DERIVATIVE INTERPOLATING SUBSPACE FRAMEWORKS FOR NONLINEAR EIGENVALUE PROBLEMS

RIFQI AZIZ*, EMRE MENGI†, AND MATTHIAS VOIGT‡

Abstract. We first consider the problem of approximating a few eigenvalues of a rational matrix-valued function closest to a prescribed target. It is assumed that the proper rational part of the rational matrix-valued function is expressed in the transfer function form $H(s) = C(sI - A)^{-1}B$, where the middle factor is large, whereas the number of rows of C and the number of columns of B are equal and small. We propose a subspace framework that performs two-sided or one-sided projections on the state-space representation of $H(\cdot)$, commonly employed in model reduction and giving rise to a reduced transfer function. At every iteration, the projection subspaces are expanded to attain Hermite interpolation conditions at the eigenvalues of the reduced transfer function closest to the target, which in turn leads to a new reduced transfer function. We prove in theory that, when a sequence of eigenvalues of the reduced transfer functions converges to an eigenvalue of the full problem, it converges at least at a quadratic rate. In the second part, we extend the proposed framework to locate the eigenvalues of a general square large-scale nonlinear meromorphic matrix-valued function $T(\cdot)$, where we exploit a representation $\mathcal{R}(s) = C(s)A(s)^{-1}B(s) - D(s)$ defined in terms of the block components of $T(\cdot)$. The numerical experiments illustrate that the proposed framework is reliable in locating a few eigenvalues closest to the target point, and that, with respect to runtime, it is competitive to established methods for nonlinear eigenvalue problems.

Key words. Nonlinear eigenvalue problems, large scale, subspace projections, Hermite interpolation, quadratic convergence, rational eigenvalue problems.

AMS subject classifications. 65F15, 65D05, 34K17

1. Introduction.

The numerical solutions of nonlinear eigenvalue problems have been a major field of research in the last twenty years [21, 13]. Numerical algorithms are proposed to estimate the eigenvalues of a nonlinear matrix-valued function either within a prescribed region, or closest to a prescribed target point in the complex plane.

Earlier works are mostly focused on polynomial and rational eigenvalue problems [26, 19, 25]. More recently, some of the attention has shifted to nonlinear eigenvalue problems that are neither polynomial nor rational. Various applications give rise to such non-polynomial, non-rational eigenvalue problems, including the stability analysis of delay systems [13], numerical solutions of elliptic PDE eigenvalue problems by the boundary element method [9], or finite element discretizations of differential equations with nonlinear boundary conditions depending on an eigenvalue parameter [6].

The nonlinear eigenvalue problem setting that we consider in this work is as follows. Let

$$T(s) := f_1(s)T_1 + \cdots + f_\kappa(s)T_\kappa, \tag{1.1}$$

where the functions $f_1, \ldots, f_\kappa : \mathbb{C} \to \mathbb{C}$ are meromorphic, and $T_1, \ldots, T_\kappa \in \mathbb{C}^{\mathsf{n}\times\mathsf{n}}$ are given matrices. Assume that the set $\{\lambda \in \mathbb{C} \mid \operatorname{rank}_{\lambda\in\mathbb{C}} T(\lambda) < \mathsf{n}\}$ consists only of

*Koç University, Department of Mathematics, Rumeli Feneri Yolu 34450, Sarıyer, Istanbul, Turkey, E-Mail: raziz14@ku.edu.tr.

†Koç University, Department of Mathematics, Rumeli Feneri Yolu 34450, Sarıyer, Istanbul, Turkey, E-Mail: emengi@ku.edu.tr.

‡Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany, E-Mail: mvoigt@math.tu-berlin.de.

isolated points. Then we want to find $\lambda \in \mathbb{C}$ and $v \in \mathbb{C}^{\mathsf{n}} \setminus \{0\}$ such that

$$T(\lambda)v = 0. \tag{1.2}$$

The scalar $\lambda \in \mathbb{C}$ satisfying (1.2) is called an eigenvalue, and the vector $v \in \mathbb{C}^{\mathsf{n}} \setminus \{0\}$ is called a corresponding eigenvector. This setting is quite general. For instance, polynomial and rational eigenvalue problems are special cases when $f_j(\cdot)$ are scalar-valued polynomials and rational functions, respectively. Delay eigenvalue problems can also be expressed in this form such that some of $f_j(\cdot)$ are exponential functions.

We propose an interpolation-based subspace framework to find a prescribed number of eigenvalues of $T(\cdot)$ closest to a given target point $\tau \in \mathbb{C}$. At every iteration, a projected small-scale nonlinear eigenvalue problem is solved. Then the projection subspaces are expanded so as to satisfy Hermite interpolation properties at the eigenvalues of the projected problem. The projections we rely on are devised from two-sided or one-sided projections commonly employed in model-order reduction [5].

Our approach could be compared with linearization-based techniques for nonlinear eigenvalue problems. However, such techniques first use either polynomial interpolation [9, 27] or rational interpolation [14] to approximate the nonlinear matrix-valued function with a polynomial or a rational matrix-valued function. Then the polynomial and rational eigenvalue problems are linearized into generalized eigenvalue problems. In order to deal with large-scale problems, typically Krylov subspace methods are applied to the generalized eigenvalue problem in an efficient manner, in particular taking the structure of the linearization into account; see for instance [28] and [18] for a one-sided and a two-sided rational Arnoldi method, respectively. Further advances in approximating a nonlinear eigenvalue problem by a rational one have been proposed very recently. In the recent works [12, 17], certain variants of the Antoulas-Anderson algorithm (AAA) are investigated for this purpose. Also [12] gives a detailed error analysis, i. e., the authors investigate how the quality of the rational approximation affects the quality of the computed eigenvalues. Based on this analysis, they propose a new termination condition. Another recent work [7] combines data-driven interpolation approaches by the Loewner framework with contour-integral methods to compute all eigenvalues inside a specified contour.

The approach proposed here is somewhat related to [7] since we also use interpolation, but otherwise it differs from the above-mentioned works. We apply subspace projections directly to the nonlinear eigenvalue problem (and hence, preserve its nonlinear structure), and the projection subspaces are not necessarily Krylov subspaces. Consequently, our approach assumes the availability of numerical techniques for the solutions of the projected small-scale nonlinear eigenvalue problems. In the case of polynomial or rational eigenvalue problems, linearization based techniques are available to our use to obtain all of the eigenvalues of the small-scale problem. Our numerical experience is that the proposed frameworks here are comparable to the state-of-the-art methods in terms of computational efficiency, and even a few times faster in some cases.

Outline. In the next section, we first describe an interpolatory subspace framework specifically for rational eigenvalue problems. For instance, for a proper rational matrix-valued function, which can always be expressed in the form $R(s) = C(sI_{\mathsf{k}} - A)^{-1}B$ for some $A \in \mathbb{C}^{\mathsf{k}\times\mathsf{k}}$, $B \in \mathbb{C}^{\mathsf{k}\times\mathsf{n}}$, and $C \in \mathbb{C}^{\mathsf{n}\times\mathsf{k}}$, the framework addresses the case when $\mathsf{k} \gg \mathsf{n}$ and reduces the dimension of the middle factor. We give formal arguments establishing the at least quadratic convergence of the proposed subspace framework. In Section 3, we extend the subspace framework idea for rational eigenvalue problems to the general nonlinear eigenvalue problem setting of (1.2).

Sections 2 and 3 present the frameworks to locate only one eigenvalue closest to the prescribed target, and employ two-sided projections. Section 4 discusses how the frameworks can be generalized to locate a prescribed number of closest eigenvalues to the target, while Section 5 describes how one-sided projections can be adopted in place of two-sided projections. Finally, Section 6 illustrates the frameworks on synthetic examples, as well as classical examples from the NLEVP data collection [6], and confirms the validity of the theoretical findings in practice.

2. Rational Eigenvalue Problems. A special important class of nonlinear eigenvalue problems are rational eigenvalue problems. There we want to find $\lambda \in \mathbb{C}$ and $v \in \mathbb{C}^{\mathsf{n}} \setminus \{0\}$ such that

$$R(\lambda)v = 0, \tag{2.1}$$

where $R(\cdot)$ is a rational matrix-valued function. We again assume that the set $\{\lambda \in \mathbb{C} \mid \operatorname{rank}_{\lambda\in\mathbb{C}} R(\lambda) < \mathsf{n}\}$ consists only of isolated points. The significance of the rational eigenvalue problem is due to several reasons. First, it is a cornerstone for the solutions of nonlinear eigenvalue problems that are not rational; such nonlinear eigenvalue problems are often approximated by rational eigenvalue problems. Secondly, there are several applications that give rise to rational eigenvalue problems such as models for vibrations of fluid-solid structure, as well as vibrating mechanical structures [21].

There are several ways to represent the function $R(\cdot)$. One possibility is to write it as

$$R(s) = P(s) + \sum_{j=1}^{\rho} \frac{p_j(s)}{d_j(s)} E_j, \tag{2.2}$$

where $d_j, p_j : \mathbb{C} \to \mathbb{C}$ are polynomials of degree k_j and strictly less than k_j, respectively. Moreover, $E_j \in \mathbb{C}^{\mathsf{n}\times\mathsf{n}}$ for $j = 1, \ldots, \rho$ are given matrices, and $P(\cdot)$ is a matrix polynomial of degree d of the form $P(s) := s^{\mathsf{d}}P_{\mathsf{d}} + \cdots + sP_1 + P_0$ for given $P_0, \ldots, P_{\mathsf{d}} \in \mathbb{C}^{\mathsf{n}\times\mathsf{n}}$.

It is usually the case that the matrices E_j in (2.2) are of low rank, hence they can be decomposed into

$$E_j = L_j U_j^* \tag{2.3}$$

for some $L_j, U_j \in \mathbb{C}^{\mathsf{n}\times\mathsf{r}_j}$ of full column rank such that $\mathsf{r}_j \ll \mathsf{n}$. In this case, the proper rational part of $R(s)$ can always be expressed as a transfer function associated with a linear time-invariant system. Formally, it can be shown that

$$\sum_{j=1}^{\rho} \frac{p_j(s)}{d_j(s)} E_j = C(sI_{\mathsf{k}} - A)^{-1}B \tag{2.4}$$

in (2.2) for some $A \in \mathbb{C}^{\mathsf{k}\times\mathsf{k}}$, $B \in \mathbb{C}^{\mathsf{k}\times\mathsf{n}}$, $C \in \mathbb{C}^{\mathsf{n}\times\mathsf{k}}$, where $\mathsf{k} := \mathsf{r}_1\mathsf{k}_1 + \cdots + \mathsf{r}_\rho\mathsf{k}_\rho$. We refer to [25] and [3, page 95] for the details of the construction of A, B, C from the polynomials p_j, d_j in (2.2) and matrices L_j, U_j in (2.3).

The expression of the rational eigenvalue problem in the form

$$R(s) = P(s) + C(sI_{\mathsf{k}} - A)^{-1}B \tag{2.5}$$

(or more generally, $R(s) = P(s) + C(s)D(s)^{-1}B(s)$ with matrix polynomials $P(\cdot)$, $A(\cdot)$, $B(\cdot)$, and $C(\cdot)$) is sometimes immediately available. Indeed, this is another way

to formulate a rational eigenvalue problem. This formulation plays a more prominent role in applications from linear systems and control theory. For example, the eigenvalues of the rational function $R(s) = C(sI_{\mathsf{k}} - A)^{-1}B$ are the so-called *transmission zeros* of the linear time-invariant system

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \tag{2.6}$$

see for example [10]. The transmission zeros play prominent roles in electronics applications such as in oscillation damping control [20] or the design of filters [23]. In the latter applications, the transmission zeros are used to specify frequency bands for which input signals are rejected.

One way of dealing with (2.1) is to convert it into a generalized eigenvalue problem. For instance, for any $\mathsf{k} \times \mathsf{k}$ matrix F, we have

$$C(\lambda F - A)^{-1}Bv = 0 \quad \Rightarrow \quad \left(\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} - \lambda \begin{bmatrix} F & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} (\lambda F - A)^{-1}Bv \\ v \end{bmatrix} = 0. \tag{2.7}$$

More generally, $\left\{P(\lambda) + C(\lambda F - A)^{-1}B\right\} v = 0$ for $P(s) = \sum_{j=0}^{\mathsf{d}} s^j P_j$ and for any $\mathsf{k} \times \mathsf{k}$ matrix F can be linearized into

$$(\mathcal{A} - \lambda\mathcal{B})z = 0, \qquad \text{where} \quad z := \begin{bmatrix} (\lambda F - A)^{-1}Bv \\ \hline \lambda^{\mathsf{d}-1}v \\ \lambda^{\mathsf{d}-2}v \\ \vdots \\ v \end{bmatrix},$$

$$\mathcal{A} := \left[\begin{array}{c|cccc} A & & & & B \\ \hline C & P_{\mathsf{d}-1} & P_{\mathsf{d}-2} & \dots & P_0 \\ & -I_{\mathsf{n}} & 0 & \dots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & -I_{\mathsf{n}} & 0 \end{array}\right], \quad \mathcal{B} := \left[\begin{array}{c|cccc} F & & & & \\ \hline & -P_{\mathsf{d}} & & & \\ & & -I_{\mathsf{n}} & & \\ & & & \ddots & \\ & & & & -I_{\mathsf{n}} \end{array}\right], \tag{2.8}$$

where we set $\mathsf{d} = 1$ and $P_1 = 0$ in the case $P(s) \equiv P_0$. For $F = I_{\mathsf{n}}$, there is a one-to-one correspondence between the eigenvalues of $L(s) := \mathcal{A} - s\mathcal{B}$ defined as in (2.8) and $R(\cdot)$ as in (2.1). In particular, for this choice of F the following holds:

- If λ is an eigenvalue of the pencil $L(s) := \mathcal{A} - s\mathcal{B}$, but not an eigenvalue of A, then λ is also an eigenvalue of $R(\cdot)$ [25].
- Conversely, if λ is an eigenvalue of $R(\cdot)$, it is also an eigenvalue of $L(\cdot)$; see also [24].

