SIAM J. SCI. COMPUT.
Vol. 38, No. 6, pp. A3410–A3429

A FAST ALGORITHM FOR FAST TRAIN PALINDROMIC QUADRATIC EIGENVALUE PROBLEMS*

LINZHANG LU†, TENG WANG‡, YUEH-CHENG KUO§, REN-CANG LI¶, AND WEN-WEI LIN‖

Abstract. In the vibration analysis of high speed trains arises such a palindromic quadratic eigenvalue problem (PQEP) $(\lambda^2 A^{\mathrm{T}} + \lambda Q + A)z = 0$, 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 $k \times k$ (thus $n = m \times k$), and Q is complex symmetric and tridiagonal block-Toeplitz, and A has only one nonzero block in the $(1, m)$th block position which is the same as the subdiagonal block of Q. This PQEP has eigenvalues 0 and ∞ each of multiplicity $(m-1)k$ just by examining A, but it is its remaining $2k$ eigenvalues, usually nonzero and finite but with an extreme wide range in magnitude, that are of interest. The problem is notoriously difficult numerically. Earlier methods that seek to deflate eigenvalues 0 and ∞ first often produce eigenvalues that are too inaccurate to be useful due to the large errors introduced in the deflation process. The solvent approach proposed by Guo and Lin in 2010 changed the situation because it can deliver sufficiently accurate eigenvalues. In this paper, we propose a fast algorithm along the line of the solvent approach. The theoretical foundation of our algorithm is the connection we establish here between this fast train PQEP and a $k \times k$ PQEP defined by the subblocks of A and Q without any computational work. This connection lends itself to a fast algorithm: solve the $k \times k$ PQEP and then use its eigenpairs to recover the eigenpairs for the original fast train PQEP. The so-called α-structured backward error analysis that preserves all possible structures in the fast train PQEP to the extreme is studied. Finally numerical examples are presented to show the effectiveness of the new fast algorithm.

Key words. palindromic quadratic eigenvalue problem, PQEP, fast train, nonlinear matrix equation, solvent approach, doubling algorithm

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

DOI. 10.1137/16M1063563

1. Introduction.

The palindromic quadratic eigenvalue problem (PQEP) [10, 11, 16] is to find scalars λ and nonzero vectors z such that

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

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 λ and a vector $z \neq 0$, we call λ a *quadratic eigenvalue*, z a corresponding *quadratic eigenvector*, and (λ, z) a *quadratic eigenpair*.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 29, 2016; accepted for publication (in revised form) August 5, 2016; published electronically November 1, 2016.

http://www.siam.org/journals/sisc/38-6/M106356.html

Funding: The first author's work was supported in part by National Natural Science Foundation of China grants 11261012, 11428104 and 11671105. The fourth author's work was supported in part by NSF grants DMS-1317330 and CCF-1527104 and NSFC grant 11428104. The fifth author's work was supported in part by the National Center of Theoretical Sciences and the ST Yau Centre at the National Chiao Tung University.

†School of Mathematics and Computer Science, Guizhou Normal University, and School of Mathematical Science, Xiamen University, People's Republic of China (lzlu@xmu.edu.cn, llz@gznu.edu.cn).

‡Department of Mathematics, and Numerical Simulation and High Performance Computing Laboratory, School of Science, Nanchang University, People's Republic of China (spring0920@163.com).

§Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 811, Taiwan (yckuo@nuk.edu.tw).

¶Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019-0408 (rcli@uta.edu).

‖Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan (wwlin@am.nctu.edu.tw).

Sometimes, we may also ask for left quadratic eigenvectors of $P(\cdot)$. By a left quadratic eigenpair (w, λ) of scalar λ and nonzero vector w, we mean

$$w^{\mathrm{T}} P(\lambda) \equiv w^{\mathrm{T}}(\lambda^2 A^{\mathrm{T}} + \lambda Q + A) = 0. \tag{1.2}$$

If both (1.1) and (1.2) hold, we call (w, λ, z) a *quadratic eigentriplet* of $P(\cdot)$. The word "*quadratic*" before *eigenvalue*, *eigenvector*, *eigenpair*, and *eigentriplet* is often dropped for convenience when no confusion arises.

In this paper, we are interested in those PQEPs (1.1) that arise from the vibration analysis of high-speed trains [3, 8, 13, 17], where the coefficients A and Q often have additional structures beyond $Q^{\mathrm{T}} = Q$. In fact in the case of high-speed trains, Q is also tridiagonal block-Toeplitz, and the subblocks in A partitioned in the same way as Q are all 0 except one subblock in the upper-right corner. Specifically, without going into too much detail, $n = mk$, and

$$P(\lambda)z \equiv (\lambda^2 A^{\mathrm{T}} + \lambda Q + A)z = 0 \quad \text{with} \tag{1.3a}$$

$$Q = \begin{array}{c} \\ k \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{ccccc} k & k & k & \cdots & k \end{array} \\ \begin{bmatrix} H_0 & H_1^{\mathrm{T}} & & & \\ H_1 & H_0 & H_1^{\mathrm{T}} & & \\ & H_1 & \ddots & \ddots & \\ & & \ddots & \ddots & H_1^{\mathrm{T}} \\ & & & H_1 & H_0 \end{bmatrix} \end{array}, \quad A = \begin{array}{c} \\ k \\ k \\ \vdots \\ k \end{array} \begin{array}{c} \begin{array}{cccc} k & \cdots & k & k \end{array} \\ \begin{bmatrix} 0 & \dots & 0 & H_1 \\ 0 & \dots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} \end{array}. \tag{1.3b}$$

Both have m block-rows and columns. Also H_0 is complex symmetric, i.e., $H_0^{\mathrm{T}} = H_0$. For future references, we will call PQEP (1.3) a *fast train PQEP*.

This fast train PQEP was first raised in a study in Germany by Hilliges, Mehl, and Mehrmann [10, 11]. There are numerous numerical difficulties in solving it: (1) most eigenvalues are 0 and ∞ and in fact, by examining A, we see that it has eigenvalues 0 and ∞ each with multiplicity $(m-1)k$; (2) the problem size n can range from 10^3 to 10^5; (3) most seriously the problem is badly scaled with finite eigenvalue magnitudes ranging from 10^{-50} to 10^{50} or to an even greater extreme; and (4) all finite nonzero eigenvalues and eigenvectors are to be computed. Systematical studies into more general palindromic eigenvalue problems including linear and polynomial ones such as PQEP (1.3) started with Mackey et al. [18, 19] and the research for efficient and robust methods remains active today. Numerically some structure-preserving algorithms were developed in [3, 12, 14, 15], and theoretically some structured backward error analysis was established in [9, 16]. Earlier methods which typically start by deflating out the known eigenvalues 0 and ∞ for the sake of efficiency do not perform as well as needed in dealing with so many eigenvalues 0 and ∞ and in particular the wide range of the eigenvalue magnitude. The key reason is due to that the deflation process involves the inverses of potentially ill-conditioned matrices (see, e.g., [3]) and consequently introduces large error into the data for the deflated eigenvalue problem from which the rest of the eigenvalues are computed. The situation, however, changed with the emergence of the solvent approach proposed in [8] and its modification in [17]. Strong numerical evidence suggests that both solvent approaches can satisfactorily handle numerous eigenvalues 0 and ∞ and the wide range of the eigenvalue magnitude in particular.

In applying their solvent approach to the fast train PQEP, Guo and Lin [8] cleverly exploited most of the inherent subblock structures in A and Q and significantly

reduced the cost per doubling iterative step for solving the $n \times n$ nonlinear matrix equation

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

which the solvent approach depends on. Recently, Lu, Yuan, and Li [17] discovered a way to get around this nonlinear matrix equation by solving, instead, another matrix equation of the same form but only of $k \times k$. Despite successful structural exploitations for much computational gain in [8, 17], the fact that Q is tridiagonal block-Toeplitz is not numerically explored in both works, not to mention that the top-right corner block in A relates to the subdiagonal block of Q. The goal of this paper is to fully exploit the structures of A and Q in (1.3b) in the best possible way. As a result, we obtain a new fast algorithm which compares favorably to the solvent approaches of [8, 17] in speed and accuracy. In fairness, the existing solvent approaches, however, have wider applicability.

The rest of this paper is organized as follows. In section 2, we review briefly the doubling algorithm used in [8] for computing the stabilizing solution Φ of (2.1). In section 3, we present our main theoretical results and devise our fast algorithm for the fast train PQEP (1.3). In section 4, we explain the advantages of our new fast algorithm in section 3 over the existing structure-preserving doubling algorithm (SDA) based solvent approaches. Section 5 presents our numerical results. 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 jth column. We usually use lowercase letters for vectors and capital letters for matrices and use λ, μ, and τ for eigenvalues. The superscripts "$\cdot^{\mathrm{T}}$" and "$\cdot^{\mathrm{H}}$" take 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 jth 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 ℓ and column i to column j, row k to row ℓ, and column i to column j, respectively. $X_1 \oplus \cdots \oplus X_k := \mathrm{diag}(X_1, \ldots, X_k)$ is the block-diagonal matrix with the diagonal blocks $X_1, \ldots, X_k$. $\|X\|_2$ and $\|X\|_{\mathrm{F}}$ are the spectral and Frobenius norms of X, respectively.

2. The doubling algorithm. The *solvent approach* in [8] depends on the so-called stabilizing solution of

$$X + A^{\mathrm{T}} X^{-1} A = Q. \tag{2.1}$$

By the *stabilizing solution*, we mean the solution X that satisfies $\rho(X^{-1}A) < 1$, where $\rho(\cdot)$ is the spectral radius of a matrix. In [7, 8], it is shown that the stabilizing solution exists if $\Im(Q) + \lambda[\Im(A)]^{\mathrm{T}} + \lambda^{-1}\Im(A)$ is positive definite for all λ on the unit circle, where $\Im(Q)$ and $\Im(A)$ are the entrywise imaginary parts of Q and A, respectively.

