SIAM J. SCI. COMPUT.
Vol. 36, No. 6, pp. A2842–A2864

NLEIGS: A CLASS OF FULLY RATIONAL KRYLOV METHODS FOR NONLINEAR EIGENVALUE PROBLEMS*

STEFAN GÜTTEL†, ROEL VAN BEEUMEN‡, KARL MEERBERGEN‡, AND WIM MICHIELS‡

Abstract. A new rational Krylov method for the efficient solution of nonlinear eigenvalue problems, $A(\lambda)x = 0$, is proposed. This iterative method, called fully rational Krylov method for nonlinear eigenvalue problems (abbreviated as NLEIGS), is based on linear rational interpolation and generalizes the Newton rational Krylov method proposed in [R. Van Beeumen, K. Meerbergen, and W. Michiels, *SIAM J. Sci. Comput.*, 35 (2013), pp. A327–A350]. NLEIGS utilizes a dynamically constructed rational interpolant of the nonlinear function $A(\lambda)$ and a new companion-type linearization for obtaining a generalized eigenvalue problem with special structure. This structure is particularly suited for the rational Krylov method. A new approach for the computation of rational divided differences using matrix functions is presented. It is shown that NLEIGS has a computational cost comparable to the Newton rational Krylov method but converges more reliably, in particular, if the nonlinear function $A(\lambda)$ has singularities nearby the target set. Moreover, NLEIGS implements an automatic scaling procedure which makes it work robustly independently of the location and shape of the target set, and it also features low-rank approximation techniques for increased computational efficiency. Small- and large-scale numerical examples are included. From the numerical experiments we can recommend two variants of the algorithm for solving the nonlinear eigenvalue problem.

Key words. nonlinear eigensolver, rational Krylov, linear rational interpolation

AMS subject classifications. 47J10, 65F15, 41A20

DOI. 10.1137/130935045

1. Introduction.

We consider the problem of finding eigenvalues $\lambda \in \Sigma$ and eigenvectors $x \in \mathbb{C}^n \backslash \{0\}$ of a nonlinear eigenvalue problem (NLEP)

$$A(\lambda)x = 0, \tag{1.1}$$

with a compact *target set* $\Sigma \subset \mathbb{C}$ and a family of matrices $A(\lambda) : \Sigma \to \mathbb{C}^{n\times n}$ depending analytically on λ, i.e., each component of $A(\lambda)$ is an analytic function of λ.

The NLEP (1.1) has been studied extensively in the literature and there exist specialized methods for different families of $A(\lambda)$; see, e.g., [28, 35]. Popular approaches can be roughly classified as Newton-type methods [29, 6, 22], methods based on contour integration [8, 7, 10], and methods based on approximations of $A(\lambda)$ [28, 11, 35, 21, 36]. The method proposed in this paper belongs to the last class.

The *infinite Arnoldi method* was first proposed in [21]. The key idea is to apply Arnoldi's method in a function setting to a linear operator eigenvalue problem, which is equivalent to the original nonlinear eigenvalue problem. Its "Taylor ver-

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 30, 2013; accepted for publication (in revised form) September 15, 2014; published electronically December 10, 2014. This work was supported by the Programme of Interuniversity Attraction Poles of the Belgian Federal Science Policy Office (IAP P6-DYSCO), by OPTEC, the Optimization in Engineering Center of the KU Leuven, by projects STRT1-09/33, OT/10/038, OT/14/074 of the KU Leuven Research Council, and by project G.0712.11N, and a travel grant of the Research Foundation-Flanders (FWO).

http://www.siam.org/journals/sisc/36-6/93504.html

†School of Mathematics, The University of Manchester, M13 9PL Manchester, United Kingdom (Stefan.Guettel@manchester.ac.uk).

‡Department of Computer Science, KU Leuven, University of Leuven, 3001 Heverlee, Belgium (Roel.VanBeeumen@cs.kuleuven.be, Karl.Meerbergen@cs.kuleuven.be, Wim.Michiels@cs.kuleuven.be).

sion" can be interpreted as a shift-and-invert Arnoldi method applied to a companion linearization obtained from a Taylor expansion of the nonlinear function $A(\lambda)$ into polynomials of λ. The degree of the expansion is not fixed in advance, resulting in a dynamical iterative algorithm. In its "Chebyshev version," the method can be interpreted as Arnoldi's method applied to a spectral discretization of the operator, which for delay eigenvalue problems has an interpretation as a rational approximation of the exponential terms. See [41] for the connection between a spectral discretization and rational approximation. In a recent work [36], the Taylor version of the infinite Arnoldi method was generalized to the case of multiple interpolation nodes, yielding the *Newton rational Krylov method*. This method uses an interpolatory expansion of $A(\lambda)$ into Newton polynomials of λ. Other modifications and extensions of the infinite Arnoldi algorithm have been proposed, for example, a restarted version [20], or a spatially adaptive version allowing for the solution of operator-valued (instead of matrix-valued) nonlinear eigenproblems [18].

For methods relying on *polynomial interpolants* of $A(\lambda)$, the convergence is limited by the convergence of polynomials. With an appropriate choice of the interpolation nodes, polynomial interpolation performs very well if $A(\lambda)$ is an entire function, in which case any polynomial interpolant is guaranteed to converge throughout the complex plane, or when singularities of $A(\lambda)$ are sufficiently far away from the target set Σ. However, if $A(\lambda)$ is difficult to approximate by polynomials, the performance of the expensive rational Krylov iteration will be limited by the accuracy of the underlying polynomial expansion. The good convergence properties of the Chebyshev version of the infinite Arnoldi method for the delay eigenvalue problem can to a large extent be attributed to the fact that the underlying approximation is a rational approximation [19]. However, this rational approximation is implicit in the sense that it is induced by the spectral discretization of the operator and, e.g., the poles cannot be chosen freely by the user, unlike in the presented method.

In this work we propose a new rational Krylov method where the nonlinearity $A(\lambda)$ is explicitly expanded in *rational functions* of λ, hence the name *fully rational Krylov method for nonlinear eigenvalue problems*, abbreviated as NLEIGS. The combination of a new linearlization, based on rational Newton basis functions, with a rational Krylov method offers a lot of flexibility:

First, in the spirit of [11] and [35], the approximation and the corresponding linearization can be first constructed, in such a way that the approximation error is guaranteed to be uniformly small on the target set Σ (assumed as a compact subset of the complex plane). The resulting linear eigenvalue problem can then be solved by any method of choice, like the standard rational Krylov method. An advantage of the proposed rational approximation is that the interpolation nodes and poles can be freely chosen, hence a choice based on arguments from potential theory (e.g., Leja–Bagby points) may lead to a fast uniform convergence of the approximation on the whole target set. Another advantage, which stems from the rational Newton expansion and the companion-like linearization, is that interpolation nodes and poles can be added incrementally in a straightforward way by simply extending the matrices of the linearization, while the convergence of the approximation can be monitored by the magnitude of the computed generalized divided differences.

Second, in the spirit of the infinite Arnoldi method and the Newton rational Krylov method, which are based on dynamic local approximation, the construction of the rational approximation and the application of the rational Krylov method can be tightly interwoven, by exploiting the structure of the linearization, a special starting vector, and the "trick" proposed in [36] of choosing the shifts of the rational

Krylov space identical to the interpolation nodes. This results in a fully dynamic method, where the underlying linearization of the resulting rational eigenvalue problem is extended at every rational Krylov iteration, thereby discovering more and more eigenpairs with ever increasing accuracy. In other words, the order of the linearization is not fixed in advance; it will increase dynamically as the iteration proceeds. Even though the dynamic property of the algorithm is a major advantage achieved by equating interpolation nodes and rational Krylov shifts, there might also be a price to pay because an optimal choice in view of achieving a fast converging linearization, which typically means choosing interpolation nodes on the boundary of the target set, might not always be favorable for the rational Krylov method to converge quickly, e.g., if the eigenvalues are not located close to the boundary of the target set, and vice versa.