2.1. The Subspace Method for Rational Eigenvalue Problems. The setting we aim to address in this section is when the rational eigenvalue problem (2.1) is given in the transfer function form (2.5), where the size of the matrix A is very large compared to $\mathsf{n} \cdot \mathsf{d}$ (i.e., the size of $R(\cdot)$ · the degree of $P(\cdot)$), that is $\mathsf{k} \gg \mathsf{n} \cdot \mathsf{d}$. In the special case when $P(s) \equiv P_0$, we aim to address the setting when $\mathsf{k} \gg \mathsf{n}$.

Here, we propose a subspace framework that replaces the proper rational part $R_{\mathrm{p}}(s) := C(sI - A)^{-1}B$ of $R(\cdot)$ with a reduced one of the form

$$R_{\mathrm{p}}^{\mathcal{W},\mathcal{V}}(s) := CV(sW^*V - W^*AV)^{-1}W^*B$$

for two subspaces $\mathcal{W}$, $\mathcal{V} \subseteq \mathbb{C}^{\mathsf{k}}$ of equal dimension, say r such that $\mathsf{r} \ll \mathsf{k}$, and matrices W, $V \in \mathbb{C}^{\mathsf{k}\times\mathsf{r}}$ whose columns form orthonormal bases for the subspaces $\mathcal{W}$, $\mathcal{V}$, respectively. We remark that the $\mathsf{r} \times \mathsf{r}$ middle factor of the reduced proper rational function

is much smaller than the $\mathsf{k} \times \mathsf{k}$ middle factor of the full problem. The full rational function $R_{\mathrm{p}}(\cdot)$ and the reduced one $R_{\mathrm{p}}^{\mathcal{W},\mathcal{V}}(\cdot)$ are the transfer functions of the linear time-invariant systems (2.6) and

$$\frac{\mathrm{d}}{\mathrm{d}t} W^*Vx(t) = W^*AVx(t) + W^*Bu(t), \quad y(t) = CVx(t),$$

respectively. Hence, in the system setting, replacing $R_{\mathrm{p}}(\cdot)$ with $R_{\mathrm{p}}^{\mathcal{W},\mathcal{V}}(\cdot)$ corresponds to restricting the state-space of (2.6) to $\mathcal{V}$, and then imposing a Petrov-Galerkin condition on the residual of the restricted state-space system to $\mathcal{W}$.

Our approach is interpolatory and inspired by model order reduction techniques [29, 8, 11, 4], as well as by a recent subspace framework proposed for the estimation of the $\mathcal{H}_\infty$ norm of a transfer function [1]. The problem at hand (2.1) can be viewed as the minimization problem

$$\min_{\lambda \in \mathbb{C}} \sigma_{\min}(R(\lambda)).$$

Rather than this problem, at every iteration, we solve a reduced problem of the form

$$\min_{\lambda \in \mathbb{C}} \sigma_{\min}\big(R^{\mathcal{W},\mathcal{V}}(\lambda)\big), \tag{2.9}$$

where $R^{\mathcal{W},\mathcal{V}}(s) := P(s) + R_{\mathrm{p}}^{\mathcal{W},\mathcal{V}}(s)$. Then we expand the subspaces $\mathcal{W}, \mathcal{V}$ to $\widetilde{\mathcal{W}}, \widetilde{\mathcal{V}}$ so that

$$R\big(\widetilde{\lambda}\big) = R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\big(\widetilde{\lambda}\big) \quad \text{and} \quad R'\big(\widetilde{\lambda}\big) = \Big[R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\Big]'\big(\widetilde{\lambda}\big) \tag{2.10}$$

at a global minimizer $\widetilde{\lambda}$ of (2.9). The procedure is repeated by solving another reduced problem as in (2.9), but with $\widetilde{\mathcal{W}}, \widetilde{\mathcal{V}}$ taking the role of $\mathcal{W}, \mathcal{V}$.

One neat issue here is that, recalling $L(s) = \mathcal{A} - s\mathcal{B}$ with $\mathcal{A}, \mathcal{B}$ as in (2.8) for $F = I_n$, finding the global minimizers of (2.9) amounts to computing the eigenvalues of the pencil

$$L^{W,V}(s) := \begin{bmatrix} W^* & 0 \\ 0 & I_{\mathsf{nd}} \end{bmatrix} L(s) \begin{bmatrix} V & 0 \\ 0 & I_{\mathsf{nd}} \end{bmatrix} = \mathcal{A}^{W,V} - s\mathcal{B}^{W,V}, \quad \text{where}$$

$$\mathcal{A}^{W,V} := \left[\begin{array}{c|cccc} W^*AV & & & & W^*B \\ \hline CV & P_{\mathsf{d}-1} & P_{\mathsf{d}-2} & \cdots & P_0 \\ & -I_{\mathsf{n}} & 0 & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & -I_{\mathsf{n}} & 0 \end{array}\right], \tag{2.11}$$

$$\mathcal{B}^{W,V} := \left[\begin{array}{c|ccccc} W^*V & & & & & \\ \hline & -P_{\mathsf{d}} & & & & \\ & & -I_{\mathsf{n}} & & & \\ & & & \ddots & & \\ & & & & -I_{\mathsf{n}} \end{array}\right],$$

which is immediate from (2.8) by replacing A, B, C, and F with W^*AV, W^*B, CV, and W^*V, respectively. We remark that the pencil $L^{W,V}(\cdot)$ in (2.11) is of size $(\mathsf{r}+\mathsf{n}\cdot\mathsf{d})\times(\mathsf{r}+\mathsf{n}\cdot\mathsf{d})$, whereas the original pencil $L(s) = \mathcal{A} - s\mathcal{B}$ is of size $(\mathsf{k}+\mathsf{n}\cdot\mathsf{d})\times(\mathsf{k}+\mathsf{n}\cdot\mathsf{d})$. As for the choice of at which eigenvalue $\widetilde{\lambda}$ of $L^{W,V}(\cdot)$ we would Hermite interpolate,

we prescribe a target τ a priori, and choose $\widetilde{\lambda}$ as the eigenvalue of $L^{W,V}(\cdot)$ closest to τ.

The only remaining issue that needs to be explained is how we expand the subspaces $\mathcal{W}, \mathcal{V}$ into $\widetilde{\mathcal{W}}, \widetilde{\mathcal{V}}$ so as to satisfy (2.10). Fortunately, the tools for this purpose have already been established as elaborated in the following result. This result is an immediate corollary of [5, Theorem 1].

LEMMA 2.1. *Suppose that $\mu \in \mathbb{C}$ is not an eigenvalue of A. Let $\widetilde{\mathcal{W}} = \mathcal{W} \oplus \mathcal{W}_\mu$ and $\widetilde{\mathcal{V}} = \mathcal{V} \oplus \mathcal{V}_\mu$, where $\mathcal{V}, \mathcal{W}$ are given subspaces of equal dimension, and $\mathcal{W}_\mu$, $\mathcal{V}_\mu$ are subspaces defined as*

$$\mathcal{V}_\mu := \bigoplus_{j=1}^{\mathsf{q}} \operatorname{Ran}\left((A-\mu I_{\mathsf{k}})^{-j}B\right) \quad \text{and} \quad \mathcal{W}_\mu := \bigoplus_{j=1}^{\mathsf{q}} \operatorname{Ran}\left(\left(C(A-\mu I_{\mathsf{k}})^{-j}\right)^*\right)$$

for some positive integer q. Let $\widetilde{V}$ and $\widetilde{W}$ be basis matrices of $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{W}}$, respectively and assume further that $\widetilde{W}^*A\widetilde{V} - \mu\widetilde{W}^*\widetilde{V}$ is invertible. Then we have

1. *$R(\mu) = R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}(\mu)$, and*
2. *$R^{(j)}(\mu) = \left[R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\right]^{(j)}(\mu)$ for $j = 1, \ldots, 2\mathsf{q}-1$,*

where $R^{(j)}(\cdot)$ and $\left[R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\right]^{(j)}(\cdot)$ denote the jth derivatives of $R(\cdot)$ and $R^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}(\cdot)$.

The resulting subspace method is described formally in Algorithm 2.1, where we assume that the proper rational part of $R(\cdot)$ is provided as an input in the transfer function form (2.4) in terms of A, B, C. At iteration ℓ, the subspaces $\mathcal{V}_{\ell-1}, \mathcal{W}_{\ell-1}$ are expanded into $\mathcal{V}_\ell, \mathcal{W}_\ell$ in order to achieve $R(\lambda_\ell) = R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)$ as well as $R^{(j)}(\lambda_\ell) = \left[R^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]^{(j)}(\lambda_\ell)$ for $j = 1, \ldots, 2\mathsf{q}-1$. Lines 3–12 of the algorithm fulfill this expansion task by augmenting $V_{\ell-1}$, $W_{\ell-1}$, matrices whose columns form orthonormal bases for $\mathcal{V}_{\ell-1}$ and $\mathcal{W}_{\ell-1}$, with additional columns as suggested by Theorem 2.1. Orthonormalizing the augmented matrices gives rise to the matrices V_ℓ, W_ℓ whose columns span the expanded subspaces $\mathcal{V}_\ell$, $\mathcal{W}_\ell$, respectively. The next interpolation point $\lambda_{\ell+1}$ is then set equal to the eigenvalue of $L^{W_\ell,V_\ell}(\cdot)$ closest to the target point τ. Note that in line 15 of Algorithm 2.1, letting $\mathsf{r} := \dim \mathcal{V}_\ell$, the vector $v_{\ell+1}$ is of size $\mathsf{r}+\mathsf{nd}$, and $v_{\ell+1}(\mathsf{r}+\mathsf{n}(\mathsf{d}-1)+1 : \mathsf{r}+\mathsf{nd})$, denoting the vector composed of the last n entries of $v_{\ell+1}$, is an eigenvector estimate according to (2.8).

Choosing orthonormal bases for $\mathcal{V}_\ell$ and $\mathcal{W}_\ell$ ensures that the norms of the projected matrices do not grow. Moreover, the projection matrices V_ℓ, W_ℓ after orthonormalization are well-conditioned, whereas, without orthonormalization, the subspace method above is likely to yield ill-conditioned projection matrices; see the discussions at the beginning of Section 6 regarding the orthogonalization of the bases for details. These are desirable properties from the numerical point of view. On the other hand, in projection-based model order reduction for linear ODE systems, often a bi-orthonormality condition $W_\ell^*V_\ell = I$ is enforced, e. g., by employing the non-symmetric Lanczos process. This ensures that the projected system is again an ODE. The benefit of such a condition in our context is however not clear a priori. In fact, to our best knowledge, an in-depth analysis of the influence of the choice of bases on the numerical properties of the projected problem is still missing in the literature.

The subsequent three subsections of this section establish the quadratic convergence of Algorithm 2.1. The arguments operate on the singular values of $R(s)$ and $R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$, especially their smallest singular values. Sections 2.2 and 2.3 focus on the interpolatory properties between these singular values, and the analytical properties of

Algorithm 2.1 Subspace method to compute a rational eigenvalue closest to a prescribed target

Input: matrices $P_1, \ldots, P_{\mathsf{d}} \in \mathbb{C}^{\mathsf{n}\times\mathsf{n}}$ as in (2.1), and $A \in \mathbb{C}^{\mathsf{k}\times\mathsf{k}}$, $B \in \mathbb{C}^{\mathsf{k}\times\mathsf{n}}$, $C \in \mathbb{C}^{\mathsf{n}\times\mathsf{k}}$ as in (2.4), interpolation parameter $\mathsf{q} \in \mathbb{Z}$ with $\mathsf{q} \geq 2$, target $\tau \in \mathbb{C}$.

Output: estimate $\lambda \in \mathbb{C}$ for the eigenvalue closest to τ and corresponding eigenvector estimate $v \in \mathbb{C}^{\mathsf{n}} \setminus \{0\}$.

$\lambda_0 \leftarrow \tau$.

% **main loop**

for $\ell = 0, 1, 2, \ldots$ **do**

  $\widehat{V}_\ell \leftarrow (A - \lambda_\ell I)^{-1}B$, $\widetilde{V}_\ell \leftarrow \widehat{V}_\ell$, $\widehat{W}_\ell \leftarrow (A - \lambda_\ell I)^{-*}C^*$, and $\widetilde{W}_\ell \leftarrow \widehat{W}_\ell$.

  for $j = 2, \ldots, \mathsf{q}$ **do**

    $\widehat{V}_\ell \leftarrow (A - \lambda_\ell I)^{-1}\widehat{V}_\ell$ and $\widetilde{V}_\ell \leftarrow \begin{bmatrix}\widetilde{V}_\ell & \widehat{V}_\ell\end{bmatrix}$.

    $\widehat{W}_\ell \leftarrow (A - \lambda_\ell I)^{-*}\widehat{W}_\ell$ and $\widetilde{W}_\ell \leftarrow \begin{bmatrix}\widetilde{W}_\ell & \widehat{W}_\ell\end{bmatrix}$.

  end for

  % **expand the subspaces to interpolate at** λ_ℓ

  if $\ell = 0$ **then**

    $V_0 \leftarrow \operatorname{orth}\left(\widetilde{V}_0\right)$ and $W_0 \leftarrow \operatorname{orth}\left(\widetilde{W}_0\right)$.

  else

    $V_\ell \leftarrow \operatorname{orth}\left(\begin{bmatrix}V_{\ell-1} & \widetilde{V}_\ell\end{bmatrix}\right)$ and $W_\ell \leftarrow \operatorname{orth}\left(\begin{bmatrix}W_{\ell-1} & \widetilde{W}_\ell\end{bmatrix}\right)$.

  end if

  Form $L^{W_\ell,V_\ell}(s) := \mathcal{A}^{W_\ell,V_\ell} - s\mathcal{B}^{W_\ell,V_\ell}$ as in (2.11).

  % **update the eigenvalue estimate**

  $\lambda_{\ell+1}, v_{\ell+1} \leftarrow$ the eigenvalue, eigenvector of $L^{W_\ell,V_\ell}(\cdot)$ closest to τ.

