (1.2). They improved the global convergence properties of Newton's method with exact line searches and gave a complete characterization of solutions in terms of the generalized Schur decomposition. Other numerical techniques, including the functional iteration methods based on Bernoulli's method, are described and compared in [14]. Recently, Long, Hu and Zhang [23] used Newton's method with $\check{S}$amanskii technique to obtain a faster convergence than Newton's method with exact line searches.

All the above-mentioned numerical techniques rely on floating point arithmetic and thus cannot provide an exact solvent of the quadratic matrix equation (1.2). Indeed, they always obtain only approximations to an exact solvent. In this paper we use interval arithmetic to provide reliable error bounds for each entry of an exact solvent of (1.2). We start with an approximate solvent $\tilde{X}$, obtained by one of the above-mentioned floating point methods, and aim at computing a tight interval matrix $\boldsymbol{X}$ which contains an exact solvent of (1.2). Actually we use interval arithmetic to find an interval matrix which is guaranteed to contain the error of an approximate solution obtained by a floating point algorithm. In this manner our approach is an example of dynamic error monitoring. The same task has been done by Frommer and

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



Hashemi for the matrix square root problem in [9]. Here, we try to generalize the approach of [9] to obtain verified solvents of (1.2).

We assume that the reader is familiar with basic results of interval analysis [2, 26]. The reader can obtain more insight into verified numerical computations based on interval analysis (also called scientific computing with automatic result verification, self-validating methods, or inclusion theory) through hundreds of related publications including many books like [1, 25]. In order to apply techniques of verified numerical computations one needs a correct implementation of machine interval arithmetic. It means that outward rounding has to be applied at each step of a numerical computation on a computer. A website that listed all the interval software is

`http://www.cs.utep.edu/intervalcomp/intsoft.html`.

An attractive interval arithmetic software is Intlab [32] which is a Matlab toolbox supporting real and complex interval scalars, vectors, and matrices, as well as sparse real and complex interval matrices. Intlab is available online at

`http://www.ti3.tu-harburg.de/rump/intlab/`.

Another new verification software is Versoft [29] which is a collection of Intlab programs and is freely available at

`http://uivtx.cs.cas.cz/~rohn/matlab/index.html`.

The organization of this paper is as follows. In Section 2, we introduce our notation and review some basic definitions and concepts. In Section 3 we present the standard Krawczyk operator and show its formulation and difficulties once it has been applied to the quadratic matrix equation (1.2). Section 4 contains our modified Krawczyk operator for enclosing solutions of the quadratic matrix equation (1.2) which dramatically reduces the computational cost of standard Krawczyk operator. In Section 5 we propose a different enclosure method based on a functional iteration method for the case that $A$ is singular. Section 6 ends this paper with some numerical examples.

**2. Notation and preliminary concepts.** Throughout this paper lower case letters are used for scalars and vectors and upper case letters for matrices. We use the standard notations of interval analysis [18]. So, all interval quantities will be typeset in boldface. $\mathbb{R}$ denotes the field of real numbers, $\mathbb{R}^{n\times n}$ the vector space of $n \times n$ matrices with real coefficients, $\mathbb{IR}$ the set of intervals and $\mathbb{IR}^{n\times n}$ the set of all $n \times n$ interval matrices. If $\boldsymbol{x} \in \mathbb{IR}$, then $\min \boldsymbol{x} := \underline{x}$ and $\max \boldsymbol{x} := \overline{x}$ are the lower and upper bounds of $\boldsymbol{x}$, respectively. The width of $\boldsymbol{x}$ is $\operatorname{wid} \boldsymbol{x} := \overline{x} - \underline{x} \geq 0$, its radius is $\operatorname{rad} \boldsymbol{x} := \frac{1}{2}(\overline{x} - \underline{x})$, its midpoint is $\operatorname{mid} \boldsymbol{x} := \frac{1}{2}(\overline{x} + \underline{x})$, and the absolute value of $\boldsymbol{x}$ is $\operatorname{abs} \boldsymbol{x} := \max\{|x| \mid x \in \boldsymbol{x}\} = \max\{|\underline{x}|, |\overline{x}|\}$. Moreover, $int(\boldsymbol{x})$ denotes the

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



topological interior of $\boldsymbol{x}$. We also use the notation $\mathbb{IC}^{n\times n}$ the set of all $n\times n$ complex interval matrices. There are at least two kinds of complex intervals: rectangular intervals and circular intervals [2]. Circular complex interval arithmetic has better algebraic properties than rectangular complex interval arithmetic [2]. We can also implement circular arithmetic completely in terms of BLAS routines to speed up interval computations [31]. This is one of the reasons why Intlab uses circular intervals while working with complex intervals. In this paper we also use circular complex intervals denoted by $\mathbb{IC}_{\text{disc}}$. So, in the remaining part of this paper we use the convention that each complex interval in $\mathbb{IC}$ is represented by a circular interval in $\mathbb{IC}_{\text{disc}}$. For interval vectors and matrices the above-mentioned operations will be applied componentwise. Hence, for a complex (real) interval matrix $\boldsymbol{A}$ we can write $\boldsymbol{A} = [\operatorname{mid}\boldsymbol{A} - \operatorname{rad}\boldsymbol{A}, \operatorname{mid}\ \boldsymbol{A} + \operatorname{rad}\boldsymbol{A}]$ where $\operatorname{mid}\boldsymbol{A}$ and $\operatorname{rad}\boldsymbol{A}$ are the complex (real) center matrix and the complex (real) radius matrix of the interval matrix $\boldsymbol{A}$, respectively.

For two real matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{k\times t}$ the Kronecker product $A \otimes B$ is given by the $mk \times nt$ block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}.$$

For $A = (a_{ij}) \in \mathbb{C}^{m\times n}$ the vector $\operatorname{vec}(A) \in \mathbb{C}^{mn}$ is obtained by stacking the columns of $A$. We use a convention on the implicit relation between upper and lower case letters when denoting variables, so $z = \operatorname{vec}(Z), u = \operatorname{vec}(U)$ etc. For an interval matrix $\boldsymbol{A} \in \mathbb{IC}^{m\times n}$ the vector $\operatorname{vec}(\boldsymbol{A})$ is analogously defined to be an interval vector $\boldsymbol{a} = \operatorname{vec}(\boldsymbol{A}) \in \mathbb{IC}^{mn}$.

