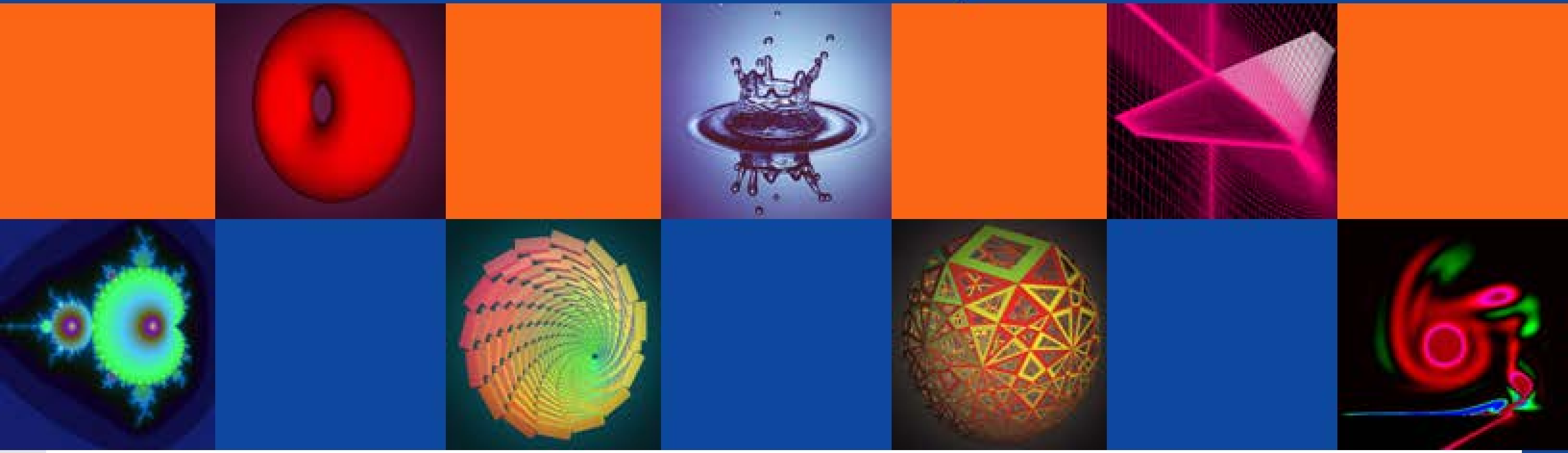

# An Improved Structure-Preserving Doubling Algorithm For a Structured Palindromic Quadratic Eigenvalue Problem

Linzhang Lu
Fei Yuan
Ren-Cang Li

Technical Report 2014-02

# An Improved Structure-preserving Doubling Algorithm for a Structured Palindromic Quadratic Eigenvalue Problem

Linzhang Lu[*] Fei Yuan[†] Ren-Cang Li [‡]

February 1, 2014

**Abstract**

In this paper, we present a numerical method to solve the palindromic quadratic eigenvalue problem (PQEP) $(\lambda^2 A^{\mathrm{T}} + \lambda Q + A)z = 0$ arising from the vibration analysis of high speed trains, where $A,\ Q \in \mathbb{C}^{n\times n}$ have special structures: both $Q$ and $A$ are $m \times m$ block matrices with each block being $k \times k$, and moreover they are complex symmetric, block tridiagonal and block Toeplitz, and also $A$ has only one nonzero block in the $(1, m)$th block position. The method is an improved version of Guo's and Lin's efficient solvent approach [*SIAM J. Matrix Anal. Appl.*, 31 (2010), 2784-2801] which solves the PQEP by computing the so-called *stabilizing solution* to the $mk\times mk$ nonlinear matrix equation $X + A^{\mathrm{T}}X^{-1}A = Q$ via the doubling algorithm. Here, we exploit the fact that the stabilizing solution $X$ differs from $Q$ only in its $(m, m)$th block position and which had also been noted and exploited by Guo and Lin there, too. What distinguishes our method from theirs is that we devise a new nonlinear matrix equation $\widetilde{X} + \widetilde{A}^{\mathrm{T}}\widetilde{X}^{-1}\widetilde{A} = \widetilde{Q}$ of only $k \times k$ in size just for computing the differing block. The new and much smaller matrix equation is also solved by the doubling algorithm at the same speed in terms of the number of doubling iterations as but about 4.8 times faster in flops than the doubling algorithm on the larger matrix equation, and its stabilizing solution $\widetilde{X}$ is used to recover the bigger stabilizing solution $X$. Numerical examples are presented to show the effectiveness of the improved method.

**Key words.** palindromic quadratic eigenvalue problem, nonlinear matrix equation, solvent approach, doubling algorithm

**AMS subject classifications.** 15A24, 65F15, 65F30

## 1 Introduction

The palindromic quadratic eigenvalue problem (PQEP) [11, 12, 15] is to find scalars $\lambda$ and nonzero vectors $z$ such that

$$P(\lambda)z \equiv (\lambda^2 A^{\mathrm{T}} + \lambda Q + A)z = 0, \quad z \neq 0, \tag{1.1}$$

[*]School of Mathematical Science, Xiamen University & School of Mathematics and Computer Science, Guizhou Normal University, P. R. China. E-mail: `lzlu@xmu.edu.cn`, `llz@gznu.edu.cn`. Supported in part by National Natural Science Foundation of China Grant No.10261012.

[†]School of Mathematical Science, Xiamen University, P. R. China. E-mail: `ranma01@163.com`.

[‡]Department of Mathematics, University of Texas at Arlington, P.O. Box 19408, Arlington, TX 76019 E-mail: `rcli@uta.edu`. Supported in part by NSF grants DMS-1115834 and DMS-1317330, and a Research Gift Grant from Intel Corporation.

where $A$ and $Q$ are $n \times n$ (real or complex) matrices and $Q^{\mathrm{T}} = Q$ (complex symmetric). When (1.1) holds for a scalar $\lambda$ and a vector $z \neq 0$, we call $\lambda$ a *quadratic eigenvalue* and $z$ a corresponding *quadratic eigenvector.* Counting multiplicities, it has $2n$ eigenvalues some of which may be infinities[1]. It can be seen that the quadratic eigenvalues of PQEP (1.1) come in reciprocal pairs $\{\lambda, 1/\lambda\}$.

Our focus in this paper is on the PQEP arising from the vibration analysis of high speed trains [4, 7, 14], where

$$Q = K_t + \iota\omega D_t - \omega^2 M_t \in \mathbb{C}^{n\times n}, \tag{1.2}$$

$$A = K_c + \iota\omega D_c - \omega^2 M_c \in \mathbb{C}^{n\times n}, \tag{1.3}$$

$\iota$ is the imaginary unit, $\omega > 0$ is the frequency of the external excitation force, and $K_t$, $D_t$, $M_t$, $K_c$, $D_c$, $M_c$ are real $m \times m$ block matrices with each block being $k \times k$:

$$K_t = \begin{array}{c} \\ k \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{ccccc} k & k & k & \dots & k \end{array} \\ \begin{bmatrix} K_0 & K_1^{\mathrm{T}} & & & \\ K_1 & K_0 & K_1^{\mathrm{T}} & & \\ & K_1 & \ddots & \ddots & \\ & & \ddots & \ddots & K_1^{\mathrm{T}} \\ & & & K_1 & K_0 \end{bmatrix} \end{array}, \quad M_t = \begin{array}{c} \\ k \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{ccccc} k & k & k & \dots & k \end{array} \\ \begin{bmatrix} M_0 & M_1^{\mathrm{T}} & & & \\ M_1 & M_0 & M_1^{\mathrm{T}} & & \\ & M_1 & \ddots & \ddots & \\ & & \ddots & \ddots & M_1^{\mathrm{T}} \\ & & & M_1 & M_0 \end{bmatrix} \end{array}, \tag{1.4}$$

$$K_c = \begin{array}{c} \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} k \\ \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array} \times \begin{array}{c} \begin{array}{cccc} k & \dots & k & k \end{array} \\ \begin{bmatrix} 0 & \dots & 0 & K_1 \end{bmatrix} \end{array}, \quad M_c = \begin{array}{c} \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} k \\ \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array} \times \begin{array}{c} \begin{array}{cccc} k & \dots & k & k \end{array} \\ \begin{bmatrix} 0 & \dots & 0 & M_1 \end{bmatrix} \end{array}, \tag{1.5}$$

$$D_t = c_1 M_t + c_2 K_t, \quad D_c = c_1 M_c + c_2 K_c \quad \text{with } c_1, c_2 > 0, \tag{1.6}$$

and $K_0 = K_0^{\mathrm{T}}$. From (1.2) – (1.6), we see that $Q$ is a complex symmetric, Toeplitz block tridiagonal matrix and $A$ is a very sparse matrix since it has only one nonzero block in its $(1, m)$th block position.

This PQEP (1.1) has been providing much motivation for studying palindromic polynomial eigenvalue problems (see, e.g. [7, 15, 16]). Several numerical methods were proposed [4, 7, 11, 12, 13] and some of them are not limited to PQEP of this form. For the case here, $A$ is of very low rank, and thus PQEP (1.1) has many infinite eigenvalues, creating numerical challenges. Most of the existing methods start by deflating out those infinite eigenvalues.