  Return $\lambda \leftarrow \lambda_{\ell+1}$, $v \leftarrow v_{\ell+1}(\mathsf{r} + \mathsf{n}(\mathsf{d}-1) + 1 : \mathsf{r} + \mathsf{nd})$, where $\mathsf{r} := \dim \operatorname{Ran}(V_\ell)$ if convergence has occurred.

end for

the singular values as a function of s. Finally, Section 2.4 deduces the main quadratic convergence result by exploiting these interpolatory and analytical properties.

2.2. Interpolation of Singular Values. Algorithm 2.1 is specifically tailored to satisfy the interpolation properties

$$R(\lambda_k) = R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k) \quad \text{and} \quad R^{(j)}(\lambda_k) = \left[R^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]^{(j)}(\lambda_k) \tag{2.12}$$

for $j = 1, \ldots, 2\mathsf{q}-1$ and $k = 1, \ldots, \ell$. It is a simple exercise to extend these interpolation properties to the singular values of $R(\lambda_k)$ and $R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k)$ for $k = 1, \ldots, \ell$.

Formally, let us consider the eigenvalues of the matrices

$$M(s) := R(s)^*R(s) \quad \text{and} \quad M^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s) := R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)^*R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$$

as functions of s which we denote with $\eta_1(s), \ldots, \eta_{\mathsf{n}}(s)$ and $\eta_1^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s), \ldots, \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$ and which are sorted in descending order. These eigenvalues correspond to the squared singular values of the matrices $R(s)$ and $R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$, respectively. By the definitions of $M(s)$ and $M^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$ and exploiting the interpolation properties (2.12), next we deduce the desired interpolation result concerning the singular values. Throughout the rest of this section, we employ the notations

$$\eta_j'(s) := \left[\frac{\partial\eta_j(s)}{\partial \operatorname{Re}(s)}, \quad \frac{\partial\eta_j(s)}{\partial \operatorname{Im}(s)}\right] \quad \text{and} \quad \left[\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]'(s) := \left[\frac{\partial\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Re}(s)}, \quad \frac{\partial\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Im}(s)}\right],$$

as well as

$$\nabla^2\eta_j(s) := \begin{bmatrix} \frac{\partial^2\eta_j(s)}{\partial \operatorname{Re}(s)^2} & \frac{\partial^2\eta_j(s)}{\partial \operatorname{Re}(s)\partial \operatorname{Im}(s)} \\ \frac{\partial^2\eta_j(s)}{\partial \operatorname{Im}(s)\partial \operatorname{Re}(s)} & \frac{\partial^2\eta_j(s)}{\partial \operatorname{Im}(s)^2} \end{bmatrix} \quad \text{and}$$

$$\nabla^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s) := \begin{bmatrix} \frac{\partial^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Re}(s)^2} & \frac{\partial^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Re}(s)\partial \operatorname{Im}(s)} \\ \frac{\partial^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Im}(s)\partial \operatorname{Re}(s)} & \frac{\partial^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial \operatorname{Im}(s)^2} \end{bmatrix}$$

for $j = 1, \ldots, \mathsf{n}$.

THEOREM 2.2 (Hermite interpolation). *Regarding Algorithm 2.1 with $\mathsf{q} \geq 2$, the following assertions hold for $k = 1, \ldots, \ell$ and $j = 1, \ldots \mathsf{n}$:*

(i) *It holds that $\eta_j(\lambda_k) = \eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k)$.*

(ii) *If $\eta_j(\lambda_k)$ is simple, then also $\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k)$ is simple. In this case,*

$$\eta_j'(\lambda_k) = \left[\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]'(\lambda_k) \quad \text{and} \quad \nabla^2\eta_j(\lambda_k) = \nabla^2\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k).$$

Proof. **(i)** The assertion immediately follows from (2.12), since $M(\lambda_k) = M^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k)$.

(ii) In Algorithm 2.1, it is required that $\mathsf{q} \geq 2$. Hence, the assertions follow from (2.12), in particular from

$$\begin{aligned} M(\lambda_k) &= M^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_k), \\ M'(\lambda_k) &= \left[M^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]'(\lambda_k), \\ M''(\lambda_k) &= \left[M^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]''(\lambda_k), \end{aligned}$$

by using the analytical formulas for the first and second derivatives of eigenvalue functions of a Hermitian matrix dependent on a real parameter [16]. ∎

The requirement that $\mathsf{q} \geq 2$ appears to be essential for quadratic convergence of the subspace framework. The arguments in the rest of this section establishing quadratic convergence does not apply for $\mathsf{q} = 1$. In practice, we observe slower convergence that is faster than linear convergence with $\mathsf{q} = 1$.

2.3. Analytical Properties of Singular Values. At an eigenvalue λ_* of $R(\cdot)$, we must have $\eta_{\mathsf{n}}(\lambda_*) = 0$. Additionally, throughout the rest of this section, the eigenvalue λ_* under consideration is assumed to be simple, i. e., $\eta_1(\lambda_*), \ldots, \eta_{\mathsf{n}-1}(\lambda_*) > 0$. There are appealing smoothness properties intrinsic to $\eta_1(\cdot), \ldots, \eta_{\mathsf{n}}(\cdot)$ as well as $\eta_1^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot), \ldots, \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ in a neighborhood of an eigenvalue λ_* of $R(\cdot)$, as long as the following assumption holds.

ASSUMPTION 2.3 (Non-defectiveness). *Let λ_* be a simple eigenvalue of $R(\cdot)$ such that, for a given $\xi > 0$, we have*

$$\sigma_{\min}(A - \lambda_* I_{\mathsf{k}}) \geq \xi \quad \text{and} \quad \sigma_{\min}(W_\ell^* A V_\ell - \lambda_* W_\ell^* V_\ell) \geq \xi, \tag{2.13}$$

where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of its matrix argument.

An implication of the assumption above, combined with the Lipschitz continuity of the singular value functions, is the boundedness of the smallest singular values in (2.13) away from zero in a vicinity of λ_*. Formally, there exists a neighborhood

$\mathcal{N}(\lambda_*)$ of λ_* – independent of the choice of the subspaces $\mathcal{V}_\ell$ and $\mathcal{W}_\ell$ as long as (2.13) is satisfied – such that

$$\sigma_{\min}(A - sI_{\mathsf{k}}) \geq \xi/2 \quad \text{and} \quad \sigma_{\min}(W_\ell^* A V_\ell - s W_\ell^* V_\ell) \geq \xi/2 \quad \forall s \in \mathcal{N}(\lambda_*), \tag{2.14}$$

see the beginning of the proof of Lemma A.1 in [2].

The matrix-valued functions $M(\cdot)$ and $M^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ are analytic in $\mathcal{N}(\lambda_*)$, which implies the following smoothness properties that we employ in the next section to analyze the rate of convergence. The proofs of the first three parts of the result below are straightforward adaptations of those for Lemma A.1 and Lemma A.2 in [2]. The proof of the fourth part is immediate from the second and third part. In the result and elsewhere, $\|\cdot\|_2$ denotes the vector or matrix 2-norm. Moreover, we make use of the notation $\mathcal{B}(\lambda_*, \delta)$ for the open ball centered at λ_* with radius $\delta > 0$, that is

$$\mathcal{B}(\lambda_*, \delta) := \{z \in \mathbb{C} \mid |z - \lambda_*| < \delta\},$$

whereas $\overline{\mathcal{B}}(\lambda_*, \delta)$ denotes the closure of $\mathcal{B}(\lambda_*, \delta)$, that is the closed ball centered at λ_* with radius $\delta > 0$, i. e.,

$$\overline{\mathcal{B}}(\lambda_*, \delta) := \{z \in \mathbb{C} \mid |z - \lambda_*| \leq \delta\}.$$

By a constant here and in the subsequent arguments in this section, we mean that the scalar does not depend on λ_j for $j = 1, \ldots, \ell$ as well as the subspaces $\mathcal{W}_\ell$, $\mathcal{V}_\ell$. Rather, it can be expressed fully in terms of the quantities related to the original rational function $R(\cdot)$.

LEMMA 2.4. *Suppose that Assumption 2.3 holds, and λ_ℓ is sufficiently close to the eigenvalue λ_* of $R(\cdot)$. There exist constants γ, $\delta > 0$ such that $\overline{\mathcal{B}}(\lambda_*, \delta) \subseteq \mathcal{N}(\lambda_*)$ satisfying the following assertions:*

- **(i)** *We have $|\eta_j(s) - \eta_j(\widehat{s})| \leq \gamma|s - \widehat{s}|$ and $\left|\eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s) - \eta_j^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\widehat{s})\right| \leq \gamma|s - \widehat{s}|$ for all s, $\widehat{s} \in \overline{\mathcal{B}}(\lambda_*, \delta)$ and for $j = 1, \ldots, \mathsf{n}$.*
- **(ii)** *The eigenvalues $\eta_{\mathsf{n}}(s)$ and $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)$ are simple for all $s \in \overline{\mathcal{B}}(\lambda_*, \delta)$. Hence, the derivatives*

$$\frac{\partial \eta_{\mathsf{n}}(s)}{\partial s_1}, \frac{\partial^2 \eta_{\mathsf{n}}(s)}{\partial s_1 \partial s_2}, \frac{\partial^3 \eta_{\mathsf{n}}(s)}{\partial s_1 \partial s_2 \partial s_3} \quad \text{and} \quad \frac{\partial \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1}, \frac{\partial^2 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1 \partial s_2}, \frac{\partial^3 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1 \partial s_2 \partial s_3}$$

  exist for every s_1, s_2, $s_3 \in \{\mathrm{Re}(s), \mathrm{Im}(s)\}$ and for all $s \in \mathcal{B}(\lambda_*, \delta)$.
- **(iii)** *We have*

$$\left|\frac{\partial \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1}\right| \leq \gamma, \quad \left|\frac{\partial^2 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1 \partial s_2}\right| \leq \gamma, \quad \left|\frac{\partial^3 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)}{\partial s_1 \partial s_2 \partial s_3}\right| \leq \gamma$$

  for every s_1, s_2, $s_3 \in \{\mathrm{Re}(s), \mathrm{Im}(s)\}$ and for all $s \in \mathcal{B}(\lambda_*, \delta)$.
- **(iv)** *We have*

$$\|\eta_{\mathsf{n}}'(s) - \eta_{\mathsf{n}}'(\widehat{s})\|_2 \leq \gamma|s - \widehat{s}|, \quad \left\|\left[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]'(s) - \left[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\right]'(\widehat{s})\right\|_2 \leq \gamma|s - \widehat{s}|$$

  and

$$\left\|\nabla^2 \eta_{\mathsf{n}}(s) - \nabla^2 \eta_{\mathsf{n}}(\widehat{s})\right\|_2 \leq \gamma|s - \widehat{s}|, \quad \left\|\nabla^2 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s) - \nabla^2 \eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\widehat{s})\right\|_2 \leq \gamma|s - \widehat{s}|$$

  for all $s, \widehat{s} \in \mathcal{B}(\lambda_*, \delta)$.

2.4. Convergence Properties. In practice, we observe that Algorithm 2.1 nearly always converges to the eigenvalue of $R(\cdot)$ closest to the target point τ. Here, we consider two consecutive iterates λ_ℓ, $\lambda_{\ell+1}$ of this subspace method, which we assume close to an eigenvalue λ_* of $R(\cdot)$. Then we prove

$$|\lambda_{\ell+1} - \lambda_*| \leq C|\lambda_\ell - \lambda_*|^2 \tag{2.15}$$

for some constant $C > 0$. The closeness of λ_ℓ, $\lambda_{\ell+1}$ to λ_* is a silent assumption that is kept throughout, even though it is not explicitly stated. In addition, we deduce the bound (2.15) under Assumption 2.3, as well as the following assumption.

ASSUMPTION 2.5 (Non-degeneracy). *The Hessian $\nabla^2\eta_{\mathsf{n}}(\lambda_*)$ is invertible.*

The main quadratic convergence result relies on the non-singularity of the Hessian of $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ in a ball centered around λ_*. This is stated formally and proven next.

LEMMA 2.6 (Uniform non-singularity of the Hessian). *Suppose that Assumptions 2.3 and 2.5 hold. Then there exist constants α, $\delta > 0$ such that*

$$\sigma_{\min}(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)) \geq \alpha \quad \forall s \in \mathcal{B}(\lambda_*, \delta). \tag{2.16}$$

Proof. Let $\beta := \sigma_{\min}(\nabla^2\eta_{\mathsf{n}}(\lambda_*)) > 0$. By the Lipschitz continuity of $\nabla^2\eta_{\mathsf{n}}(\cdot)$ around λ_* (which follows from part **(iv)** of Lemma 2.4), there exists a $\widehat{\delta} > 0$ such that $\sigma_{\min}(\nabla^2\eta_{\mathsf{n}}(s)) \geq \beta/2$ for all $s \in \mathcal{B}\big(\lambda_*, \widehat{\delta}\big)$. Without loss of generality, we may also assume that $\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ is Lipschitz continuous in $\mathcal{B}\big(\lambda_*, \widehat{\delta}\big)$ with the Lipschitz constant γ (once again due to part **(iv)** of Lemma 2.4).

Setting $\delta := \min\big\{\beta/(8\gamma), \widehat{\delta}\big\}$, we additionally assume, without loss of generality, that $\lambda_\ell \in \mathcal{B}(\lambda_*, \delta)$. But then the Hermite interpolation property, specifically part **(ii)** of Theorem 2.2, implies

$$\sigma_{\min}\big(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)\big) = \sigma_{\min}(\nabla^2\eta_{\mathsf{n}}(\lambda_\ell)) \geq \beta/2.$$

Moreover,

$$\begin{aligned}\big|\sigma_{\min}\big(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)\big) - \sigma_{\min}\big(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)\big)\big| &\leq \big\|\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell) - \nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)\big\|_2 \\ &\leq \gamma|\lambda_\ell - s| \leq \beta/4\end{aligned}$$

for all $s \in \overline{\mathcal{B}}(\lambda_*, \delta)$, where the first inequality follows from Weyl's theorem [15, Theorem 4.3.1], whereas the second inequality is due to the Lipschitz continuity of $\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$. Hence, we deduce $\sigma_{\min}\big(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(s)\big) \geq \beta/4 =: \alpha$ for all $s \in \mathcal{B}(\lambda_*, \delta)$ as desired. ∎

Now we are ready to present the main quadratic convergence result, where the notation $\mathcal{R}^2 : \mathbb{C} \to \mathbb{R}^2$ refers to the linear map defined by $\mathcal{R}^2(z) := \big[\mathrm{Re}(z),\ \mathrm{Im}(z)\big]^\top$.