The Hadamard (pointwise) division of two matrices $A, B \in \mathbb{C}^{n\times m}$ which we denote as $\cdot/$ is

$$A \cdot /B = C \in \mathbb{C}^{n\times m}, \text{ where } C = (c_{ij}) \text{ with } c_{ij} = a_{ij}/b_{ij}.$$

For $d = (d_1, \dots, d_n)^T \in \mathbb{C}^n$, the matrix $\operatorname{Diag}(d)$ denotes the diagonal matrix in $\mathbb{C}^{n\times n}$ whose $i$-th diagonal entry is $d_i$. Also for $D \in \mathbb{C}^{n\times m}$ we put $\operatorname{Diag}(D) = \operatorname{Diag}(\operatorname{vec}(D)) \in \mathbb{C}^{nm\times nm}$. The following properties of the Kronecker product, vec operator and Hadamard division will be used several times in this paper.

LEMMA 2.1. [16, 9] *For real matrices $A, B, C$ and $D$ with compatible sizes we have*

*a)* $(A \otimes B)(C \otimes D) = (AC \otimes BD)$.
*b)* $\operatorname{vec}(ABC) = (C^T \otimes A)\operatorname{vec}(B)$.
*c)* $\operatorname{Diag}(A)^{-1}\operatorname{vec}(B) = \operatorname{vec}(B \cdot /A)$.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



The above identities do not hold if we replace the matrices by interval matrices. However, the following enclosure properties are still valid.

LEMMA 2.2. [9] *Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ be interval matrices of compatible sizes. Then*

$$\{(C^T \otimes A)\mathrm{vec}(B) : A \in \boldsymbol{A}, B \in \boldsymbol{B}, C \in \boldsymbol{C}\} \subseteq \mathrm{vec}((\boldsymbol{AB})\boldsymbol{C}).$$

$$\{(C^T \otimes A)\mathrm{vec}(B) : A \in \boldsymbol{A}, B \in \boldsymbol{B}, C \in \boldsymbol{C}\} \subseteq \mathrm{vec}(\boldsymbol{A}(\boldsymbol{BC}).$$

The Fréchet derivative of a matrix function $G : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at $X \in \mathbb{C}^{n\times n}$ is a linear mapping

$$\mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n},$$

$$E \longmapsto G'(X)E,$$

such that for all $E \in \mathbb{C}^{n\times n}$

$$G(X + E) - G(X) - G'(X)E = o(||E||),$$

and $G'(X)E$ is said to be the Fréchet derivative of $G$ applied to the direction $E$ [13].

**3. Standard Krawczyk operator and its use for the quadratic matrix equation (1.2).** Let $f(x) = 0$, $f : D \subseteq \mathbb{C}^n \to \mathbb{C}^n$ be a nonlinear system of equation with a continuously differentiable function $f$, $\tilde{x} \in \mathbb{C}^n$, and $\boldsymbol{x} \in \mathbb{IC}^n$. A mapping $S : D \times D \to \mathbb{C}^{n\times n}$ is called a slope for $f$ if

$$f(y) - f(x) = S(y, x)(y - x) \text{ for all } x, y \in D.$$

Let $\boldsymbol{S}$ be an interval matrix containing all slopes $S(y, x)$ for $y \in \boldsymbol{x}$. If $x \in \boldsymbol{x}$ the standard choice is $\boldsymbol{S} = \boldsymbol{f}'(\boldsymbol{x})$, the interval arithmetic evaluation of $f'(x)$ which contains the set $\{f'(y) : y \in \boldsymbol{x}\}$. For real case this follows from the mean-value theorem applied to each component $f_i$, but for the complex case this is not correct and we cannot use the mean-value theorem. However, for the complex function under consideration in this paper, we can still enclose slopes by the interval arithmetic evaluation of the derivative. This has been proved in Theorem 3.1 in the following.

Various fixed-point theorems applicable in finite or infinite dimensional spaces, state roughly that, if a mapping maps a set into itself, then that mapping has a fixed-point within that set [17]. For example, the Brouwer fixed-point theorem states that, if $D$ is homeomorphic to the closed unit ball in $\mathbb{R}^n$ and $g$ is a continuous mapping such that $g$ maps $D$ into $D$, then $g$ has a fixed-point in $D$, i.e., there is an $x \in D$ with

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010



$x = g(x)$. An interval extension $\boldsymbol{g}$ of $g$ has the property that, if $\boldsymbol{x}$ is an interval vector with $\boldsymbol{x} \subseteq D$, then $\boldsymbol{g}(\boldsymbol{x})$ contains the range $\{g(x) : x \in \boldsymbol{x}\}$. This interval extension once evaluated with outward rounding can be used so that the floating point intervals rigorously contain the actual range of $g$. Thus, if $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, we conclude that $g$ has a fixed point within $\boldsymbol{x}$ [5, 17, 28].

As a most popular fixed point form for $f(x) = 0$ one can define

$$g(x) = x - Rf(x), \tag{3.1}$$

where $R$ is usually taken to be a nonsingular matrix (a nonsingular linear operator in $D$ if $D$ is a general linear space [28]). However, it is clear that one can define other fixed point forms for $f(x) = 0$ depending on the function. *Krawczyk interval operator* $\boldsymbol{k}(\tilde{x}, \boldsymbol{x})$ is actually based on a mean value extension of the fixed point form $g$ in (3.1). For a given matrix $R \in \mathbb{C}^{n\times n}$, the Krawczyk operator $\boldsymbol{k}(\tilde{x}, \boldsymbol{x})$ defined by

$$\boldsymbol{k}(\tilde{x}, \boldsymbol{x}) = \tilde{x} - Rf(\tilde{x}) + (I - R \cdot \boldsymbol{S})(\boldsymbol{x} - \tilde{x}),\ \tilde{x} \in \boldsymbol{x} \subset D. \tag{3.2}$$

can then be used to find an enclosures for the solution of the nonlinear system of equations $f(x) = 0$. Assume that $\boldsymbol{S}$ is an interval matrix containing all slopes $S(y, \tilde{x})$ for $y \in \boldsymbol{x}$. If

$$\boldsymbol{k}(\tilde{x}, \boldsymbol{x}) \subseteq \operatorname{int} \boldsymbol{x}, \tag{3.3}$$

then $f$ has a zero $x^*$ in $\boldsymbol{k}(\tilde{x}, \boldsymbol{x})$. Moreover, if $\boldsymbol{S}$ also contains all slopes $S(y, x)$ for $x, y \in \boldsymbol{x}$, the zero $x^*$ is the only zero of $f$ in $\boldsymbol{x}$ [21, 22, 24].

Let us note that the relation (3.3) is likely to hold only if $R \cdot \boldsymbol{S}$ is close to the identity, (i.e., $R$ is a good approximation to the inverse of mid $\boldsymbol{S}$ which is the standard choice for $R$) and $\tilde{x}$ is a good a good approximation to a zero of $f$. A method for obtaining trial interval vectors $\boldsymbol{x}$ around $\tilde{x}$ for which the relation (3.3) can be expected to hold is the so-called $\epsilon$-inflation [30].