In [4, 7], the so-called *solvent approach* was explored for numerically solving PQEP (1.1). The approach consists of two steps:

1. Compute the stabilizing solution $\varPhi$ of the matrix equation

$$X + A^{\mathrm{T}} X^{-1} A = Q \tag{1.7}$$

by the doubling algorithm [4], where $A$ and $Q$ are given by (1.2) – (1.6). By the *stabilizing solution*, we mean the spectral radius $\rho(\varPhi^{-1}A) < 1$. Guo and Lin [7]

[1]Infinite eigenvalues are defined through the quadratic eigenvalue 0 of $\lambda^2 P(1/\lambda) = A^{\mathrm{T}} + \lambda Q + \lambda^2 A$.

showed that $P(\lambda)$ with $Q$ and $A$ given by (1.2) – (1.6) has no eigenvalues on the unit circle and thus the stabilizing solution $\varPhi$ exists.

$\varPhi$ is one of many possible solutions of (1.7), called the *solvent matrices*. Any solvent matrix $X$ gives rise to the following factorization for $P(\lambda)$:

$$P(\lambda) = \lambda^2 A^{\mathrm{T}} + \lambda Q + A = (\lambda A^{\mathrm{T}} + X)X^{-1}(\lambda X + A). \tag{1.8}$$

2. Solve the (linear) eigenvalues for matrix pencils $\lambda A^{\mathrm{T}} + \varPhi$ and $\lambda\varPhi + A$. Not the eigenvalues of $\lambda A^{\mathrm{T}} + \varPhi$ and those of $\lambda\varPhi + A$ enjoy the reciprocal relation: if $\mu$ is an eigenvalue of one, then $1/\mu$ is an eigenvalue of the other.

Finally the quadratic eigenvalues of $P(\lambda)$ are the multiset union of the eigenvalues of the two matrix pencils $\lambda A^{\mathrm{T}} + \varPhi$ and $\lambda\varPhi + A$. The eigenvectors of $\lambda\varPhi + A$ are also the quadratic eigenvectors of $P(\lambda)$, but those of $\lambda A^{\mathrm{T}} + \varPhi$ need to be processed to yield the corresponding quadratic eigenvectors of $P(\lambda)$ [7]. This solvent approach works on the whole PQEP directly without deflating out the infinite eigenvalues, and numerical tests suggest that it delivers more accurate numerical solutions than other existing methods [7].

The complete Guo-Lin algorithm [7] is essentially this solvent approach with clever exploitations of the structures in $A$ and $Q$ to dramatically reduce the cost in solving the $mk \times mk$ matrix equation (1.7) by the doubling algorithm. In this paper, we will exploit the structures even further by proposing a new and more efficient implementation for the part of computing the stabilizing solution $\varPhi$. It is made possible by an observation in the structure of any solution $X$ of (1.7): *it differs from $Q$ only in the $(m,m)$th block position.* Thus it may be unnecessary to solve (1.7) but potentially some matrix equation for the $(m,m)$th block alone. We prove that this is indeed the case and devise a new matrix equation

$$\widetilde{X} + \widetilde{A}^{\mathrm{T}}\widetilde{X}^{-1}\widetilde{A} = \widetilde{Q}$$

of $k \times k$ in size only just for determining that block. The new and much smaller matrix equation can also be solved by the doubling algorithm and because of its much smaller size, it can be solved much faster.

We point out that the solvent approach for more general quadratic eigenvalue problems was explored before (see, e.g., [5, 9, 8, 10, 18]).

The rest of this paper is organized as follows. In section 2, we review briefly the doubling algorithm that used in [7] for computing the stabilizing solution $\varPhi$ of (1.7). In section 3, we exploit the structure of the stabilizing solution and devise a $k \times k$ matrix equation whose solution can be used to recover $\varPhi$. In section 4, we outline implementation details about our improved method. Section 5 presents our numerical results comparing the improved method here with the Guo-Lin method in [7]. Finally concluding remarks are given in section 6.

**Notation.** $\mathbb{C}^{n\times m}$ is the set of all $n \times m$ complex matrices, $\mathbb{C}^n = \mathbb{C}^{n\times 1}$, and $\mathbb{C} = \mathbb{C}^1$. $I_n$ (or simply $I$ if its dimension is clear from the context) is the $n \times n$ identity matrix, and $e_j$ is its $j$th column. The superscripts "$\cdot^{\mathrm{T}}$" and "$\cdot^{\mathrm{H}}$" takes the transpose and complex conjugate transpose of a matrix or vector, respectively. We shall also adopt MATLAB-like convention to access the entries of vectors and matrices. Let $i:j$ be the set of integers from $i$ to $j$ inclusive. For a vector $u$ and a matrix $X$, $u_{(j)}$ is $u$'s $j$th entry, $X_{(i,j)}$ is $X$'s $(i,j)$th entry; $X$'s submatrices $X_{(k:\ell,i:j)}$, $X_{(k:\ell,:)}$, and $X_{(:,i:j)}$ consist of intersections of row $k$ to row $\ell$ and column $i$ to column $j$, row $k$ to row $\ell$, and column $i$ to column $j$, respectively.

## 2 The doubling algorithm

The key and novel step of the Guo-Lin solvent approach is to use the stabilizing solution $X = \Phi$ of (1.7) to factorize $P(\lambda)$ as in (1.8). Once (1.8) with $X = \Phi$ is gotten, the QZ algorithm implemented in LAPACK [1] and in MATLAB as `eig(...)` is readily applicable to solve the eigenvalue problems[2] for $\lambda A^{\mathrm{T}} + \Phi$ and $\lambda\Phi + A$ whose combined eigen-solutions gives the full spectral information for PQEP (1.1).

The doubling algorithm [3], Algorithm 2.1, was used to solve the matrix equation (1.7) for its stabilizing solution [4, 7].

**Algorithm 2.1** The doubling algorithm for solving (1.7)

Given $A, Q = Q^{\mathrm{T}} \in \mathbb{C}^{n\times n}$, this algorithm computes a solution of (1.7).

```
1: $A_0 = A$, $X_0 = Q$, $Y_0 = 0$.
2: for $i = 1, 2, \dots,$ do
3:     $A_{i+1} = A_i(X_i - Y_i)^{-1}A_i$;
4:     $X_{i+1} = X_i - A_i^{\mathrm{T}}(X_i - Y_i)^{-1}A_i$;
5:     $Y_{i+1} = Y_i + A_i(X_i - Y_i)^{-1}A_i^{\mathrm{T}}$;
6: end for
7: return $X_i$ as the computed solution at convergence.
```

We mentioned in section 1 that for the PQEP arising from the vibration analysis of high speed trains, (1.7) has the stabilizing solution $\Phi$ which is unique [7]. Once $\Phi$ is computed, the $n$ quadratic eigenvalues of $P(\lambda)$ inside the unique circle can be computed by solving the eigenvalues of $\lambda\Phi + A$, and the other $n$ quadratic eigenvalues which are outside the unique circle are their reciprocals.

It is shown in [7] that $X_i$ generated by Algorithm 2.1 converges to the stabilizing solution $\Phi$ quadratically, and

$$\limsup_{i\to\infty} \sqrt[2^i]{\|X_i - \Phi\|} \le [\rho(\Phi^{-1}A)]^2, \tag{2.1}$$

where $\rho(\cdot)$ is the spectral radius of a matrix, and $\|\cdot\|$ is any matrix norm. As a by-product, $Q - Y_i$ converges to the stabilizing solution $\Psi$ of the complementary matrix equation

$$Y + AY^{-1}A^{\mathrm{T}} = Q \tag{2.2}$$

of (1.7), and

$$\limsup_{i\to\infty} \sqrt[2^i]{\|(Q - Y_i) - \Psi\|} \le [\rho(\Phi^{-1}A)]^2.$$

Because of the quadratic convergence, $X_{i+1} - X_i$ is usually a good indicator of error in $X_i$ as an approximation to $\Phi$. This can be seen as follows. Let $\epsilon = \|X_{i+1} - X_i\|$. For $i$ sufficiently large such that the quadratic convergent behavior shows, i.e.,

$$\|X_{j+2} - X_{j+1}\| \le \alpha\, \|X_{j+1} - X_j\|^2 \quad \text{for } j \ge i,$$

[2]By exploiting the sparsity structure of $A$, Guo and Lin [7] showed how the $mk\times mk$ eigenvalue problems can be solved via two eigenvalue problems of only $k \times k$ in size.

where $\alpha$ is some constant, then

$$\Phi - X_i = \sum_{j=i}^{\infty}(X_{j+1} - X_j) \quad \Rightarrow \quad \|\Phi - X_i\| \le \epsilon + \sum_{j=1}^{\infty} \alpha\epsilon^{2j} = \epsilon + \frac{\alpha\epsilon^2}{1-\epsilon^2} \approx \epsilon.$$

Therefore a reasonable stopping criteria for Algorithm 2.1 is

$$\frac{\|X_{i+1} - X_i\|}{\|X_i\|} \le \texttt{rtol}, \tag{2.3}$$

where `rtol` is a given relative tolerance which, in our numerical tests, was set to a modest multiple of $2^{-52}$, the machine unit roundoff of the IEEE double precision since our tests were carried within MATLAB.

## 3 Structure of the stabilizing solution

It is evident that the success of this solvent approach lies critically in whether the stabilizing solution $\Phi$ can be efficiently computed by the doubling algorithm. In [7], it was shown that, by taking advantage of the special sparse structures of the coefficient matrices $Q$ and $A$, each iterative step in Algorithm 2.1 costs about $(154/3)k^3$, despite all involved matrices are $n \times n$, where $n = mk$.