THEOREM 2.7 (Quadratic convergence). *Suppose that Assumptions 2.3 and 2.5 hold. Then for the iterates of Algorithm 2.1 with $\mathsf{q} \geq 2$, there exists a constant $C > 0$ such that (2.15) is satisfied.*

Proof. Let δ be such that the assertions of Lemmas 2.4 and 2.6 hold in the ball $\mathcal{B}(\lambda_*, \delta)$. In particular, the eigenvalues $\eta_{\mathsf{n}}(\cdot)$ and $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ are simple, $\nabla^2\eta_{\mathsf{n}}(\cdot)$ and $\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ are Lipschitz continuous with Lipschitz constant $\gamma > 0$, and the lower bound (2.16) is satisfied for some constant $\alpha > 0$ in $\mathcal{B}(\lambda_*, \delta)$. Without loss of generality, assume that λ_ℓ, $\lambda_{\ell+1} \in \mathcal{B}(\lambda_*, \delta)$.

The iterate $\lambda_{\ell+1}$, by definition, is an eigenvalue of $R^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$, hence we have $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_{\ell+1}) = 0$. Indeed, $\lambda_{\ell+1}$ is a smooth global minimizer of $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ (i. e., the smoothness follows from part **(ii)** of Lemma 2.4), implying also $\big[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\big]'(\lambda_{\ell+1}) = 0$.

By employing the Lipschitz continuity of $\nabla^2\eta_{\mathsf{n}}(\cdot)$ in $\mathcal{B}(\lambda_*,\delta)$, we have

$$0 = \eta_{\mathsf{n}}'(\lambda_*) = \eta_{\mathsf{n}}'(\lambda_\ell) + \int_0^1 \mathcal{R}^2(\lambda_* - \lambda_\ell)\nabla^2\eta_{\mathsf{n}}(\lambda_\ell + t(\lambda_* - \lambda_\ell))\,\mathrm{d}t,$$

which, by exploiting $\nabla^2\eta_{\mathsf{n}}(\lambda_\ell) = \nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)$ (see part **(ii)** of Theorem 2.2), could be arranged to

$$\begin{aligned}0 = \eta_{\mathsf{n}}'(\lambda_\ell) + \mathcal{R}^2(\lambda_* - \lambda_\ell)\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell) \\ + \int_0^1 \mathcal{R}^2(\lambda_* - \lambda_\ell)\left(\nabla^2\eta_{\mathsf{n}}(\lambda_\ell + t(\lambda_* - \lambda_\ell)) - \nabla^2\eta_{\mathsf{n}}(\lambda_\ell)\right)\,\mathrm{d}t.\end{aligned} \tag{2.17}$$

Moreover, by a Taylor expansion of $\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\cdot)$ about λ_ℓ, we obtain

$$0 = \big[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\big]'(\lambda_{\ell+1}) = \big[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\big]'(\lambda_\ell) + \mathcal{R}^2(\lambda_{\ell+1} - \lambda_\ell)\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell) + \mathcal{O}\big(|\lambda_{\ell+1} - \lambda_\ell|^2\big),$$

which, combined with $\big[\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}\big]'(\lambda_\ell) = \eta_{\mathsf{n}}'(\lambda_\ell)$ (again due to part **(ii)** of Theorem 2.2), give rise to

$$\begin{aligned}\eta_{\mathsf{n}}'(\lambda_\ell) + \mathcal{R}^2(\lambda_* - \lambda_\ell)\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell) \\ = \mathcal{R}^2(\lambda_* - \lambda_{\ell+1})\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell) + \mathcal{O}\big(|\lambda_{\ell+1} - \lambda_\ell|^2\big).\end{aligned} \tag{2.18}$$

In (2.17), by plugging the right-hand side of (2.18), then exploiting the Lipschitz continuity of $\nabla^2\eta_{\mathsf{n}}(\cdot)$, and taking the norm, we deduce

$$\big\|\mathcal{R}^2(\lambda_* - \lambda_{\ell+1})\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)\big\|_2 \le \frac{\gamma}{2}|\lambda_\ell - \lambda_*|^2 + \mathcal{O}\big(|\lambda_{\ell+1} - \lambda_\ell|^2\big), \tag{2.19}$$

where γ is the Lipschitz constant for $\nabla^2\eta_{\mathsf{n}}(\cdot)$. Finally, by employing

$$\sigma_{\min}\big(\nabla^2\eta_{\mathsf{n}}^{\mathcal{W}_\ell,\mathcal{V}_\ell}(\lambda_\ell)\big) \ge \alpha$$

in (2.19) (which is implied by Lemma 2.6) for $\alpha > 0$, we obtain

$$\alpha|\lambda_{\ell+1} - \lambda_*| \le \frac{\gamma}{2}|\lambda_\ell - \lambda_*|^2 + \mathcal{O}\big(|\lambda_{\ell+1} - \lambda_\ell|^2\big).$$

The desired inequality (2.15) is now immediate from $|\lambda_{\ell+1} - \lambda_\ell|^2 \le 2\left(|\lambda_{\ell+1} - \lambda_*|^2 + |\lambda_\ell - \lambda_*|^2\right)$. ∎

3. General Nonlinear Eigenvalue Problem Setting. Inspired by the ideas of the previous section for rational eigenvalue problems, we present a subspace framework for the more general setting of a nonlinear eigenvalue problem of the form (1.2). Let us consider $T(\cdot)$ and T_j for $j = 1, \ldots, \kappa$ in (1.2) and (1.1) in the partitioned forms

$$T(s) = \begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix} \quad \text{and} \quad T_j = \begin{bmatrix} A_j & B_j \\ C_j & D_j \end{bmatrix}, \tag{3.1}$$

where $A(s)$, $A_j \in \mathbb{C}^{\mathsf{k}\times\mathsf{k}}$, $B(s)$, $B_j \in \mathbb{C}^{\mathsf{k}\times\mathsf{m}}$, $C(s)$, $C_j \in \mathbb{C}^{\mathsf{m}\times\mathsf{k}}$, $D(s)$, $D_j \in \mathbb{C}^{\mathsf{m}\times\mathsf{m}}$ for all $s \in \mathbb{C}$ such that $\mathsf{k} + \mathsf{m} = \mathsf{n}$ and $\mathsf{k} \gg \mathsf{m}$. It is a simple exercise to deduce that every

finite eigenvalue $\lambda \in \mathbb{C}$ of $T(\cdot)$ that is not an eigenvalue of $A(\cdot)$, is also an eigenvalue of the function

$$\mathcal{R}(s) := C(s)A(s)^{-1}B(s) - D(s).$$

Conversely, every finite eigenvalue of $\mathcal{R}(\cdot)$ is an eigenvalue of $T(\cdot)$.

Similar to the rational eigenvalue problem setting, the large-scale nature of $\mathcal{R}(\cdot)$ is hidden in the middle factor $A(\cdot)$. Hence, we define the reduced matrix-valued function corresponding to $\mathcal{R}(\cdot)$ by

$$\mathcal{R}^{\mathcal{W},\mathcal{V}}(s) \;:=\; C^V(s)A^{W,V}(s)^{-1}B^W(s) - D(s)$$

in terms of two subspaces $\mathcal{W}$, $\mathcal{V} \subseteq \mathbb{C}^{\mathsf{k}}$ of equal dimension, say $\mathsf{r} \ll \mathsf{k}$, and matrices W, V whose columns form orthonormal bases for them, where

$$\begin{aligned} A^{W,V}(s) &:= W^*A(s)V = f_1(s)(W^*A_1V) + \cdots + f_\kappa(s)(W^*A_\kappa V), \\ B^W(s) &:= W^*B(s) = f_1(s)(W^*B_1) + \cdots + f_\kappa(s)(W^*B_\kappa), \quad \text{and} \\ C^V(s) &:= C(s)V = f_1(s)(C_1V) + \cdots + f_\kappa(s)(C_\kappa V). \end{aligned} \tag{3.2}$$

The middle factor $A^{W,V}(\cdot)$ of the reduced matrix-valued function is of dimension $\mathsf{r} \times \mathsf{r}$ and much smaller than $A(\cdot)$.

Again, we benefit from the optimization point of view, that is we consider the minimization problem

$$\min_{\lambda\in\mathbb{C}} \sigma_{\min}(\mathcal{R}(\lambda)). \tag{3.3}$$

In particular, assuming that the spectra of $A(\cdot)$ and $T(\cdot)$ are disjoint, the eigenvalue of $T(\cdot)$ closest to a prescribed target $\tau \in \mathbb{C}$ is the global minimizer of the optimization problem above closest to τ. At every subspace iteration, instead of (3.3), we solve

$$\min_{\lambda\in\mathbb{C}} \sigma_{\min}\big(\mathcal{R}^{\mathcal{W},\mathcal{V}}(\lambda)\big). \tag{3.4}$$

Specifically, we determine the global minimizer of $\mathcal{R}^{\mathcal{W},\mathcal{V}}(\cdot)$ closest to the prescribed target τ. The eigenvalues of $\mathcal{R}^{\mathcal{W},\mathcal{V}}(\cdot)$ are the same as those of the function

$$T^{W,V}(s) := \begin{bmatrix} A^{W,V}(s) & B^W(s) \\ C^V(s) & D(s) \end{bmatrix} = \begin{bmatrix} W^* & 0 \\ 0 & I_{\mathsf{m}} \end{bmatrix} T(s) \begin{bmatrix} V & 0 \\ 0 & I_{\mathsf{m}} \end{bmatrix}, \tag{3.5}$$

except possibly those that are the eigenvalues of $A^{W,V}(\cdot)$. Hence, to retrieve the global minimizer $\widetilde{\lambda}$ of $\sigma_{\min}\big(\mathcal{R}^{\mathcal{W},\mathcal{V}}(\cdot)\big)$ closest to τ, we find an eigenvalue of $T^{W,V}(\cdot)$ closest to this target point.

We expand the subspaces $\mathcal{W}, \mathcal{V}$ into $\widetilde{\mathcal{W}}, \widetilde{\mathcal{V}}$ so that

$$\mathcal{R}\big(\widetilde{\lambda}\big) = \mathcal{R}^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\big(\widetilde{\lambda}\big) \quad \text{and} \quad \mathcal{R}^{(j)}\big(\widetilde{\lambda}\big) = \Big[\mathcal{R}^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\Big]^{(j)}\big(\widetilde{\lambda}\big) \tag{3.6}$$

hold for $j = 1, \ldots, \mathsf{q}$ and for a prescribed positive integer q. The following generalization of Lemma 2.1 indicates how this Hermite interpolation property can be attained. This result is also a corollary of [5, Theorem 1].

LEMMA 3.1. *Suppose that $\mu \in \mathbb{C}$ is not an eigenvalue of $A(\cdot)$. Let $\widetilde{\mathcal{W}} = \mathcal{W} \oplus \mathcal{W}_\mu$ and $\widetilde{\mathcal{V}} = \mathcal{V} \oplus \mathcal{V}_\mu$, where $\mathcal{V}, \mathcal{W}$ are given subspaces of equal dimension, and $\mathcal{W}_\mu$, $\mathcal{V}_\mu$ are subspaces defined as*

$$\mathcal{V}_\mu := \bigoplus_{j=0}^{\mathsf{q}-1} \operatorname{Ran}\left(\frac{\mathrm{d}^j}{\mathrm{d}s^j}(A(s)^{-1}B(s))\Big|_{s=\mu}\right), \; \mathcal{W}_\mu := \bigoplus_{j=0}^{\mathsf{q}-1} \operatorname{Ran}\left(\frac{\mathrm{d}^j}{\mathrm{d}s^j}\big(C(s)A(s)^{-1}\big)^*\Big|_{s=\mu}\right)$$

for some positive integer q. *Let* $\widetilde{V}$ *and* $\widetilde{W}$ *be basis matrices of* $\widetilde{\mathcal{V}}$ *and* $\widetilde{\mathcal{W}}$, *respectively and assume further that* $\widetilde{W}^*A(\mu)\widetilde{V}$ *is invertible. Then we have*

1. $\mathcal{R}(\mu) = \mathcal{R}^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}(\mu)$, *and*
2. $\mathcal{R}^{(j)}(\mu) = \left[\mathcal{R}^{\widetilde{\mathcal{W}},\widetilde{\mathcal{V}}}\right]^{(j)}(\mu)$ *for* $j = 1, \ldots, 2\mathsf{q}-1$.

Based on the discussions and the subspace expansion strategy above, we outline the subspace framework to locate the eigenvalue of $T(\cdot)$ closest to the target point $\tau \in \mathbb{C}$ in Algorithm 3.1. At iteration ℓ of the algorithm, first the subspaces $\mathcal{W}_{\ell-1}, \mathcal{V}_{\ell-1}$ are expanded to $\mathcal{W}_\ell, \mathcal{V}_\ell$ to achieve Hermite interpolation conditions at the current candidate λ_ℓ for the eigenvalue in lines 4–13. Then the next candidate $\lambda_{\ell+1}$ is retrieved by computing the eigenvalue of $T^{W_\ell,V_\ell}(\cdot)$ closest to the target point. The termination condition employed at the end in line 16 is specified in Section 6 in a way that also sheds light into the choice of the eigenvector estimate v returned in line 16.

The quick convergence result of Theorem 2.7 extends to Algorithm 3.1 in a straightforward fashion. Two consecutive iterates λ_ℓ, $\lambda_{\ell+1}$ of Algorithm 3.1 satisfy

$$|\lambda_{\ell+1} - \lambda_*| \leq C|\lambda_\ell - \lambda_*|^2$$

for some constant $C > 0$, provided λ_ℓ, $\lambda_{\ell+1}$ are sufficiently close to an eigenvalue λ_* and under non-defectiveness and non-degeneracy assumptions analogous to Assumptions 2.3 and 2.5.