Given the prominent role of the stabilizing solution of (2.1) and to distinguish it from other solutions, we designate the symbol Φ for it, i.e., we will use Φ for the stabilizing solution of (2.1). It is complex symmetric, i.e., $\Phi^{\mathrm{T}} = \Phi$.

The stabilizing solution Φ (as well as any other solutions of (2.1) called the *solvent matrices*) gives rise to the following factorization for $P(\cdot)$:

$$P(\lambda) = \lambda^2 A^{\mathrm{T}} + \lambda Q + A = (\lambda A^{\mathrm{T}} + \Phi)\Phi^{-1}(\lambda\Phi + A). \tag{2.2}$$

Algorithm 2.1. The doubling algorithm for solving (2.1).

Input: $A, Q = Q^{\mathrm{T}} \in \mathbb{C}^{n\times n}$;
Output: A solution of (2.1);

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

In summary, the solvent approach consists of two steps:

1. Compute the stabilizing solution, if it exists, of the nonlinear matrix equation (2.1) with the doubling algorithm.
2. Solve the (linear) eigenvalue problems for matrix pencils $\lambda A^{\mathrm{T}} + \Phi$ or $\lambda\Phi + A$. Note the eigenvalues of $\lambda A^{\mathrm{T}} + \Phi$ and those of $\lambda\Phi + A$ satisfy the reciprocal relation: if μ is an eigenvalue of one, then $1/\mu$ is an eigenvalue of the other.

Guo and Lin [8] used SDA [2] to solve the matrix equation (2.1) for its stabilizing solution Φ (see also [3]). For convenience we outline SDA here in Algorithm 2.1. Once Φ is computed, the QZ algorithm [20] implemented in LAPACK [1] and in MATLAB as `eig(`$\cdots$`)` is applied to solve the eigenvalue problem[1] for $\lambda A^{\mathrm{T}} + \Phi$ or $\lambda\Phi + A$. Finally, the n eigenvalues of $P(\,\cdot\,)$ inside the unit circle are the n eigenvalues of $\lambda\Phi + A$, and the other n eigenvalues which are outside the unit circle are their reciprocals. The eigenvectors of $\lambda\Phi + A$ are also the eigenvectors of $P(\,\cdot\,)$, but those of $\lambda A^{\mathrm{T}} + \Phi$ need to be processed to yield the corresponding eigenvectors of $P(\,\cdot\,)$. Similarly, the left eigenvectors of $\lambda A^{\mathrm{T}} + \Phi$ are the left eigenvectors of $P(\,\cdot\,)$, and those of $\lambda\Phi + A$ have to be processed to yield the corresponding left eigenvectors of $P(\,\cdot\,)$.

It is shown in [8] that X_i generated by Algorithm 2.1 converges to the stabilizing solution Φ quadratically, and

$$\limsup_{i\to\infty} \sqrt[2^i]{\|X_i - \Phi\|} \le [\rho(\Phi^{-1}A)]^2, \tag{2.3}$$

where $\|\cdot\|$ is any matrix norm. Because of the quadratic convergence, as argued in [17], a reasonable stopping criteria for Algorithm 2.1 to use at line 2 is

$$\frac{\|X_{i+1} - X_i\|}{\|X_i\|} \le \mathtt{rtol}, \tag{2.4}$$

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

The solvent approach in [17] follows the same framework as outlined above but instead of $mk \times mk$ equation (2.1), it devises another nonlinear matrix equation,

$$\widetilde{X} + \widetilde{A}^{\mathrm{T}}\widetilde{X}^{-1}\widetilde{A} = \widetilde{Q}, \tag{2.5}$$

which takes the same form as (2.1) but is only of $k \times k$. The newly devised nonlinear matrix equation (2.5) is again solved by Algorithm 2.1. It was proved that the

[1] By exploiting the sparsity structure of A, Guo and Lin [8] showed how the $mk \times mk$ eigenvalue problems can be solved via two eigenvalue problems of only $k \times k$ in size.

approximations by Algorithm 2.1 converge at the same rate for both (2.1) and (2.5). We point out in passing that Guo and Lin [8] cleverly exploited the structures of A and Q in their implementation of Algorithm 2.1 to solve (2.1).

3. The fast algorithm. In this section, we consider the fast train PQEP (1.3). Structurally, it always has the eigenvalue 0 and ∞, each of multiplicity $n-k = (m-1)k$. In fact, if $P(0)z = 0$, which is the same as $Az = 0$, then $H_1 z_{(n-k+1:n)} = 0$ immediately yielding $n-k$ linearly independent eigenvectors $e_1, e_2, \ldots, e_{n-k}$. Similarly by considering the PQEP $\lambda^2 P(1/\lambda)z = 0$, we find $n-k$ linearly independent eigenvectors $e_{k+1}, e_{k+2}, \ldots, e_n$ associated with the eigenvalue ∞. When H_1 is nonsingular, these are the only linearly independent eigenvectors associated with the eigenvalue 0 and ∞, respectively. Therefore without any computational effort, we already know $2(n-k)$ eigenvalues and their corresponding eigenvectors: 0 and ∞, each of multiplicity $n-k = (m-1)k$. There are $2k$ remaining eigenvalues and, if needed, their associated eigenvectors, to be found.

Owing to this observation, instead of straightforwardly applying the solvent approach as we outlined in section 1, we will skip computing these zero and infinite eigenvalues by establishing a theoretical result that exposes a $k \times k$ PQEP whose solution leads to the remaining $2k$ eigenvalues and eigenvectors. We introduce $k \times k$ PQEP:

$$\widehat{P}(\lambda)y := (\lambda^2 H_1^{\mathrm{T}} + \lambda H_0 + H_1)y = 0, \tag{3.1}$$

where H_0 and H_1 are the same as the ones in (1.3b). We further assume that

$$P(\,\cdot\,) \text{ and } \widehat{P}(\,\cdot\,) \text{ are } \textit{regular}, \tag{3.2}$$

i.e., $\det P(\lambda) \not\equiv 0$ and $\det \widehat{P}(\lambda) \not\equiv 0$, which are true for real test problems and thus reasonable to have. Since $\widehat{P}(\,\cdot\,)$ is assumed regular, by [5, Theorem 7.3] $\widehat{P}(\,\cdot\,)$ has a decomposable pair:

$$\Big((J_0 \oplus J_1) \oplus J_\infty, \quad \overset{\qquad\quad k_0 \qquad\quad k_1 \qquad\quad k_\infty}{\left[\begin{array}{ccc} [Y_0 & Y_1] & Y_\infty \end{array}\right]}\Big) \tag{3.3}$$

with $k_0 = k_\infty$ and $k_0 + k_1 + k_\infty = 2k$, where

1. the matrix
$$\left[\begin{array}{cc|c} Y_0 & Y_1 & Y_\infty J_\infty \\ Y_0 J_0 & Y_1 J_1 & Y_\infty \end{array}\right] \in \mathbb{C}^{2k \times 2k} \tag{3.4}$$
is nonsingular; and
2. $H_1^{\mathrm{T}} Y_\infty + H_0 Y_\infty J_\infty + H_1 Y_\infty J_\infty^2 = 0$, and $H_1^{\mathrm{T}} Y_i J_i^2 + H_0 Y_i J_i + H_1 Y_i = 0$ for $i = 0, 1$ which are equivalent to
$$H_1^{\mathrm{T}}[Y_0 \;\; Y_1](J_0 \oplus J_1)^2 + H_0[Y_0 \;\; Y_1](J_0 \oplus J_1) + H_1[Y_0 \;\; Y_1] = 0;$$
3. J_0 corresponds to the eigenvalue 0, J_1 corresponds to all nonzero finite eigenvalues, and J_∞ corresponds to the eigenvalue ∞. Thus $J_0 \oplus J_1$ corresponds to all finite eigenvalues. J_0 and J_∞ appear if and only if H_1 is singular.

In (3.3) both J_0 and J_∞ are nilpotent matrices,

$$J_0^k = J_\infty^k = 0_{k_0 \times k_0}.$$

THEOREM 3.1. *Consider the fast train PQEP* (1.3) *and define* $\widehat{P}(\,\cdot\,)$ *by* (3.1) *and assume* (3.2) *with* $\widehat{P}(\,\cdot\,)$ *having the decomposable pair* (3.3). *Let*

$$Z_0 = \begin{matrix} \\ n-k \\ k \end{matrix} \begin{matrix} \begin{matrix} n-k & k_0 \end{matrix} \\ \begin{bmatrix} I & 0 \\ 0 & Y_0 \end{bmatrix} \end{matrix}, \quad Z_1 = \begin{bmatrix} Y_1 \\ Y_1 J_1 \\ \vdots \\ Y_1 J_1^{m-1} \end{bmatrix}, \quad Z_\infty = \begin{matrix} \\ k \\ n-k \end{matrix} \begin{matrix} \begin{matrix} k_\infty & n-k \end{matrix} \\ \begin{bmatrix} Y_\infty & 0 \\ 0 & I \end{bmatrix} \end{matrix}. \tag{3.5}$$

If[2]

$$J_0 = J_\infty = 0_{k_0 \times k_0}, \tag{3.6}$$

then $P(\,\cdot\,)$ *has the decomposable pair*

$$\Big(\boldsymbol{J}_{\text{finite}} \oplus \boldsymbol{J}_\infty,\, [\boldsymbol{Z}_{\text{finite}} \;\; Z_\infty]\Big), \tag{3.7a}$$

where

$$\boldsymbol{J}_{\text{finite}} = (0_{(n-k)\times(n-k)} \oplus 0_{k_0 \times k_0}) \oplus J_1^m, \quad \boldsymbol{J}_\infty = 0_{k_\infty \times k_\infty} \oplus 0_{(n-k)\times(n-k)}, \tag{3.7b}$$

$$\boldsymbol{Z}_{\text{finite}} = \begin{matrix} \begin{matrix} n-(k-k_0) & 2(k-k_0) \end{matrix} \\ \begin{bmatrix} Z_0 & Z_1 \end{bmatrix} \end{matrix}. \tag{3.7c}$$

That is to say,

1. *the matrix*

$$\left[\begin{array}{cc|c} Z_0 & Z_1 & 0 \\ 0 & Z_1 J_1^m & Z_\infty \end{array}\right] \in \mathbb{C}^{2n\times 2n} \tag{3.8}$$

   is nonsingular;
2. $AZ_0 = 0$, $A^{\mathrm{T}} Z_1 J_1^{2m} + Q Z_1 J_1^m + A Z_1 = 0$, *and* $A^{\mathrm{T}} Z_\infty = 0$. *Together, they are equivalent to*

$$A^{\mathrm{T}} \boldsymbol{Z}_{\text{finite}} \boldsymbol{J}_{\text{finite}}^2 + Q \boldsymbol{Z}_{\text{finite}} \boldsymbol{J}_{\text{finite}} + A \boldsymbol{Z}_{\text{finite}} = 0, \tag{3.9a}$$

$$A^{\mathrm{T}} Z_\infty + Q Z_\infty \boldsymbol{J}_\infty + A Z_\infty \boldsymbol{J}_\infty^2 = 0; \tag{3.9b}$$

3. $\boldsymbol{J}_{\text{finite}}$ *corresponds to the finite eigenvalues, and* $\boldsymbol{J}_\infty$ *corresponds to the eigenvalue* ∞.