Let $\mathbb{F}$ be the set of floating-point numbers following IEEE standard 754. By $A \in \mathbb{F}^{n\times n}$ we mean that $A$ is an $n\times n$ matrix with entries that are exactly representable by a floating-point number in $\mathbb{F}$. In this paper we consider the quadratic matrix equation (1.2) and suppose that $A, B, C \in \mathbb{F}^{n\times n}$. The solution matrix $X$ can, however, be a matrix in $\mathbb{C}^{n\times n}$ and we aim at computing enclosures for each entry of $X$. The quadratic matrix equation (1.2) can be reformulated, interpreting matrices as vectors in $\mathbb{C}^{n^2}$ via $x = \operatorname{vec}(X), a = \operatorname{vec}(A), b = \operatorname{vec}(B), c = \operatorname{vec}(C)$ and using Lemma 2.1 as

$$q(x) = (X^T \otimes A)x + (I_n \otimes B)x + c = 0. \tag{3.4}$$

Because

$$Q(X + E) = Q(X) + AEX + (AX + B)E + AE^2,$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



the Fréchet derivative of the quadratic matrix equation (1.2) at $X$ applied to the direction $E$ is given by

$$Q'(X)E = AEX + (AX + B)E.$$

Higham and Kim [15] note that if $A$ is nonsingular then $Q'(X)$ is nonsingular at a dominant or minimal solvent and also at all solvents $X$ if the eigenvalues of $Q(\lambda)$ are distinct. By Lemma 2.1, this translates into

$$q'(x)e = [X^T \otimes A + I_n \otimes (AX + B)]e, \quad \text{and } q'(x) = X^T \otimes A + I_n \otimes (AX + B).$$

THEOREM 3.1. *Consider the quadratic matrix equation (1.2). Then the interval arithmetic evaluation of the derivative of $q(x)$, i.e., the interval matrix $\boldsymbol{X}^T \otimes A + I \otimes (A\boldsymbol{X} + B)$ contains slopes $S(y, x)$ for all $x, y \in \boldsymbol{x}$.*

*Proof.* For $x, y \in \boldsymbol{x}$ we have

$$Q(Y) - Q(X) = AY^2 + BY + C - AX^2 - BX - C = A(Y^2 - X^2) + B(Y - X)$$

$$= \frac{1}{2}A(Y + X)(Y - X) + \frac{1}{2}A(Y - X)(Y + X) + B(Y - X)$$

$$= [\frac{1}{2}A(Y + X) + B](Y - X) + \frac{1}{2}A(Y - X)(Y + X).$$

So, using part b) of Lemma 2.1 we have

$$q(y) - q(x) = [I \otimes (\frac{1}{2}A(Y + X) + B)](y - x) + (\frac{1}{2}(Y + X)^T \otimes A)(y - x),$$

which means that

$$S(y, x) = I \otimes [\frac{1}{2}A(Y + X) + B] + \frac{1}{2}(Y + X)^T \otimes A,$$

is a slope for $q$. Therefore, for $x, y \in \boldsymbol{x}$, the slope $S(y, x)$ is contained in

$$S(\boldsymbol{x}, \boldsymbol{x}) = I \otimes (A\boldsymbol{X} + B) + \boldsymbol{X}^T \otimes A,$$

which is the interval arithmetic evaluation of derivative of the function $q(x) = 0$. □

We note in passing that this theorem also justifies the use of interval arithmetic evaluation of the derivative of the nonlinear function $X^2 - A = 0$ for enclosing its slopes in [9].

The standard Krawczyk operator (3.2) for the particular function $q(x)$ is given as

$$\boldsymbol{k}(\tilde{x}, \boldsymbol{x}) = \tilde{x} - R\left((\tilde{X}^T \otimes A)\tilde{x} + (I_n \otimes B)\tilde{x} + c\right) + \tag{3.5}$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



$$[I_{n^2} - R\Big(\operatorname{mid} \boldsymbol{X}^T \otimes A + I \otimes (A \cdot \operatorname{mid} \boldsymbol{X} + B)\Big)](\boldsymbol{x} - \tilde{x}),$$

where $R \in \mathbb{C}^{n^2 \times n^2}$ is an approximate inverse of $q'(\boldsymbol{x}) = \operatorname{mid} \boldsymbol{X}^T \otimes I + I \otimes (A \cdot \operatorname{mid} \boldsymbol{X} + B)$. Generally computing such an approximate inverse $R$ requires $\mathcal{O}(n^6)$ operations. On the other hand, the $n^2 \times n^2$ matrix $R$ does not have a nice Kronecker structure, and it will usually be a full matrix. Each of the $n^2$ columns of $(I \otimes (A\boldsymbol{X} + B) + \boldsymbol{X}^T \otimes A)$ has $n^2$ non-zeros. So, computing the product of $R$ with $(I \otimes (A\boldsymbol{X} + B) + \boldsymbol{X}^T \otimes A)$ will require $n^4$ operations for each entry, i.e. a total cost of $\mathcal{O}(n^6)$. Overall, the dominant cost in evaluating $\boldsymbol{k}(\tilde{x}, \boldsymbol{x})$ is $\mathcal{O}(n^6)$. Therefore, the main disadvantage of the standard Krawczyk operator (3.5) for enclosing a solvent of the quadratic matrix equation (1.2) is its huge computational complexity.

So, we need an enclosing method which has been specially designed for the quadratic matrix equation (1.2) and can exploit its structure so that we would be able to obtain an enclosure more cheaply. To our best knowledge there is not any other verification algorithm available in the literature which can be used for the quadratic matrix equation (1.2). The next section is aimed at introducing such an approach.

**4. A modified Krawczyk operator for the quadratic matrix equation (1.2).** The following slight generalization of (3.3) which expresses the essence of all Krawczyk type verification methods has been proved in [9]. In its formulation we represent $\boldsymbol{x}$ as $\tilde{x} + \boldsymbol{z}$, thus separating the approximate zero $\tilde{x}$ from the enclosure of its error, $\boldsymbol{z}$.