4. Computing Multiple Eigenvalues. The proposed subspace frameworks, Algorithms 2.1 and 3.1 for rational eigenvalue problems and general nonlinear eigenvalue problems, are meant to estimate only one eigenvalue closest to the prescribed target τ. However, they have natural extensions to compute k eigenvalues closest to the target for a prescribed integer $k \geq 2$. These extensions are based on extracting multiple eigenvalues of the projected problems, and expanding the projection spaces so as to ensure Hermite interpolation at some of these eigenvalues.

Before proposing three alternatives for the interpolation points, let us remark a subtle issue. There is the possibility that some of the eigenvalues of the projected problems $L^{W,V}(\cdot)$ and $T^{W,V}(\cdot)$ are indeed also the eigenvalues of their top-left blocks $L_A^{W,V}(s) := W^*AV - sW^*V$ and $A^{W,V}(\cdot)$. Even though this situation seems unlikely, we observe in practice that it sometimes occurs when multiple eigenvalues of the projected problems are extracted. We do not take such eigenvalues of $L^{W,V}(\cdot)$ and $T^{W,V}(\cdot)$ into consideration; such eigenvalues may correspond to the poles of $R^{\mathcal{W},\mathcal{V}}(\cdot)$ and $\mathcal{R}^{\mathcal{W},\mathcal{V}}(\cdot)$ rather than the eigenvalues of $R^{\mathcal{W},\mathcal{V}}(\cdot)$ and $\mathcal{R}^{\mathcal{W},\mathcal{V}}(\cdot)$.

To summarize, in lines 14 and 15 of Algorithms 2.1 and 3.1, we choose the interpolation points for the next iteration from the set Λ^{W_ℓ,V_ℓ} consisting of all (finite) eigenvalues of $L^{W_\ell,V_\ell}(\cdot)$ and $T^{W_\ell,V_\ell}(\cdot)$ that are not eigenvalues of $L_A^{W_\ell,V_\ell}(\cdot)$ and $A^{W_\ell,V_\ell}(\cdot)$. Specifically, we employ one of the following three viable strategies for the selection of the interpolation points at the next iteration among $\lambda_{\ell+1}^{(1)}, \ldots, \lambda_{\ell+1}^{(k)}$, the k closest to the target point τ in Λ^{W_ℓ,V_ℓ}:

ALL: **Interpolate at up to all of the k closest eigenvalues:** Hermite interpolation is performed at the next iteration at each $\lambda_{\ell+1}^{(j)}$ unless the corresponding residual is below the convergence threshold for $j = 1, \ldots, k$; see (6.1) below for the specification of the residual corresponding to $\lambda_{\ell+1}^{(j)}$.

BR: **Interpolate at the eigenvalue among the k closest with the best residual:** Among the points $\lambda_{\ell+1}^{(1)}, \ldots, \lambda_{\ell+1}^{(k)}$ with residuals greater than the convergence threshold, we choose only the one with the smallest residual

Algorithm 3.1 Subspace method to compute a nonlinear eigenvalue closest to a prescribed target

Input: matrices $T_1, \ldots, T_\kappa \in \mathbb{C}^{\mathsf{n}\times\mathsf{n}}$, meromorphic functions $f_1, \ldots, f_\kappa : \mathbb{C} \to \mathbb{C}$ as in (1.1), partition parameter $\mathsf{m} \in \mathbb{Z}^+$, interpolation parameter $\mathsf{q} \in \mathbb{Z}$ with $\mathsf{q} \geq 2$, target $\tau \in \mathbb{C}$.

Output: estimate $\lambda \in \mathbb{C}$ for the eigenvalue closest to τ and corresponding eigenvector estimate $v \in \mathbb{C}^{\mathsf{n}} \setminus \{0\}$.

Partition $T(s)$ as in (3.1) so that $A(s) \in \mathbb{C}^{\mathsf{k}\times\mathsf{k}}$, $B(s) \in \mathbb{C}^{\mathsf{k}\times\mathsf{m}}$, $C(s) \in \mathbb{C}^{\mathsf{m}\times\mathsf{k}}$, $D(s) \in \mathbb{C}^{\mathsf{m}\times\mathsf{m}}$ for all $s \in \mathbb{C}$, where $\mathsf{k} := \mathsf{n} - \mathsf{m}$.
$\lambda_0 \leftarrow \tau$.
% **main loop**
for $\ell = 0, 1, 2, \ldots$ **do**
  $\widetilde{V}_\ell \leftarrow A(\lambda_\ell)^{-1}B(\lambda_\ell)$ and $\widetilde{W}_\ell \leftarrow A(\lambda_\ell)^{-*}C(\lambda_\ell)^*$.
  for $j = 1, \ldots, \mathsf{q}-1$ **do**
    $\widehat{V}_\ell \leftarrow \frac{\mathrm{d}^j}{\mathrm{d}s^j}(A(s)^{-1}B(s))\big|_{s=\lambda_\ell}$ and $\widetilde{V}_\ell \leftarrow \begin{bmatrix}\widetilde{V}_\ell & \widehat{V}_\ell\end{bmatrix}$.
    $\widehat{W}_\ell \leftarrow \frac{\mathrm{d}^j}{\mathrm{d}s^j}(A(s)^{-*}C(s)^*)\big|_{s=\lambda_\ell}$ and $\widetilde{W}_\ell \leftarrow \begin{bmatrix}\widetilde{W}_\ell & \widehat{W}_\ell\end{bmatrix}$.
  end for
  % **expand the subspaces to interpolate at** λ_ℓ
  if $\ell = 0$ **then**
    $V_0 \leftarrow \mathrm{orth}\left(\widetilde{V}_0\right)$ and $W_0 \leftarrow \mathrm{orth}\left(\widetilde{W}_0\right)$.
  else
    $V_\ell \leftarrow \mathrm{orth}\left(\begin{bmatrix}V_{\ell-1} & \widetilde{V}_\ell\end{bmatrix}\right)$ and $W_\ell \leftarrow \mathrm{orth}\left(\begin{bmatrix}W_{\ell-1} & \widetilde{W}_\ell\end{bmatrix}\right)$.
  end if
  Form $T^{W_\ell,V_\ell}(s) := \begin{bmatrix}A^{W_\ell,V_\ell}(s) & B^{W_\ell}(s)\\ C^{V_\ell}(s) & D(s)\end{bmatrix}$, where $A^{W_\ell,V_\ell}(\cdot)$, $B^{W_\ell}(\cdot)$, $C^{V_\ell}(\cdot)$ are defined as in (3.2).
  % **update the eigenvalue estimate**
  $\lambda_{\ell+1}, v_{\ell+1} \leftarrow$ the eigenvalue, eigenvector of $T^{W_\ell,V_\ell}(\cdot)$ closest to τ.
  Return $\lambda \leftarrow \lambda_{\ell+1}$ and $v \leftarrow \begin{bmatrix}V_\ell & 0\\ 0 & I_{\mathsf{m}}\end{bmatrix} v_{\ell+1}$ if convergence has occurred.
end for

for Hermite interpolation at the next iteration.

WR: **Interpolate at the eigenvalue among the k closest with the worst residual:** We perform Hermite interpolation at only one of $\lambda_{\ell+1}^{(1)}, \ldots, \lambda_{\ell+1}^{(k)}$, whichever has the largest residual.

5. One-Sided Variations. Variants of the subspace methods introduced for the rational eigenvalue problems and general nonlinear eigenvalue problems are obtained by forming V_ℓ as suggested in the proposed frameworks, but setting $W_\ell = V_\ell$. Interpolation results in Lemmas 2.1 and 3.1 hold even with $\mathcal{V}_\mu$ as stated in those lemmas and $\mathcal{W}_\mu = \mathcal{V}_\mu$, but with the equality of the derivatives holding in the second parts up to the $(\mathsf{q}-1)$st derivative (rather than the $(2\mathsf{q}-1)$st derivative). In particular, provided $\mathsf{q} \geq 3$, the interpolation properties between the full and reduced matrix-valued functions, as well as their first two derivatives are attained at μ. This paves the way for an analysis analogous to that in Sections 2.3 and 2.4, and leads to an at least quadratic convergence result for the sequences of eigenvalue estimates.

In our experience, these one-sided variants sometimes tend to be quicker. As an example, for a proper rational eigenvalue problem $R_{\mathsf{p}} = C(sI_{\mathsf{k}} - A)^{-1}B$ with $\mathsf{q} = 3$, one-sided variants expand the projection subspace with the directions

$$\begin{bmatrix}(A - \lambda_\ell I_{\mathsf{k}})^{-1}B & (A - \lambda_\ell I_{\mathsf{k}})^{-2}B & (A - \lambda_\ell I_{\mathsf{k}})^{-3}B\end{bmatrix}$$

for interpolation at λ_ℓ, while the original two sided subspace method to achieve quadratic convergence needs to expand the left and right subspaces with the directions

$$\begin{bmatrix}(A - \lambda_\ell I_{\mathsf{k}})^{-1}B & (A - \lambda_\ell I_{\mathsf{k}})^{-2}B\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}((A - \lambda_\ell I_{\mathsf{k}})^*)^{-1}C^* & ((A - \lambda_\ell I_{\mathsf{k}})^*)^{-2}C^*\end{bmatrix},$$

respectively. Both of these expansion tasks require one LU decomposition, but the latter requires additional back and forward substitutions. Two-sided method also needs to orthogonalize both of the projection matrices at every subspace iteration, as opposed to orthogonalization of only one projection matrix for the one-sided variant. Yet, it appears that the two-sided subspace method is usually more reliable and numerically more stable in practice.

In what follows, we refer to one-sided variations of ALL, BR, WR as ALL1, BR1, WR1, respectively.

6. Numerical Results. In this section, we apply the proposed subspace frameworks to several large-scale nonlinear eigenvalue problems. Our implementation and numerical experiments are performed in Matlab R2020b on an iMac with Mac OS 11.3.1 operating system, Intel® Core™ i5-9600K CPU and 32GB RAM.

In the subsequent three subsections, we present numerical results on proper rational eigenvalue problems given in the transfer function form, polynomial eigenvalue problems, and three other nonlinear eigenvalue problems that are neither polynomial nor rational. In these subsections, when reporting the runtime, number of iterations, number of LU decompositions for a problem, we always run the algorithm five times, and present the average over the five runs. Before presenting the numerical results, we spell out the important implementation details below. For the rest, recall that k is the prescribed number of eigenvalues sought closest to the target point.

Termination. The algorithms are terminated when the norms of the relative residuals of the Ritz pairs associated with the k closest eigenvalues of the projected problems are less than a prescribed tolerance `tol`. Formally, letting $\lambda_\ell^{(j)}$ and $v_\ell^{(j)}$ denote the jth closest eigenvalue of $T^{W_\ell,V_\ell}(\cdot)$ to τ and corresponding eigenvector for $j = 1, \ldots, k$, we terminate if

$$\text{Rs}\left(\lambda_\ell^{(j)}, v_\ell^{(j)}\right) := \frac{\left\| T\left(\lambda_\ell^{(j)}\right) \begin{bmatrix} V_\ell & 0 \\ 0 & I_{\mathsf{m}} \end{bmatrix} v_\ell^{(j)} \right\|_\infty \Big/ \left\| v_\ell^{(j)} \right\|_\infty}{\left|f_1\left(\lambda_\ell^{(j)}\right)\right| \|T_1\|_\infty + \cdots + \left|f_\kappa\left(\lambda_\ell^{(j)}\right)\right| \|T_\kappa\|_\infty} < \texttt{tol} \tag{6.1}$$

for $j = 1, \ldots, k$ for the general nonlinear eigenvalue problem setting of (1.2).

A similar termination condition is adopted for the proper rational eigenvalue problems in the transfer function form, i.e., $R(s) = R_{\mathsf{p}}(s) := C(sI_{\mathsf{k}} - A)^{-1}Bv$. To be precise, if $\lambda_\ell^{(j)}$ and $v_\ell^{(j)}$ denote the the jth closest eigenvalue of $L^{W_\ell,V_\ell}(\cdot)$ to τ and a corresponding eigenvector for $j = 1, \ldots, k$, we terminate when

$$\text{Rs}_{\mathsf{r}}\left(\lambda_\ell^{(j)}, v_\ell^{(j)}\right) := \frac{\left\| L\left(\lambda_\ell^{(j)}\right) \begin{bmatrix} V_\ell & 0 \\ 0 & I_{\mathsf{nd}} \end{bmatrix} v_\ell^{(j)} \right\|_\infty \Big/ \left\| v_\ell^{(j)} \right\|_\infty}{\left|\lambda_\ell^{(j)}\right| + \left\| \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \right\|_\infty} < \texttt{tol} \tag{6.2}$$

for $j = 1, \ldots, k$.

Initial Subspaces. We require the initial projected matrices $A^{W_0,V_0}(\cdot)$ and $W_0^* A V_0$ (in the general nonlinear eigenvalue problem setting and in the proper rational eigenvalue problem setting, respectively) to be of size $k \times k$ at least. To make sure this is the case, we form the initial projected problem by interpolating the full problem at the target point τ, as well as at randomly selected points close to the target point unless otherwise specified.

Orthogonalization of the Bases for the Subspaces. The orthogonalization of the bases for the expansion subspaces (i.e., the columns of $\widetilde{V}_\ell, \widetilde{W}_\ell$ in lines 11 and 12 of Algorithms 2.1 and 3.1) with respect to the existing projection subspaces (spanned by the columns of $V_{\ell-1}, W_{\ell-1}$) via

$$\widetilde{V}_\ell - V_{\ell-1}\big(V_{\ell-1}^* \widetilde{V}_\ell\big), \qquad \widetilde{W}_\ell - W_{\ell-1}\big(W_{\ell-1}^* \widetilde{W}_\ell\big)$$

are performed several times in practice. This reorthogonalization strategy seems to improve the stability of the subspace frameworks, especially close to convergence. In particular, the column spaces of $(A - sI_{\mathsf{k}})^{-1}B$ in the case of Algorithm 2.1 and $A(s)^{-1}B$ in the case of Algorithm 3.1 at $s = \lambda_\ell$ and $s = \lambda_{\ell+1}$ are close to each other when $\lambda_\ell \approx \lambda_{\ell-1}$. Similar remarks also hold for the derivatives of $(A - sI_{\mathsf{k}})^{-1}B$ and $A(s)^{-1}B$. Hence, the columns of $[\, V_{\ell-1} \;\; \widetilde{V}_\ell \,]$ are nearly linearly dependent, and these projection matrices are ill-conditioned. Analogously, the matrix $[\, W_{\ell-1} \;\; \widetilde{W}_\ell \,]$ for left projections becomes ill-conditioned when λ_ℓ is close to convergence. In our experience, the reorthogonalization strategy, in the presence of rounding errors, appears to yield well-conditioned projection matrices with nearly orthonormal columns.