In this section, we will show that the complexity of computing the solvent can be further reduced to solving a matrix equation having the same form as (1.7) but of only $k \times k$ instead of $n \times n$, the size of original (1.7). This is made possible by the observation that any solvent matrix of (1.7) is the same as $Q$ in all its blocks except the one in the $(m, m)$th block position.

For the ease of presentation, we partition $A$ and $Q$ in (1.2) and (1.3) as

$$A = \begin{array}{c} \\ k \\ (m-2)k \\ k \end{array} \begin{array}{c} \begin{array}{ccc} k & (m-2)k & k \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & A_{13} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right] \end{array}, \quad Q = \begin{array}{c} \\ k \\ (m-2)k \\ k \end{array} \begin{array}{c} \begin{array}{ccc} k & (m-2)k & k \end{array} \\ \left[\begin{array}{ccc} Q_{11} & Q_{12} & 0 \\ Q_{12}^{\mathrm{T}} & Q_{22} & Q_{23} \\ 0 & Q_{23}^{\mathrm{T}} & Q_{33} \end{array}\right] \end{array}. \tag{3.1}$$

The following two theorems are the main theoretical results of this paper.

**Theorem 3.1.** *Let $X$ be a solution to* (1.7)*, and partition $X$ and $X^{-1}$ in the same way as in* (3.1) *for $A$ and $Q$. Then*

$$X = \begin{array}{c} \\ k \\ (m-2)k \\ k \end{array} \begin{array}{c} \begin{array}{ccc} k & (m-2)k & k \end{array} \\ \left[\begin{array}{ccc} Q_{11} & Q_{12} & 0 \\ Q_{12}^{\mathrm{T}} & Q_{22} & Q_{23} \\ 0 & Q_{23}^{\mathrm{T}} & X_{33} \end{array}\right] \end{array}, \tag{3.2}$$

*i.e., $X_{ij} = Q_{ij}$ for $i$, $j = 1, 2, 3$, except for $i = j = 3$, and $X_{33}$ satisfies*

$$X_{33} + A_{13}^{\mathrm{T}}(X^{-1})_{11}A_{13} = Q_{33}, \tag{3.3}$$

*where $(X^{-1})_{11} \in \mathbb{C}^{k\times k}$ is the $(1, 1)$st block of $X^{-1}$.*

*Proof.* Since

$$A = \begin{bmatrix} I_k \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & A_{13} \end{bmatrix},$$

the matrix equation (1.7) becomes

$$X + \begin{bmatrix} 0 \\ 0 \\ A_{13}^{\mathrm{T}} \end{bmatrix} (X^{-1})_{11} \begin{bmatrix} 0 & 0 & A_{13} \end{bmatrix} = Q. \tag{3.4}$$

Equate the corresponding blocks in the two sides of (3.4) to get (3.2) and (3.3). $\square$

Despite being $n \times n$, Theorem 3.1 says that only the $k \times k$ submatrix of $\Phi$ in the lower-right corner needs to be computed, the rest of $\Phi$ is known for free. But (3.3) doesn't have the same form as (1.7) and thus the doubling algorithm is not readily applicable. The next theorem transforms (3.3) into a $k \times k$ matrix equation which does have the same form as (1.7) and thus makes the doubling algorithm applicable.

**Theorem 3.2.** *Under the conditions of* Theorem 3.1*, let*

$$C_{22} = \begin{array}{c c} & \begin{array}{cc} k & (m-2)k \end{array} \\ \begin{array}{c} k \\ (m-2)k \end{array} & \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^{\mathrm{T}} & Q_{22} \end{bmatrix} \end{array}. \tag{3.5}$$

*Suppose* $Q_{11}$ *and* $C_{22}$ *are nonsingular, and view* $C_{22}^{-1} \in C^{(m-1)k\times(m-1)k}$ *as an* $(m-1) \times (m-1)$ *block matrix with each block being* $k \times k$ *and denote by* $(C_{22}^{-1})_{ij}$ *its* $(i,j)$*th block. Then the matrix equation* (3.3) *can be transformed into*

$$\widetilde{X} + \widetilde{A}^{\mathrm{T}} \widetilde{X}^{-1} \widetilde{A} = \widetilde{Q}, \tag{3.6}$$

*where*

$$\widetilde{X} = X_{33} - A_{13}^{\mathrm{T}} (C_{22}^{-1})_{m-1,m-1} A_{13}, \tag{3.7a}$$

$$\widetilde{A} = A_{13} (C_{22}^{-1})_{1,m-1} A_{13}, \tag{3.7b}$$

$$\widetilde{Q} = Q_{33} - A_{13}^{\mathrm{T}} (C_{22}^{-1})_{1,1} A_{13} - A_{13}^{\mathrm{T}} (C_{22}^{-1})_{m-1,m-1} A_{13}. \tag{3.7c}$$

*Proof.* By Theorem 3.1,

$$X = \begin{array}{c c} & \begin{array}{cc} (m-1)k & k \end{array} \\ \begin{array}{c} (m-1)k \\ k \end{array} & \begin{bmatrix} C_{22} & C_{23} \\ C_{23}^{\mathrm{T}} & X_{33} \end{bmatrix} \end{array} \quad \text{with} \quad C_{23} = \begin{array}{c c} & k \\ \begin{array}{c} (m-1)k \\ k \end{array} & \begin{bmatrix} 0 \\ Q_{23} \end{bmatrix} \end{array}. \tag{3.8}$$

It can be seen that $Q_{23} = A_{13}$ by (1.2) – (1.6). Recall[3]

$$X^{-1} = \begin{bmatrix} C_{22} & C_{23} \\ C_{23}^{\mathrm{T}} & X_{33} \end{bmatrix}^{-1} = \begin{bmatrix} C_{22}^{-1} + C_{22}^{-1} C_{23} \widehat{C}_{22}^{-1} C_{23}^{\mathrm{T}} C_{22}^{-1} & -C_{22}^{-1} C_{23} \widehat{C}_{22}^{-1} \\ -\widehat{C}_{22}^{-1} C_{23}^{\mathrm{T}} C_{22}^{-1} & \widehat{C}_{22}^{-1} \end{bmatrix}, \tag{3.9}$$

[3]This is well-known. See, e.g., [6, pp.102-103], [19, Page 23].

where

$$\begin{aligned}
\widehat{C}_{22} &= X_{33} - C_{23}^{\mathrm{T}} C_{22}^{-1} C_{23} \\
&= X_{33} - \begin{bmatrix} 0 & A_{13} \end{bmatrix} C_{22}^{-1} \begin{bmatrix} 0 \\ A_{13}^{\mathrm{T}} \end{bmatrix} \\
&= X_{33} - A_{13}^{\mathrm{T}} (C_{22}^{-1})_{m-1,m-1} A_{13} \\
&= \widetilde{X}.
\end{aligned} \tag{3.10}$$

Noticing that $C_{22}^{-1}$ is symmetric since $C_{22}$ is symmetric, we have

$$\begin{aligned}
(X^{-1})_{1,1} &= (C_{22}^{-1} + C_{22}^{-1} C_{23} \widehat{C}_{22}^{-1} C_{23}^{\mathrm{T}} C_{22}^{-1})_{1,1} \\
&= (C_{22}^{-1})_{1,1} + (C_{22}^{-1} C_{23} \widehat{C}_{22}^{-1} C_{23}^{\mathrm{T}} C_{22}^{-1})_{1,1} \\
&= (C_{22}^{-1})_{1,1} + \left( C_{22}^{-1} \begin{bmatrix} 0 \\ A_{13}^{\mathrm{T}} \end{bmatrix} \right)_{1,1} \widehat{C}_{22}^{-1} \left( \begin{bmatrix} 0 & A_{13} \end{bmatrix} C_{22}^{-1} \right)_{1,1} \\
&= (C_{22}^{-1})_{1,1} + (C_{22}^{-1})_{1,m-1} A_{13}^{\mathrm{T}} \widehat{C}_{22}^{-1} A_{13} (C_{22}^{-1})_{m-1,1}.
\end{aligned} \tag{3.11}$$

Plug in the expression for $\widehat{C}_{22}^{-1}$ in (3.10) into (3.11) and then plug in the resulting expression for $(X^{-1})_{1,1}$ into (3.3) to get (3.6). □

By Theorem 3.2, it suffices to solve the $k \times k$ matrix equation (3.6), instead of the $n \times n$ matrix equation (1.7) that is solved in [7], in order to compute the stabilizing solution $\Phi$. That is where we improve the Guo-Lin solvent approach [7].

## 4 Computations of $\Phi$ and eigenpairs

By Theorems 3.1 and 3.2, to compute the solvent matrix $\Phi$, we need to calculate the matrices $\widetilde{A}$ and $\widetilde{Q}$ in (3.7b) and (3.7c) and then solve the matrix equation (3.6) to recover $X_{33}$ by (3.7a) and thus $\Phi$. Finally, we can solve the eigenvalue problems for $\lambda A^{\mathrm{T}} + \Phi$ and $\lambda \Phi + A$.

### 4.1 Set up (3.6)

For this purpose, we have to compute

$$(C_{22}^{-1})_{1,1}, \quad (C_{22}^{-1})_{1,m-1}, \quad \text{and} \quad (C_{22}^{-1})_{m-1,m-1} \tag{4.1}$$

since $A_{13}$ and $Q_{33}$ are known.