Therefore a combination of the two approaches is sometimes necessary, as we shall illustrate in the paper. For instance, we can start with the dynamic variant where shifts/interpolation nodes are determined by the underlying approximation problem, and at the moment that the approximation has converged to sufficient accuracy, we can freeze the linearization and continue the rational Krylov iterations on the resulting matrices with the selection of shifts as for the standard rational Krylov method.

We now outline our contributions and the structure of this work. In section 2 we will give a motivating example, showcasing the possible advantages of rational over polynomial interpolation. In section 3 we show how the NLEP (1.1) can be approximated by linear rational interpolation, including a new theorem on the computation of rational divided differences via matrix functions. We then present a new linearization of the resulting rational eigenvalue problem. In section 4 we discuss how this linearization can be combined with the rational Krylov iteration. The resulting algorithm, called *fully rational Krylov method*, generalizes the Newton rational Krylov method of [36] (when all poles ξ_j are infinite), and the Taylor version of the infinite Arnoldi method of [21] (when all poles ξ_j are infinite and all interpolation nodes σ_j coincide). Our approach is particularly efficient when some information about the (approximate) location of the singularities of $A(\lambda)$ is available, described by a set Ξ, and we will discuss the problem of choosing near-optimal Leja–Bagby parameters in section 5. These Leja–Bagby points may not always be advantageous for the rational Krylov iteration to quickly find the targeted eigenvalues of the linearization. However, due to its fast convergence, the rational expansion can be truncated after some iterations, which ultimately allows us to freely choose the shifts of the rational Krylov space. This truncation procedure will be discussed in section 6, along with a strategy for exploiting low-rank structure in the NLEP. We also present in that section an automated scaling procedure via an estimation of the logarithmic capacity of (Σ, Ξ) using control points on the boundary of Σ. Combined with the choice of near-optimal interpolation nodes, which guarantees that the rational expansion of $A(\lambda)$ is a good approximation on Σ, this scaling procedure avoids numerical underflow or overflow in the sampling phase of the NLEP and allows the residual norm of the eigenpairs of the approximate linearization of $A(\lambda)$ to be controlled. We consider our algorithm to be robust in that sense. Note that, as with Krylov methods for linear eigenvalue problems, this robust choice of approximation does not guarantee that *all* eigenvalues in Σ are found. Convergence merely depends on the choice of parameters of the rational Krylov method, such as the shifts and number of iterations. All experiments in section 7 are done with our MATLAB implementation `nleigs` which is available for download, with the weblink to be found in section 7.

Throughout this paper, we denote vectors v by lowercase Roman characters, matrices A by capital Roman characters, scalars α by lowercase Greek letters, and sets Σ

by capital Greek letters. For block vectors and block matrices we use $\boldsymbol{v}$ and $\boldsymbol{A}$, respectively, and a superscript as in $v^{[j]}$ denotes the jth block of the block vector $\boldsymbol{v}$. The conjugate transpose of a matrix A is denoted by A^*. If not stated otherwise, V_j denotes a matrix with j columns and $A_{j,k}$ is a matrix of size $j \times k$. We omit subscripts when the dimensions of matrices are clear from the context. Column j of the matrix V is denoted by v_j. A superscript such as $\lambda^{(j)}$ is used to denote the value at iteration j of a quantity that may change from one iteration to the next.

2. Motivating example. We now give some motivation for the approach of the paper by means of the simple scalar NLEP

$$A(\lambda) = 0.2\sqrt{\lambda} - 0.6\sin(2\lambda), \tag{2.1}$$

which illustrates the improvements that can potentially be achieved by using a rational expansion instead of a polynomial expansion. Suppose we want to find all 3 real eigenvalues of $A(\lambda)$ on the interval $\Sigma = [\alpha, \beta] = [10^{-2}, 4]$. Of course, this is in fact just a root finding problem. A natural solution approach is to first approximate $A(\lambda)$ by a polynomial interpolant $P_N(\lambda)$ of degree N, interpolating at nodes $\sigma_0, \sigma_1, \ldots, \sigma_N \in \Sigma$, and then to compute the roots of $P_N(\lambda)$ on Σ. See [9] for a recent review of polynomial root finding. Let λ^* be such a root, i.e., $P_N(\lambda^*) = 0$. Then the residual $A(\lambda^*)$ is bounded by the uniform approximation error of $P_N(\lambda)$ for $A(\lambda)$:

$$|A(\lambda^*)| \le \max_{\lambda\in\Sigma} |A(\lambda) - P_N(\lambda)| =: \|A(\lambda) - P_N(\lambda)\|_\Sigma.$$

In view of this inequality, it is natural to make the error $\|A(\lambda) - P_N(\lambda)\|_\Sigma$ small. The asymptotic convergence of $P_N(\lambda)$ to $A(\lambda)$ in the uniform norm is determined by the location of the singularities of $A(\lambda)$ relative to Σ, and by the distribution of interpolation nodes on Σ. For interpolation nodes which are chosen asymptotically optimal on Σ we can show that convergence takes place at a geometric rate

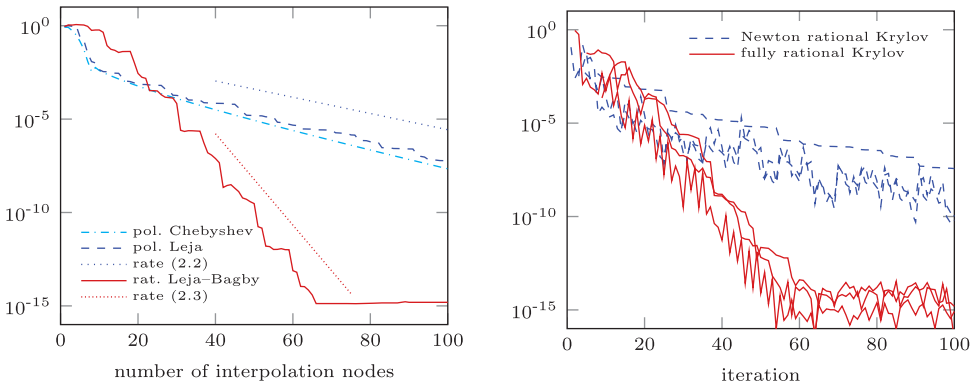
$$\limsup_{N\to\infty} \|A(\lambda) - P_N(\lambda)\|_\Sigma^{1/N} \le \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right) \lesssim \exp(-2/\sqrt{\kappa}), \tag{2.2}$$

with $\kappa = \beta/\alpha$ and $\lesssim$ denoting an approximate upper bound that is asymptotically sharp for large κ. This is the best possible asymptotic convergence that can be achieved by polynomial interpolation. Examples of interpolation nodes for which this convergence is achieved are the Chebyshev points $\sigma_j = \frac{\alpha+\beta}{2} + \frac{\alpha-\beta}{2}\cos(\pi j/N)$, $j = 0, 1, \ldots, N$, and Leja points (see, e.g., [31]).

It is well known that rational interpolants $Q_N(\lambda)$ potentially exhibit faster convergence than polynomials, in particular, if the function to be approximated has singularities nearby Σ. In example (2.1), the *singularity set* Ξ of $A(\lambda)$ is the branch cut of the square root, $\Xi = (-\infty, 0]$. In this case it can be shown that there exist asymptotically optimal sequences of interpolation nodes σ_j in Σ and poles ξ_j in Ξ, so-called *Leja–Bagby points* [3], such that the resulting rational interpolants $Q_N(\lambda)$ converge considerably faster than (2.2), namely, like

$$\limsup_{N\to\infty} \|A(\lambda) - Q_N(\lambda)\|_\Sigma^{1/N} \le \exp(-1/\mathrm{cap}(\Sigma, \Xi)) \lesssim \exp(-\pi^2/\log(16\kappa)), \tag{2.3}$$

with $\mathrm{cap}(\Sigma, \Xi)$ denoting a logarithmic capacity; see section 5 for details. In Figure 1(a) we show a numerical comparison of polynomial and rational interpolation for (2.1), and in Figure 1(b) the convergence of the resulting root-finding iterations.