THEOREM 4.1. [9] *Assume that $f : D \subset \mathbb{C}^n \to \mathbb{C}^n$ is continuous in $D$. Let $\tilde{x} \in D$ and $\boldsymbol{z} \in \mathbb{IC}^n$ be such that $\tilde{x} + \boldsymbol{z} \subseteq D$. Moreover, assume that $\mathcal{S} \subset \mathbb{C}^{n \times n}$ is a set of matrices containing all slopes $S(\tilde{x}, y)$ of $f$ for $y \in \tilde{x} + \boldsymbol{z} =: \boldsymbol{x}$. Finally, let $R \in \mathbb{C}^{n \times n}$. Define the set $\mathcal{K}_f(\tilde{x}, R, \boldsymbol{z}, \mathcal{S})$ by*

$$\mathcal{K}_f(\tilde{x}, R, \boldsymbol{z}, \mathcal{S}) := \{-Rf(\tilde{x}) + (I - RS)z : S \in \mathcal{S}, z \in \boldsymbol{z}\}. \tag{4.1}$$

*Then, if*

$$\mathcal{K}_f(\tilde{x}, R, \boldsymbol{z}, \mathcal{S}) \subseteq \operatorname{int} \boldsymbol{z}, \tag{4.2}$$

*the function $f$ has a zero $x^*$ in $\tilde{x} + \mathcal{K}_f(\tilde{x}, R, \boldsymbol{z}, \mathcal{S}) \subseteq \boldsymbol{x}$. Moreover, if $\mathcal{S}$ also contains all slope matrices $S(y, x)$ for $x, y \in \boldsymbol{x}$, then this zero is unique in $\boldsymbol{x}$.*

We now develop another version of the Krawczyk operator, relying on Theorem 4.1, which has a reasonable computational cost of $\mathcal{O}(n^3)$, provided that $A$ is nonsingular, and $\tilde{X}$ and $\tilde{X} + A^{-1}B$ are diagonalizable. The non-singularity condition of the matrix $A$ in not so restrictive, because the matrix $A$ in the quadratic matrix equation (1.2) is often nonsingular, e.g., in applications in the overdamped quadratic eigenvalue problems [19], gyroscopic systems [12, 27], noisy Wiener-Hopf problems for

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



Markov chains [11] and for the special case of matrix square roots. Assume that we have the following spectral decompositions for $X$ and $X + A^{-1}B$

$$X = V_X D_X W_X, \text{ with } D_X = \text{Diag}\,(\lambda_1, \ldots, \lambda_n) \text{ diagonal}, V_X W_X = I, \tag{4.3}$$

and

$$X + A^{-1}B = V_T D_T W_T, \text{ with } D_T = \text{Diag}\,(\mu_1, \ldots, \mu_n) \text{ diagonal}, V_T W_T = I, \tag{4.4}$$

in which $V_X, D_X, W_X, V_T, D_T, W_T \in \mathbb{C}^{n\times n}$. Here, $V_X$ and $V_T$ are the matrices of right eigenvectors, and $W_X$, $W_T$ are the matrices of left eigenvectors.

If $X$ is an accurate approximate solvent of (1.2) then $V_X^{-1} X V_X$ and $W_T(X + A^{-1}B)W_T^{-1}$ will be close to the diagonal matrices $D_X$ and $D_T$, respectively. Because

$$q'(x) = X^T \otimes A + I_n \otimes (AX + B) = (I_n \otimes A) \cdot \big(X^T \otimes I_n + I_n \otimes (X + A^{-1}B)\big) =$$

$$(I_n \otimes A) \cdot (W_X^T \otimes V_T) \cdot \big(V_X^T X^T V_X^{-T} \otimes I + I \otimes (W_T(X + A^{-1}B)W_T^{-1})\big) \cdot (V_X^T \otimes W_T),$$

an approximate inverse for $q'(x)$ would be in factorized form

$$R = (V_X^{-T} \otimes W_T^{-1}) \cdot \Delta^{-1} \cdot (V_X^T \otimes W_T) \cdot (I_n \otimes A^{-1}), \tag{4.5}$$

where $\Delta = I \otimes D_T + D_X \otimes I$. We assume that $V_X, W_X, D_X$ and $V_T, W_T, D_T$ are approximated using a floating point method for computing the spectral decompositions (4.3) and (4.4) like MATLAB's `eig` function, So, we do not assume that $V_X W_X = I$ and $V_T W_T = I$ hold exactly. Moreover, the diagonal matrices $D_X$ and $D_T$ will generally not have the exact eigenvalues of $X$ and $T$ on their diagonal. In other words, $V_X, W_X, D_X$ and $V_T, W_T, D_T$ are all approximations (not the exact quantities) obtained by a floating point algorithm.

We now introduce Algorithm 1 which by using Part b of Lemma 2.1 efficiently compute

$$l = \text{vec}(L) := -R \cdot q(x) = -(V_X^{-T} \otimes W_T^{-1})\Delta^{-1}(V_X^T \otimes W_T)(I_n \otimes A^{-1}) \cdot q(x).$$

**Algorithm 1** Efficient computation of $l = -Rq(x)$

1: Compute $Q = AX^2 + BX + C$
2: Compute $G_1 = A^{-1}Q$
3: Compute $G_2 = W_T G_1 V_X$
4: Compute $H = G_2./D$
5: Compute $L = -W_T^{-1} H V_X^{-1}$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



On the other hand we need to compute $(I - Rq'(x))z$. For any matrix $X \in \mathbb{C}^{n\times n}$ and any vector $z \in \mathbb{C}^{n^2}$ we have

$$\Big(I - R\cdot\big(X^T \otimes A + I \otimes (AX + B)\big)\Big)\cdot z =$$

$$(V_X^{-T}\otimes W_T^{-1})\cdot\Delta^{-1}\cdot\big(\Delta - I\otimes(W_T(X + A^{-1}B)W_T^{-1}) - (V_X^{-1}XV_X)^T\otimes I\big)\cdot\big(V_X^T\otimes W_T\big)\cdot z.$$

The latter expression is rich in Kronecker products, so that using Lemma 2.1 b) we can efficiently compute $u = (I - R(X^T \otimes A + I \otimes (AX + B)))z$. See Algorithm 2.