Proof. Through elimination among its columns, we see that the matrix in (3.8) is nonsingular if and only if

$$I_{n-k} \oplus \begin{bmatrix} Y_0 & Y_1 J_1^{m-1} & 0 \\ 0 & Y_1 J_1^m & Y_\infty \end{bmatrix} \oplus I_{n-k}$$

is nonsingular. This matrix is nonsingular if and only if

$$\begin{bmatrix} Y_0 & Y_1 J_1^{m-1} & 0 \\ 0 & Y_1 J_1^m & Y_\infty \end{bmatrix} = \begin{bmatrix} Y_0 & Y_1 & 0 \\ 0 & Y_1 J_1 & Y_\infty \end{bmatrix} \begin{bmatrix} I_{k_0} & & \\ & J_1^{m-1} & \\ & & I_{k_\infty} \end{bmatrix}$$

[2]That $k_0 = k_\infty = 0$ is allowed. This happens when H_1 is nonsingular. As a result, Y_0 and Y_∞ are empty matrices and thus $Z_0 = (I_n)_{(:,1:n-k)}$ and $Z_\infty = (I_n)_{(:,k+1:n)}$ in (3.5).

is nonsingular. The last matrix is nonsingular because J_1 has only nonzero eigenvalues and thus nonsingular and $\left[\begin{smallmatrix} Y_0 & Y_1 & 0 \\ 0 & Y_1 J_1 & Y_\infty \end{smallmatrix}\right]$ is nonsingular because it is the same as (3.4) due to $J_0 = J_\infty = 0$ by assumption. This proves item 1.

For item 2, we note $H_1 Y_0 = 0$, and $H_1^{\mathrm{T}} Y_\infty = 0$ again due to $J_0 = J_\infty = 0$ by assumption. Thus $AZ_0 = 0$ and $A^{\mathrm{T}} Z_\infty = 0$ follow from the structure of A in (1.3b). We now prove $A^{\mathrm{T}} Z_1 J_1^{2m} + QZ_1 J_1^m + AZ_1 = 0$. Expand this equation by block-rows to get

$$H_0 Y_1 J_1^m + H_1^{\mathrm{T}} Y_1 J_1^{m+1} + H_1 Y_1 J_1^{m-1} = 0, \tag{3.10a}$$

$$H_1 Y_1 J_1^{m+i-2} + H_0 Y_1 J_1^{m+i-1} + H_1^{\mathrm{T}} Y_1 J_1^{m+i} = 0 \quad \text{for } 2 \le i \le m-1, \tag{3.10b}$$

$$H_1^{\mathrm{T}} Y_1 J_1^{2m} + H_1 Y_1 J_1^{2m-2} + H_0 Y_1 J_1^{2m-1} = 0. \tag{3.10c}$$

It suffices to show (3.10), instead. To this end, we recall $H_1^{\mathrm{T}} Y_1 J_1^2 + H_0 Y_1 J_1 + H_1 Y_1 = 0$. Now for (3.10a), we have

$$\begin{aligned} H_0 Y_1 J_1^m + H_1^{\mathrm{T}} Y_1 J_1^{m+1} + H_1 Y_1 J_1^{m-1} &= (H_1^{\mathrm{T}} Y_1 J_1^2 + H_0 Y_1 J + H_1 Y_1) J_1^{m-1} \\ &= 0. \end{aligned}$$

For (3.10b), we have

$$\begin{aligned} H_1 Y_1 J_1^{m+i-2} + H_0 Y_1 J_1^{m+i-1} + H_1^{\mathrm{T}} Y_1 J_1^{m+i} &= (H_1 Y_1 + H_0 Y_1 J_1 + H_1^{\mathrm{T}} Y_1 J_1^2) J^{m+i-2} \\ &= 0. \end{aligned}$$

Finally for (3.10c), we have

$$\begin{aligned} H_1^{\mathrm{T}} Y_1 J_1^{2m} + H_1 Y_1 J_1^{2m-2} + H_0 Y_1 J_1^{2m-1} &= (H_1 Y_1 + H_0 Y_1 J_1 + H_1^{\mathrm{T}} Y_1 J_1^2) J^{2m-2} \\ &= 0. \end{aligned}$$

This proves (3.10). To see that $AZ_0 = 0$, $A^{\mathrm{T}} Z_1 J_1^{2m} + QZ_1 J_1^m + AZ_1 = 0$, and $A^{\mathrm{T}} Z_\infty = 0$ together are equivalent to (3.9). We notice that (3.9b) is the same as $A^{\mathrm{T}} Z_\infty = 0$ because $\boldsymbol{J}_\infty = 0$. Equation (3.9a) is two equations in one:

$$\begin{aligned} A^{\mathrm{T}} Z_0 J_0^{2m} + QZ_0 J_0^m + AZ_0 &= 0, \\ A^{\mathrm{T}} Z_1 J_1^{2m} + QZ_1 J_1^m + AZ_1 &= 0, \end{aligned}$$

where $J_0 = 0_{(n-k)\times(n-k)}$. Both have been proven.

Item 3 follows from items 1 and 2. ∎

Besides the key assumption (3.2), (3.6) is another one that makes the construction of the decomposable pair (3.7) work. The latter means that all Jordan blocks, if any, of $\widehat{P}(\,\cdot\,)$ corresponding to its eigenvalues 0 and ∞ are 1-by-1. Both (3.2) and (3.6) hold for all the test problems in section 5, where in fact H_1 is nonsingular to yield $k_0 = k_\infty = 0$. What this theorem says, under the key assumptions, is that all nonzero and finite eigenvalues[3] of the $n \times n$ fast train PQEP (1.3) come from the nonzero and finite eigenvalues of the $k \times k$ PQEP (3.1). In particular, all nonzero and finite eigenpairs of $P(\,\cdot\,)$ are given by

$$\left(\mu^m, \begin{bmatrix} y \\ y\mu \\ \vdots \\ y\mu^{m-1} \end{bmatrix} \right), \tag{3.11}$$

[3]By a nonzero and finite eigenpair, we mean the associated eigenvalue is nonzero and finite.

Algorithm 3.1. Fast algorithm for fast train PQEP (1.3).

Input: A and Q as in (1.3b);
Output: all nonzero and finite eigenpairs of fast train PQEP (1.3);

1: use Algorithm 2.1 with input $A \leftarrow H_1$ and $Q \leftarrow H_0$ to compute the stablizing solution $\widehat{\Phi}$ of (3.13) and, as a result, $\widehat{P}(\lambda) = (\lambda H_1^{\mathrm{T}} + \widehat{\Phi})\widehat{\Phi}^{-1}(\lambda\widehat{\Phi} + H_1)$.
2: solve the eigenvalue problem for matrix pencil $\lambda\widehat{\Phi} + H_1$;
3: for each nonzero and finite eigenpair (μ, y) of $\lambda\widehat{\Phi} + H_1$ (i.e., $H_1 y = \mu\,\widehat{\Phi} y$ and $\mu \neq 0$), construct an eigenpair as in (3.11) of the fast train PQEP (1.3);
4: if desired, for each nonzero and finite left eigenpair $(\hat{w}, \mu)$ of $\lambda\widehat{\Phi}+H_1$ (i.e., $\hat{w}^{\mathrm{T}} H_1 = \mu\,\hat{w}^{\mathrm{T}}\widehat{\Phi}$ and $\mu \neq 0$), solve $w^{\mathrm{T}}(\mu H_1^{\mathrm{T}} + \widehat{\Phi}) = \hat{w}^{\mathrm{T}}\widehat{\Phi}$ for w and construct a left eigenpair as in (3.12) of the fast train PQEP (1.3);
5: **return** all constructed eigenpairs (3.11), and, if desired, all constructed left eigenpairs (3.12).

where (μ, y) is a nonzero and finite eigenpair of $\widehat{P}(\,\cdot\,)$. Correspondingly, it is not hard to introduce the notation of the *left* decomposable pair and develop a version of Theorem 3.1 about the *left* decomposable pair of the fast train PQEP (1.3) constructed from that of PQEP (3.1). We omit the details, but we will say that all nonzero and finite left eigenpairs of $P(\,\cdot\,)$ are given by

$$\left(\begin{bmatrix} w\mu^{m-1} \\ \vdots \\ w\mu \\ w \end{bmatrix}, \mu^m \right), \tag{3.12}$$

where (w, μ) is a nonzero and finite left eigenpair of $\widehat{P}(\,\cdot\,)$. We are now ready to present our fast algorithm, Algorithm 3.1, in which the $k \times k$ nonlinear matrix equation

$$\widehat{X} + H_1^{\mathrm{T}} \widehat{X}^{-1} H_1 = H_0 \tag{3.13}$$

needs to be solved instead of the $n \times n$ equation (2.1).