6.1. Proper Rational Eigenvalue Problems.

6.1.1. A Banded Example. We first employ the variants of Algorithm 2.1 to locate several eigenvalues of a proper rational matrix-valued function $R_{\mathsf{p}}(s) = C(sI_{\mathsf{k}} - A)^{-1}B$ closest to given target points, where $A \in \mathbb{R}^{10^5 \times 10^5}$ is a sparse banded random matrix with bandwidth five, and $B \in \mathbb{R}^{10^5 \times 2}$, $C \in \mathbb{R}^{2 \times 10^5}$ are also random matrices[1]. We perform experiments with two target points, namely $\tau_1 = -2 + \mathrm{i}$ and $\tau_2 = 3 - 7\mathrm{i}$; among these two, τ_1 is close to the eigenvalues of $R_{\mathsf{p}}(\cdot)$, while τ_2 is away from the eigenvalues of $R_{\mathsf{p}}(\cdot)$.

Parameters. We terminate when condition (6.2) is met for $\texttt{tol} = 10^{-12}$. The interpolation parameter that determines how many derivatives will be interpolated is chosen as $\mathsf{q} = 2$ and $\mathsf{q} = 5$ for the two-sided and one-sided variants, respectively.

Estimation of One Eigenvalue. The iterates λ_ℓ of Algorithm 2.1 and the corresponding relative residuals to compute the eigenvalues closest to τ_1 and τ_2 are listed in Table 6.1(a) and Table 6.1(b), respectively. Note that the stopping criterion is met after 2 and 7 subspace iterations for τ_1 and τ_2, respectively, and only the last two estimates for τ_2 by Algorithm 2.1 and the corresponding residuals are given in Table 6.1(b). The results reported in this table are consistent with the quadratic convergence assertion of Theorem 2.7. A comparison of the runtimes and accuracy of the computed results of Algorithm 2.1 and its one-sided variant with `eigs` is provided in Table 6.2.

[1]The precise matrices A, B, C for this proper rational eigenvalue problem, as well as for the two random examples in Section 6.1.2, are publicly available at https://zenodo.org/record/5811971.

TABLE 6.1
The iterates and corresponding residuals of Algorithm 2.1 to locate the eigenvalue of the proper rational matrix-valued function in Section 6.1.1 closest to the target point τ_1 and target point τ_2.

(a) $\tau_1 = -2 + \mathrm{i}$.

ℓ	λ_ℓ	$\mathrm{Rs}_{\mathsf{r}}(\lambda_\ell, v_\ell)$
1	$-2.000383015796 + 0.976443733019\mathrm{i}$	$2.866 \cdot 10^{-7}$
2	$-2.000388770264 + 0.976430802383\mathrm{i}$	$2.648 \cdot 10^{-16}$

(b) $\tau_2 = 3 - 7\mathrm{i}$.

ℓ	λ_ℓ	$\mathrm{Rs}_{\mathsf{r}}(\lambda_\ell, v_\ell)$
6	$1.749889546050 - 3.144267907227\mathrm{i}$	$1.330 \cdot 10^{-9}$
7	$1.749889549609 - 3.144267920104\mathrm{i}$	$3.263 \cdot 10^{-19}$

TABLE 6.2
Runtimes in seconds of the subspace methods to compute the eigenvalue of the proper rational matrix-valued function in Section 6.1.1 closest to $\tau_1 = -2 + \mathrm{i}$ and $\tau_2 = 3 - 7\mathrm{i}$ compared with `eigs`*. The differences in the computed results with* `eigs` *in absolute value are also reported.*

	time in s		difference with `eigs`	
method	τ_1	τ_2	τ_1	τ_2
two-sided	0.55	1.90	4×10^{-14}	3×10^{-14}
one-sided	0.35	0.80	4×10^{-9}	9×10^{-14}
`eigs`	4.20	5.85	–	–

Decay in the Residuals when Estimating Multiple Eigenvalues. Next we estimate the five eigenvalues closest to the target point τ_1 by employing the variants ALL, BR, WR of Algorithm 2.1. All of these variants, as well as `eigs` return exactly the same five closest eigenvalues up to twelve decimal digits. In order to compare and illustrate the progresses of ALL, WR, BR, we present the relative residuals with respect to the number of iterations (until all five eigenvalues converge up to the prescribed tolerance) in Table 6.3. The eigenvalue estimates corresponding to the residuals typed in blue italic letters are selected as interpolation points at the next subspace iteration. For all three variants, typically, the relative residual of an eigenvalue estimate that is selected as an interpolation point decreases dramatically in the next iteration.

6.1.2. Comparison of Runtimes on Several Examples. We also test the subspace methods on **(i)** two randomly generated sparse examples that are not banded, which we refer as `R1` and `R2`, as well as **(ii)** the `Eady` example from the SLICOT benchmark collection[2] for model reduction. The non-banded matrix A in `R1` and `R2` is of size 5000 and 10000 with about 0.1% and 0.05% nonzero entries, respectively, whereas B and C have two columns and two rows, respectively. The matrix A in `Eady` is of size 598 and dense, while B and C are column and row vectors, respectively.

Parameters. In these experiments, we set the termination tolerance $\mathtt{tol} = 10^{-8}$. For a fair comparison, we run `eigs` also with termination tolerance equal to 10^{-8}. The interpolation parameter (see Algorithm 2.1) is $\mathtt{q} = 5$ for all of the one-sided variants of the subspace method, and $\mathtt{q} = 2$ for the the two-sided subspace methods excluding the `Eady` example. In the applications of the two-sided subspace methods to the `Eady`

[2]see http://slicot.org/20-site/126-benchmark-examples-for-model-reduction

TABLE 6.3

The residuals of the iterates of the three variants of Algorithm 2.1 to compute the five eigenvalues of the proper rational matrix-valued function in Section 6.1.1 closest to $\tau_1 = -2+\mathrm{i}$. Interpolation is performed at the eigenvalue estimates whose residuals are typed in blue italic letters.

(a) Results for ALL.

ℓ	$\mathrm{Rs_r}\left(\lambda_\ell^{(1)}, v_\ell^{(1)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(2)}, v_\ell^{(2)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(3)}, v_\ell^{(3)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(4)}, v_\ell^{(4)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(5)}, v_\ell^{(5)}\right)$
1	*$4.66 \cdot 10^{-8}$*	*$2.86 \cdot 10^{-6}$*	*$1.85 \cdot 10^{-6}$*	*$9.95 \cdot 10^{-7}$*	*$6.76 \cdot 10^{-6}$*
2	$2.42 \cdot 10^{-19}$	*$7.11 \cdot 10^{-12}$*	*$1.44 \cdot 10^{-8}$*	*$2.13 \cdot 10^{-9}$*	*$3.48 \cdot 10^{-8}$*
3	$3.91 \cdot 10^{-19}$	$2.48 \cdot 10^{-18}$	$2.06 \cdot 10^{-18}$	$9.08 \cdot 10^{-19}$	$1.85 \cdot 10^{-18}$

(b) Results for BR.

ℓ	$\mathrm{Rs_r}\left(\lambda_\ell^{(1)}, v_\ell^{(1)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(2)}, v_\ell^{(2)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(3)}, v_\ell^{(3)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(4)}, v_\ell^{(4)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(5)}, v_\ell^{(5)}\right)$
1	*$1.36 \cdot 10^{-10}$*	$3.10 \cdot 10^{-6}$	$4.11 \cdot 10^{-6}$	$3.21 \cdot 10^{-6}$	$1.65 \cdot 10^{-6}$
2	$5.80 \cdot 10^{-19}$	*$4.40 \cdot 10^{-7}$*	$3.21 \cdot 10^{-6}$	$1.08 \cdot 10^{-6}$	$5.70 \cdot 10^{-6}$
3	$1.45 \cdot 10^{-18}$	$5.25 \cdot 10^{-14}$	$5.46 \cdot 10^{-6}$	*$1.16 \cdot 10^{-6}$*	$5.89 \cdot 10^{-6}$
4	$4.33 \cdot 10^{-19}$	$2.04 \cdot 10^{-15}$	*$1.23 \cdot 10^{-7}$*	$9.23 \cdot 10^{-13}$	$2.82 \cdot 10^{-7}$
5	$2.83 \cdot 10^{-19}$	$8.67 \cdot 10^{-17}$	$9.51 \cdot 10^{-19}$	$3.88 \cdot 10^{-14}$	*$9.41 \cdot 10^{-10}$*
6	$8.17 \cdot 10^{-19}$	$9.08 \cdot 10^{-18}$	$9.80 \cdot 10^{-19}$	$3.10 \cdot 10^{-15}$	$9.42 \cdot 10^{-19}$

(c) Results for WR.

ℓ	$\mathrm{Rs_r}\left(\lambda_\ell^{(1)}, v_\ell^{(1)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(2)}, v_\ell^{(2)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(3)}, v_\ell^{(3)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(4)}, v_\ell^{(4)}\right)$	$\mathrm{Rs_r}\left(\lambda_\ell^{(5)}, v_\ell^{(5)}\right)$
3	$5.58 \cdot 10^{-10}$	$1.33 \cdot 10^{-6}$	$3.64 \cdot 10^{-6}$	$2.21 \cdot 10^{-6}$	*$4.77 \cdot 10^{-6}$*
4	$4.03 \cdot 10^{-11}$	*$7.34 \cdot 10^{-8}$*	$1.22 \cdot 10^{-8}$	$7.31 \cdot 10^{-8}$	$5.20 \cdot 10^{-11}$
5	$2.77 \cdot 10^{-12}$	$1.08 \cdot 10^{-18}$	$1.56 \cdot 10^{-9}$	*$1.95 \cdot 10^{-8}$*	$1.04 \cdot 10^{-11}$
6	$6.65 \cdot 10^{-13}$	$1.18 \cdot 10^{-18}$	*$7.95 \cdot 10^{-11}$*	$7.21 \cdot 10^{-19}$	$5.75 \cdot 10^{-13}$
7	$2.60 \cdot 10^{-14}$	$2.42 \cdot 10^{-18}$	$1.21 \cdot 10^{-18}$	$5.77 \cdot 10^{-19}$	$1.51 \cdot 10^{-15}$

example, we use $\mathtt{q} = 5$ to reduce the number of LU decomposition computations, which is considerably expensive compared to back and forward substitutions as this is a dense example.

Comparison of Runtimes. Runtimes of the subspace methods and `eigs` are reported in Table 6.4. For the random examples, the target point $\tau = 3 + 2\mathrm{i}$ is close to the spectrum, whereas $\tau = 1 + 9\mathrm{i}$ is away. Similarly, the target points $\tau = -4 + 4\mathrm{i}$ and $\tau = -100 + 20\mathrm{i}$ are close to and away from the spectrum of the `Eady` example.

All of the computed eigenvalues by the subspace methods and `eigs` differ by amounts around the prescribed tolerance 10^{-8} with one exception. The fifth closest eigenvalue to $\tau = 1 + 9\mathrm{i}$ for `R2` computed by the subspace methods $(0.2352 + 2.2138\mathrm{i})$ and `eigs` $(0.8429 + 2.1611\mathrm{i})$ differ significantly and are located at a distance 6.8292 and 6.8407 to $\tau = 1 + 9\mathrm{i}$, respectively. We have verified that the absolute residuals at both of these computed eigenvalues are very small.

The subspace methods on these examples have lower runtimes compared to `eigs` when the target point is away from the spectrum, and there is no notable difference in the runtimes on these examples when the target point is closer to the spectrum.

6.2. Polynomial Eigenvalue Problems. Next we consider large-scale polynomial eigenvalue problems available in the NLEVP collection [6]. All of these involve quadratic matrix polynomials of the form $P(s) = P_0 + sP_1 + s^2P_2$ for given square matrices P_0, P_1, $P_2 \in \mathbb{C}^{\mathtt{n}\times\mathtt{n}}$.

Throughout this section, the termination condition in (6.1) is employed with the

TABLE 6.4

Runtimes of the subspace methods vs. `eigs` *in seconds on the test examples of Section 6.1.2 are listed, where # eigs refers to the number of closest eigenvalues sought.*

example, target, # eigs	ALL	BR	WR	ALL1	BR1	WR1	eigs
R1, $3+2$i, 5	5.15	3.55	4.82	3.40	2.22	2.50	2.27
R1, $3+2$i, 10	7.85	5.79	7.25	5.50	3.17	3.82	2.90
R1, $1+9$i, 5	7.20	5.53	7.64	4.39	4.51	4.70	9.85
R1, $1+9$i, 10	11.79	11.46	10.94	8.36	7.02	5.70	14.60
R2, $3+2$i, 5	39.04	30.41	38.95	28.41	15.63	30.64	16.35
R2, $1+9$i, 5	45.79	44.15	52.21	34.21	16.46	33.96	104.44
Eady, $-4+4$i, 5	0.13	0.15	0.13	0.13	0.26	0.17	0.39
Eady, $-4+4$i, 10	0.19	0.34	0.26	0.34	0.85	0.68	0.50
Eady, $-100+20$i, 5	0.23	0.35	0.29	0.15	0.13	0.13	2.24
Eady, $-100+20$i, 10	0.37	0.73	0.43	0.32	0.29	0.32	2.90

tolerance $\texttt{tol} = 10^{-8}$, the partition parameter in (3.1) is $\mathsf{m} = 2$, and the interpolation parameter (see Algorithm 3.1) is $\mathsf{q} = 2$ and $\mathsf{q} = 3$ for the two-sided and one-sided subspace frameworks, respectively. The reduced polynomial eigenvalue problems are solved by using a companion form linearization. In Sections 6.2.1 and 6.2.2 below, in comparisons of the proposed frameworks with CORK, we use the default parameter values for CORK, except its termination tolerance is set equal to 10^{-8}. By default, CORK uses only one shift, which is the target point.