First we note that $C_{22}$ defined in (3.5) is the $(m-1)k \times (m-1)k$ principal submatrix of $Q$ in (1.2). Recall (1.2) – (1.6), and let

$$H_0 = K_0 + i\omega D_0 - \omega^2 M_0 \in \mathbb{C}^{k \times k}, \quad H_1 = K_1 + i\omega D_1 - \omega^2 M_1 \in \mathbb{C}^{k \times k}. \tag{4.2}$$

It follows from (1.2) that

$$C_{22} = \begin{array}{c|ccccc} & k & k & k & \dots & k \\ \hline k & H_0 & H_1^{\mathrm{T}} & & & \\ k & H_1 & H_0 & H_1^{\mathrm{T}} & & \\ k & & H_1 & \ddots & \ddots & \\ \vdots & & & \ddots & \ddots & H_1^{\mathrm{T}} \\ k & & & & H_1 & H_0 \end{array}, \tag{4.3}$$

an $(m-1)\times(m-1)$ block tridiagonal matrix. To compute the three blocks in (4.1), we use the QR decomposition of $C_{22}$ with the Q-factor stored in its factor form, as in [7]. It goes as follows.

1. Initially (step $j=1$), perform the QR decomposition

$$\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = \widehat{U}_1 \times \begin{array}{c|c} & k \\ \hline k & R_{11} \\ k & 0 \end{array},$$

where $\widehat{U}_1 \in \mathbb{C}^{2k\times 2k}$ is unitary and $R_{11}$ is upper triangular, and set

$$\begin{array}{c|cc} & k & k \\ \hline k & R_{12} & R_{13} \\ k & \widehat{R}_{22} & \widehat{R}_{23} \end{array} = \widehat{U}_1^{\mathrm{H}} \begin{bmatrix} H_1^{\mathrm{T}} & 0 \\ H_0 & H_1^{\mathrm{T}} \end{bmatrix}.$$

It can be verified that

$$U_1^{\mathrm{H}} C_{22} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & & \\ & \widehat{R}_{22} & \widehat{R}_{23} & & \\ & H_1 & \ddots & \ddots & \\ & & \ddots & \ddots & H_1^{\mathrm{T}} \\ & & & H_1 & H_0 \end{bmatrix} \quad \text{with} \quad U_1 = \operatorname{diag}(\widehat{U}_1, I_{(m-3)k}). \tag{4.4}$$

2. At the beginning of step $j$ ($j \geq 2$), we have computed

$$U_{j-1}^{\mathrm{H}} \cdots U_2^{\mathrm{H}} U_1^{\mathrm{H}} C_{22} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & & & & \\ & R_{22} & R_{23} & R_{24} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \widehat{R}_{jj} & \widehat{R}_{j,j+1} & & \\ & & & H_1 & \ddots & \ddots & \\ & & & & \ddots & \ddots & H_1^{\mathrm{T}} \\ & & & & & H_1 & H_0 \end{bmatrix}, \tag{4.5}$$

where for $1 \leq i \leq j-1$

$$U_i = \operatorname{diag}(I_{(i-1)k}, \widehat{U}_i, I_{(m-i-2)k}) \tag{4.6}$$

with $\widehat{U}_i \in \mathbb{C}^{2k\times 2k}$ being unitary. Now perform the QR decomposition

$$\begin{bmatrix} \widehat{R}_{jj} \\ H_1 \end{bmatrix} = \widehat{U}_j \times \begin{array}{c c} & k \\ k & \left[\begin{array}{c} R_{jj} \\ 0 \end{array}\right. \\ k & \end{array}\!\!\left.\vphantom{\begin{array}{c} R_{jj} \\ 0 \end{array}}\right],$$

where $\widehat{U}_j \in \mathbb{C}^{2k\times 2k}$ is unitary and $R_{jj}$ is upper triangular, and set

$$\begin{array}{c c c} & k & k \\ k & R_{j,j+1} & R_{j,j+2} \\ k & \widehat{R}_{j+1,j+1} & \widehat{R}_{j+1,j+2} \end{array} = \widehat{U}_j^{\mathrm{H}} \begin{bmatrix} \widehat{R}_{j,j+1} & 0 \\ H_0 & H_1^{\mathrm{T}} \end{bmatrix}, \quad \text{for } j \le m-3,$$

$$\begin{array}{c c} & k \\ k & R_{j,j+1} \\ k & \widehat{R}_{j+1,j+1} \end{array} = \widehat{U}_j^{\mathrm{H}} \begin{bmatrix} \widehat{R}_{j,j+1} \\ H_0 \end{bmatrix}, \qquad \text{for } j = m-2.$$

It can be verified that (4.5) remains valid with $j-1$ replaced by $j$.

At the end of this process, we have computed

$$C_{22} = UR, \tag{4.7a}$$

where

$$U = U_1U_2\cdots U_{m-2}, \tag{4.7b}$$

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & & & & \\ & R_{22} & R_{23} & R_{24} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & R_{m-3,m-3} & R_{m-3,m-2} & R_{m-3,m-1} \\ & & & & R_{m-2,m-2} & R_{m-2,m-1} \\ & & & & & \widehat{R}_{m-1,m-1} \end{bmatrix}, \tag{4.7c}$$

$U_i$ is as in (4.6), $R_{ii}$ are upper triangular, except that $\widehat{R}_{m-1,m-1}$ is a $k\times k$ dense matrix. We point out that in actual implementation, $U$ should not be explicitly computed but stored in its factor form by simply storing all $\widehat{U}_j$ into a $2k$-by-$2k(m-2)$ array, for example.

With (4.7), the three blocks in (4.1) can now be readily computed. Let

$$E_1 = \begin{bmatrix} I_k \\ 0_{(m-2)k\times k} \end{bmatrix} \in \mathbb{C}^{(m-1)q}, \quad E_{m-1} = \begin{bmatrix} 0_{(m-2)k\times k} \\ I_k \end{bmatrix} \in \mathbb{C}^{(m-1)k}.$$

Note $C_{22} = UR = R^{\mathrm{T}}U^{\mathrm{T}}$ since $C_{22}$ is complex symmetric, and thus

$$C_{22}^{-1} = R^{-1}U^{\mathrm{H}} = \bar{U}R^{-\mathrm{T}},$$

where $\bar{U}$ is its complex conjugate. We have

$$(C_{22}^{-1})_{1,1} = E_1^{\mathrm{T}}C_{22}^{-1}E_1 = E_1^{\mathrm{T}}R^{-1}U^{\mathrm{H}}E_1, \tag{4.8}$$

$$(C_{22}^{-1})_{1,m-1} = E_1^{\mathrm{T}} C_{22}^{-1} E_{m-1} = E_1^{\mathrm{T}} \bar{U} R^{-\mathrm{T}} E_{m-1} = E_1^{\mathrm{T}} \bar{U} \begin{bmatrix} 0_{k\times k} \\ \vdots \\ 0_{k\times k} \\ \widehat{R}_{m-1,m-1}^{-\mathrm{T}} \end{bmatrix}, \tag{4.9}$$

$$(C_{22}^{-1})_{m-1,m-1} = E_{m-1}^{\mathrm{T}} C_{22}^{-1} E_{m-1} = E_{m-1}^{\mathrm{T}} \bar{U} R^{-\mathrm{T}} E_{m-1} = E_{m-1}^{\mathrm{T}} \bar{U} \begin{bmatrix} 0_{k\times k} \\ \vdots \\ 0_{k\times k} \\ \widehat{R}_{m-1,m-1}^{-\mathrm{T}} \end{bmatrix}. \tag{4.10}$$

So we need

1. $E_1^{\mathrm{T}} U$, the first block row of $U$; The following piece of pseudo-code in MATLAB-like notation illustrating how it can be computed.

```
**compute** $Z = E_1^{\mathrm{T}} U$:
1  $Z = [(\widehat{U}_1^{\mathrm{H}})_{(1:k,:)}, 0_{k\times(m-2)k}]$;
2  **for** $j = 2 : m-2$
3      $Z_{(:,(j-1)k+1:(j+1)k)} = Z_{(:,(j-1)k+1:jk)} \times (\widehat{U}_j^{\mathrm{H}})_{(1:k,:)}$;
4  **end for**.
```

2. $\widehat{R}_{m-1,m-1}^{-\mathrm{T}}$; It is also needed in computing $E_1^{\mathrm{T}} R^{-1}$ next.

3. $E_1^{\mathrm{T}} R^{-1}$, the first block row of $R^{-1}$; This can be implemented through solving a block triangular system $E_1 = ZR$, where $Z$ is $k \times (m-1)k$.

4. the last $k \times k$ block of $E_{m-1}^{\mathrm{T}} U$; Since

$$E_{m-1}^{\mathrm{T}} U = E_{m-1}^{\mathrm{T}} U_1^{\mathrm{H}} U_2^{\mathrm{H}} \cdots U_{m-2}^{\mathrm{H}} = E_{m-1}^{\mathrm{T}} U_{m-2}^{\mathrm{H}},$$

   the last $k \times k$ block of $E_{m-1}^{\mathrm{T}} U$ is just the bottom-right $k \times k$ submatrix of $\widehat{U}_{m-2}^{\mathrm{H}}$, i.e., $(\widehat{U}_{m-2}^{\mathrm{H}})_{(k+1:2k,k+1:2k)}$.

## 4.2 Solve (3.6) by the doubling algorithm

Having computed the three $k\times k$ blocks in (4.1), the coefficient matrices for (3.6) are readily available. We propose to solve (3.6) by applying Algorithm 2.1 in a straightforward way. For future notation reference, we restate Algorithm 2.1 for (3.6) into Algorithm 4.1.