For the eigenvalue 0 and ∞, we have, according to Theorem 3.1, that each eigentriplet $(\hat{w}, 0, y)$ of $\lambda\widehat{\Phi} + H_1$ (i.e., $H_1 y = 0$ and $\hat{w}^{\mathrm{T}} H_1 = 0$) leads to two eigentriplets,

$$\left(\begin{bmatrix} \hat{w} \\ 0 \end{bmatrix}, 0, \begin{bmatrix} 0 \\ y \end{bmatrix} \right), \quad \left(\begin{bmatrix} 0 \\ y \end{bmatrix}, \infty, \begin{bmatrix} \hat{w} \\ 0 \end{bmatrix} \right), \tag{3.14}$$

for $P(\,\cdot\,)$, where each vector in (3.14) lies in $\mathbb{C}^n$, provided both (3.2) and (3.6) hold.

Remark 3.1. There are several comments in order.

1. As mentioned before, the eigenvalues of $\lambda\widehat{\Phi} + H_1$ are inside the unit circle, and those of $\lambda H_1^{\mathrm{T}} + \widehat{\Phi}$ are outside of the unit circle.
2. At line 2, we can solve the eigenvalue problem for matrix pencil $\lambda H_1^{\mathrm{T}} + \widehat{\Phi}$ instead. This is because the eigenvalues of $\lambda\widehat{\Phi} + H_1$ and those of the eigenvalues of $\lambda H_1^{\mathrm{T}} + \widehat{\Phi}$ satisfy the reciprocal relationship: if μ is an eigenvalue of $\lambda\widehat{\Phi} + H_1$, then $1/\mu$ is an eigenvalue of $\lambda H_1^{\mathrm{T}} + \widehat{\Phi}$ and vice versa. Also their eigenvectors are related, too.
3. At line 4 for recovering the left eigenvectors of $P(\,\cdot\,)$ from those of $\lambda\widehat{\Phi} + H_1$ via

$$w^{\mathrm{T}}(\mu H_1^{\mathrm{T}} + \widehat{\Phi}) = \hat{w}^{\mathrm{T}}\widehat{\Phi} \tag{3.15}$$

which, upon taking transpose and noting that $\widehat{\Phi}$ is complex symmetric, becomes

$$(\widehat{\Phi} + \mu H_1)w = \widehat{\Phi}\hat{w}. \tag{3.16}$$

Straightforward computations by (3.16) cost $O(k^4)$ for all k eigenpairs. That's too costly. But there is a more efficient way. Recall that at line 2 in solving the eigenvalue problem for matrix pencil $\lambda\widehat{\Phi} + H_1$ by, e.g., the QZ algorithm, H_1 and $\widehat{\Phi}$ are first reduced to an upper Hessenberg matrix and an upper triangular matrix, respectively,

$$U^{\mathrm{H}} H_1 V = G \quad \text{and} \quad U^{\mathrm{H}}\widehat{\Phi}V = T, \tag{3.17}$$

where U, $V \in \mathbb{C}^{k\times k}$ are unitary, and G and T are upper Hessenberg and upper triangular, respectively. Utilizing (3.17) which has already been computed at line 2, we propose to solve (3.16) as follows:

$$\text{solve } (\mu G + T)g = T(V^{\mathrm{H}}\hat{w}) \text{ for } g \text{ and then } w = Vg.$$

This incurs a total cost of $16k^3/3$ flops for recovering all k left eigenvectors of $P(\,\cdot\,)$.

4. Although at line 3 and optionally at line 4, only the eigenvectors of $P(\,\cdot\,)$ associated with its eigenvalues within the unit circle are explicitly constructed, the eigenvectors associated with the eigenvalues outside the unit circle can be made readily available, too. Recall that the reciprocals of all eigenvalues of $\lambda\widehat{\Phi} + H_1$ yield all eigenvalues of PQEP (1.3) outside the unit circle. It follows from $H_1 y = \mu\,\widehat{\Phi}y$ and $\hat{w}^{\mathrm{T}}H_1 = \mu\,\hat{w}^{\mathrm{T}}\widehat{\Phi}$ that

$$y^{\mathrm{T}}\widehat{\Phi} = \mu^{-1}y^{\mathrm{T}}H_1^{\mathrm{T}}, \quad \widehat{\Phi}\hat{w} = \mu^{-1}H_1^{\mathrm{T}}\hat{w} \tag{3.18}$$

for each nonzero eigentriplet $(\hat{w}, \mu, y)$ of $\lambda\widehat{\Phi} + H_1$. The equations in (3.18) imply that $(y, \mu^{-1}, \hat{w})$ is an eigentriplet of $\lambda H_1^{\mathrm{T}} + \widehat{\Phi}$. Consequently, by (3.11) and (3.12),

$$\left(\begin{bmatrix} y\mu^{-(m-1)} \\ \vdots \\ y\mu^{-1} \\ y \end{bmatrix}, \mu^{-m}, \begin{bmatrix} w \\ w\mu^{-1} \\ \vdots \\ w\mu^{-(m-1)} \end{bmatrix} \right) \tag{3.19}$$

is an eigentriplet of $P(\,\cdot\,)$, where w solves (3.16). Essentially, no additional work is needed after the left and right eigenvectors associated with the eigenvalues within the unit circles are known.

4. Compare to existing doubling-based solvent approaches. Previously, there were two variations of solvent approaches to solve (1.3) based on the doubling algorithm:

- SDA_GL [8]. It uses the doubling iteration to solve (2.1) and then cleverly solve the eigenvalue problems for matrix pencils $\lambda A^{\mathrm{T}} + \Phi$ and $\lambda\Phi + A$ by reducing them to two $k \times k$ eigenvalue problems.
- SDA_LYL [17]. It also uses the doubling iteration but to solve some $k \times k$ nonlinear matrix equation [17, (3.6)],

$$\widetilde{X} + \widetilde{A}^{\mathrm{T}}\widetilde{X}^{-1}\widetilde{A} = \widetilde{Q} \tag{4.1}$$

derived from A and Q, and then to solve the PQEP for $\widetilde{P}(\lambda) \equiv \lambda^2 \widetilde{A}^{\mathrm{T}} + \lambda \widetilde{Q} + \widetilde{A}$ through the eigenvalue problems for $k \times k$ matrix pencils $\lambda \widetilde{A}^{\mathrm{T}} + \widetilde{\Phi}$ and $\lambda \widetilde{\Phi} + \widetilde{A}$.

Our approach here in Algorithm 3.1 can also be categorized as a solvent approach:

- SDA_LWKLL (Algorithm 3.1). It uses the same doubling iteration but to solve (3.13) instead. There is an essential difference between SDA_LYL and SDA_LWKLL: SDA_LYL doesn't really need to solve the eigenvalue problem for $\widetilde{P}(\,\cdot\,)$ but rather the stabilizing solution $\widehat{\Phi}$ of (4.1) and use it to recover the stabilizing solution Φ of (2.1), while SDA_LWKLL does compute the eigenvalues of $\widehat{P}(\,\cdot\,)$ and then raises them to their mth powers to recover the eigenvalues of $P(\,\cdot\,)$.

All three approaches share a common feature: after the doubling iteration converged, two $k \times k$ linear eigenvalue problems will have to be solved. The two eigenvalue problems are related in that the eigenvalues of one are inside the unit circle and their reciprocals give all eigenvalues of the other.

The numerical solution of a dense linear eigenvalue problem are now considered standard and solved, e.g., by the QZ algorithm [6, 20]. Therefore we will not delve into that part in all three approaches. What we will discuss among the three are (1) the rates of convergence in the doubling iterations, (2) the flop counts, (3) the structured backward errors in computed eigenpairs of the fast train PQEP (1.3) to preserve either simply the palindromic form in (1.1) or additionally all the finer subblock structures in (1.3), and (4) accuracy in the computed eigenvalues for $P(\,\cdot\,)$.

4.1. Rate of convergence for doubling iterations. Adopt the notation convention of using X_i, $\widetilde{X}_i$, and $\widehat{X}_i$ for the ith approximations by the doubling algorithm on (2.1), (4.1), and (3.13), respectively, whose stabilizing solutions are denoted by Φ, $\widetilde{\Phi}$, and $\widehat{\Phi}$, respectively.

Recall our discussion on the rate of convergence of the doubling algorithm in the second half of section 2. It was shown [17, Theorem 4.1] that the doubling iterations converge at the same rate for (2.1) and for (4.1) [17, (3.6)]:

$$\|X_i - \Phi\|,\ \|\widetilde{X}_i - \widetilde{\Phi}\| \lessapprox \gamma^{2^{i+1}} \quad \text{with} \quad \gamma = \rho\big(\widetilde{\Phi}^{-1}\widetilde{A}\big) = \rho(\Phi^{-1}A), \tag{4.2}$$

where, and in what follows, "$\lessapprox$" means the upper bound holds modulo some constant factor that is independent of iterative index i. But on (3.13), we have

$$\|\widehat{X}_i - \widehat{\Phi}\| \lessapprox \gamma^{2^{i+1}/m} \tag{4.3}$$

because $\gamma = \rho(\Phi^{-1}A) = [\rho(\widehat{\Phi}^{-1}H_1)]^m$ by Theorem 3.1. So the doubling algorithm is slower on (3.13) than on (2.1) and (4.1). But exactly how much slower? Roughly speaking, (4.2) and (4.3) imply that

(4.4) | for the same accuracy, (3.13) needs about $\log_2 m$ more doubling iterative steps than (2.1) and (4.1) do.

We now substantiate this claim. For simplicity, let us ignore the constant factors hidden in the notation $\lessapprox$. To reduce the norm error between each approximation by the doubling algorithm and their target to below some tolerance $\epsilon < 1$, for (2.1) and (4.1) we need by (4.2) that

$$\gamma^{2^{i+1}} \le \epsilon \quad \Rightarrow \quad i + 1 \ge \log_2 \frac{\log_2 \epsilon}{\log_2 \gamma},$$

TABLE 1
Comparison of flop counts ($m \geq 2$).