6.2.1. Schrodinger Example. The first example is the `schrodinger` example with $\mathsf{n} = 1998$ that arises from a discretization of the Schrödinger operator.

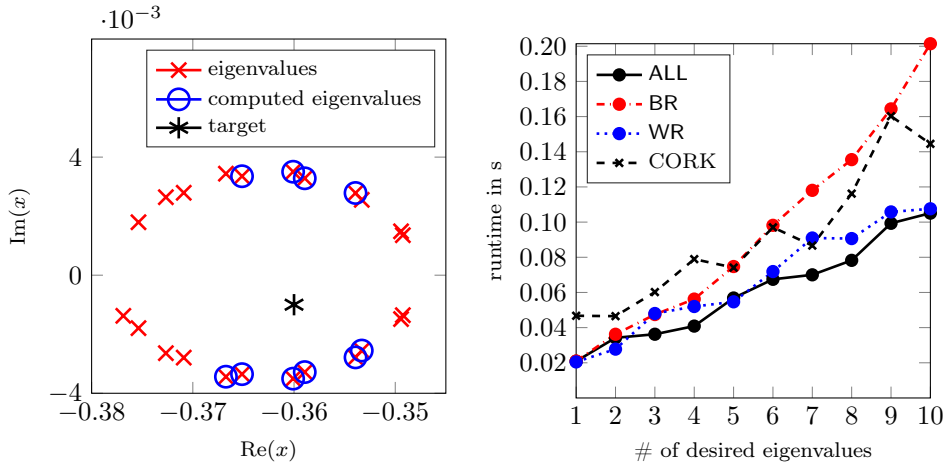
Estimation of Multiple Eigenvalues. We compute the k closest eigenvalues to the target point $\tau = -0.36 - 0.001$i for $k = 1, \ldots, 10$ using the variants ALL, BR, and WR of Algorithm 3.1, as well as the latest version of a free Matlab implementation[3] of the CORK algorithm [28]. In all cases, the computed eigenvalues by all these methods match exactly up to at least eight decimal digits. In particular, in Figure 6.1(a) the eigenvalues near the target point (computed by applying `eigs` to a linearization of P) are displayed with red crosses, and the ten closest eigenvalues computed by the subspace methods and CORK are encircled in blue.

Comparison of Runtimes. In Figure 6.1(b), the runtimes in seconds for the three variants of Algorithm 3.1 and the CORK algorithm are plotted as the function of the prescribed number of eigenvalues. The runtimes appear to be similar, though the variants ALL and WR look slightly faster.

Decay in the Residuals. For the ALL variant of Algorithm 3.1 and to compute the ten eigenvalues closest to τ, the termination criterion is satisfied after two iterations. The residuals for the ten eigenvalue estimates at each of these two iterations are given in Table 6.5. Interpolation is performed at every one of the ten eigenvalue estimates at the second iteration, and all residuals decrease dramatically.

6.2.2. Other Quadratic Eigenvalue Problems. We have also experimented with various other quadratic eigenvalue problems (QEPs) from the NLEVP collection. In Table 6.6, the runtimes of the variants of Algorithm 3.1 to compute the five eigenvalues closest to prescribed target points for several QEPs are listed together with the runtimes of the CORK algorithm. The computed eigenvalues by all of the approaches are the same up to nearly eight decimal digits.

[3] available at http://twr.cs.kuleuven.be/research/software/nleps/cork.html



(a) Eigenvalues near the target $\tau = -0.36 - 0.001\mathrm{i}$ and the computed ten eigenvalues closest to τ by the variants of Algorithm 3.1.

(b) Runtimes in seconds as functions of prescribed number of eigenvalues.

FIG. 6.1. *The variants of Algorithm 3.1 applied to the* `schrodinger` *example.*

TABLE 6.5

The residuals of the eigenvalue estimates of the ALL *variant of Algorithm 3.1 on the* `schrodinger` *example with the target point* $\tau = -0.36 - 0.001\mathrm{i}$.

ℓ	$\mathrm{Rs}\left(\lambda_\ell^{(1)}, v_\ell^{(1)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(2)}, v_\ell^{(2)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(3)}, v_\ell^{(3)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(4)}, v_\ell^{(4)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(5)}, v_\ell^{(5)}\right)$
1	$1.05 \cdot 10^{-8}$	$1.08 \cdot 10^{-8}$	$1.64 \cdot 10^{-8}$	$1.77 \cdot 10^{-8}$	$3.02 \cdot 10^{-8}$
2	$8.71 \cdot 10^{-16}$	$9.37 \cdot 10^{-16}$	$2.08 \cdot 10^{-15}$	$5.70 \cdot 10^{-16}$	$1.93 \cdot 10^{-15}$
ℓ	$\mathrm{Rs}\left(\lambda_\ell^{(6)}, v_\ell^{(6)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(7)}, v_\ell^{(7)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(8)}, v_\ell^{(8)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(9)}, v_\ell^{(9)}\right)$	$\mathrm{Rs}\left(\lambda_\ell^{(10)}, v_\ell^{(10)}\right)$
1	$1.26 \cdot 10^{-8}$	$4.32 \cdot 10^{-8}$	$2.68 \cdot 10^{-8}$	$1.38 \cdot 10^{-8}$	$5.88 \cdot 10^{-8}$
2	$2.34 \cdot 10^{-15}$	$2.83 \cdot 10^{-15}$	$4.38 \cdot 10^{-15}$	$1.34 \cdot 10^{-15}$	$2.59 \cdot 10^{-15}$

6.2.3. Comparison with a Rational Krylov Method with Adaptive Shifts. The comparisons in the previous two subsections are with CORK that uses a static shift, namely the prescribed target point, at every iteration. As Algorithm 3.1 selects new interpolation points at every iteration, it has similarities with a rational Krylov method for nonlinear eigenvalue problems that chooses shifts adaptively, where shifts correspond to the interpolation points for the polynomial or rational approximation [27, 14].

Here, we compare Algorithm 3.1 with a modification of CORK that uses adaptive shifts, equivalently adaptive interpolation points. Specifically, every shift is used a few times. Then the shift is set equal to the Ritz value closest to the target point. The reason to use every shift more than once is to interpolate not only the function values but also the derivatives. Also, an alternative for the shift selection is to use the Ritz value with the smallest residual. We have experimented with such alternatives only to observe that they lead to approaches that are often less reliable.

The results to compute an eigenvalue closest to a target point on a few quadratic eigenvalue problems are given in Table 6.7. In these examples, the computed eigenvalue estimates by the two approaches are the same up to about four decimal digits. We have consistently observed that Algorithm 3.1 requires fewer number of LU de-

TABLE 6.6

Runtimes of the subspace methods and CORK in seconds on several quadratic eigenvalue problems. The examples marked with (D) are dense examples. The sizes of the problems are $\mathsf{n} = 2472, 2000, 2000, 100172, 10000, 35955, 1331$ *from top to bottom.*

example	target	ALL	BR	WR	ALL1	BR1	WR1	CORK
`concrete`	$1+5\mathrm{i}$,	0.12	0.07	0.07	0.04	0.06	0.06	0.07
`dirac` (D)	$1+0.2\mathrm{i}$	2.87	3.90	3.81	2.05	2.64	2.55	13.96
`gen_hyper2` (D)	$3+3\mathrm{i}$	2.37	3.49	3.48	1.98	3.58	2.93	3.77
`acoustic_wave_2d`	$4+0.1\mathrm{i}$	4.24	4.71	3.67	3.54	2.84	2.74	3.89
`pdde_stability`	$-0.1+0.03\mathrm{i}$	0.56	0.57	0.57	0.39	0.46	0.37	0.76
`railtrack2`	$3-\mathrm{i}$	23.40	20.47	18.99	17.36	10.95	12.42	11.24
`utrecht1331`	$300\mathrm{i}$	0.15	0.15	0.13	0.13	0.11	0.08	0.11

TABLE 6.7

A comparison of Algorithm 3.1 with adaptive CORK on a few quadratic eigenvalue problems. The computed closest eigenvalues to the target points are reported in the column of λ*. For both methods, the runtime in seconds and # LU decompositions are listed.*

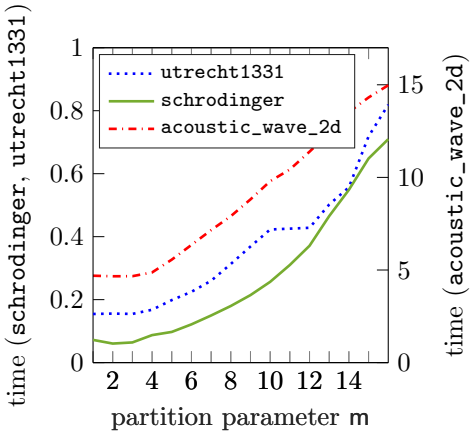
			Alg. 3.1		Adap. CORK	
example	target	λ	time	# lu	time	# lu
`concrete`	$-0.2+5\mathrm{i}$	$-0.2067+5.2350\mathrm{i}$	0.04	2	0.26	6.2
`acoustic_wave_2d`	4	$4.0281+0.0429\mathrm{i}$	1.63	3.2	4.34	4.8
`pdde_stability`	-0.1	$-0.1028-0.0001\mathrm{i}$	0.15	3	0.29	3.6
`railtrack2`	$3-1\mathrm{i}$	$3.1433-1.7621\mathrm{i}$	10.17	3	22.87	5

compositions until termination as compared to adaptive CORK. A difficulty we have encountered is that our version of adaptive CORK often fails to converge to the correct eigenvalue unless the prescribed target point is close to an eigenvalue. This is the reason why the target point in these examples are chosen close to an eigenvalue.

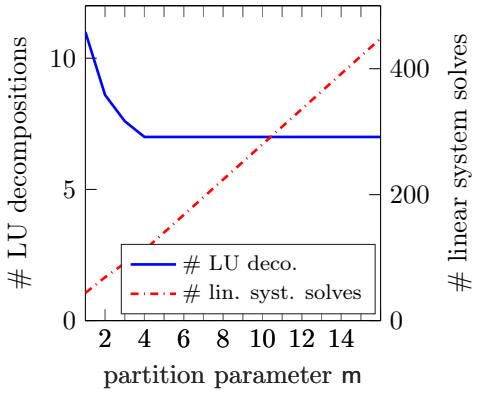
6.2.4. The Effect of the Partition Parameter. We investigate the effect of the partition parameter m on the ALL variant numerically. In Figure 6.2(a), the runtimes are reported as m varies in $[1, 16]$ for three quadratic eigenvalue problems. The runtimes do not change much for smaller values of m, i.e., for $\mathsf{m} \in [1, 4]$. But then for larger values of m the runtimes increase gradually as m is increased.

This dependence of the runtimes on m is partly explained by the number of LU decomposition and linear system solves performed, which are depicted in Figure 6.2(b). As m is increased, the number of subspace iterations decreases slightly initially, possibly since larger values of m result in more accurate interpolating reduced problems. But for $\mathsf{m} \geq 4$ the number of subspace iterations stagnate and do not decrease anymore. This variation in the number of iterations is directly reflected into the number of LU decompositions shown in Figure 6.2(b). On the other hand, the number of linear system solves increases consistently as a function of m, also visible in Figure 6.2(b). For smaller values of m the decrease in the number of *LU* decompositions is offset by the increase in the number of linear system solves, leading to a nearly constant dependence of the runtime on m. But for $\mathsf{m} \geq 4$, there is no offset for the increasing cost of linear system solves, so the runtime increases.

Apart from runtime considerations, there appears to be a second good reason to choose m small. For larger values of m, we have occasionally witnessed problems with convergence, whereas for smaller values of m convergence is almost always guaranteed. This is merely a practical observation as of now, which we hope to be able to reason in the future.



(a) Runtimes in seconds of the ALL variant as a function of m.



(b) Number of LU decompositions and linear system solves by the ALL variant as functions of m on the acoustic_wave_2d example.

FIG. 6.2. *The effect of* m *(the partition parameter) to compute the five closest eigenvalues. The target points* $\tau = -0.36 - 0.001\mathrm{i}$, $300\mathrm{i}$, $4 + 0.1\mathrm{i}$ *are used for the* ***schrodinger***, ***utrecht1331***, ***acoustic_wave_2d*** *examples, respectively.*

6.3. Non-Rational, Non-Polynomial Nonlinear Eigenvalue Problems. Now we apply the ALL variant of Algorithm 3.1 to three nonlinear eigenvalue problems from the NLEVP collection, that are neither polynomial nor rational.

The termination tolerance in (6.1) is $\mathtt{tol} = 10^{-8}$ unless otherwise specified, while the interpolation parameter in Algorithm 3.1 and partition parameter in (3.1) are $\mathsf{q} = 2$ and $\mathsf{m} = 2$ throughout this section. The eigenvalues of the reduced problems are computed using the Matlab implementation of NLEIGS [14] that is available on the internet[4].