(a) uniform interpolation error (b) convergence of residuals

FIG. 1. *Scalar NLEP* (2.1). (a) *Uniform error of polynomial interpolation at Chebyshev and Leja points on* Σ*, as well as linear rational interpolation at Leja–Bagby points. The curve for Chebyshev interpolation is only shown for reference; we do not actually use it in the algorithms. The predicted convergence slopes given by* (2.2) *and* (2.3) *are indicated.* (b) *Convergence history for the* 3 *real roots on* Σ *computed with the rational Krylov methods using polynomial interpolation (dashed blue) and linear rational interpolation (solid red).*

3. Linearization of the NLEP. In order to solve the NLEP (1.1), we will approximate $A(\lambda)$ by a rational function $Q_N(\lambda)$, resulting in a rational eigenvalue problem. The approximation will be constructed with a linear rational interpolation procedure to be detailed in section 3.1. In section 3.2 we show how the resulting rational eigenvalue problem is equivalent to a generalized eigenvalue problem in companion form.

3.1. Linear rational interpolation. Given a sequence of interpolation nodes $\sigma_0, \sigma_1, \ldots$, and another sequence of nonzero poles $\xi_1, \xi_2, \ldots$, we consider a sequence of *rational basis functions*

$$b_j(\lambda) = \frac{1}{\beta_0} \prod_{k=1}^{j} \frac{\lambda - \sigma_{k-1}}{\beta_k(1 - \lambda/\xi_k)}, \quad j = 0, 1, \ldots, \tag{3.1}$$

where the numbers $\beta_0, \beta_1, \ldots$ are nonzero scaling parameters to be specified later. Note that if all poles ξ_j are at infinity, then the functions $b_j(\lambda)$ reduce to Newton basis polynomials. Also note that there is a trivial recursion

$$b_0(\lambda) = \frac{1}{\beta_0}, \qquad b_{j+1}(\lambda) = \frac{(\lambda - \sigma_j)}{\beta_{j+1}(1 - \lambda/\xi_{j+1})} b_j(\lambda), \quad j = 0, 1, \ldots. \tag{3.2}$$

Our aim is to compute a sequence of matrices $D_j \in \mathbb{C}^{n \times n}$, $j = 0, 1, \ldots$, such that for each $N = 0, 1, \ldots$ the rational eigenvalue problem

$$Q_N(\lambda) = b_0(\lambda)D_0 + b_1(\lambda)D_1 + \cdots + b_N(\lambda)D_N \tag{3.3}$$

interpolates $A(\lambda)$ in Hermite's sense (that is, counting multiplicities) at the nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$. Note that the poles $\xi_1, \xi_2, \ldots, \xi_N$ are prescribed and we will assume that they are all distinct from the nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$. In this case, $Q_N(\lambda)$ is a linear rational interpolant of type $[N, N]$ and hence guaranteed to exist uniquely. In particular, if $A(\lambda)$ itself is a rational eigenvalue problem of type $[N, N]$ with poles $\xi_1, \xi_2, \ldots, \xi_N$, then the interpolant $Q_N(\lambda) = A(\lambda)$ will be exact.

In what follows we will refer to the D_j as *(rational) divided difference matrices*, as they become the standard Newton divided differences in the case of polynomial interpolation (that is, when all $\xi_j = \infty$). A straightforward way for computing these matrices D_j when all the interpolation nodes σ_j are distinct can be derived from (3.3) and (3.1). By the interpolation condition $Q_0(\sigma_0) = A(\sigma_0)$ and the formula for $b_0(\lambda)$ we have $D_0 = \beta_0 A(\sigma_0)$. From the interpolation conditions $A(\sigma_j) = Q_j(\sigma_j)$, $j \geq 1$, we then find recursively

$$D_j = \frac{A(\sigma_j) - b_0(\sigma_j)D_0 - \cdots - b_{j-1}(\sigma_j)D_{j-1}}{b_j(\sigma_j)} = \frac{A(\sigma_j) - Q_{j-1}(\sigma_j)}{b_j(\sigma_j)}.$$

Note that the matrix-valued numerator in D_j can be evaluated via the Horner scheme starting with the coefficient D_{j-1}, using the fact that each $b_{j-1}(\lambda)$ divides $b_j(\lambda)$. Computing the matrices D_j this way is mathematically equivalent to computing the diagonal entries of a divided-difference tableau with matrix entries. In the confluent case, when some of the interpolation nodes coincide, derivatives of $A(\lambda)$ will enter the formulas. We will not discuss this further to keep the presentation simple. Instead we will show that the confluent case can often be handled conveniently using matrix functions.

In situations where $A(\lambda)$ is given explicitly in the form of a linear combination involving scalar functions $f_i(\lambda)$ and constant matrices C_i,

$$A(\lambda) = f_1(\lambda)C_1 + f_2(\lambda)C_2 + \cdots + f_m(\lambda)C_m,$$

it suffices to compute the scalar divided differences $d_{i,j}$ of all $f_i(\lambda)$ such that

$$q_{i,N}(\lambda) = d_{i,0}b_0(\lambda) + d_{i,1}b_1(\lambda) + \cdots + d_{i,N}b_N(\lambda), \quad i = 1, 2, \ldots, m,$$

satisfy the prescribed interpolation conditions. The divided differences of $A(\lambda)$ are then given by linearity as

$$D_j = \sum_{i=1}^{m} d_{i,j}C_i, \qquad j = 0, 1, \ldots. \tag{3.4}$$

The following theorem is inspired by the rational interpolation procedure underlying the "poles and interpolation nodes (PAIN) method" for matrix function approximation; see [15, section 5.4.2] and [16]. It also contains as a special case (when all $\xi_j = \infty$) a result of Opitz [30] about the relation between matrix functions and (Newton) divided differences.

THEOREM 3.1. *Let $f(\lambda)$ be a scalar function, and let $q_N(\lambda) = d_0b_0(\lambda) + d_1b_1(\lambda) + \cdots + d_Nb_N(\lambda)$ be its rational interpolant of type $[N, N]$ with prescribed interpolation nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$ and poles $\xi_1, \xi_2, \ldots, \xi_N$. Then the rational divided differences $d_0, d_1, \ldots, d_N$ can be computed as*

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_N \end{bmatrix} = f\big(H_N K_N^{-1}\big)(\beta_0 e_1), \tag{3.5}$$

where $f(H_N K_N^{-1})$ is a matrix function, $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{N+1}$, and K_N and H_N

are the upper $(N+1)\times(N+1)$ *parts of the* $(N+2)\times(N+1)$ *matrices,*

$$\underline{K_N} = \left[\begin{array}{cccc} 1 & & & \\ \beta_1/\xi_1 & 1 & & \\ & \beta_2/\xi_2 & \ddots & \\ & & \ddots & 1 \\ \hline & & & \beta_{N+1}/\xi_{N+1} \end{array}\right] \quad \text{and} \quad \underline{H_N} = \left[\begin{array}{cccc} \sigma_0 & & & \\ \beta_1 & \sigma_1 & & \\ & \beta_2 & \ddots & \\ & & \ddots & \sigma_N \\ \hline & & & \beta_{N+1} \end{array}\right].$$

Proof. Define the matrix $V_{N+1}(\lambda) = [b_0(\lambda), b_1(\lambda), \ldots, b_N(\lambda), b_{N+1}(\lambda)]$. With this matrix it can easily be verified by multiplication that

$$\lambda V_{N+1}(\lambda)\underline{K_N} = V_{N+1}(\lambda)\underline{H_N},$$

which is a so-called *rational Krylov decomposition* in the sense defined in [15, Definition 5.5]. The entries $b_j(\lambda)$ in $V_{N+1}(\lambda)$ are rational functions of type $[N+1, N+1]$ with the common denominator $p_N(\lambda) := (1-\lambda/\xi_1)(1-\lambda/\xi_2)\cdots(1-\lambda/\xi_{N+1})$. Therefore the columns of $V_{N+1}(\lambda)$ span a linear space of rational functions. It has been shown in [15, Theorem 5.8] that

$$q_N(\lambda) = V_N(\lambda) f\big(H_N K_N^{-1}\big)(\beta_0 e_1)$$

is a rational function of type $[N, N]$ with denominator $p_N(\lambda)$, interpolating in Hermite's sense the function $f(\lambda)$ at the eigenvalues of $H_N K_N^{-1}$. The eigenvalues of $H_N K_N^{-1}$ are precisely the nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$, which concludes the proof. ∎

3.2. Rational companion linearization. The following theorem generalizes Theorem 3.1 and formula (3.21) in [1] from polynomial to linear rational interpolation.