In [7], it is showed that Algorithm 2.1 produces a convergent sequence $X_i$ that converges to the stabilizing solution $\varPhi$ at the rate of $[\rho(\varPhi^{-1}A)]^2$ as in (2.1). Naturally, we would expect Algorithm 4.1 also produces a convergent sequence $\widetilde{X}_i$ that converges to the solution

$$\widetilde{\varPhi} := \varPhi_{33} - A_{13}^{\mathrm{T}}\,(C_{22}^{-1})_{m-1,m-1}\,A_{13} \tag{4.11}$$

at likely the same rate, where $\varPhi_{33}$ is the bottom-right $k \times k$ block of $\varPhi$, i.e.,

$$\varPhi_{33} = \varPhi_{((m-1)k+1:mk,(m-1)k+1:mk)}.$$

**Algorithm 4.1** The doubling algorithm for solving (3.6)

Given $\widetilde{A}$, $\widetilde{Q} = \widetilde{Q}^{\mathrm{T}} \in \mathbb{C}^{k\times k}$, this algorithm computes a solution of (3.6).

```
1: $\widetilde{A}_0 = \widetilde{A}$, $\widetilde{X}_0 = \widetilde{Q}$, $\widetilde{Y}_0 = 0$.
2: for $i = 1, 2, \ldots,$ do
3:     $\widetilde{A}_{i+1} = \widetilde{A}_i(\widetilde{X}_i - \widetilde{Y}_i)^{-1}\widetilde{A}_i$;
4:     $\widetilde{X}_{i+1} = \widetilde{X}_i - \widetilde{A}_i^{\mathrm{T}}(\widetilde{X}_i - \widetilde{Y}_i)^{-1}\widetilde{A}_i$;
5:     $\widetilde{Y}_{i+1} = \widetilde{Y}_i - \widetilde{A}_i(\widetilde{X}_i - \widetilde{Y}_i)^{-1}\widetilde{A}_i^{\mathrm{T}}$;
6: end for
7: return $\widetilde{X}_i$ as the computed solution at convergence.
```

**Lemma 4.1.** *$\widetilde{\Phi}$ defined by* (4.11) *is the stabilizing solution of* (3.6) *and, moreover,*

$$\rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big) = \rho(\Phi^{-1}A) < 1.$$

*Proof.* Recall, by Theorem 3.1,

$$\Phi = \begin{array}{c c} & \begin{array}{cc} (m-1)k & k \end{array} \\ \begin{array}{c} (m-1)k \\ k \end{array} & \begin{bmatrix} C_{22} & C_{23} \\ C_{23}^{\mathrm{T}} & \Phi_{33} \end{bmatrix} \end{array}. \tag{4.12}$$

We note all blocks of $A$ as an $m \times m$ block matrix are zeros, except its $(1, m)$th block $A_{(1:k,(m-1)k+1:mk)} =: A_{13}$, where $A_{13}$ is as defined in (3.1). Thus

$$\Phi^{-1}A = \begin{array}{c c} & \begin{array}{cc} (m-1)k & k \end{array} \\ \begin{array}{c} (m-1)k \\ k \end{array} & \begin{bmatrix} 0 & \cdots \\ 0 & (\Phi^{-1})_{((m-1)k+1:mk,1:k)}A_{13} \end{bmatrix} \end{array},$$

where $(\Phi^{-1})_{((m-1)k+1:mk,1:k)}$ is the last $k \times k$ block in the first block column of $\Phi^{-1}$. This implies

$$\rho(\Phi^{-1}A) = \rho((\Phi^{-1})_{((m-1)k+1:mk,1:k)}A_{13}). \tag{4.13}$$

For the same reason as for (3.9), we have

$$\Phi^{-1} = \begin{bmatrix} C_{22}^{-1} + C_{22}^{-1}C_{23}\widehat{C}_{22}^{-1}C_{23}^{\mathrm{T}}C_{22}^{-1} & -C_{22}^{-1}C_{23}\widehat{C}_{22}^{-1} \\ -\widehat{C}_{22}^{-1}C_{23}^{\mathrm{T}}C_{22}^{-1} & \widehat{C}_{22}^{-1} \end{bmatrix},$$

where $\widehat{C}_{22} = \Phi_{33} - A_{13}\,(C_{22}^{-1})_{m-1,m-1}\,A_{13}^{\mathrm{T}}$ by (3.10). Therefore

$$\begin{aligned} (\Phi^{-1})_{((m-1)k+1:mk,1:k)} &= -\widehat{C}_{22}^{-1}(C_{23}^{\mathrm{T}}C_{22}^{-1})_{1,1} \\ &= -[\Phi_{33} - A_{13}\,(C_{22}^{-1})_{m-1,m-1}\,A_{13}^{\mathrm{T}}]^{-1}A_{13}(C_{22}^{-1})_{m-1,1}, \\ (\Phi^{-1})_{((m-1)k+1:mk,1:k)}A_{13} &= -[\Phi_{33} - A_{13}\,(C_{22}^{-1})_{m-1,m-1}\,A_{13}^{\mathrm{T}}]^{-1}A_{13}(C_{22}^{-1})_{m-1,1}A_{13} \\ &= -[\Phi_{33} - A_{13}\,(C_{22}^{-1})_{m-1,m-1}\,A_{13}^{\mathrm{T}}]^{-1}\widetilde{A} \\ &= -\widetilde{\Phi}^{-1}\widetilde{A}. \end{aligned}$$

Therefore, $\rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big) = \rho((\Phi^{-1})_{((m-1)k+1:mk,1:k)}A_{13}) = \rho(\Phi^{-1}A) < 1$ by (4.13), i.e., $\widetilde{\Phi}$ is the stabilizing solution of (3.6). □

We now show that Algorithm 4.1 will not break down, and $\widetilde{X}_i$ converges quadratically to the stabilizing solution $\widetilde{\varPhi}$ of (3.6) under certain nonsingularity assumption. Let $\mathscr{T}_i(X, Z, Y)$ be $i \times i$ block tridiagonal Toeplitz matrix

$$\mathscr{T}_i(X,Z,Y) := \begin{bmatrix} Z & X & & & \\ Y & Z & X & & \\ & \ddots & \ddots & \ddots & \\ & & Y & Z & X \\ & & & Y & Z \end{bmatrix} \in \mathbb{C}^{ik\times ik},$$

defined for any given $X$, $Y$, $Z \in \mathbb{C}^{k\times k}$. In particular, $\mathscr{T}_1(X, Z, Y) = Z$. The complementary equation of (3.6) is

$$\widetilde{Y} + \widetilde{A}\widetilde{Y}^{-1}\widetilde{A}^{\mathrm{T}} = \widetilde{Q}. \tag{4.14}$$

**Theorem 4.1.** *Let $\widetilde{A}$ and $\widetilde{Q}$ be given by* (3.7b) *and* (3.7c). *Let $\widetilde{\varPhi}$ be the stabilizing solution of* (3.6) *and $\widetilde{\varPsi}$ be the stabilizing solution of the complementary equation* (4.14). *Suppose $\mathscr{T}_{2^\ell-1}(-\widetilde{A}^{\mathrm{T}}, \widetilde{Q}, -\widetilde{A})$ is invertible for all $\ell \geq 1$. Then*

*(a) The sequences $\{\widetilde{A}_i\}$, $\{\widetilde{X}_i\}$, $\{\widetilde{Y}_i\}$ in* Algorithm 4.1 *are well-defined, and $\widetilde{X}_i$ and $\widetilde{Y}_i$ are complex symmetric;*

*(b) $\widetilde{X}_i$ converges to $\widetilde{\varPhi}$ quadratically, $\widetilde{A}_i$ converges to* 0 *quadratically, and $\widetilde{Q}-\widetilde{Y}_i$ converges to $\widetilde{\varPsi}$ quadratically. Moreover,*

$$\limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{X}_i - \widetilde{\varPhi}\|} \leq [\rho(\widetilde{\varPhi}^{-1}\widetilde{A})]^2, \qquad \limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{A}_i\|} \leq \rho(\widetilde{\varPhi}^{-1}\widetilde{A}),$$
$$\limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{Q} - \widetilde{Y}_i - \widetilde{\varPsi}\|} \leq [\rho(\widetilde{\varPhi}^{-1}\widetilde{A})]^2,$$

*where $\|\cdot\|$ is any matrix norm.*

*Proof.* Let $Z_i = \widetilde{X}_i - \widetilde{Y}_i$. Then the sequence $\{Z_i\}$ satisfies

$$Z_{i+1} = Z_i - \widetilde{A}_i^{\mathrm{T}} Z_i^{-1}\widetilde{A}_i - \widetilde{A}_i Z_i^{-1}\widetilde{A}_i^{\mathrm{T}} \tag{4.15}$$

with $Z_0 = \widetilde{Q}$. Similarly to [2, Theorem 13 and also (9)], we can prove that $Z_i$ is nonsingular for all $i \geq 0$. In fact, we will prove a stronger statement:

$$\boxed{\mathscr{T}_{2^\ell-1}(-\widetilde{A}_i^{\mathrm{T}}, Z_i, -\widetilde{A}_i) \text{ for all } i \geq 0 \text{ and } \ell \geq 1 \text{ are nonsingular.}} \tag{4.16}$$

It is stronger than $Z_i$ being nonsingular for all $i$ because $Z_i = \mathscr{T}_{2^\ell-1}(-\widetilde{A}_i^{\mathrm{T}}, Z_i, -\widetilde{A}_i)$ for $\ell = 1$. We proceed by induction on $i$. By the assumption of the theorem,

$$\mathscr{T}_{2^\ell-1}(-\widetilde{A}_0^{\mathrm{T}}, Z_0, -\widetilde{A}_0) = \mathscr{T}_{2^\ell-1}(-\widetilde{A}^{\mathrm{T}}, \widetilde{Q}, -\widetilde{A}) \quad \text{for all } \ell \geq 0$$

are nonsingular. The statement (4.16) holds for $i = 0$.