6.3.1. The Gun Problem. The gun problem, originating from modeling of a radio-frequency gun cavity, concerns the solution of a nonlinear eigenvalue problem of the form

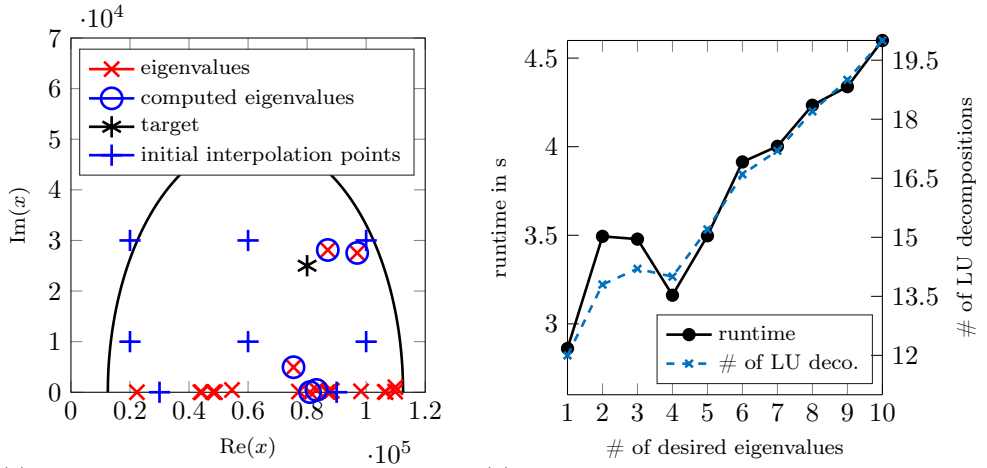
$$T(\lambda)v = \left(P_0 - \lambda P_1 + \mathrm{i}\sqrt{\lambda - \sigma_1^2}\, W_1 + \mathrm{i}\sqrt{\lambda - \sigma_2^2}\, W_2\right) v = 0$$

for given real symmetric matrices P_0, P_1, W_1, $W_2 \in \mathbb{R}^{9956\times 9956}$ with positive semidefinite P_0 and positive definite P_1. With the parameters $\sigma_1 = 0$, $\sigma_2 = 108.8774$, the eigenvalues inside the upper half of the disk in the complex plane centered at 250^2 on the real axis with radius $300^2 - 250^2$ are reported in several works in the literature.

Estimation of Eigenvalues. Here, we compute the k eigenvalues closest to the target $\tau = (8 + 2.5\mathrm{i}) \cdot 10^4$ for $k = 1, \ldots, 10$ inside the specified upper half-disk by employing the ALL variant. The initial interpolation points are not chosen randomly anymore. Rather, in addition to the target point τ, we employ the following eight interpolation points initially: $3 \cdot 10^4, 9 \cdot 10^4, (2+\mathrm{i}) \cdot 10^4, (6+\mathrm{i}) \cdot 10^4, (10+\mathrm{i}) \cdot 10^4, (2+3\mathrm{i}) \cdot 10^4, (6+3\mathrm{i}) \cdot 10^4, (10+3\mathrm{i}) \cdot 10^4$.

The computed five closest eigenvalues (encircled in blue) together with all eigenvalues (marked with red crosses) inside the half-disk are shown in Figure 6.3(a). In all

[4]available at http://twr.cs.kuleuven.be/research/software/nleps/nleigs.html



(a) The eigenvalues of the gun example inside the upper semi-circle, and the computed five closest eigenvalues to the target $\tau = (8+2.5\mathrm{i})\cdot 10^4$ by the ALL variant of Algorithm 3.1.

(b) The runtime in seconds and the number of LU decompositions required by the ALL variant of Algorithm 3.1 as functions of the number of closest eigenvalues sought.

FIG. 6.3. *Performance of Algorithm 3.1 on the* **gun** *example.*

cases, the computed eigenvalues are the same as those returned by NLEIGS applied directly to the full problem up to prescribed tolerances. Figure 6.3(b) depicts the runtime and LU decompositions required by the ALL variant as functions of k. The runtime of the ALL variant on this gun example is mainly affected by the number of LU decompositions, and linear system solves. As a result, the runtimes and number of LU decompositions vary more or less in harmony in Figure 6.3(b) as k increases.

Comparison with a Direct Application of NLEIGS. A direct application of NLEIGS with a tolerance of 10^{-4} on the residual for termination leads to 21 eigenvalues inside the half disk. This is also consistent with what is reported in [27]. We apply the ALL variant of Algorithm 3.1 with tolerance $\texttt{tol} = 10^{-4}$ to compute all 21 eigenvalues inside the specified region closest to $\tau = 146.71^2$. The computed eigenvalues are nearly the same with those returned by NLEIGS with relative differences about 10^{-8} or smaller.

Runtimes for both approaches are reported in Table 6.8. The total times in the last column are listed disregarding the time for orthogonalization. This is because the implementation of NLEIGS does not exploit the Kronecker structure of the linearization when orthogonalizing the subspaces, and it is likely that orthogonalization costs would be negligible for NLEIGS just like it is the case for the ALL variant, had the Kronecker structure been taken into account. Still, the total time for the ALL variant is substantially smaller. The table also reveals that linear system solves dominate the computation time. The difference of the total time and time for linear system solves for the ALL variant mainly corresponds to the time spent for the construction of the reduced problems and their solution.

6.3.2. The Particle in a Canyon Problem. This problem arises from a finite element discretization of the Schrödinger equation for a particle in a canyon-shaped

TABLE 6.8
A comparison of the runtimes of the ALL *variant and NLEIGS. The times for linear system solves and orthogonalization, as well as the total time excluding the time for orthogonalization are listed in seconds for both of the approaches.*

	linear systems	orthogonalization	total time w/o orth.
NLEIGS	7.89	31.03	13.18
Algorithm 3.1, ALL	2.98	0.10	4.17

potential well, and is of the form

$$T(\lambda)v = \left(H - \lambda I - \sum_{k=1}^{81} e^{\mathrm{i}\sqrt{2(\lambda-\alpha_k)}} A_k \right) v = 0,$$

where $H \in \mathbb{R}^{16281\times 16281}$ is sparse, $A_1, \ldots, A_{81} \in \mathbb{R}^{16281\times 16281}$ are sparse with rank two, and $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ are given branch points. Indeed, the decompositions of $A_k = L_k U_k^\top$ for $L_k, U_k \in \mathbb{R}^{16281\times 2}$ are available in NLEVP, but in our implementation we do not exploit this low-rank structure. The eigenvalues of interest are those on the real axis in the interval (α_1, α_2), where $\alpha_1 \approx -0.1979$ and $\alpha_2 \approx -0.1320$.

Estimation of Eigenvalues. We compute the closest eigenvalue and the two closest eigenvalues to the target points $\tau_1 = -0.135$ and $\tau_2 = -0.180$ by the ALL variant of the subspace method. Five initial interpolation points are chosen equidistantly in the interval (α_1, α_2). The results are summarized in Table 6.9. The computed eigenvalues in the second columns, displayed to a five decimal digit accuracy, are the same as those returned by a direct application of NLEIGS. Solutions of the projected subproblems by NLEIGS take nearly half of the computation time.

TABLE 6.9
Results for the ALL *variant of Algorithm 3.1 on the particle in a canyon problem. In the last three columns, times for linear system solves, subproblems, and total runtimes are listed in seconds.*

target, # eigs	computed eigs	linear systems	subproblems	total time
−0.135, 1	−0.13254	0.37	0.90	1.98
−0.135, 2	−0.13254, −0.14060	0.38	0.92	2.11
−0.180, 1	−0.14060	0.37	0.91	2.04
−0.180, 2	−0.13254, −0.14060	0.42	0.94	2.20

6.3.3. The Partial Delay Differential Equation Problem. Our final test is on a problem that comes from a finite difference discretization of a delay partial differential equation. This problem is abbreviated as `Pdde_symmetric` in the NLEVP collection, and has the form

$$T(\lambda)v = \left(F - \lambda I + e^{-2\lambda} G\right) v = 0$$

for sparse and banded $F, G \in \mathbb{R}^{16129\times 16129}$. The eigenvalues close to zero are of interest. In particular, we seek eigenvalues in the interval $[-1, 1]$.

Estimation of Eigenvalues. We compute the six eigenvalues closest to $\tau = 0.2$. Five initial interpolation points are chosen randomly on the real axis from a normal distribution with zero mean and variance equal to 0.2. The six eigenvalue estimates computed and runtimes are reported in Table 6.10. All of the six eigenvalue estimates retrieved have residuals smaller than $5 \cdot 10^{-12}$. The computation time is once again dominated by the solutions of the projected small-scale eigenvalue problems.

TABLE 6.10
ALL *variant of Algorithm 3.1 on the partial delay differential equation problem.*

(a) Computed closest six eigenvalues to $\tau = 0.2$.

-0.00248	-0.51908	-0.56141	-0.84591	-0.89726	-0.92237

(b) Computation times in seconds.

linear system solves	subproblems	total runtime
0.40	0.72	1.30

7. Concluding Remarks. We have proposed subspace frameworks based on Hermite interpolation to deal with the estimation of a few eigenvalues of a large-scale nonlinear matrix-valued function closest to a prescribed target. At every subspace iteration, first a reduced nonlinear eigenvalue problem is obtained by employing two-sided or one-sided projections inspired from interpolatory model-order reduction techniques, then the eigenvalues of the reduced eigenvalue problem are extracted, and finally the projection subspaces are expanded to attain Hermite interpolation between the full and reduced problem at the eigenvalues of the reduced problem. We have proven that the proposed framework converges at least at a quadratic rate in theory under a non-defectiveness and a non-degeneracy assumption.

There are several directions that are open to improvement. One of them is the initial selection of the interpolation points. This may affect the number of subspace iterations. At the moment, we choose the initial interpolation points randomly around the target. A more careful selection of them, for instance using the AAA algorithm [22], may reduce the number of subspace iterations, and improve the reliability. Another issue is the partitioning of the nonlinear matrix-valued function $T(\cdot)$ as in (3.1), where $B(\cdot)$ has few columns and $C(\cdot)$ has few rows. This partitioning may affect the convergence and stability properties of the framework. It seems even possible to permute the rows and columns of $T(\cdot)$, and more generally apply unitary transformations from left or right in advance with the purpose of enhancing the convergence and stability properties. We hope to address these issues in a future work.

Software. Matlab implementations of ALL, BR, WR, ALL1, BR1, WR1 variants of Algorithms 2.1 and 3.1, and the banded rational eigenvalue problem example in Section 6.1.1, as well as two sparse random rational eigenvalue problems (i.e., `R1`, `R2`) in Section 6.1.2 that are experimented on are publicly available at https://zenodo.org/record/5811971.

Other nonlinear eigenvalue problem examples on which we perform experiments in Sections 6.2 and 6.3 are also publicly available in the NLEVP collection [6].

Acknowledgements. The authors are grateful to two anonymous referees who provided invaluable comments about the initial versions of this manuscript.

REFERENCES

[1] A. ALIYEV, P. BENNER, E. MENGI, P. SCHWERDTNER, AND M. VOIGT, *Large-scale computation of $\mathcal{L}_\infty$-norms by a greedy subspace method*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1496–1516.

[2] A. ALIYEV, P. BENNER, E. MENGI, AND M. VOIGT, *A subspace framework for $\mathcal{H}_\infty$-norm minimization*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 928–956.

[3] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, vol. 6 of Adv. Des. Con-

trol, SIAM Publications, Philadelphia, PA, 2005.
[4] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective*, Arch. Comput. Methods Eng., 21 (2014), pp. 331–358.
[5] C. Beattie and S. Gugercin, *Interpolatory projection methods for structure-preserving model reduction*, Systems Control Lett., 58 (2009), pp. 225–232.
[6] T. Betcke, N. J. Higham, V. Mehrmann, , C. Schröder, and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, ACM Trans. Math. Software, 39 (2013), pp. 7:1–7:28.
[7] M. Brennan, M. Embree, and S. Gugercin, *Contour integral methods for nonlinear eigenvalue problems: A systems theoretic approach*, Preprint arXiv:2012.14979, 2020.
[8] C. De Villemagne and R. E. Skelton, *Model reductions using a projection formulation*, Internat. J. Control., 46 (1987), pp. 2141–2169.
[9] C. Effenberger and D. Kressner, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT, 52 (2012), pp. 933–951.
[10] B. A. Francis and W. M. Wonham, *The role of transmission zeros in linear multivariable regulators*, Internat. J. Control, 22 (1975), pp. 657–681.
[11] S. Gugercin, T. Stykel, and S. Wyatt, *Model reduction of descriptor systems by interpolatory projection methods*, SIAM J. Sci. Comput., 35 (2013), pp. B1010–B1033.
[12] S. Güttel, G. M. N. Porzio, and F. Tisseur, *Robust rational approximations of nonlinear eigenvalue problems*, MIMS EPrint 2020.24, 2020. Available at http://eprints.maths.manchester.ac.uk/2796/.
[13] S. Güttel and F. Tisseur, *The nonlinear eigenvalue problem*, Acta Numer., 26 (2017), pp. 1–94.
[14] S. Güttel, R. Van Beeumen, K. Meerbergen, and W. Michiels, *NLEIGS: A class of fully rational eigenvalue Krylov methods for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 36 (2014), pp. A2842–A2864.
[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2nd ed., 2013.
[16] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math., 6 (1964), pp. 377–387.
[17] P. Lietaert, K. Meerbergen, J. Pérez, and B. Vandereycken, *Automatic rational approximation and linearization of nonlinear eigenvalue problems*, IMA J. Numer. Anal., (2021).
[18] P. Lietaert, K. Meerbergen, and K. Tisseur, *Compact two-sided Krylov methods for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 40 (2014), pp. A2801–A2829.
[19] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.
[20] N. Martins, P. C. Pellanda, and J. Rommes, *Computation of transfer function dominant zeros with applications to oscillations damping control of large power systems*, IEEE Trans. Power Syst., 22 (2007), pp. 1657–1664.
[21] V. Mehrmann and H. Voss, *Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods*, GAMM-Mitt., 27 (2004), pp. 121–152.
[22] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, *The AAA algorithm for rational approximation*, SIAM J. Sci. Comput., 40 (2018), pp. A1494–A1522.
[23] C. . Quendo, E. Rius, and C. Person, *An original topology of dual-band filter with transmission zeros*, IEEE MTT-S International Microwave Symposium Digest, 2 (2003), pp. 1093–1096.
[24] J. Rommes and N. Martins, *Efficient computation of transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1218–1226.
[25] Y. Su and Z. Bai, *Solving rational eigenvalue problems via linearization*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 201–216.
[26] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[27] R. Van Beeumen, K. Meerbergen, and W. Michiels, *A rational krylov method based on hermite interpolation for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A327–A350.
[28] R. Van Beeumen, K. Meerbergen, and W. Michiels, *Compact rational Krylov methods for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2014), pp. 820–838.
[29] A. Yousuff, D. A. Wagie, and R. E. Skelton, *Linear system approximation via covariance equivalent realizations*, J. Math. Anal. Appl., 106 (1985), pp. 91–115.