	Before DA iteration	Each DA iteration	Finding eigenpairs
SDA_GL [8]	$\frac{113}{3}mk^3$	$\frac{154}{3}k^3$	$14mk^3 + \frac{214}{3}k^3$
SDA_LYL [17]	$\frac{113}{3}(m-1)k^3$	$\frac{32}{3}k^3$	$14mk^3 + \frac{214}{3}k^3$
SDA_LWKLL	0	$\frac{32}{3}k^3$	$2mk^2 + \frac{214}{3}k^3$

and for (3.13) we need by (4.3)

$$\gamma^{2^{i+1}/m} \leq \epsilon \quad \Rightarrow \quad i+1 \geq \log_2 \frac{\log_2 \epsilon}{\log_2 \gamma} + \log_2 m,$$

yielding the claim.

4.2. Flop counts. We begin by estimating the flop counts in Table 1, where those for SDA_GL and SDA_LYL are taken from [17, Table 6.1] but with the cost $\frac{214}{3}k^3$ flops (in the fourth column of Table 1), for one QZ run (including forming the Q- and Z-matrices in the QZ algorithm and solving (3.15) as outlined in item 3 of Remark 3.1) [6, p. 385] on one of the two $k \times k$ linear eigenvalue problems from factorizing $P(\,\cdot\,)$ and $\widetilde{P}(\,\cdot\,)$ for their eigenvalues and corresponding left and right eigenvectors. The costs incurred by the QZ runs are not negligible, especially for m not too big. The expression $2mk^2$ in the fourth column for SDA_LWKLL is for forming left and right eigenvectors as in (3.11) and (3.12). Evidently, SDA_LWKLL uses the fewest flops. In particular, its cost does not contain the term $O(mk^3)$, but just $O(mk^2)$. This makes SDA_LWKLL particularly attractive when k is not small. In particular, if m and k are comparable, then both SDA_GL and SDA_LYL cost $O(k^4)$, whereas it is $O(k^3)$ for SDA_LWKLL.

Now what does the claim (4.4) mean to the overall cost for solving PQEP (1.3)? Recalling Table 1, we see that the extra doubling steps taken by SDA_LWKLL will cost

$$\frac{32}{3}k^3 \log_2 m$$

flops, which is far fewer than the cost terms $O(mk^3)$ in the flop counts for SDA_GL and SDA_LYL.

Suppose that it takes i doubling iterative steps to solve (2.1) and (4.1). Our analysis above says that it will take $i + \log_2 m$ doubling iterative steps to solve (3.13). With the help of Table 1, we can calculate the needed flops by SDA_GL, SDA_LYL, and SDA_LWKLL for solving PQEP for $P(\,\cdot\,)$ in terms of i. Consequently, we can calculate the speedups of SDA_LWKLL over SDA_GL and SDA_LYL again in terms of i. Previously in [8, 17], it was observed that it would take about 7 to 10 doubling iterative steps to solve (2.1) and (4.1) for various fast train PQEPs there. So it is reasonable for us to look at the speedups for i, say, from 5 to 12, in order to get some idea of the effectiveness of SDA_LWKLL compared to SDA_GL and SDA_LYL. In Figure 1, we plot these speedups. What we can see is that SDA_LWKLL is at least 3 times faster than SDA_LYL and at least 5 times faster than SDA_GL.

Last, we must put this flop comparison in perspective. Both SDA_GL and SDA_LYL have much wider applicability than SDA_LWKLL, namely, SDA_GL and SDA_LYL do not need the block-Toeplitz structure in Q to work so long as it is block-tridiagonal, not

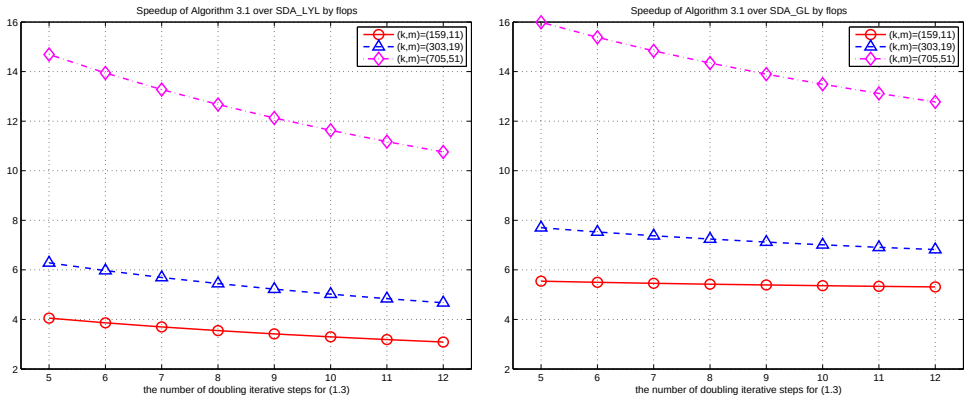

FIG. 1. *Left plot: speedup of Algorithm* 3.1 *over* SDA_LYL*; right plot: speedup of Algorithm* 3.1 *over* SDA_GL*. Algorithm* 3.1 *is at least* 3 *times faster than* SDA_LYL *and at least* 5 *times faster than* SDA_GL*.*

to mention that there is no need to require that the upper-right block in A is the same as the subdiagonal block in Q, either. It is just that in the case of (1.3), SDA_LWKLL performs the best, as a trade-off for being so specialized.

4.3. Structured backward errors. In [16], the authors investigated backward errors for four kinds of palindromic polynomial eigenvalue problems, one of which is PQEP (1.1), where A and Q do not necessarily have the finer subblock structure as displayed in (1.3b) for the so-called fast train PQEP. While the main results in [16] are stated in a more general setting, in what follows we shall specify them to PQEPs in the form of (1.1).

Let (τ, z) be a computed eigenpair of PQEP (1.1) and set

$$r = (\tau^2 A^{\mathrm{T}} + \tau Q + A)z.$$

Ideally r should be 0, in which case (τ, z) is an exact computed eigenpair, but in general $r \neq 0$. The structured backward error analysis [16] is about seeking perturbations ΔA and ΔQ to A and Q such that

$$\big[\tau^2(A + \|A\|_{\mathrm{F}}\Delta A)^{\mathrm{T}} + \tau(Q + \|Q\|_{\mathrm{F}}\Delta Q) + (A + \|A\|_{\mathrm{F}}\Delta A)\big]z = 0 \tag{4.5}$$

subject to $(\Delta Q)^{\mathrm{T}} = \Delta Q$. In other words, (τ, z) is an exact eigenpair of the perturbed PQEP for

$$P_\Delta(\lambda) \equiv \lambda^2(A + \|A\|_{\mathrm{F}}\Delta A)^{\mathrm{T}} + \lambda(Q + \|Q\|_{\mathrm{F}}\Delta Q) + (A + \|A\|_{\mathrm{F}}\Delta A). \tag{4.6}$$

Usually there exist infinitely many ΔA and ΔQ for this purpose, but we are interested in the smallest perturbations in the certain sense. Theorem 4.1 of [16] says that the *optimal backward error* is

$$\begin{aligned} \varepsilon :&= \min_{\Delta A,\, \Delta Q} \sqrt{\|\Delta A\|_{\mathrm{F}}^2 + \|\Delta Q\|_{\mathrm{F}}^2} \quad \text{subject to (4.5) and } (\Delta Q)^{\mathrm{T}} = \Delta Q \\ &= \sqrt{\frac{\delta_1^2}{\|A\|_2|1+\tau^2|^2 + \|Q\|_2|\tau|^2} + \frac{\delta_2^2}{\|A\|_2(1+|\tau|^4) + \|Q\|_2|\tau|^2/2}}, \end{aligned} \tag{4.7a}$$

where

$$\delta_1 = \frac{|z^{\mathrm{T}} r|}{\|z\|_2^2}, \quad \delta_2 = \frac{\sqrt{\|r\|_2^2\|z\|_2^2 - |z^{\mathrm{T}} r|^2}}{\|z\|_2^2}, \quad \sqrt{\delta_1^2 + \delta_2^2} = \frac{\|r\|_2}{\|z\|_2}. \tag{4.7b}$$

We caution the reader that the perturbations ΔA and ΔQ in (4.5) honor $(\Delta Q)^{\mathrm{T}} = \Delta Q$ only, but when applied to (1.3) it disregards the subblock patterns presented in A and Q of (1.3b). In particular, $A + \|A\|_{\mathrm{F}}\Delta A$ is no longer with just one nonzero block in the upper-right corner, not to mention the tridiagonal block-Toeplitz structure in Q. We argue that the indiscriminate fill-ins in A by $\|A\|_{\mathrm{F}}\Delta A$ to all subblock positions are the most damaging because they structurally destroy the eigenvalues 0 and ∞, each of multiplicity $n - k = (m-1)k$, of the fast train PQEP (1.3).

It would be interesting if we could enforce the perturbations ΔA and ΔQ to respect all the finer structures in (1.3b):

(4.8) $A+\|A\|_{\mathrm{F}}\Delta A$ and $Q+\|Q\|_{\mathrm{F}}\Delta Q$ take the same form as A and Q but with H_1 and H_0 changed to $H_1+\|H_1\|_{\mathrm{F}}\Delta H_1$ and $H_0+\|H_0\|_{\mathrm{F}}\Delta H_0$, respectively, and $(\Delta H_0)^{\mathrm{T}} = \Delta H_0$.

To distinguish this kind of extremely structured backward perturbation error analysis, we call it the *α-structured backward error analysis*, and accordingly $\sqrt{\|\Delta A\|_{\mathrm{F}}^2+\|\Delta Q\|_{\mathrm{F}}^2}$ is called the *α-structured backward error*.