**Algorithm 2** Efficient computation of $u = (I - Rq'(x))z$

1: Compute $Y = W_T Z V_X$ {The $j$-th column of $Y$ will be denoted $Y_j$}
2: Compute $S = V_X^{-1}XV_X$ {$S$ is an $n \times n$ matrix with entries $S_{ij}$}
3: Compute $T = W_T(X + A^{-1}B)W_T^{-1}$
4: **for** $i = 1, \ldots, n$ **do** {Compute columns $f_i$ of matrix $F$}
5: Compute $f_i = (\mathrm{Diag}\,(d_i) - S_{ii}I - T)Y_i$ $\{\Delta = \mathrm{Diag}\,(D), D = [d_1|\ldots|d_n] \in \mathbb{C}^{n\times n}\}$
6: **end for**
7: Compute $P = -YS_0 + [f_1|\ldots|f_n]$ where $S_0 = S - \mathrm{Diag}\,(S_{11}, \ldots, S_{nn})$
8: Compute $N = P \cdot /D$
9: Compute $U = W_T^{-1}NV_X^{-1}$

Lines 4-7 in Algorithm 2 compute

$$p = \mathrm{vec}(P) := \Big(\Delta - I\otimes\big(W_T(X + A^{-1}B)W_T^{-1}\big) - (V_X^{-1}XV_X)^T\otimes I\Big)y.$$

We defined the matrix $S_0$ in line 7 of the algorithm so that the diagonal entries of $S$ are replaced by zeros in $S_0$. The reason for defining $S_0$ is to prevent the use of Level 1 BLAS, since machine interval arithmetic as implemented in Intlab is particularly efficient if the Level 2 and Level 3 BLAS are used as much as possible.

We are now in a position to use Algorithm 1 and Algorithm 2 to efficiently compute an interval vector containing the set $\mathcal{K}_f(\tilde{x}, R, \boldsymbol{z}, \boldsymbol{S})$ in (4.1) with $f(x)$ replaced by $q(x)$

$$\boldsymbol{S} = \boldsymbol{q'}(\tilde{x} + \boldsymbol{z}) = (\tilde{X} + \boldsymbol{Z})^T \otimes I_n + I_n \otimes \big(A(\tilde{X} + \boldsymbol{Z})B\big).$$

Algorithm 3 obtains an interval vector $\boldsymbol{k} = \mathrm{vec}(\boldsymbol{K}) = \mathrm{vec}(\boldsymbol{L}) + \mathrm{vec}(\boldsymbol{U})$ containing

$$\mathcal{K}_q(\tilde{x}, R, \boldsymbol{z}, \boldsymbol{q'}(\tilde{x} + \boldsymbol{z})) = \{-Rq(\tilde{x}) + (I - RS)z : S \in \boldsymbol{q'}(\tilde{x} + \boldsymbol{z}), z \in \boldsymbol{z}\}.$$

We need to replace the point quantities $X$ and $Z$ from Algorithm 1 by $\tilde{X} + \boldsymbol{Z}$ and $\boldsymbol{Z}$ in Algorithm 3 because of the inclusion property of interval arithmetic. Moreover, note

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



that $V_X^{-1}$ and $W_T^{-1}$ will usually not be available as exact inverses of the computed matrices $V_X$ and $W_T$, because they have been obtained using a floating point algorithm. Actually we need to use enclosures for the exact values of $V_X^{-1}$ and $W_T^{-1}$. Such enclosures can be obtained by using methods of interval analysis like the `verifylss` command of Intlab. So, we replace $V_X^{-1}$ and $W_T^{-1}$ by interval enclosures $\boldsymbol{I}_{V_X}, \boldsymbol{I}_{W_T}$ which we know that contain their exact values, respectively. The same holds for lines 2 and 8 of Algorithm 3, where we replaced $A^{-1}$ by the interval matrix $\boldsymbol{I}_A$. Note also that in general for three interval matrices $\boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}$ we have $(\boldsymbol{E} \cdot \boldsymbol{F}) \cdot \boldsymbol{G} \neq \boldsymbol{E} \cdot (\boldsymbol{F} \cdot \boldsymbol{G})$ because of the subdistributive law of interval arithmetic. But in Algorithm 3 we do not need to indicate the order of interval matrix multiplications; see Lemma 2.2 which guarantees $\boldsymbol{k} \supseteq \mathcal{K}_q(\tilde{x}, R, \boldsymbol{z}, \boldsymbol{q}'(\tilde{x} + \boldsymbol{z}))$ for whatever order we choose for the interval matrix multiplications.

**Algorithm 3** Computation of an interval matrix $\boldsymbol{K}$ such that $\mathrm{vec}(\boldsymbol{K})$ contains $\mathcal{K}_q$ with $\tilde{x}$ instead of $x$

1: Compute $\boldsymbol{Q} = A \cdot \tilde{X}^2 + B \cdot \tilde{X} + C$ {$\boldsymbol{Q}$ is an interval matrix due to outward rounding}
2: Compute $\boldsymbol{G}_1 = \boldsymbol{I}_A \boldsymbol{Q}$
3: Compute $\boldsymbol{G}_2 = W_T \boldsymbol{G}_1 V_X$ {$W_T$ and $V_X$ are obtained from the spectral decompositions of $\tilde{X} + A^{-1}B$ and $\tilde{X}$, respectively}
4: Compute $\boldsymbol{H} = \boldsymbol{G}_2 \cdot /D$
5: Compute $\boldsymbol{L} = -\boldsymbol{I}_{W_T} \boldsymbol{H} \boldsymbol{I}_{V_X}$
6: Compute $\boldsymbol{Y} = W_T \boldsymbol{Z} V_X$ {The $j$-th column of $\boldsymbol{Y}$ will be denoted $\boldsymbol{Y}_j$}
7: Compute $\boldsymbol{S} = \boldsymbol{I}_{V_X}(\boldsymbol{Z} + \tilde{X})V_X$ {$\boldsymbol{S}$ is an $n \times n$ interval matrix with entries $\boldsymbol{S}_{ij}$}
8: Compute $\boldsymbol{T} = W_T(\boldsymbol{Z} + \tilde{X} + \boldsymbol{I}_A B)\boldsymbol{I}_{W_T}$
9: **for** $i = 1, \ldots, n$ **do**
10: compute $\boldsymbol{f}_i = (\mathrm{Diag}\,(d_i) - \boldsymbol{S}_{ii} I - \boldsymbol{T})\boldsymbol{Y}_i$
11: **end for**
12: Compute $\boldsymbol{P} = -\boldsymbol{Y}\boldsymbol{S}_0 + [\boldsymbol{f}_1 | \ldots | \boldsymbol{f}_n]$
13: Compute $\boldsymbol{N} = \boldsymbol{P} \cdot /D$
14: Compute $\boldsymbol{U} = \boldsymbol{I}_{W_T} \boldsymbol{N} \boldsymbol{I}_{V_X}$
15: Compute $\boldsymbol{K} = \boldsymbol{L} + \boldsymbol{U}$

The following theorem analyzes the cost of Algorithm 3.