Suppose it holds for $i$. We now prove it must hold for $i+1$, i.e.,

$$\mathscr{T}_{2^\ell-1}(-\widetilde{A}_{i+1}^{\mathrm{T}}, Z_{i+1}, -\widetilde{A}_{i+1}) \quad \text{for all } \ell \geq 1 \text{ are nonsingular.} \tag{4.17}$$

To this end, for any given matrix $X$ and $Y$, we introduce the following notations: $\mathscr{D}_j(X) = \operatorname{diag}(\underbrace{X, \dots, X}_{j})$, $\mathscr{L}_j(X,Y)$ is the $(j+1)\times j$ block lower bidiagonal matrix with $X$ on the main-diagonal and $Y$ on the sub-diagonal, and $\mathscr{U}_j(X,Y)$ is the $j \times (j+1)$ block upper bidiagonal matrix with $X$ on the main diagonal and $Y$ on the sup-diagonal.

Applying the even-odd block row-and-column permutation

$$[1, 3, \dots, 2^{\ell+1}-1, 2, 4, \dots, 2^{\ell+1}-2]$$

to $\mathscr{T}_{2^{\ell+1}-1}(-\widetilde{A}_i^{\mathrm{T}}, Z_i, -\widetilde{A}_i)$ yields

$$\begin{bmatrix} \mathscr{D}_{2^\ell}(Z_i) & \mathscr{L}_{2^\ell-1}(-\widetilde{A}_i^{\mathrm{T}}, -\widetilde{A}_i) \\ \mathscr{U}_{2^\ell-1}(-\widetilde{A}_i, -\widetilde{A}_i^{\mathrm{T}}) & \mathscr{D}_{2^\ell-1}(Z_i) \end{bmatrix}.$$

The Schur complement of $\mathscr{D}_{2^\ell}(Z_i)$, obtained by one step of block Gaussian elimination on this matrix, is

$$\begin{aligned}
&\mathscr{D}_{2^\ell-1}(Z_i) - \mathscr{U}_{2^\ell-1}(-\widetilde{A}_i, -\widetilde{A}_i^{\mathrm{T}})\mathscr{D}_{2^\ell}(Z_i^{-1})\mathscr{L}_{2^\ell-1}(-\widetilde{A}_i^{\mathrm{T}}, -\widetilde{A}_i) \\
&= \begin{bmatrix} Z_i & & \\ & \ddots & \\ & & Z_i \end{bmatrix} - \begin{bmatrix} -\widetilde{A}_i & -\widetilde{A}_i^{\mathrm{T}} & & \\ & \ddots & \ddots & \\ & & -\widetilde{A}_i & -\widetilde{A}_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} Z_i^{-1} & & \\ & \ddots & \\ & & Z_i^{-1} \end{bmatrix} \begin{bmatrix} -\widetilde{A}_i^{\mathrm{T}} & & \\ -\widetilde{A}_i & \ddots & \\ & \ddots & -\widetilde{A}_i^{\mathrm{T}} \\ & & -\widetilde{A}_i \end{bmatrix} \\
&= \begin{bmatrix} Z_{i+1} & -\widetilde{A}_i^{\mathrm{T}} Z_i^{-1} \widetilde{A}_i^{\mathrm{T}} & & & \\ -\widetilde{A}_i Z_i^{-1} \widetilde{A}_i & Z_{i+1} & -\widetilde{A}_i^{\mathrm{T}} Z_i^{-1} \widetilde{A}_i^{\mathrm{T}} & & \\ & \ddots & \ddots & \ddots & \\ & & -\widetilde{A}_i Z_i^{-1} \widetilde{A}_i & Z_{i+1} & -\widetilde{A}_i^{\mathrm{T}} Z_i^{-1} \widetilde{A}_i^{\mathrm{T}} \\ & & & -\widetilde{A}_i Z_i^{-1} \widetilde{A}_i & Z_{i+1} \end{bmatrix}
\end{aligned}$$

which is $\mathscr{T}_{2^\ell-1}(-\widetilde{A}_{i+1}^{\mathrm{T}}, Z_{i+1}, -\widetilde{A}_{i+1})$ and must be nonsingular because, by the inductive assumption, both $\mathscr{T}_{2^{\ell+1}-1}(-\widetilde{A}_i^{\mathrm{T}}, Z_i, -\widetilde{A}_i)$ and $\mathscr{D}_{2^\ell}(Z_i)$ are nonsingular. So (4.17) holds, i.e., (4.16) holds for $i+1$. This completes the inductive proof.

It is evident that $\{\widetilde{X}_i\}$ and $\{\widetilde{Y}_i\}$ are complex symmetric since $\widetilde{Q}$ is complex symmetric. This proves item (a).

We now prove item (b). It can be verified that

$$N_0 \begin{bmatrix} I \\ \widetilde{\Phi} \end{bmatrix} = L_0 \begin{bmatrix} I \\ \widetilde{\Phi} \end{bmatrix} \widetilde{\Phi}^{-1}\widetilde{A} \quad \text{with} \quad N_0 = \begin{bmatrix} \widetilde{A} & 0 \\ \widetilde{Q} & -I \end{bmatrix}, \quad L_0 = \begin{bmatrix} 0 & I \\ \widetilde{A}^{\mathrm{T}} & 0 \end{bmatrix}. \tag{4.18}$$

Define the sequences $\{N_i\}$ and $\{L_i\}$ by

$$N_i = \begin{bmatrix} \widetilde{A}_i & 0 \\ \widetilde{X}_i & -I \end{bmatrix}, \quad L_i = \begin{bmatrix} -\widetilde{Y}_i & I \\ \widetilde{A}_i^{\mathrm{T}} & 0 \end{bmatrix}. \tag{4.19}$$

By the arguments in [3], we have for each $i \geq 0$

$$N_i \begin{bmatrix} I \\ \widetilde{\Phi} \end{bmatrix} = L_i \begin{bmatrix} I \\ \widetilde{\Phi} \end{bmatrix} \big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}. \tag{4.20}$$

Substituting (4.19) into (4.20) yields

$$\widetilde{A}_i = (\widetilde{\Phi} - \widetilde{Y}_i)\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}, \quad \widetilde{X}_i - \widetilde{\Phi} = \widetilde{A}_i^{\mathrm{T}}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}. \tag{4.21}$$

Similarly we have

$$\widehat{N}_0 \begin{bmatrix} I \\ \widetilde{\Psi} \end{bmatrix} = \widehat{L}_0 \begin{bmatrix} I \\ \widetilde{\Psi} \end{bmatrix} \widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}} \quad \text{with} \quad \widehat{N}_0 = \begin{bmatrix} \widetilde{A}^{\mathrm{T}} & 0 \\ \widetilde{Q} & -I \end{bmatrix}, \quad \widehat{L}_0 = \begin{bmatrix} 0 & I \\ \widetilde{A} & 0 \end{bmatrix}.$$

The pencil $\widehat{N}_0 - \lambda\widehat{L}_0$ is a linearization of $\lambda^2\widetilde{A} - \lambda\widetilde{Q} + \widetilde{A}^{\mathrm{T}}$, which has the same eigenvalues as $\lambda^2\widetilde{A}^{\mathrm{T}} - \lambda\widetilde{Q} + \widetilde{A}$. Thus $\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}$ and $\widetilde{\Phi}^{-1}\widetilde{A}$ have the same eigenvalues, and hence

$$\rho\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big) = \rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big).$$

Run Algorithm 4.1 with the inputs $\widetilde{A}^{\mathrm{T}}$ and $\widetilde{Q}$ to yield new $\widetilde{A}$-, $\widetilde{X}$-, and $\widetilde{Y}$-sequences which we denote by $\widehat{A}_i$, $\widehat{X}_i$, and $\widehat{Y}_i$, respectively. It can be verified (by induction, for example) that

$$\widehat{A}_i = \widetilde{A}_i^{\mathrm{T}}, \quad \widehat{Y}_i = \widetilde{Q} - \widetilde{X}_i, \quad \widehat{X}_i = \widetilde{Q} - \widetilde{Y}_i. \tag{4.22}$$

Indeed, the relations in (4.22) are true for $i = 0$. Assuming (4.22) for $i$, we have

$$\begin{aligned} \widehat{X}_{i+1} &= \widehat{X}_i - \widehat{A}_i^{\mathrm{T}}(\widehat{X}_i - \widehat{Y}_i)^{-1}\widehat{A}_i \\ &= \widetilde{Q} - \widetilde{Y}_i - \widetilde{A}_i(\widetilde{X}_i - \widetilde{Y}_i)^{-1}\widetilde{A}_i^{\mathrm{T}} \\ &= \widetilde{Q} - \widetilde{Y}_{i+1}, \end{aligned}$$

and similarly we have $\widehat{A}_{i+1} = \widetilde{A}_{i+1}^{\mathrm{T}}$ and $\widehat{Y}_{i+1} = \widetilde{Q} - \widetilde{X}_{i+1}$. Define the sequences $\{\widehat{N}_i\}$ and $\{\widehat{L}_i\}$ by

$$\widehat{N}_i = \begin{bmatrix} \widehat{A}_i & 0 \\ \widehat{X}_i & -I \end{bmatrix}, \quad \widehat{L}_i = \begin{bmatrix} -\widehat{Y}_i & I \\ \widehat{A}_i^{\mathrm{T}} & 0 \end{bmatrix}.$$