Theorem 3.1 allows us to establish the α-backward error analysis for the fast train PQEP. Consider a computed eigenpair (τ, z), in the form (3.11), of PQEP (1.3), where[4]

$$(\mu, y) := (\tau^{1/m}, z_{(1:k)}) \tag{4.9a}$$

is an approximate eigenpair of $\widehat{P}(\,\cdot\,)$. Let

$$\hat{r} = (\mu^2 H_1^{\mathrm{T}} + \mu H_0 + H_1)y. \tag{4.9b}$$

First we seek perturbations ΔH_1 and ΔH_0 to H_1 and H_0 such that

$$\big[\mu^2(H_1 + \|H_1\|_{\mathrm{F}}\Delta H_1)^{\mathrm{T}} + \mu(H_0 + \|H_0\|_{\mathrm{F}}\Delta H_0) + (H_1 + \|H_1\|_{\mathrm{F}}\Delta H_1)\big]y = 0 \tag{4.10}$$

subject to $(\Delta H_0)^{\mathrm{T}} = \Delta H_0$. In other words, (μ, y) is an exact eigenpair of the perturbed PQEP for

$$\widehat{P}_{\Delta}(\lambda) \equiv \lambda^2(H_1 + \|H_1\|_{\mathrm{F}}\Delta H_1)^{\mathrm{T}} + \lambda(H_0 + \|H_0\|_{\mathrm{F}}\Delta H_0) + (H_1 + \|H_1\|_{\mathrm{F}}\Delta H_1). \tag{4.11}$$

As before we are interested in the smallest perturbations ΔH_1 and ΔH_0. By Theorem 4.1 of [16], we conclude the optimal backward error is

$$\hat{\varepsilon} := \min_{\Delta H_1,\, \Delta H_0} \sqrt{\|\Delta H_1\|_{\mathrm{F}}^2 + \|\Delta H_0\|_{\mathrm{F}}^2} \quad \text{subject to (4.10) and } (\Delta H_0)^{\mathrm{T}} = \Delta H_0 \tag{4.12a}$$

$$= \sqrt{\frac{\hat{\delta}_1^2}{\|H_1\|_2|1+\mu^2|^2 + \|H_0\|_2|\mu|^2} + \frac{\hat{\delta}_2^2}{\|H_1\|_2(1+|\mu|^4) + \|H_0\|_2|\mu|^2/2}}, \tag{4.12b}$$

[4]Since not every mth root of τ is an eigenvalue of $\widehat{P}(\,\cdot\,)$, here we use $\tau^{1/m}$ to stand for the one that is an eigenvalue of $\widehat{P}(\,\cdot\,)$.

where

$$\hat{\delta}_1 = \frac{|y^{\mathrm{T}}\hat{r}|}{\|y\|_2^2}, \quad \hat{\delta}_2 = \frac{\sqrt{\|\hat{r}\|_2^2\|y\|_2^2 - |y^{\mathrm{T}}\hat{r}|^2}}{\|y\|_2^2}, \quad \sqrt{\hat{\delta}_1^2 + \hat{\delta}_2^2} = \frac{\|\hat{r}\|_2}{\|y\|_2}. \tag{4.12c}$$

As a consequence, we have the following theorem.

THEOREM 4.1. *Let* (τ, z)*, in the form* (3.11)*, be an approximate eigenpair of PQEP* (1.3)*, where* (μ, y) *is the corresponding approximate eigenpair as defined by* (4.9a) *of PQEP* (3.1) *with residual* $\hat{r}$ *by* (4.9b)*. Assume the conditions of* Theorem 3.1*. If* $\|\hat{r}\|_2$ *is sufficiently tiny,*[5] *then* (τ, z) *is an exact eigenpair of a perturbed PQEP* (4.6) *subject to* (4.8)*, and the optimal* α*-structured backward error*

$$\begin{aligned} \varepsilon_\alpha :&= \min_{\Delta A,\, \Delta Q} \sqrt{\|\Delta A\|_{\mathrm{F}}^2 + \|\Delta Q\|_{\mathrm{F}}^2} \quad \text{subject to (4.8)} \\ &\le \sqrt{1 + \frac{(2m-2)\|H_1\|_{\mathrm{F}}^2}{m\|H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2}}\,\hat{\varepsilon}, \end{aligned} \tag{4.13}$$

where $\hat{\varepsilon}$ *is the optimal structured backward error to PQEP* (3.1) *satisfying* (4.12)*.*

Proof. We have (4.9)–(4.12). Let ΔH_0 and ΔH_1 be the optimal ones in the sense of (4.12a), and let $A + \|A\|_{\mathrm{F}}\Delta A$ and $Q + \|Q\|_{\mathrm{F}}\Delta Q$ take the same form as A and Q in (1.3b) but with H_1 and H_0 changed to $H_1 + \|H_1\|_{\mathrm{F}}\Delta H_1$ and $H_0 + \|H_0\|_{\mathrm{F}}\Delta H_0$, respectively. We have perturbed PQEPs for $P_\Delta(\,\cdot\,)$ and $\widehat{P}_\Delta(\,\cdot\,)$ in (4.6) and (4.11), both of which are regular if $\|\hat{r}\|_2$ is sufficiently tiny. Suppose this is the case. We then apply Theorem 3.1 to conclude the proof for all except (4.13), which we shall now prove. We have $\|A\|_{\mathrm{F}} = \|H_1\|_{\mathrm{F}}$ and $\|\Delta A\|_{\mathrm{F}} = \|\Delta H_1\|_{\mathrm{F}}$ and

$$\begin{aligned} \|Q\|_{\mathrm{F}}^2 &= m\|H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2, \\ \|Q\|_{\mathrm{F}}^2\|\Delta Q\|_{\mathrm{F}}^2 &= m\|H_0\|_{\mathrm{F}}^2\|\Delta H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2\|\Delta H_1\|_{\mathrm{F}}^2. \end{aligned}$$

Therefore

$$\begin{aligned} \|\Delta A\|_{\mathrm{F}}^2 + \|\Delta Q\|_{\mathrm{F}}^2 &= \|\Delta H_1\|_{\mathrm{F}}^2 + \frac{m\|H_0\|_{\mathrm{F}}^2\|\Delta H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2\|\Delta H_1\|_{\mathrm{F}}^2}{m\|H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2} \\ &\le \left(1 + \frac{(2m-2)\|H_1\|_{\mathrm{F}}^2}{m\|H_0\|_{\mathrm{F}}^2 + (2m-2)\|H_1\|_{\mathrm{F}}^2}\right)\left(\|\Delta H_1\|_{\mathrm{F}}^2 + \|\Delta H_0\|_{\mathrm{F}}^2\right), \end{aligned}$$

which, together with (4.12), yield (4.13) immediately. ∎

All approximate eigenpairs (τ, z) of PQEP (1.3) solved by Algorithm 3.1 are naturally in the form (3.11). It is reasonable to expect that they should be accurate enough to satisfy "$\|\hat{r}\|_2$ is sufficiently tiny" to ensure $P_\Delta(\,\cdot\,)$ and $\widehat{P}_\Delta(\,\cdot\,)$ in the proof are regular. On the other hand, all approximate eigenpairs (τ, z) of PQEP (1.3) solved by existing methods such as SDA_GL and SDA_LYL are not in the form (3.11) because the eigenvalues τ are not produced as μ^m for some eigenvalues μ of $\widehat{P}(\,\cdot\,)$. However, one can always recover μ as one of the mth roots of τ by, say, checking $\widehat{P}(\tau^{1/m})z_{(1:k)}$ for each mth root $\tau^{1/m}$ and picking one that minimizes $\|\widehat{P}(\tau^{1/m})z_{(1:k)}\|_2$, at a cost of $4k^2 + 4mk$ for each eigenpair (τ, z).

[5] This condition is made clear in the proof: with the optimal ΔH_0 and ΔH_1 in the sense of (4.12a), $P_\Delta(\,\cdot\,)$ and $\widehat{P}_\Delta(\,\cdot\,)$ are regular.

4.4. Accuracy of computed eigenvalues of $P(\cdot)$. We argue that Algorithm 3.1, through solving the eigenvalue problem for $\widehat{P}(\cdot)$, produces more accurate eigenvalues for $P(\cdot)$ than all previous methods, including the ones in [8, 17], through solving the eigenvalue problem for $P(\cdot)$ *directly*. We shall now explain. Hypothetically, suppose that we have the exact stablizing solutions Φ and $\widehat{\Phi}$ to (2.1) and (3.13). Then in theory the eigenvalues τ of $\lambda\Phi + A$ are precisely those of $P(\cdot)$ inside the unit circle, but numerically their computed counterparts will suffer absolute errors of $O(\mathfrak{u})$ in the best scenario but often bigger, which means in general at best

$$\tau_{\text{computed}} = \tau_{\text{exact}} + O(\mathfrak{u}) = \tau_{\text{exact}} \left[1 + O\left(\frac{\mathfrak{u}}{|\tau_{\text{exact}}|}\right)\right], \tag{4.14}$$

where τ_{exact} is the corresponding exact eigenvalue of $P(\cdot)$, where $\mathfrak{u}$ is the machine unit roundoff. On the other hand, in theory the eigenvalues μ of $\lambda\widehat{\Phi}+H_1$ are precisely those of $\widehat{P}(\cdot)$ inside the unit circle. Let us assume that numerically the eigenvalue problem from $\lambda\widehat{\Phi}+H_1$ is solved equally accurately as the one for $\lambda\Phi+A$, i.e., computed μ also suffers an absolute error of $O(\mathfrak{u})$. Consequently, eigenvalues μ inside the unit circle recovered in Algorithm 3.1 via the relation $\tau = \mu^m$ will have various degree of relative accuracy. In fact, write $\mu_{\text{computed}} = \mu_{\text{exact}} + O(\mathfrak{u})$, where μ_{exact} is the corresponding exact eigenvalue of $\widehat{P}(\cdot)$. Then the computed τ_{computed} satisfies

$$\tau_{\text{computed}} = \mu_{\text{exact}}^m [1 + O(\mathfrak{u}/\mu_{\text{exact}})]^m = \tau_{\text{exact}} \left[1 + O\left(m\frac{\mathfrak{u}}{|\tau_{\text{exact}}|^{1/m}}\right)\right]. \tag{4.15}$$

The estimates in (4.14) and (4.15) are very far apart even for a modest m. For example, with $m = 10$ the estimate (4.15) means all eigenvalues of $P(\cdot)$ with magnitudes $10^{-80} < |\tau| \le 1$ are computed with at least eight correct decimal digits in the IEEE double precision! On the other hand, the estimate (4.14) says all eigenvalues of $P(\cdot)$ with magnitudes $|\tau| \le 10^{-16}$ will likely have no correct decimal digits at all.