THEOREM 3.2. *Given a rational eigenvalue problem of the form* (3.3)*, with the rational functions* $b_j(\lambda)$ *defined in* (3.1)*, and matrices* $D_j \in \mathbb{C}^{n\times n}$*. Then the linear pencil*

$$\boldsymbol{L}_N(\lambda) = \boldsymbol{A}_N - \lambda \boldsymbol{B}_N, \tag{3.6}$$

where

$$\boldsymbol{A}_N = \begin{bmatrix} D_0 & D_1 & \cdots & D_{N-2} & D_{N-1} - \sigma_{N-1} D_N/\beta_N \\ \sigma_0 I & \beta_1 I & & & \\ & \ddots & \ddots & & \\ & & \sigma_{N-3} I & \beta_{N-2} I & \\ & & & \sigma_{N-2} I & \beta_{N-1} I \end{bmatrix}, \tag{3.7}$$

$$\boldsymbol{B}_N = \begin{bmatrix} D_0/\xi_N & D_1/\xi_N & \cdots & D_{N-2}/\xi_N & D_{N-1}/\xi_N - D_N/\beta_N \\ I & \beta_1/\xi_1 I & & & \\ & \ddots & \ddots & & \\ & & I & \beta_{N-2}/\xi_{N-2} I & \\ & & & I & \beta_{N-1}/\xi_{N-1} I \end{bmatrix}, \tag{3.8}$$

is a strong linearization of $Q_N(\lambda)$*. Next, if* $(\lambda_\star, x)$ *is an eigenpair of* $Q_N(\lambda)$ *then* $(\lambda_\star, b(\lambda_\star)\otimes x)$ *with* $b(\lambda) := \begin{bmatrix} b_0(\lambda) & b_1(\lambda) & \cdots & b_{N-1}(\lambda) \end{bmatrix}^T$ *is an eigenpair of* $\boldsymbol{L}_N(\lambda)$*. Furthermore, if* $(\lambda_\star, \boldsymbol{y}_N)$ *is an eigenpair of* $\boldsymbol{L}_N(\lambda)$ *then there exists a vector* x *such that* $\boldsymbol{y}_N = b(\lambda_\star)\otimes x$ *and* $(\lambda_\star, x)$ *is an eigenpair of* $Q_N(\lambda)$*.*

Proof. The proof for a strong linearization is analogous to [1, Theorem 3.1] and the eigenpair equivalence between $Q_N(\lambda)$ and $\boldsymbol{L}_N(\lambda)$ follows from the identity $\boldsymbol{L}_N(\lambda)\,(b(\lambda) \otimes I) = e_1 \otimes Q_N(\lambda)$. ∎

Note that the last pole ξ_N plays a special role in Theorem 3.2. In what follows it will be convenient to choose $\xi_N = \infty$. In this case the linearization has the same structure as in the Newton-type companion form used in [36], and we can run exactly the same rational Krylov algorithm for the growing pencil $(\boldsymbol{A}_N, \boldsymbol{B}_N)$.

4. Fully rational Krylov methods for the NLEP. In this section we introduce a class of fully rational Krylov methods for NLEPs (1.1), abbreviated as NLEIGS. All these methods use the new rational companion linearization proposed in section 3.2. The main difference lies in the way the construction of the linear rational approximation of $A(\lambda)$ is connected with the rational Krylov iteration for computing the eigenpairs.

Once we have chosen N nodes and poles leading to the linearization pencil (3.6), we can compute eigenvalue estimates of the NLEP by solving $\boldsymbol{A}_N \boldsymbol{y}_N = \lambda \boldsymbol{B}_N \boldsymbol{y}_N$ using the rational Krylov method. This is the standard method for solving linear eigenvalue problems. This approach corresponds to the *discretize first and then solve approach*, which is advocated by [11]. We call it the *static NLEIGS variant*. The shifts of the rational Krylov space are chosen to solve this linear problem efficiently. They are not necessarily related to the interpolation nodes of the linearization. The difference with [35] is that NLEIGS does not require low-rank nonlinear terms and can adopt polynomials of high degree as numerical examples will illustrate. In addition, it will be shown further that NLEIGS can also greatly benefit from low-rank nonlinear terms as in [35].

On the other hand, when the shifts of the rational Krylov space are chosen equal to the interpolation nodes, we can use a dynamic variant of the method, i.e., the nodes and poles can be chosen dynamically during the execution of the NLEIGS method. This method we call the *dynamic NLEIGS variant* and is a direct generalization of the Newton rational Krylov method [36] from polynomial to rational interpolation. This method has the same computational cost per iteration as the Newton rational Krylov method, but it may require considerably fewer iterations. The new method uses the companion-type linearization of Theorem 3.2. Similarly to [36], we choose the shifts of the rational Krylov space equal to the interpolation nodes of the interpolant $Q_N(\lambda)$ defined in (3.3). We also use a specific starting vector. This allows us to dynamically expand $Q_N(\lambda)$ and to exploit the structure of the generalized eigenvalue problem $\boldsymbol{A}_N \boldsymbol{y}_N = \lambda \boldsymbol{B}_N \boldsymbol{y}_N$.

In the remainder of this section we describe the dynamic variant in more detail: in section 4.1 we outline some properties of the associated rational Krylov space and in section 4.2 we describe the algorithm. Some of the proofs are very similar or identical to those in [36, section 4.1] and we will often refer to this paper.

4.1. Building the rational Krylov space. We start with the following lemma.

LEMMA 4.1. *Let $\boldsymbol{A}_N$ and $\boldsymbol{B}_N$ be defined by* (3.7)–(3.8), *setting $\xi_N = \infty$, and*

$$\boldsymbol{y}_j = \operatorname{vec}\left(y_j^{[1]}, y_j^{[2]}, \ldots, y_j^{[j+1]}, 0, \ldots, 0\right),$$

where $\boldsymbol{y}_j \in \mathbb{C}^{Nn}$ and $y_j^{[i]} \in \mathbb{C}^n$ for $i = 1, \ldots, j+1$. Then for all j, $0 \le j \le N-2$, the solution $\boldsymbol{x}_j$ of the linear system of equations with shift σ_j,

$$(\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N)\boldsymbol{x}_j = \boldsymbol{y}_j, \tag{4.1}$$

is of the structure

$$\boldsymbol{x}_j = \mathrm{vec}\Big(x_j^{[1]},\, x_j^{[2]},\, \ldots,\, x_j^{[j+1]},\, 0,\, \ldots,\, 0\Big),$$

where $\boldsymbol{x}_j \in \mathbb{C}^{Nn}$ *and* $x_j^{[i]} \in \mathbb{C}^n$ *for* $i = 1, \ldots, j+1$.

Proof. This lemma follows directly from the fact that the matrix $\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N$ in (4.1) is block triangular. ∎

The main difference to a standard rational Krylov method (see [32, 33]) is that the shifts are not free parameters, but they are implicitly prescribed by the matrices $\boldsymbol{A}_N$ and $\boldsymbol{B}_N$ in (3.7)–(3.8), namely, the nodes $\sigma_1, \sigma_2, \ldots$. We need the following definition.