THEOREM 4.2. *Algorithm 3 requires $\mathcal{O}(n^3)$ arithmetic operations.*

*Proof.* Since the main operations in lines 1-8 include interval matrix-matrix multiplications, the computation of $\boldsymbol{Q}, \boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{H}, \boldsymbol{L}, \boldsymbol{Y}, \boldsymbol{S}$ and $\boldsymbol{T}$ costs $\mathcal{O}(n^3)$. In lines 9-11, the cost for each $i$ is $\mathcal{O}(n^2)$, since we have an interval matrix-vector multiplication for a matrix of size $n \times n$ and a vector of size $n \times 1$. These matrix-vector multiplications have to be done $n$ times. So, overall the for-loop also has cost $\mathcal{O}(n^3)$.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010



Since lines 12-15 have also cost $\mathcal{O}(n^3)$, the theorem is proved. □

We now use Algorithm 3 to construct our final algorithm that states the use of Krawczyk operator for enclosing an exact solvent of the quadratic matrix equation (1.2). See Algorithm 4.

**Algorithm 4** If successful this algorithm provides an interval matrix $\boldsymbol{X}$ containing an exact solvent of the quadratic matrix equation (1.2)

1: Use a floating point algorithm to get an approximate solvent $\tilde{X}$ of (1.2).
{Use Newton's method with exact line searches, e.g.}
2: Use a floating point algorithm to get approximations for $V_X, W_X, D_X$ and $V_T, W_T, D_T$ in the spectral decomposition of $\tilde{X}$ and $\tilde{X} + A^{-1}B$, resp.
{Use Matlab's `eig`, e.g.}
3: Compute interval matrices $\boldsymbol{I}_{V_X}$, $\boldsymbol{I}_{W_T}$ and $\boldsymbol{I}_A$ containing $V_X^{-1}$, $W_T^{-1}$, and $A^{-1}$ resp. {Take `verifylss.m` from INTLAB, e.g.}
4: Compute $\boldsymbol{L}$, an interval matrix containing $-Rq(\tilde{x})$ as in lines 1-5 of Algorithm 3.
5: $\boldsymbol{Z} = \boldsymbol{L}$
6: **for** $k = 1, \ldots k_{\max}$ **do**
7: $\epsilon$-inflate $\boldsymbol{Z}$
8: compute $\boldsymbol{U}$ for input $\tilde{X}, \boldsymbol{Z}$ as in lines 6-14 of Algorithm 3
9: **if** $\boldsymbol{K} := \boldsymbol{L} + \boldsymbol{U} \subseteq \operatorname{int} \boldsymbol{Z}$ **then** {successful}
10: output $\boldsymbol{X} = \tilde{X} + \boldsymbol{K}$ and **stop**
11: **else** {second try}
12: put $\boldsymbol{Z}^{(2)} = \boldsymbol{Z} \cap \boldsymbol{K}$
13: compute $\boldsymbol{U}^{(2)}$ for input $\tilde{X}, \boldsymbol{Z}^{(2)}$ as in lines 6-14 of Algorithm 3
14: **if** $\boldsymbol{K}^{(2)} = \boldsymbol{L} + \boldsymbol{U}^{(2)} \subseteq \operatorname{int} \boldsymbol{Z}^{(2)}$ **then** {successful}
15: output $\boldsymbol{X} = \tilde{X} + \boldsymbol{K}^{(2)}$ and **stop**
16: **else**
17: overwrite $\boldsymbol{Z}$ as $\boldsymbol{Z} \cap \boldsymbol{K}^{(2)}$
18: **end if**
19: **end if**
20: **end for**

**5. Enclosures based on a functional iteration method in the case that $A$ is singular.** We can use iterative interval methods, like the Krawczyk operator, for enclosing solutions to an equation in fixed-point form. It may happen that the matrix $A$ in (1.2) is nearly singular or even singular as in the second example given in the next section. In such cases the product $R \cdot \boldsymbol{S}$ in the standard Krawczyk operator (3.2) will not be close to the identity matrix and so is likely to fail. Our modified Krawczyk operator cannot also be used, since the nonsingularity of $A$ was necessary

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



for the formulation of our algorithms in Section 4. So, in the case that $A$ is singular or nearly singular, we need an alternative approach.

In Section 3 we noted that Krawczyk interval operators are essentially based on the fixed point function $g(x) = x - Rq(x)$ with $q(x) = 0$ in (3.4). However, we can define other fixed point functions $g$ for $q(x) = 0$ in a manner which makes possible the computation of enclosures without inverting the matrix $A$. In general, given a fixed point equation $x = g(x)$ we can take an interval extension $\boldsymbol{g}$ of $g$ and set up an iterative procedure of the following form [25]

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k) \cap \boldsymbol{x}_k : \quad k = 0, 1, 2, ...$$

When $A$ in the quadratic matrix equation (1.2) is singular but $B$ is nonsingular, a good choice for the fixed point function $g$ is

$$G(X) = -B^{-1}(AX^2 + C),$$

i.e.,

$$g(x) = -(I_n \otimes B^{-1}) \left((X^T \otimes A)x + c\right),$$

in the vector form. Higham and Kim [14] used the above function $G(X)$ to construct a functional iteration based on the Bernoulli's method. The floating point iterative method of [14] is

$$X_{k+1} = -B^{-1}(AX_k^2 + C),$$

with $X_0 = 0_{n\times n}$.

If $\boldsymbol{I}_B$ is an enclosure for $B^{-1}$, then $\boldsymbol{G}(\boldsymbol{X}) = -\boldsymbol{I}_B(A\boldsymbol{X}^2 + C)$ is an interval extension to the function $G$ in the above. Indeed $\mathrm{vec}(\boldsymbol{G}(\boldsymbol{X}))$ contains the range of $g$ over $\boldsymbol{x}$. We define the following iterative interval scheme

$$\boldsymbol{X}_{k+1} = -\boldsymbol{I}_B(A\boldsymbol{X}_k^2 + C) \cap \boldsymbol{X}_k : \quad k = 0, 1, 2, ... \tag{5.1}$$

If we start with an interval matrix $\boldsymbol{X}_0$ such that $\boldsymbol{G}(\boldsymbol{X}_0) \subseteq \boldsymbol{X}_0$, then (5.1) produces a nested sequence of interval matrices $\{\boldsymbol{X}_k\}$ convergent to an interval matrix $\boldsymbol{X}^*$ which encloses an exact solvent of (1.2) with $\boldsymbol{X}^* = \boldsymbol{G}(\boldsymbol{X}^*)$ and $\boldsymbol{X}^* \subseteq \boldsymbol{X}_k$ for all $k = 0, 1, 2, ....$

THEOREM 5.1.