5. Numerical experiments. Previously in [8], it was demonstrated that SDA_GL produced results with much better accuracy than earlier existing methods and that in [17], SDA_LYL generated equally accurate results. For this reason, we will only numerically compare SDA_LWKLL to SDA_LYL in this article. All numerical experiments are carried out within MATLAB with machine unit roundoff $\mathfrak{u} = 2^{-53} \approx 1.11 \times 10^{-16}$. We use (2.4) with $\texttt{rtol} = 10^{-16}$ to stop all doubling iterations.

As in [17], we will report numerical results on three sets of test data, generated by a finite element method, with

$$(k, m) = (159, 11), \quad (303, 19), \quad (705, 51), \tag{5.1}$$

respectively. The finite element method generates real $k \times k$ matrices K_i and M_i to give

$$H_i = K_i + \iota\omega D_i - \omega^2 M_i \quad \text{with} \quad D_i = c_1 M_i + c_2 K_i$$

for $i = 0, 1$, where ι is the imaginary unit, $\omega > 0$ is the frequency of the external excitation force, and c_1 and c_2 are two positive parameters. For more detail on these examples, see [3, 8, 13, 17]. Table 2 displays the spectral radii

$$\hat{\gamma} := \rho(\widehat{\Phi}^{-1}H_1), \quad \gamma := \rho(\Phi^{-1}A) = \rho(\widetilde{\Phi}^{-1}\widetilde{A}) = \hat{\gamma}^m$$

for the three pairs (k, m) in (5.1). These radii determine the rates of convergence by the doubling algorithms on the respective nonlinear matrix equations (2.1), [17, (3.7)],

TABLE 2
$(\gamma, \hat{\gamma})$ *for various* (k, m) *and* ω.

(k,m) \ ω	100	1000	3000	5000
$(159, 11)$	$(0.959, 0.996)$	$(0.875, 0.988)$	$(0.793, 0.979)$	$(0.741, 0.973)$
$(303, 19)$	$(0.931, 0.996)$	$(0.793, 0.988)$	$(0.669, 0.979)$	$(0.595, 0.973)$
$(705, 51)$	$(0.962, 0.999)$	$(0.883, 0.998)$	$(0.806, 0.996)$	$(0.757, 0.995)$

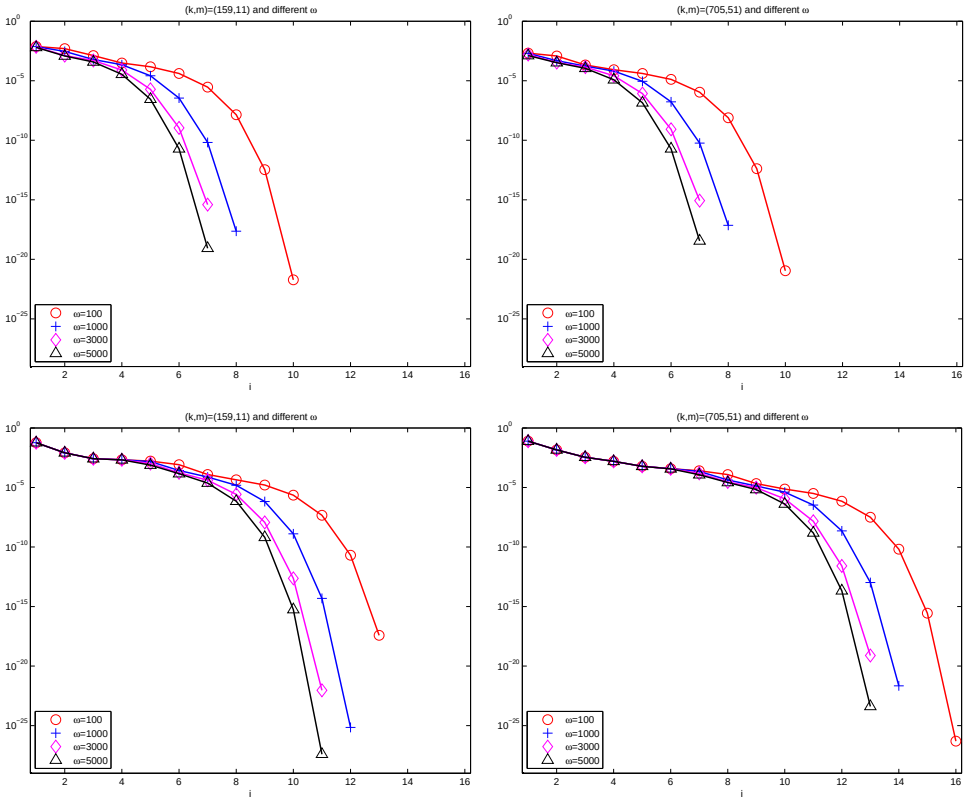

FIG. 2. *Convergence of doubling iterations. Top:* $\frac{\|\widetilde{X}_{i+1}-\widetilde{X}_i\|_2}{\|\widetilde{X}_i\|_2}$ *(*SDA_LYL*); bottom:* $\frac{\|\widehat{X}_{i+1}-\widehat{X}_i\|_2}{\|\widehat{X}_i\|_2}$ *(*SDA_LWKLL*).* $\widehat{X}_i$ *needs about* $\log_2 m$ *more doubling steps to converge that* $\widetilde{X}_i$.

and (3.13). Unfortunately, the degradation from γ to $\hat{\gamma} = \gamma^{1/m}$ in terms of their closeness to 1 from below is substantial, but the good news is, because of the quadratically convergent behavior, the degradation delays the convergence by only about $\log_2 m$ (which are 3.5, 4.2, and 5.7 for $m = 11$, 19, and 51, respectively) doubling steps. For illustrating the convergence history of the doubling iterations, Figure 2 plots the ratios

$$\frac{\|\widetilde{X}_{i+1} - \widetilde{X}_i\|_2}{\|\widetilde{X}_i\|_2}, \quad \frac{\|\widehat{X}_{i+1} - \widehat{X}_i\|_2}{\|\widehat{X}_i\|_2} \tag{5.2}$$

for the first and last pairs of (k, m) in (5.1). What we can see from the figure are that (1) both $\widetilde{X}_i$ and $\widehat{X}_i$ converge very fast to their respective targets, and (2) for $m = 11$ it takes about $\log_2 m \approx 4$ more doubling steps for $\widehat{X}_i$ to be regarded as converged than for $\widetilde{X}_i$ and for $m = 51$ it is about $\log_2 m \approx 6$, as predicted by our earlier analysis. Although not plotted here, the same can be said for $m = 19$.

To measure the accuracy of an approximate eigenpair (τ, z) of $P(\,\cdot\,)$, we use two relative residuals [17]:

$$\text{RRes} := \frac{\|\tau^2 A^{\mathrm{T}} z + \tau Q z + A z\|_2}{(|\tau|^2\|A\|_{\mathrm{F}} + |\tau|\|Q\|_{\mathrm{F}} + \|A\|_{\mathrm{F}})\|z\|_2}, \tag{5.3}$$

$$\text{RRes}_{\text{new}} := \frac{\|\tau^2 A^{\mathrm{T}} z + \tau Q z + A z\|_2}{|\tau|^2\|H_1\|_{\mathrm{F}}\|z_1\|_2 + |\tau|\|Q\|_{\mathrm{F}}\|z\|_2 + \|H_1\|_{\mathrm{F}}\|z_m\|_2}, \tag{5.4}$$

where the Frobenius norm $\|\cdot\|_{\mathrm{F}}$ is chosen for easy computation but it could be replaced by any other matrix norm, and z is partitioned into an m-block vector: $z = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}, \ldots, z_m^{\mathrm{T}}]^{\mathrm{T}}$ with $z_i \in \mathbb{C}^k$. Some comments are in order. RRes is generic and is commonly used. Usually the best one could hope for RRes is to reduce it to about $O(\mathfrak{u}) = O(10^{-16})$. But as analyzed and argued in [17], this RRes for the current problem is not suitable. The more appropriate one is RRes_{new} due to the special subblock structure in A that alters rounding error characteristics in evaluating $P(\tau)z$. In fact, RRes can uncharacteristically reach down to $O(10^{-35})$ or smaller.

Similarly to (5.3), the relative residual for an approximate eigenpair (μ, y) of $\widehat{P}(\,\cdot\,)$ in (3.1) is

$$\text{RRes} := \frac{\|\mu^2 H_1^{\mathrm{T}} y + \mu H_0 y + H_1 y\|_2}{(|\mu|^2\|H_1\|_{\mathrm{F}} + |\mu|\|H_0\|_{\mathrm{F}} + \|H_1\|_{\mathrm{F}})\|y\|_2}. \tag{5.5}$$

In Figure 3, we use the first and last pairs of (k, m) in (5.1) as examples and also for $\omega = 1000$ only. We plot RRes (5.3) and RRes_{new} (5.4) for all approximate eigenpairs of $P(\,\cdot\,)$ and also RRes (5.5) for all approximate eigenpairs of $\widehat{P}(\,\cdot\,)$ in (3.1). As it shows, RRes (5.5) for $\widehat{P}(\,\cdot\,)$ is always about $O(\mathfrak{u})$ for the computed eigenpairs, regardless of its eigenvalue magnitudes. But for $P(\,\cdot\,)$, RRes_{new} (5.4) is always about $O(\mathfrak{u})$, while RRes (5.3) for τ with tiny or huge magnitude are skewed up to even $O(10^{-160})$. The reason behind it is that for tiny $|\tau|$, $\|z_m\|_2/\|z\|_2$ is also tiny, while for huge $|\tau|$, $\|z_1\|_2/\|z\|_2$ is tiny.