DEFINITION 4.2. *Let* $\boldsymbol{A}_N$ *and* $\boldsymbol{B}_N$ *be given by* (3.7) *and* (3.8)*, respectively. Given the* shifts $\sigma_1, \sigma_2, \ldots, \sigma_{k-1}$*, we define by*

$$\begin{aligned}\mathcal{Q}_k &:= \mathrm{span}\big\{\boldsymbol{v}_1,\, (\boldsymbol{A}_N - \sigma_1 \boldsymbol{B}_N)^{-1}\boldsymbol{B}_N\boldsymbol{v}_1,\, (\boldsymbol{A}_N - \sigma_2 \boldsymbol{B}_N)^{-1}\boldsymbol{B}_N\boldsymbol{v}_2,\, \ldots \\ &\qquad\qquad \ldots,\, (\boldsymbol{A}_N - \sigma_{k-1} \boldsymbol{B}_N)^{-1}\boldsymbol{B}_N\boldsymbol{v}_{k-1}\big\} \\ &:= \mathrm{span}\{\boldsymbol{v}_1,\, \boldsymbol{v}_2,\, \boldsymbol{v}_3,\, \ldots,\, \boldsymbol{v}_k\}\end{aligned} \tag{4.2}$$

the rational Krylov space *constructed with the matrices* $\boldsymbol{A}_N$*,* $\boldsymbol{B}_N$ *and the starting vector* $\boldsymbol{v}_1 \in \mathbb{C}^{Nn}$.

We assume that the starting vector $\boldsymbol{v}_1$ and the shifts $\sigma_1, \ldots, \sigma_{k-1}$ are chosen such that $\mathcal{Q}_k$ is of dimension $k \le N$. The vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ will be orthonormalized using a rational Krylov sequence method [32, 33]; see also Algorithm 1. The special structure of the matrices $\boldsymbol{A}_N$ and $\boldsymbol{B}_N$ can be exploited by choosing the particular starting vector

$$\boldsymbol{v}_1 := \mathrm{vec}\left(v_1^{[1]},\, 0,\, 0, \ldots,\, 0\right) \quad \text{with} \quad v_1^{[1]} \in \mathbb{C}^n. \tag{4.3}$$

The advantageous consequences of choosing (4.3) as starting vector for (4.2) are summarized in the following lemmas.

LEMMA 4.3. *Suppose that a starting vector* $\boldsymbol{v}_1$ *of the form* (4.3) *is used for the rational Krylov method. Then for all* j*,* $1 \le j \le N-1$*, the vector*

$$\boldsymbol{v}_{j+1} = (\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N)^{-1}\boldsymbol{B}_N\boldsymbol{w}_j, \tag{4.4}$$

where $\boldsymbol{w}_j = \boldsymbol{V}_j t_j$*, is of the structure*

$$\boldsymbol{v}_{j+1} = \mathrm{vec}\left(v_{j+1}^{[1]},\, v_{j+1}^{[2]}, \ldots,\, v_{j+1}^{[j+1]},\, 0, \ldots,\, 0\right) \quad \textit{with} \quad v_{j+1}^{[i]} \in \mathbb{C}^n \ (i = 1, \ldots, j+1).$$

Proof. The proof is analogous to the one of Lemma 4.3 in [36]. ∎

LEMMA 4.4. *Let the rational Krylov space* $\mathcal{Q}_k$ *be constructed as in Definition* 4.2 *and Lemma* 4.3*. Then at each iteration* j *of the rational Krylov method only the upper-left parts of the matrices* $\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N$ *are used to compute the leading nonzero parts of the vectors* $\boldsymbol{v}_{j+1}$*, i.e.,*

$$(\boldsymbol{A}_j - \sigma_j \boldsymbol{B}_j)\widetilde{\boldsymbol{v}}_{j+1} = \boldsymbol{B}_j \widetilde{\boldsymbol{w}}_j, \tag{4.5}$$

where

$$\widetilde{\boldsymbol{v}}_{j+1} = \mathrm{vec}\left(v_{j+1}^{[1]},\, v_{j+1}^{[2]}, \ldots,\, v_{j+1}^{[j+1]}\right) \quad \textit{and} \quad \widetilde{\boldsymbol{w}}_j = \mathrm{vec}\left(w_j^{[1]},\, w_j^{[2]}, \ldots,\, w_j^{[j]},\, 0\right).$$

Proof. The proof is a direct consequence of Lemma 4.1 and Lemma 4.3. ∎

At each iteration j of the rational Krylov method, following Lemma 4.4, we only have to solve system (4.5) of dimension $(j+1)n \times (j+1)n$, instead of (4.4), which is of dimension $Nn \times Nn$. This results in a significant reduction of computational cost. In fact, we can formally set $N = \infty$ because Lemma 4.4 tells us that at any finite iteration number j only a finite leading principal submatrix of $\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N$ is required. The companion-type form of the matrix $\boldsymbol{A}_j - \sigma_j \boldsymbol{B}_j$ can be exploited further to efficiently solve (4.5) using operations on $n \times n$ sparse matrices only.

LEMMA 4.5. *The linear system* (4.5) *can be solved using the equations*

$$A(\sigma_j) v_{j+1}^{[1]} = z_0^{(j)},$$

where

$$z_0^{(j)} = -\frac{1}{\beta_0} \sum_{i=1}^{j} D_i \sum_{k=1}^{i} \frac{\prod_{\ell=k}^{i-1} (\sigma_j - \sigma_\ell)}{\prod_{\ell=k}^{i} \beta_\ell (1 - \sigma_j/\xi_\ell)} \left(w_j^{[k]} + \frac{\beta_k}{\xi_k} w_j^{[k+1]} \right)$$

with $\prod_{\ell=i}^{i-1}(\cdot) := 1$ *and* $w_j^{[j+1]} := 0$, *and*

$$v_{j+1}^{[k]} = \frac{\mu_{k-2}^{(j)}}{\nu_{k-1}^{(j)}} v_{j+1}^{[k-1]} + \frac{1}{\nu_{k-1}^{(j)}} \left(w_j^{[k-1]} + \frac{\beta_{k-1}}{\xi_{k-1}} w_j^{[k]} \right)$$

for $k = 2, \ldots, j+1$.

Proof. The proof is analogous to the one of Lemma 4.5 in [36]. ∎

COROLLARY 4.6. *At each iteration j of the rational Krylov method one only has to perform an LU factorization of $A(\sigma_j) \in \mathbb{C}^{n \times n}$, instead of an LU factorization of $\boldsymbol{A}_N - \sigma_j \boldsymbol{B}_N \in \mathbb{C}^{Nn \times Nn}$. With repeated interpolation nodes σ_j one can reuse LU factors for several iterations.*

PROPOSITION 4.7. *The Ritz values $\lambda_i^{(j)}$, computed at iteration j of the rational Krylov method (see section* 4.2 *and Algorithm* 1*), are independent of N as long as $j < N$. These Ritz values are also independent of $\sigma_{j+1}, \ldots, \sigma_N$ and $\xi_{j+1}, \ldots, \xi_N$.*

Proof. At iteration j, the Ritz values are computed from the upper $j \times j$ parts of two Hessenberg matrices obtained by orthogonalization of $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j+1}$. Following Lemmas 4.3–4.5, only the first $j+1$ nodes $\sigma_0, \ldots, \sigma_j$, and the first j poles $\xi_1, \ldots, \xi_j$, are used for the construction of the rational Krylov vectors $\boldsymbol{V}_{j+1}$. Therefore the approximated eigenvalues are independent of the nodes $\sigma_{j+1}, \ldots, \sigma_N$ and the poles $\xi_{j+1}, \ldots, \xi_N$. Hence they are also independent of N. ∎

Remark 4.8. It is neither necessary to choose all the nodes σ_j and poles ξ_j in advance, nor the degree N of the rational interpolant $Q_N(\lambda)$. Instead, at each iteration we can choose the next node and pole based on information of the previous iterations. Therefore, the rational Krylov method can be implemented in an adaptive and incremental way. The rational Krylov method is initialized with a node $\sigma_0 \in \Sigma$ and a particular starting vector, and can run until convergence by dynamically adding a node σ_j and pole ξ_j at each iteration.

4.2. Algorithm. Based on the previous section, our dynamic NLEIGS algorithm for solving the NLEP (1.1) can be implemented efficiently. Algorithm 1 gives an outline. Similar to [36, Algorithm 2], we can subdivide each iteration j of NLEIGS into two phases: an expansion phase and a rational Krylov phase.

First, in the expansion phase (lines 2–4), we choose at iteration j the next interpolation node σ_j, pole ξ_j, and scaling parameter β_j. We then compute the corresponding rational divided difference D_j in order to extend the linearization matrices

ALGORITHM 1. NLEIGS: DYNAMIC VARIANT.