Then for each $i \geq 0$

$$\widehat{N}_i \begin{bmatrix} I \\ \widetilde{\Psi} \end{bmatrix} = \widehat{L}_i \begin{bmatrix} I \\ \widetilde{\Psi} \end{bmatrix} \big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}. \tag{4.23}$$

By (4.23) and (4.22), we have

$$\widetilde{A}_i^{\mathrm{T}} = (\widetilde{\Psi} - \widehat{Y}_i)\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}, \quad \widehat{X}_i - \widetilde{\Psi} = \widetilde{A}_i\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}. \tag{4.24}$$

By (4.21), (4.24) and (4.22), we have

$$\widetilde{X}_i - \widetilde{\Phi} = \widetilde{A}_i^{\mathrm{T}}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}$$

$$
\begin{aligned}
&= (\widetilde{\Psi} - \widehat{Y}_i)\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}\\
&= \big[\widetilde{X}_i - \widetilde{\Phi} + (\widetilde{\Phi} + \widetilde{\Psi} - \widetilde{Q})\big]\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}
\end{aligned}
$$

from which we conclude

$$
(\widetilde{X}_i - \widetilde{\Phi})\big[I - \big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}\big] = (\widetilde{\Phi} + \widetilde{\Psi} - \widetilde{Q})\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)^{2^i}\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)^{2^i}.
$$

It follows that

$$
\limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{X}_i - \widetilde{\Phi}\|} \le \rho\big(\widetilde{\Psi}^{-1}\widetilde{A}^{\mathrm{T}}\big)\rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big) = [\rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)]^2 < 1.
$$

So $\widetilde{X}_i$ converges to $\widetilde{\Phi}$ quadratically. Then by (4.22), we know $\widehat{Y}_i$ is uniformly bounded in $i$ and thus by the first equation in (4.24)

$$
\limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{A}_i\|} \le \rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big) < 1,
$$

i.e., $\widetilde{A}_i$ converges to 0 quadratically. By the second equation in(4.24) and (4.22), we get

$$
\limsup_{i\to\infty} \sqrt[2^i]{\|\widetilde{Q} - \widetilde{Y}_i - \widetilde{\Psi}\|} \le [\rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big)]^2 < 1,
$$

i.e., $\widetilde{Q} - \widetilde{Y}_i$ converges to $\widetilde{\Psi}$ quadratically. This completes the proof of item (b). $\square$

Because of the quadratic convergence claim in this theorem, we can use (2.3) with $X_i$ and $X_{i+1}$ replaced by $\widetilde{X}_i$ and $\widetilde{X}_{i+1}$, respectively, as the stopping criteria for Algorithm 4.1. For our numerical test in the next section, the spectral norm $\|\cdot\|_2$ is used.

## 4.3 Solve the eigenvalue problem for $P(\lambda)$

At this point, we have recovered the stabilizing solution $\Phi$ for (1.7) through solving (3.6) by Algorithm 4.1. The PQEP (1.1) arising from the vibration analysis of high speed trains is now transformed to the eigenvalue problems for $\lambda A^{\mathrm{T}} + \Phi$ and $\lambda\Phi + A$.

Recall (4.12) and $C_{22} = UR$ in (4.7). We have

$$
\begin{bmatrix} U & \\ & I_k \end{bmatrix}^{\mathrm{H}} \Phi = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{R} & U^{\mathrm{H}}C_{23} \\ \hline 0_{k\times(m-2)k} & \multicolumn{1}{|c|}{H_1} & \Phi_{33} \end{array}\right]
$$

and notice $U^{\mathrm{H}}C_{23} = U_{m-2}^{\mathrm{H}}C_{23}$. Now similarly to what we did in subsection 4.1, we find a unitary matrix $U_{m-1} = \operatorname{diag}(I_{(m-2)k}, \widehat{U}_{m-1})$ so that $U_{m-1}^{\mathrm{H}}\operatorname{diag}(U^{\mathrm{H}}, I_k)\,\Phi$ is block upper triangular, where $\widehat{U}_{m-1} \in \mathbb{C}^{2k\times 2k}$. On the other hand, only the last $k$ columns of $U_{m-1}^{\mathrm{H}}\operatorname{diag}(U^{\mathrm{H}}, I_k)\,A$ are possibly nonzero. Thus the eigenvalue problem for $\lambda\Phi + A$ is reduced to the one for $\lambda Z_1 + Z_2$, where $Z_1$ and $Z_2$ are the right-bottom $k\times k$ blocks of $U_{m-1}^{\mathrm{H}}\operatorname{diag}(U^{\mathrm{H}}, I_k)\,\Phi$ and $U_{m-1}^{\mathrm{H}}\operatorname{diag}(U^{\mathrm{H}}, I_k)\,A$, respectively. The rest, i.e., recovering the eigenvectors for (1.1) from those of the $k\times k$ eigenvalue problem for $\lambda Z_1 + Z_2$, is the same as in [7]. We omit the rest of the detail.

Table 5.1: Comparison of cost ($m \geq 2$)

| | before DA iteration | each DA iteration | finding eigenpairs |
|---|---|---|---|
| SDA_GL | $\frac{113}{3}mk^3$ | $\frac{154}{3}k^3$ | $14mk^3$ |
| SDA_LYL | $\frac{113}{3}(m-1)k^3$ | $\frac{32}{3}k^3$ | $14mk^3 + \frac{86}{3}k^3$ |

## 5 Numerical experiments

For the comparison purpose, we will identify our implementation as SDA_LYL and the one in [7] as SDA_GL. Since SDA_GL has been shown in [7] to have much better accuracy than earlier existing methods, we will only compare ours to SDA_GL.

We begin by estimating the flop counts for SDA_LYL. The QR decomposition of $C_{22}$ requires about $\frac{86}{3}(m-1)k^3$ flops; computing the three blocks in (4.1) require $4k^3$ and $(9m-5)k^3$; each iteration of the doubling algorithm for (3.6) requires about $8k^3$ flops; recovering $\varPhi$ and finding all eigenpairs requires about $14mk^3 + \frac{86}{3}k^3$ flops. Table 5.1 summarizes the flop counts for SDA_LYL as well as those for SDA_GL taken from [7]. It is evident that the save in our new implementation comes from solving (3.6) which is of $k \times k$ instead of (1.7) which is of $mk \times mk$. In fact, as far as solving (3.6) and (1.7) by the doubling algorithm is concerned, ours on (3.6) is $154/32 = 4.8$ times faster than theirs on (1.7). But overall, while SDA_LYL is always faster, its speed potential gradually drops as $m$ increases. In fact, according to Table 5.1, the flops ratio is

$$\frac{\text{SDA\_GL}}{\text{SDA\_LYL}} = \frac{165m + 154\ell}{165m + 32\ell - 27}, \tag{5.1}$$

assuming the doubling algorithm takes $\ell$ iterative steps to finish. It decreases as $m$ increases. For example, with $\ell = 10$, the ratio (5.1) is 3.00 at $m = 2$ and decreases to 1.14 at $m = 50$.

Numerically, we tested SDA_GL and SDA_LYL on three sets of test data[4], generated by a finite element method, with

$$(k, m) = (159, 11), \quad (303, 19), \quad (705, 51), \tag{5.2}$$

respectively. The matrices $Q = K_t + \imath\omega D_t - \omega^2 M_t$ and $A = K_c + \imath\omega D_c - \omega^2 M_c$, where $M_t$, $M_c$, $K_t$ and $K_c$ are as in (1.4) – (1.6), and $\omega > 0$. All numerical experiments are carried out within MATLAB 7.0 with machine unit roundoff $2^{-52} \approx 2.22 \times 10^{-16}$.

In all cases, the doubling algorithm on (3.6) shows rapid convergence, as expected, because in theory the doubling algorithm on (1.7) and on the transformed (3.6) has the same rate of convergence. Table 5.2 displays the spectral radiuses

$$\rho = \rho(\varPhi^{-1}A) = \rho(\widetilde{\varPhi}^{-1}\widetilde{A})$$

for the three pairs of $(k, m)$ in (5.2) and for

$$\omega = 100, \quad 1000, \quad 3000, \quad 5000,$$

[4]We thank Prof. Wen-Wei Lin of National Chiao Tung University, Taiwan and Prof. Tiexiang Li of Southeast University, China for generously providing us with the data and their code.