Last, for each computed eigenpair (τ, z) of PQEP (1.3), we check the structured backward errors (4.7) and the α-structured backward errors (4.13). This is done in Figure 4, where in calculating (4.7) for quite many largest $|\lambda_i|$, the denominators in (4.7a) overflow to ∞ and thus the corresponding ε are computed to 0. Consequently, these ε do not show up in the log-log plots but they would continue the trend if overflows hadn't happened. It is interesting to see how tiny ε of (4.7a) can get if backward perturbations are allowed in every possible position in A and Q. On the other hand, the α-structured backward errors always hover around $O(\mathfrak{u})$.

6. Conclusion. The fast train PQEP for $P(\lambda) = \lambda^2 A^{\mathrm{T}} + \lambda Q + A$ in (1.3) has additional structures in its coefficient matrices A and Q beyond the usual "palindromic" appearance, namely, both A and Q are partitioned block matrices, Q is complex symmetric and tridiagonal block-Toeplitz, and A has only one nontrivial block in its upper-right corner and that block is the same as the subdiagonal block of Q (see (1.3)). In previous SDA-based solvent approaches [8, 17], these rich structures

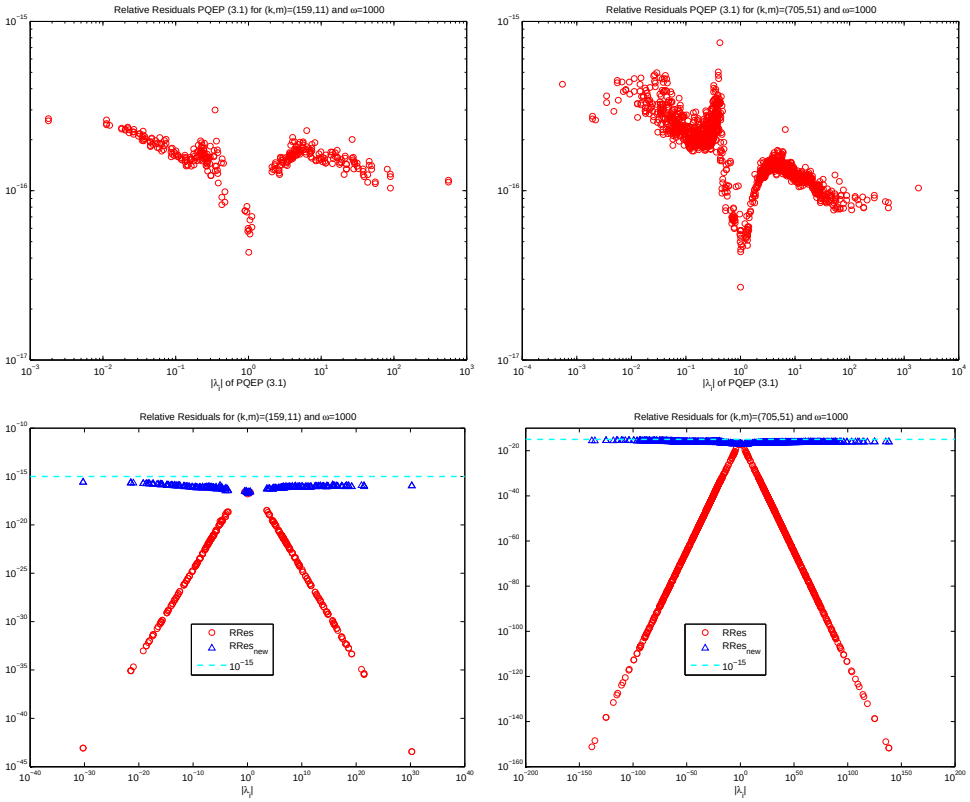

FIG. 3. *Relative residuals. Top row:* RRes (5.5) *for approximate eigenpairs of* $\widehat{P}(\,\cdot\,) = \lambda^2 H_1^{\mathrm{T}} + \lambda H_0 + H_1$ *in* (3.1)*; bottom row:* RRes (5.3) *and* $\mathrm{RRes}_{\mathrm{new}}$ (5.4) *for approximate eigenpairs of* $P(\,\cdot\,) = \lambda^2 A^{\mathrm{T}} + \lambda Q + A$ *in* (1.3).

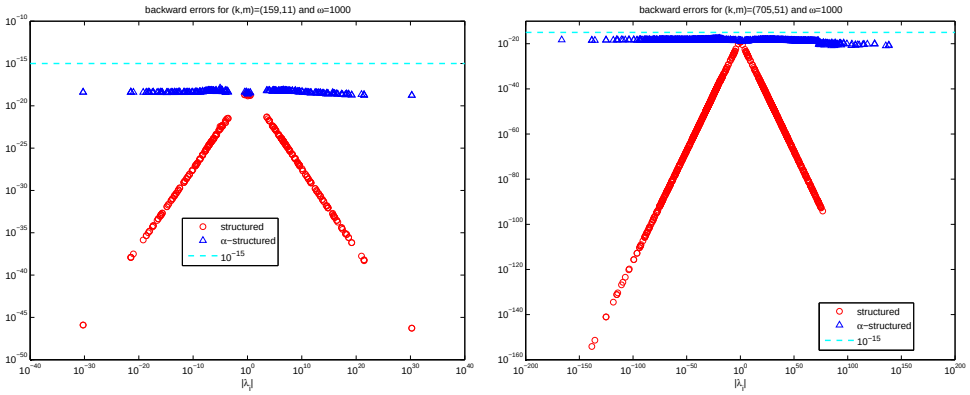

FIG. 4. *Structured backward errors* (4.7) *and* α*-structured backward errors* (4.13)*. For quite many largest* $|\lambda_i|$ *for the case* $k = 705$*, the denominators in* (4.7a) *overflow to* ∞ *to give* $\varepsilon = 0$*. These* ε *do not show up. But they would continue the trend in the plot if overflows hadn't happened.*

were taken advantage of but not to their fullest, namely, the block-Toeplitz property and that the nontrivial block of A is the same as the subdiagonal block of Q went unused. In this article, we take advantage of all that we know. In fact, our main theoretical results in Theorem 3.1 that relate the eigen-information of $P(\lambda)$ to that of $\widehat{P}(\lambda) = \lambda^2 H_1^{\mathrm{T}} + \lambda H_0 + H_1$ are the direct consequences of these additional structural properties previously unused. Based on the results, we propose a new fast algorithm SDA_LWKLL to find all interesting eigenpairs of $P(\lambda)$ through solving the PQEP for $\widehat{P}(\lambda)$. In terms of flops, SDA_LWKLL is at least 5 times faster than SDA_GL [8] and 3 times faster than SDA_LYL [17]. Nonetheless, SDA_GL and SDA_LYL have much wider applicability than SDA_LWKLL in that the first two methods do not require that Q be tridiagonal block-Toeplitz as in (1.3) but just block-tridiagonal.

We also explored the α-structured backward error analysis that precisely honors all possible subblock structures in A and Q displayed in (1.3b). It is so named in order to distinguish the previous structured backward error analysis in [16] for general palindromic polynomial eigenvalue problems.

Acknowledgment. The authors wish to thank two anonymous referees for their constructive suggestions and comments that considerably improve the presentation of the article. We are especially indebted to one of the referees for reminding us the availability of (3.17) at the time of solving (3.16), leading to a more efficient way than what we previously had in mind.

REFERENCES

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

[2] C.-Y. Chiang, E. K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, and S.-F. Xu, *Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 227–247.

[3] E. K.-W. Chu, T.-M. Hwang, W.-W. Lin, and C.-T. Wu, *Vibration of fast trains, palindromic eigenvalue problems and structure-preserving doubling algorithms*, J. Comput. Appl. Math., 219 (2008), pp. 237–252.

[4] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[5] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Classics in Appl. Math. 58, SIAM, Philadelphia, 2009.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[7] C.-H. Guo, Y.-C. Kuo, and W.-W. Lin, *Numerical solution of nonlinear matrix equations arising from Green's function calculations in nano research*, J. Comput. Appl. Math., 236 (2012), pp. 4166–4180.

[8] C.-H. Guo and W.-W. Lin, *Solving a structured quadratic eigenvalue problem by a structure-preserving doubling algorithm*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2784–2801.

[9] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1218–1241.

[10] A. Hilliges, *Numerische Lösung von quadratischen Eigenwertproblemen mit Anwendung in der Schienendynamik*, diplomarbeit, Institut für Mathematik, Technical University Berlin, 2004.

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

[12] T.-M. Huang, W.-W. Lin, and J. Qian, *Structure-preserving algorithms for palindromic quadratic eigenvalue problems arising from vibration on fast trains*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1566–1592.

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

[14] D. Kressner, C. Schröder, and D. S. Watkins, *Implicit QR algorithms for palindromic and even eigenvalue problems*, Numer. Algorithms, 51 (2009), pp. 209–238.

[15] P. Lancaster, U. Prells, and L. Rodman, *Canonical structures for palindromic matrix polynomials*, Oper. Matrices, 1 (2007), pp. 469–489.
[16] R.-C. Li, W.-W. Lin, and C.-S. Wang, *Structured backward error for palindromic polynomial eigenvalue problems*, Numer. Math., 116 (2010), pp. 95–122.
[17] L. Lu, F. Yuan, and R.-C. Li, *A new look at the doubling algorithm for a structured palindromic quadratic eigenvalue problem*, Numer. Linear Algebra Appl., 22 (2015), pp. 393–409.
[18] 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 (2006), pp. 1029–1051.
[19] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular Schur form*, Numer. Linear Algebra Appl., 16 (2009), pp. 63–86.
[20] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.