1 Choose node σ_0, scaling parameter $\beta_0 = 1$, and starting vector v_1.
for $j = 1, 2, \ldots$ **do**
  EXPANSION PHASE:
2 Choose node σ_j, pole ξ_j, and scaling parameter β_j.
3 Compute rational divided difference matrix D_j.
4 Expand $\boldsymbol{A}_j$, $\boldsymbol{B}_j$, and $\boldsymbol{V}_j$.
  RATIONAL KRYLOV PHASE:
5 Compute $\boldsymbol{w} := (\boldsymbol{A}_j - \sigma_j \boldsymbol{B}_j)^{-1} \boldsymbol{B}_j \boldsymbol{v}_j$.
6 Orthogonalize $\boldsymbol{w} := \boldsymbol{w} - \boldsymbol{V}_j h_j$, where $h_j = \boldsymbol{V}_j^* \boldsymbol{w}$.
7 Get new basis vector $\boldsymbol{v}_{j+1} = \boldsymbol{w}/h_{j+1,j}$, where $h_{j+1,j} = \|\boldsymbol{w}\|$.
8 Compute Ritz pairs $(\lambda_i, \boldsymbol{u}_i := \boldsymbol{V}_{j+1} \underline{H_j} s_i)$.
9 Nonlinear eigenpairs $\left(\lambda_i, u_i^{[1]}\right)$ and test for convergence.
end

$\boldsymbol{A}_{j-1}$ and $\boldsymbol{B}_{j-1}$ to $\boldsymbol{A}_j$ and $\boldsymbol{B}_j$, respectively. We also extend the matrix $\boldsymbol{V}_j$ with a zero block at the bottom.

Second, in the rational Krylov phase, we compute a new basis vector of the rational Krylov space (line 5). This vector is then orthogonalized against $\boldsymbol{V}_j$ and normalized in order to append the rational Krylov basis with $\boldsymbol{v}_{j+1}$ (lines 6–7). In line 8, the Ritz values are computed as eigenvalues of the low-dimensional generalized eigenproblem

$$K_j s_i = \lambda_i H_j s_i, \qquad s_i \neq 0,$$

where H_j is the upper $j \times j$ part of the Hessenberg matrix $\underline{H_j}$ obtained from the orthogonalization process and $K_j = H_j \operatorname{diag}(\sigma_1, \ldots, \sigma_j) + I_{j+1,j}$. The Ritz vectors $\boldsymbol{u}_i$ are obtained by left multiplication of s_i with $\boldsymbol{V}_{j+1}\underline{H_j}$.

Finally, in line 9, we take the first blocks of $\boldsymbol{u}_i$ as approximations for the nonlinear eigenvectors and check for convergence of the nonlinear eigenpairs $(\lambda_i, u_i^{[1]})$. For more details we refer to [36, Algorithm 2].

5. The choice of parameters. Our algorithm requires a choice of the interpolation nodes σ_j, poles ξ_j, and scaling parameters β_j. The choice of the parameters β_j is dictated by numerical stability considerations and will be discussed in section 6.1.

Choosing the parameters σ_j and ξ_j in a (near) optimal way is closely related to rational approximation problems on the target set Σ. These problems are in turn very closely related to logarithmic potential theory; see [25, 34] for introductions. For the purpose of this paper we only focus on linear rational interpolation with *prescribed* poles and nodes (as opposed to interpolation with *free* poles, a well-known special case of which is Padé approximation). This is a classical problem that has been studied extensively since the late 1960s by Bagby [3], Walsh [38, 40, 39], and others.

Let us assume that $\Sigma \subset \mathbb{C}$ is a simply connected compact set, and that $A(\lambda) = [a_{i,j}(\lambda)]$ is analytic in a simply connected open set $\Omega \supset \Sigma$. Let $Q_N(\lambda)$ be a rational interpolant of $A(\lambda)$ with interpolation nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$ in Σ, and poles $\xi_1, \ldots, \xi_N$ outside Σ. Let each component of $Q_N(\lambda) = [q_{i,j}^{(N)}(\lambda)]$ have accuracy ε on Σ, i.e., $\max_{i,j} \|a_{i,j}(\lambda) - q_{i,j}^{(N)}(\lambda)\|_\Sigma \leq \varepsilon$. Assume further that (λ^*, x) with $\lambda^* \in \Sigma$ and $\|x\|_2 = 1$

is an eigenpair for $Q_N(\lambda)$, i.e., $Q_N(\lambda^*)x = 0$. Then from

$$\|A(\lambda^*)x\|_2 = \|(A(\lambda^*) - Q_N(\lambda^*))x\|_2 \leq \|A(\lambda^*) - Q_N(\lambda^*)\|_F \leq n\varepsilon \tag{5.1}$$

we find that a small component-w uniform error of $Q_N(\lambda)$ for $A(\lambda)$ implies that (λ^*, x) has a small residual for the original NLEP (1.1). To obtain a reliable algorithm which attempts to find *all* eigenvalues in Σ, it is sensible to sample $A(\lambda)$ at nodes σ_j that guarantee a component-wise small error ε in $Q_N(\lambda)$ for all $\lambda \in \Sigma$.

In what follows we will drop the element indices from $a_{i,j}(\lambda) = a(\lambda)$ and $q_{i,j}^{(N)}(\lambda) = q^{(N)}(\lambda)$. The uniform interpolation error $\|a(\lambda) - q^{(N)}(\lambda)\|_\Sigma$ can be studied conveniently using the nodal rational functions

$$s_j(\lambda) = \frac{(\lambda - \sigma_0)(\lambda - \sigma_1)\cdots(\lambda - \sigma_j)}{(1 - \lambda/\xi_1)\cdots(1 - \lambda/\xi_j)}, \quad j = 0, 1, \ldots.$$

Let Γ be a rectifiable closed curve in $\Omega \setminus \Sigma$, winding around Σ exactly once. Then by the Walsh–Hermite integral representation of the interpolation error (see, e.g., [39, p. 50]) and standard estimation of integrals we have for all $\lambda \in \Sigma$

$$|a(\lambda) - q^{(N)}(\lambda)| = \left| \frac{1}{2\pi i} \int_\Gamma \frac{s_N(\lambda)}{s_N(\zeta)} \frac{a(\zeta)}{(\zeta - \lambda)} \, \mathrm{d}\zeta \right| \leq C \frac{|s_N(\lambda)|}{\min_{\zeta \in \Gamma} |s_N(\zeta)|} \tag{5.2}$$

for a constant C that only depends on Γ and $a(\lambda)$. The pair (Σ, Γ) is called a *condenser* [2, 13]. It can be shown [13, 24] that there exists a number $\mathrm{cap}(\Sigma, \Gamma) > 0$, called the *condenser capacity* of (Σ, Γ), such that

$$\limsup_{N \to \infty} \left(\frac{\max_{\lambda \in \Sigma} |s_N(\lambda)|}{\min_{\lambda \in \Gamma} |s_N(\lambda)|} \right)^{1/N} \geq \exp(-1/\mathrm{cap}(\Sigma, \Gamma)) \tag{5.3}$$

with equality if the points σ_j and ξ_j are distributed according to the so-called *signed equilibrium measure* on (Σ, Γ). A sequence of points that follow this distribution are the *Leja–Bagby points* for (Σ, Γ) [38, 3], which can be constructed as follows: start with an arbitrary $\sigma_0 \in \Sigma$, and then define the nodes $\sigma_j \in \Sigma$ and poles $\xi_j \in \Gamma$ recursively such that the following conditions are satisfied:

$$\max_{\lambda \in \Sigma} |s_j(\lambda)| = |s_j(\sigma_{j+1})| \quad \text{and} \quad \inf_{\lambda \in \Gamma} |s_j(\lambda)| = |s_j(\xi_{j+1})|, \quad j = 0, 1, \ldots.$$

By the maximum modulus principle for analytic functions, the points σ_j lie on $\partial\Sigma$, the boundary of Σ, and Γ can be replaced by its closed exterior, say Ξ, without changing the capacity of $\mathrm{cap}(\Sigma, \Gamma) = \mathrm{cap}(\Sigma, \Xi)$. Combining the inequality (5.2) and (5.3) (with equality), we arrive at the asymptotic convergence result

$$\limsup_{N \to \infty} \|A(\lambda) - Q_N(\lambda)\|_\Sigma^{1/N} \leq \exp(-1/\mathrm{cap}(\Sigma, \Xi))$$

for linear rational interpolation at Leja–Bagby points. The convergence is thus geometric with a rate depending on the target set Σ and the poles on Ξ, which should stay away from Σ. In our numerical experiments we will typically use for Ξ (a discretization of) the singularity set of $f(\lambda)$. The determination of the numerical value $\mathrm{cap}(\Sigma, \Xi)$ is difficult for general condensers (Σ, Ξ). However, in some cases, including the example $(\Sigma, \Xi) = \big([\alpha, \beta], (-\infty, 0]\big)$ from the introduction, there are known closed formulas derived from conformal maps; see [16] for some examples, including the formula (2.3).