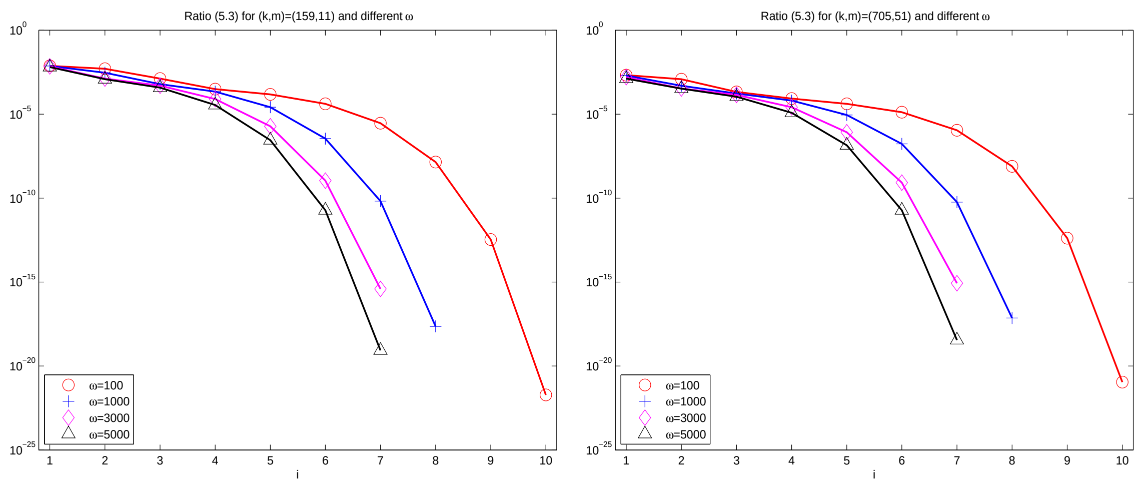

Figure 5.1: $\|\widetilde{X}_{i+1} - \widetilde{X}_i\|_2 / \|\widetilde{X}_i\|_2$ for $(k,m) = (159, 11)$ (*left* plot) and $(705, 51)$ (*right* plot)

Table 5.2: $\rho = \rho(\Phi^{-1}A) = \rho(\widetilde{\Phi}^{-1}\widetilde{A})$

| $(k,m) \setminus \omega$ | 100 | 1000 | 3000 | 5000 |
|---|---|---|---|---|
| $(159, 11)$ | 0.9593 | 0.8745 | 0.7925 | 0.7406 |
| $(303, 19)$ | 0.9307 | 0.7933 | 0.6692 | 0.5953 |
| $(705, 51)$ | 0.9622 | 0.8831 | 0.8060 | 0.7569 |

respectively. For illustrating the convergence history of the doubling algorithm, Figure 5.1 plots the ratio

$$\frac{\|\widetilde{X}_{i+1} - \widetilde{X}_i\|_2}{\|\widetilde{X}_i\|_2}, \tag{5.3}$$

where $\widetilde{X}_i$ is defined by Algorithm 4.1 for the first and last pair of $(k, m)$ in (5.2). What we can see from the figure is that in no more than 10 doubling iterations, this ratio reaches $O(10^{-16})$ or much less. There is no significant difference in the numbers of doubling iterations for different values of $(k, m)$, but for the larger $\omega$, $\rho$ becomes much smaller and consequently fewer doubling iterations are recorded.

Lastly, we assess accuracies of an approximate eigenpair $(\lambda, z)$ of $P(\lambda)$ by the relative residual

$$\text{RRes} := \frac{\|\lambda^2 A^{\mathrm{T}} z + \lambda Q z + A z\|_2}{(|\lambda|^2 \|A\|_{\mathrm{F}} + |\lambda| \|Q\|_{\mathrm{F}} + \|A\|_{\mathrm{F}}) \|z\|_2} \tag{5.4}$$

as in [7], where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. As in Figure 5.1, we use the first and last pair of $(k, m)$ in (5.2) as examples and also for $\omega = 1000$ only. We plot this RRes for all approximate eigenpairs of $P(\lambda)$ in Figure 5.2. These RRes for SDA_LYL and SDA_GL are indistinguishable.

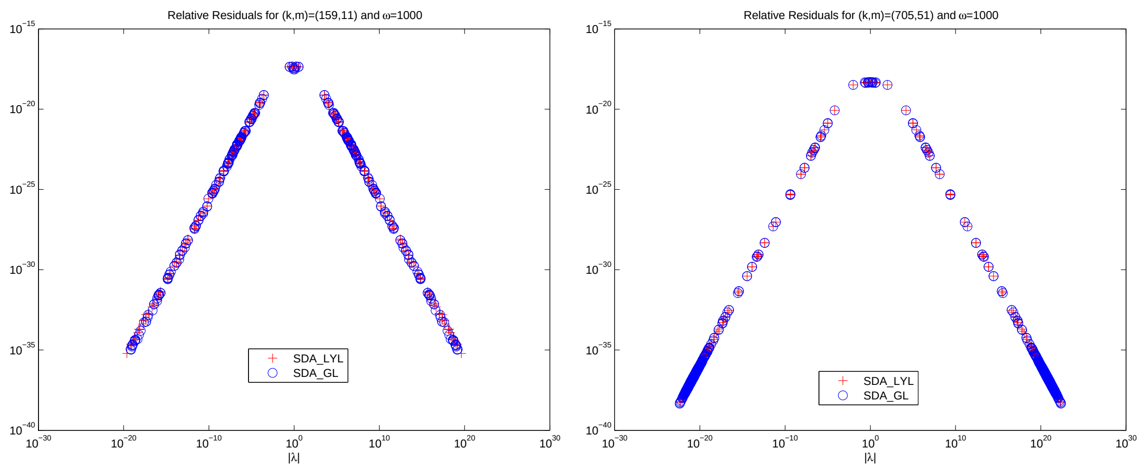

Figure 5.2: RRes for approximate eigenpairs when $(k, m) = (159, 11)$ (*left* plot) and $(705, 51)$ (*right* plot)

## 6 Concluding remarks

We have presented a numerical method to solve the palindromic quadratic eigenvalue problem (1.1) arising from the vibration analysis of high speed trains [4, 7, 14]. The method is based on the Guo-Lin method [7] but with an improvement in the part of the application of the doubling algorithm: we solve a $k \times k$ nonlinear matrix equation (3.6) while Guo and Lin solved an $mk \times mk$ nonlinear matrix equation (1.7). Despite their clever effort in exploiting the structures in $A$ and $Q$, each doubling iteration in [7] is about 4.8 times as expensive as here for large $k$.

Numerical tests suggest that the quality of computed eigenpairs of $P(\lambda)$ by either the original Guo-Lin method and its improved version here is about the same.

So far, we have focused on PQEP (1.1) from the vibration analysis of high speed trains. The idea here, as well as the one in [7], is easily carried over to the case that $Q$ is block tridiagonal but not necessarily block Toeplitz, provided that $P(\lambda)$ has no eigenvalues on the the unit circle and Algorithm 4.1 does not breakdown, i.e., all inverses exist. Other extensions are conceivably possible. We omit the detail.

## References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, 3rd edition, 1999.

[2] D.A. Bini, L. Gemignani, and B. Meini. Computations with infinite Toeplitz matrices and polynomials. *Linear Algebra Appl.*, 343-344:21–61, 2002.

[3] Chun-Yueh Chiang, Eric King-Wah Chu, Chun-Hua Guo, Tsung-Ming Huang, Wen-Wei Lin, and Shu-Fang Xu. Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. *SIAM J. Matrix Anal. Appl.*, 31(2):227–247, 2009.

[4] Eric King-Wah Chu, Tsung-Min Hwang, Wen-Wei Lin, and Chin-Tien Wu. Vibration of fast trains, palindromic eigenvalue problems and structure-preserving doubling algorithms. *J. Comput. Appl. Math.*, 219:237–252, 2008.

[5] G. Davis. Numerical solution of a quadratic matrix equation. *SIAM J. Sci. Statist. Comput.*, 2(2):164–175, 1981.

[6] V. N. Faddeeva. *Computational methods of linear algebra.* Dover Publications, New York, 1959. Translated from the Russian by Curtis D. Benster.

[7] Chun-Hua Guo and Wen-Wei Lin. Solving a structured quadratic eigenvalue problem by a structure-preserving doubling algorithm. *SIAM J. Matrix Anal. Appl.*, 31(5):2784–2801, 2010.

[8] Chun-Hua Guo and P. Lancaster. Algorithms for hyperbolic quadratic eigenvalue problems. *Math. Comp.*, 74:1777–1791, 2005.

[9] Chun-Hua Guo. Numerical solution of a quadratic eigenvalue problem. *Linear Algebra Appl.*, 385(0):391–406, 2004.

[10] N. J. Higham and Hyun-Min Kim. Numerical analysis of a quadratic matrix equation. *IMA J. Numer. Anal.*, 20(4):499–519, 2000.

[11] A. Hilliges. Numerische Lösung von quadratischen Eigenwertproblemen mit Anwendung in der Schienendynamik. Diplomarbeit, Technical University Berlin, Inst. f. Mathematik, Germany, 2004.

[12] A. Hilliges, C. Mehl, and V. Mehrmann. On the solution of palindromic eigenvalue problems. 4th European Congress on Computational Methods in Applied Sciences and Engineerings (ECCOMAS), Jyväskylä, Finland, 2004.

[13] Tsung-Ming Huang, Wen-Wei Lin, and Jiang Qian. Structure-preserving algorithms for palindromic quadratic eigenvalue problems arising from vibration of fast trains. *SIAM J. Matrix Anal. Appl.*, 30(4):1566–1592, 2009.

[14] I. C. F. Ipsen. Accurate eigenvalues for fast trains. *SIAM News*, 37(9), November 2004.

[15] Ren-Cang Li, Wen-Wei Lin, and Chern-Shuh Wang. Structured backward error for palindromic polynomial eigenvalue problems. *Numer. Math.*, 116(1):95–122, 2010.

[16] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Structured polynomial eigenvalue problems: Good vibrations from good linearizations. *SIAM J. Matrix Anal. Appl.*, 28(4):1029–1051, 2006.

[17] J. Schur. Über potenzreihen, die im Innern des Einheitskreises beschränkt sind. *Journal für die reine und angewandte Mathematik*, 147:205–232, 1917.

[18] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43(2):235–386, 2001.

[19] Kemin Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice Hall, Upper Saddle River, New Jersey, 1995.