*a) If there exists a solvent $X^*$ of (1.2) in $\boldsymbol{X}_k$, then $X^* \in \boldsymbol{X}_{k+1}$ defined by (5.1).*
*b) If $\boldsymbol{X}_{k+1}$ obtained by (5.1) is empty, then there is no solvent of (1.2) in $\boldsymbol{X}_k$.*

*Proof.* Firstly suppose that there exists a solvent $X^*$ of (1.2) such that $X^* \in \boldsymbol{X}_k$. Then $X^* = G(X^*) = -B^{-1}(AX^{*^2} + C) \in -\boldsymbol{I}_B(A\boldsymbol{X}_k^2 + C)$, i.e., $X^* \in G(\boldsymbol{X}_k)$. So,

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



$X^* \in \boldsymbol{G}(\boldsymbol{X}_k) \cap \boldsymbol{X}_k$, i.e., $X^* \in \boldsymbol{X}_{k+1}$.
Now consider second part of theorem and suppose that there is a solvent $X^*$ of (1.2) in $\boldsymbol{X}_k$. Then the first part of this theorem shows that $X^* \in \boldsymbol{X}_{k+1}$ which is a contradiction. So, the proof is completed. □

We emphasize that the iterative method (5.1) cannot be applied to examples with a singular or nearly singular matrix $B$, like the matrix square root problem which has $B = 0_{n\times n}$, or the last example in the next section where the condition number of $B$ is $1.1 \times 10^{18}$. Another advantage of Algorithm 4 over (5.1) is that different theoretical and computational aspects of Krawczyk-type methods have been fully analyzed in the literature. For example the choice of an initial interval vector using $\epsilon$-inflation has been a subject of research by itself [30].

As a final remark we note that there are still other possibilities for choosing the fixed point function $G$ like

$$G(X) = (-A^{-1}(BX + C))^{1/2} \tag{5.2}$$

as described in [14]. Here, we decided not to work with (5.2), even if $A$ is nonsingular. The reasons for this are two fold: Firstly, we have Algorithm 4 which works quite well in the case that $A$ is nonsingular. Secondly, if we want to define an interval iteration by using $G(X) = (-A^{-1}(BX + C))^{1/2}$, then we need a definition and method of efficient computation for *enclosures for square roots of interval matrices* which has not been fully examined in the literature. This can also be a direction for a future research.

**6. Numerical experiments.** Here we present a comparison of the results obtained by the standard Krawczyk operator (3.5) and our modified Krawczyk operator (Algorithm 4). For the special case of matrix square roots there is an alternative algorithm available in the `vermatfun` routine of Rohn's Versoft [29]. But, for the case of quadratic matrix equation (1.2) we could not find any other algorithm to compare with ours. We performed all our experiments on a 2.00 GHz Pentium 4 with 1 GB of RAM. $t_0$ will denote the time spent only for computing the approximation $\tilde{X}$ via Newton's method with exact line searches and *time* will represent the total computing time, i.e. the time for computing $\tilde{X}$ plus the time needed for the verification. To show the quality of the enclosures obtained, we also report the maximum radius $mr$ of the components of the enclosing interval matrix $\boldsymbol{X}$, i.e.

$$mr = \max_{i,j=1}^{n} \operatorname{rad}(\boldsymbol{X}_{ij}).$$

So, the value $-\log_{10} mr$ is the minimum number of correct decimal digits (including leading zeros) among all the components that our algorithms can guarantee.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



| $n$ | $t_0$ | Standard Krawczyk | | | Algorithm 4 | | |
|---|---|---|---|---|---|---|---|
| | | time | $k$ | mr | time | $k$ | mr |
| 10 | 0.02 | 0.10 | 1 | $6.3 \cdot 10^{-16}$ | 0.12 | 1 | $3.5 \cdot 10^{-15}$ |
| 20 | 0.06 | 2.7 | 1 | $6.7 \cdot 10^{-16}$ | 0.21 | 1 | $7.6 \cdot 10^{-15}$ |
| 40 | 0.29 | 145.53 | 1 | $7.6 \cdot 10^{-16}$ | 0.58 | 1 | $1.5 \cdot 10^{-14}$ |
| 50 | 0.60 | 658.42 | 1 | $8.1 \cdot 10^{-16}$ | 1.40 | 1 | $1.9 \cdot 10^{-14}$ |
| 100 | 2.50 | - | - | - | 5.73 | 1 | $4.0 \cdot 10^{-14}$ |
| 200 | 18.48 | - | - | - | 39.92 | 1 | $8.3 \cdot 10^{-14}$ |

TABLE 6.1
*Results for the damped mass-spring example; all times are in seconds*

For the special case of matrix square roots we obtain similar results as those of Algorithm 4 in Table A.1 of [9].
Our first example is a problem given from [14, 19] which arises in a damped mass-spring system with $A = I_n$,

$$B = \begin{bmatrix} 20 & -10 & & & & \\ -10 & 30 & -10 & & & \\ & -10 & 30 & -10 & & \\ & & -10 & \ddots & \ddots & \\ & & & & 30 & -10 \\ & & & & -10 & 20 \end{bmatrix}, \quad C = \begin{bmatrix} 15 & -5 & & & & \\ -5 & 15 & -5 & & & \\ & -5 & \ddots & \ddots & & \\ & & \ddots & \ddots & -5 & \\ & & & -5 & 15 & \end{bmatrix}.$$

Since Newton's method with exact line searches obtains an approximation to the minimal solvent in this example [14], our algorithms verify existence and uniqueness of the exact value of minimal solvent in a quite narrow interval matrix. Table 6.1 reports the numerical results obtained for different values of $n$. An isolated dash (-) in Table 6.1 means that the corresponding test needs more than 15 minutes and we didn't wait for the standard Krawczyk operator to be completed.

Our second example arises in a quasi-birth-death process with the following matrices [14, 19]

$$A = \begin{bmatrix} 0 & 0.05 & 0.055 & 0.08 & 0.1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0.22 & 0 & 0 \\ 0 & 0 & 0 & 0.32 & 0.4 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 0.01 & 0.02 & 0.01 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0.04 & -1 & 0 & 0 \\ 0 & 0 & 0.08 & -1 & 0 \\ 0 & 0 & 0 & 0.04 & -1 \end{bmatrix},$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



$$C = \begin{bmatrix} 0.1 & 0.04 & 0.025 & 0.01 & 0 \\ 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.16 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.04 & 0 \end{bmatrix}.$$