6. Computational considerations. In this section we will discuss various ingredients of an efficient and reliable computer realization of our algorithm. We have implemented these techniques in a MATLAB code `nleigs`.

6.1. Scaling. Once the points σ_j and ξ_j have been specified, as described in section 5, it remains to choose appropriate scaling parameters β_j. Note that, for a given fixed linearization $Q_N(\lambda)$ of $A(\lambda)$, changing a parameter β_j to $\alpha\beta_j$ has no influence other than scaling the divided difference matrices $D_j, D_{j+1}, \ldots$ to $\alpha D_j, \alpha D_{j+1}, \ldots$. Although scaling has no effect on the eigenvalues of the linearization $(\boldsymbol{A}_N, \boldsymbol{B}_N)$ in Theorem 3.2 in exact arithmetic, the following choice of β_j can dramatically improve both the stability and the convergence of the rational Krylov method for finding these eigenvalues: choose the β_j such that all $b_j(\lambda)$ defined in (3.1) are (approximately) of unit uniform norm on Σ, i.e., $\|b_j(\lambda)\|_\Sigma = 1$. This choice also guarantees that the evaluation of the functions $b_j(\lambda)$ during the sampling procedure is robust in the sense that it is not affected by numerical overflow or underflow.

This choice is motivated by the well-known fact that for polynomial interpolation at Leja points the set Σ should be scaled to unit capacity for stability [31] or, alternatively, we can scale the Newton basis polynomials at each iteration. We are following the second approach, with the difference that we are now dealing with rational functions instead of polynomials. Our scaling also seems natural by inspecting the block components of the eigenvectors of the linearization in Theorem 3.2. If all $b_j(\lambda)$ have the same order of magnitude on Σ, then the eigenvectors corresponding to eigenvalues λ in the target set Σ have evenly balanced block entries $b_j(\lambda)x$.

Under the condition that the poles ξ_j are away from Σ, the value $\|b_j(\lambda)\|_\Sigma$ is attained on the boundary $\Gamma = \partial\Sigma$ (see section 5). In order to practically implement the scaling procedure, we only need a sufficiently fine discretization $\Gamma_\nu = \{\gamma_1, \ldots, \gamma_\nu\}$ of Γ, and then choose each β_j such that $\max_{\lambda\in\Gamma_\nu} |b_j(\lambda)| = 1$. In our numerical experiments we evaluated each $b_j(\lambda)$ at $\nu = 1000$ equispaced control points, a scalar computation which is of negligible constant cost when using the recursion (3.2).

6.2. Truncating the expansion. The scaling such that $\|b_j(\lambda)\|_\Sigma = 1$ for all j is also convenient for error estimation: from a convergent expansion $Q_j(\lambda)$ of $A(\lambda)$ we find $\max_{\lambda\in\Sigma} \|A(\lambda) - Q_j(\lambda)\|_F \le \|D_{j+1}\|_F + \|D_{j+2}\|_F + \cdots$. Since the divided differences D_j are computed as in (3.4), we use the scalar divided differences $d_{i,j}$ for checking the accuracy of the expansion

$$\delta_j := \max_i |d_{i,j}|.$$

This scalar quantity is readily available in our algorithm and can be computed accurately via matrix functions. Assume that we wish to compute eigenpairs (λ^*, x) with $\lambda^* \in \Sigma$ and $\|x\|_2 = 1$ such that $\|A(\lambda^*)x\|_2 \le$ `tol`, with a user-specified residual tolerance `tol`. Then we will stop expanding $Q_j(\lambda)$ if the last ℓ quantities $\delta_j \le$ `tol`$/10$, where 10 is a safety factor. In our experiments we found that $\ell = 5$ is a practical choice.

6.3. Residual stopping criterion. In our numerical experiments reported in section 7 we have used a very simple stopping criterion for NLEIGS. First of all, we will allow the user to specify a maximal iteration number, with 250 as default. While the iteration is running, we test for the accuracy of the expansion $Q_j(\lambda)$ using the estimator δ_j described in section 6.2. We will never stop iterating as long as $\delta_j >$ `tol`$/10$, because in this case $Q_j(\lambda)$ cannot be accepted as a sufficiently accurate

approximation of $A(\lambda)$. Otherwise, we perform at least a minimal number of extra iterations, with 20 as default. Next, we stop the iteration if all Ritz pairs $(\lambda_i, u_i^{[1]})$ with $\lambda_i \in \Sigma$ have a residual norm below `tol`.

6.4. Exploiting low-rank structure. In several applications the NLEP (1.1) consists of a polynomial part and a nonlinear part which is of low rank. For self-containedness of the paper, we now review and generalize the exploitation of low-rank structure in the coefficient matrices [36, section 4.4].

Suppose that the NLEP is defined as follows:

$$A(\lambda)x = \left(\sum_{i=0}^{p} B_i\lambda^i + \sum_{i=1}^{m} C_i f_i(\lambda)\right) x = 0, \tag{6.1}$$

where $B_i, C_i \in \mathbb{C}^{n\times n}$ are constant matrices, $f_i(\lambda)$ are scalar functions of λ, $p \ll n^2$, and $m \ll n^2$. Furthermore, we assume that the matrices C_i have rank-revealing factorizations $C_i = L_i U_i^*$, where $L_i, U_i \in \mathbb{C}^{n\times r_i}$ are of full column rank $r_i \ll n$.

Approximating the scalar functions $f_i(\lambda)$ of (6.1) by linear rational interpolants with nodes $\sigma_0, \sigma_1, \ldots, \sigma_N$ and poles $\xi_1, \xi_2, \ldots, \xi_N$ yields

$$\widetilde{Q}_N(\lambda) = \sum_{i=0}^{N} \widetilde{D}_i b_i(\lambda) = \sum_{i=0}^{p} \left(\widetilde{B}_i + \widetilde{C}_i\right) b_i(\lambda) + \sum_{i=p+1}^{N} \widetilde{C}_i b_i(\lambda), \tag{6.2}$$

where

$$\widetilde{B}_i = \sum_{j=0}^{p} \beta_{ij} B_j \quad \text{and} \quad \widetilde{C}_i = \sum_{j=1}^{m} \gamma_{ij} C_j = \sum_{j=1}^{m} \gamma_{ij} L_j U_j^*,$$

with scalars β_{ij} and γ_{ij}. Define

$$\widetilde{L}_i = \begin{bmatrix} \gamma_{i1}L_1 & \gamma_{i2}L_2 & \cdots & \gamma_{im}L_m \end{bmatrix} \quad \text{and} \quad \widetilde{U} = \begin{bmatrix} U_1 & U_2 & \cdots & U_m \end{bmatrix},$$

where the size of $\widetilde{L}_i$ and $\widetilde{U}$ is $n \times r$ and $r = r_1 + r_2 + \cdots + r_m$. Similarly as in Theorem 3.2, we obtain a companion-type reformulation, where the pair $(\lambda, x \neq 0)$ is an eigenpair of the rational eigenvalue problem (6.2) if and only if