Here, both matrices $A$ and $C$ are singular and so we cannot apply Algorithm 4. Even the standard Krawczyk operator fails to verify the answer. Here, we can use the interval iteration (5.1). Choosing $\boldsymbol{X}_0$ to be `intval(zeros(5));` in Intlab notation we obtain after 0.15 seconds an interval matrix with $mr = 9.7 \times 10^{-17}$. In particular the elements of the first row of the obtained enclosure are

$$\boldsymbol{X}_{11} = [0.11186117330535, 0.11186117330536],$$

$$\boldsymbol{X}_{12} = [0.04596260121747, 0.04596260121748],$$

$$\boldsymbol{X}_{13} = [0.02710477934505, 0.02710477934506],$$

$$\boldsymbol{X}_{14} = [0.01026428479283, 0.01026428479284],$$

$$\boldsymbol{X}_{15} = [0.00000000000000, 0.00000000000000].$$

In our last example we choose $A$ to be the identity matrix, and $B$ and $C$ to be the matrices `frank` and `gcdmat`, respectively. These are matrices from Matlab's gallery. We set the size of matrices to $n = 20$. The matrix $B$ is ill-conditioned with a condition number of $1.1 \times 10^{18}$. Both the standard Krawczyk operator and the iterative scheme (5.1) fail to enclose a solvent. However, Algorithm 4 obtains after 0.8 seconds an enclosure with $mr = 2.4 \times 10^{-10}$. Here, we see that Algorithm 4 is not sensitive to the condition of $B$, while both the standard Krawczyk method and the iterative method (5.1) need the matrix $B$ to be well-conditioned.

**Acknowledgment.** The authors would like to offer particular thanks to the referee for his/her many valuable suggestions and constructive comments which improved this paper in several manners, particularly in formulating Theorem 3.1. We are also grateful to Professor Dr. Andreas Frommer for his helpful advice concerning this paper.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



REFERENCES

[1] E. Adams and U. Kulisch. (eds.) *Scientific Computing with Automatic Result Verification*. Vol. 189 of Mathematics in Science and Engineering, Academic Press Inc., Boston, MA, 1993.

[2] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Computer Science and Applied Mathematics, Academic Press, New York, 1983.

[3] M. Binder and M. Hashem Pesaran. Multivariate rational expectations models and macroeconometric modelling: A review and some new results. in *Handbook of Applied Econometrics: Macroeconomics*. M. Hashem Pesaran and M. Wickens. (eds.), Basil Blackwell, pp. 139–187, 1999.

[4] D. Bini, B. Meini, and F. Poloni. From algebraic Riccati equations to unilateral quadratic matrix equations: old and new algorithms. In *Dagstuhl Seminar Proceedings 07461, Numerical Methods for Structured Markov Chains*. 2008.

[5] O. Caprani and K. Madsen. Iterative methods for interval inclusion of fixed points. *BIT Numerical Mathematics*, 18:42–51, 1978.

[6] G.J. Davis. Numerical solution of a quadratic matrix equation. *SIAM Journal on Scientific and Statistical Computing*, 2:164–175, 1981.

[7] G.J. Davis. Algorithm 598: An algorithm to compute solvents of the matrix equation $AX^2 + BX + C = 0$. *ACM Transactions on Mathematical Software*, 9:246–254, 1983.

[8] J.E. Dennis, Jr., J.F. Traub, and R.P. Weber. Algorithms for solvents of matrix polynomials. *SIAM Journal on Numerical Analysis*, 15:523–533, 1978.

[9] A. Frommer and B. Hashemi. Verified computation of square roots of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 31:1279–1302, 2009. Preprint available as technical report BUW-SC 09/2, University of Wuppertal, `(www-ai.math.uni-wuppertal.de/SciComp/preprints/SC0902.pdf)`

[10] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix polynomials*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982. Computer Science and Applied Mathematics.

[11] C.-H. Guo. On a quadratic matrix equation associated with an M-matrix. *IMA Journal of Numerical Analysis*, 23:11–27, 2003.

[12] C.-H. Guo. Numerical solution of a quadratic eigenvalue problem. *Linear Algebra and its Applications*, 385:391–406, 2004.

[13] N.J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia, 2008.

[14] N.J. Higham and H.-M. Kim. Numerical analysis of a quadratic matrix equation. *IMA Journal of Numerical Analysis*, 20:499–519, 2000.

[15] N.J. Higham and H.-M. Kim. Solving a quadratric matrix equation by Newton's method with exact line searches. *SIAM Journal on Matrix Analysis and Applications*, 23:303–316, 2001.

[16] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1994.

[17] R. Kearfott. Interval analysis: Interval fixed point theory. in *Encyclopedia of Optimization*. Vol. 3, Dordrecht, Netherlands, Kluwer, pp. 45-51, 2001.

[18] R.B. Kearfott, M. Nakao, A. Neumaier, S. Rump, S. Shary, and P. van Hentenryck. Standardized notation in interval analysis, 2005. `(www.mat.univie.ac.at/~neum/ms/notation.pdf)`

[19] H.-M. Kim. *Numerical Methods for Solving a Quadratic Matrix Equation*. PhD thesis, University of Manchester, 2000.

[20] H.-M. Kim. Minimization method for solving a quadratic matrix equation. *Kyungpook Mathematical Journal*, 47:239–251, 2007.

[21] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4:187–201, 1969.

[22] R. Krawczyk and A. Neumaier. Interval slopes for rational functions and associated centered

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 519-536, August 2010

ELA



forms. *SIAM Journal on Numerical Analysis*, 22:604–616, 1985.

[23] J.-H. Long, X.-Y. Hu, and L. Zhang. Improved Newton's method with exact line searches to solve quadratic matrix equation. *Journal of Computational and Applied Mathematics*, 222:645–654, 2008.

[24] R.E. Moore. A test for existence of solutions to nonlinear systems. *SIAM Journal on Numerical Analysis*, 14:611–615, 1977.

[25] R.E. Moore, R.B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.

[26] A. Neumaier. *Interval Methods for Systems of Equations*. No. 37 in Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, 1990.

[27] J. Qian and W.-W. Lin. A numerical method for quadratic eigenvalue problems of gyroscopic systems. *Journal of Sound and Vibration*, 306:284–296, 2007.

[28] L.B. Rall. A theory of interval iteration. *Proceedings of the American Mathematical Society*, 86:625–631, 1982.

[29] J. Rohn. VERSOFT: Verification Software in MATLAB/INTLAB. `(uivtx.cs.cas.cz/~rohn/matlab)`

[30] S.M. Rump. A note on epsilon-inflation. *Reliable Computing*, 4:371–375, 1998.

[31] S.M. Rump. Fast and parallel interval arithmetic. *BIT Numerical Mathematics*, 39:534–554, 1999.

[32] S.M. Rump. INTLAB – INTerval LABoratory. in *Developments in Reliable Computing*. T. Csendes. (ed.), Dordrecht, 1999, Kluwer Academic Publishers, pp. 77-104.