$$\widetilde{\boldsymbol{A}}_N \widetilde{\boldsymbol{y}}_N = \lambda \widetilde{\boldsymbol{B}}_N \widetilde{\boldsymbol{y}}_N,$$

where

$$\widetilde{\boldsymbol{A}}_N = \begin{bmatrix} \widetilde{B}_0 + \widetilde{C}_0 & \widetilde{B}_1 + \widetilde{C}_1 & \ldots & \widetilde{B}_p + \widetilde{C}_p & \widetilde{L}_{p+1} & \widetilde{L}_{p+2} & \ldots \\ \sigma_0 I & \beta_1 I & & & & & \\ & \ddots & \ddots & & & & \\ & & \sigma_{p-1} I & \beta_p I & & & \\ & & & \sigma_p \widetilde{U}^* & \beta_{p+1} I & & \\ & & & & \sigma_{p+1} I & \beta_{p+2} I & \\ & & & & & \ddots & \ddots \end{bmatrix},$$

$$\widetilde{\boldsymbol{B}}_N = \begin{bmatrix} (\widetilde{B}_0+\widetilde{C}_0)/\xi_N & (\widetilde{B}_1+\widetilde{C}_1)/\xi_N & \dots & (\widetilde{B}_p+\widetilde{C}_p)/\xi_N & \widetilde{L}_{p+1}/\xi_N & \widetilde{L}_{p+2}/\xi_N & \dots \\ I & \beta_1/\xi_1 I & & & & & \\ & \ddots & \ddots & & & & \\ & & I & \beta_p/\xi_p I & & & \\ & & & \widetilde{U}^* & \beta_{p+1}/\xi_{p+1} I & & \\ & & & & I & \beta_{p+2}/\xi_{p+2} I & \\ & & & & & \ddots & \ddots \end{bmatrix},$$

and

$$\widetilde{\boldsymbol{y}}_N = \mathrm{vec}\big(b_0(\lambda)x,\, b_1(\lambda)x,\, \dots,\, b_p(\lambda)x,\, b_{p+1}(\lambda)\widetilde{U}^*x,\, b_{p+2}(\lambda)\widetilde{U}^*x,\, \dots\big).$$

For this type of linearization we can also prove Lemmas 4.1–4.5.

7. Numerical experiments. In the introduction we demonstrated NLEIGS as a root finder for a scalar problem and illustrated the differences between polynomial and rational interpolation in Algorithm 1. We now apply NLEIGS to two large-scale problems. For the `'gun'` problem, we first discuss polynomial versus rational interpolation and compare it to the Newton rational Krylov method. We then illustrate the use of Leja–Bagby points in Algorithm 1 and discuss the truncation strategy proposed in section 6.2. Finally, for an application from physics, we use NLEIGS for computing real eigenvalues very close to singularities.

All numerical experiments are performed in MATLAB version 7.14.0 (R2012a) on a Dell Latitude notebook running an Intel(R) Core(TM) i5-2540M CPU @ 2.60GHz quad core processor with 4 GB RAM. Our experiments can be reproduced with the NLEIGS code available from

http://twr.cs.kuleuven.be/research/software/nleps/nleigs.html

The syntax of NLEIGS resembles that of the sparse eigensolver `eigs` of MATLAB: in its simplest version we call `[V,D] = nleigs(A,Sigma,Xi)`, where `A` is a structure representing $A(\lambda)$, `Sigma` is a vector of vertices of a polygonal target set Σ, and `Xi` is a discretization of the singularity set Ξ.

Before presenting the results of the numerical experiments, we first describe four variants of Algorithm 1 to be compared below.

Variant P: a polynomial version of Algorithm 1, which is in fact the Newton rational Krylov method introduced in [36]. In this variant, $A(\lambda)$ is approximated by interpolating polynomials $P_j(\lambda)$ with cyclically repeated interpolation nodes $\sigma_j \in \Omega_{\mathrm{cycl}} \subset \Sigma$ and all poles $\xi_j = \infty$. The shifts of the rational Krylov space are chosen equal to the interpolation nodes σ_j in order to make the algorithm dynamic.

Variant R_1: a rational version of Algorithm 1, whereby $A(\lambda)$ is approximated by rational interpolants $Q_j(\lambda)$ with cyclically repeated interpolation nodes $\sigma_j \in \Omega_{\mathrm{cycl}}$ and poles $\xi_j \in \Xi$ selected in Leja–Bagby style. Again, the shifts of the rational Krylov space are chosen equal to the interpolation nodes σ_j.

Variant R_2: a rational version of Algorithm 1, whereby $A(\lambda)$ is also approximated by rational interpolants $Q_j(\lambda)$, but the selection of interpolation nodes and poles is different. As long as the interpolant $Q_j(\lambda)$ has not yet converged, we choose Leja–Bagby interpolation nodes $\sigma_j \in \Sigma$ and poles $\xi_j \in \Xi$. Upon convergence we truncate and freeze the rational expansion $Q_j(\lambda)$. Next, we switch to cyclically repeated interpolation nodes $\sigma_j \in \Omega_{\mathrm{cycl}}$ in order to obtain faster convergence of the rational Krylov method to eigenvalues located in the interior of Σ. The poles ξ_j are still selected in

Leja–Bagby style. The shifts of the rational Krylov space are again chosen equal to the interpolation nodes σ_j in order to make the algorithm dynamic.

Variant S: a static rational version of Algorithm 1. We first determine the rational approximation $Q_N(\lambda)$ such that $Q_N(\lambda) \approx A(\lambda)$ for all $\lambda \in \Sigma$. For the computation of $Q_N(\lambda)$ we select Leja–Bagby interpolation nodes $\sigma_j \in \Sigma$ and poles $\xi_j \in \Xi$, and use the same truncation criterion as for *Variant* R_2. Then, once the linearization is fixed, we use the rational Krylov method with shifts in Ω_{cycl} for solving the generalized eigenvalue problem in a nondynamic way.

7.1. Gun problem. We consider the 'gun' problem of the NLEVP collection [5] (see also [26]). This is a large-scale problem that models a radio-frequency gun cavity and is of the form

$$A(\lambda)x = \left(K - \lambda M + i\sqrt{\lambda - \sigma_1^2}\, W_1 + i\sqrt{\lambda - \sigma_2^2}\, W_2\right) x = 0, \tag{7.1}$$

where M, K, W_1, and W_2 are real symmetric matrices of size 9956×9956, K is positive semidefinite, and M is positive definite. As in [5], we take $\sigma_1 = 0$ and $\sigma_2 = 108.8774$. The complex square root $\sqrt{\cdot}$ corresponds to the principal branch. For measuring the convergence of an approximate eigenpair (λ, x), we used the relative residual norm defined in [27],

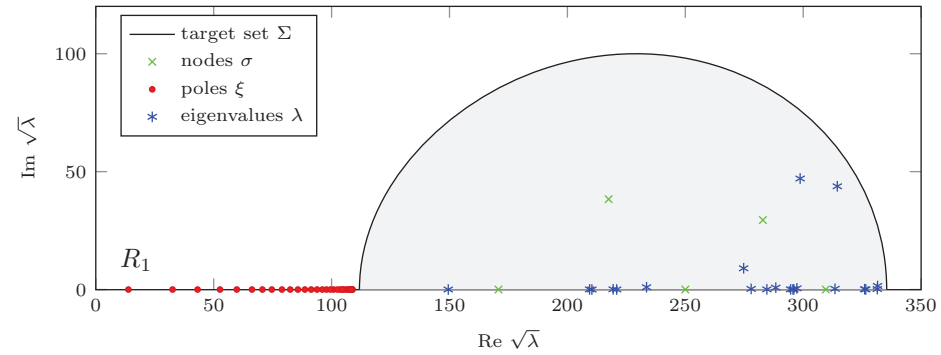
$$E(\lambda, x) = \frac{\|A(\lambda)x\|_2 / \|x\|_2}{\|K\|_1 + |\lambda|\,\|M\|_1 + \sqrt{|\lambda - \sigma_1^2|}\,\|W_1\|_1 + \sqrt{|\lambda - \sigma_2^2|}\,\|W_2\|_1}.$$

The target set Σ is the upper half-disk with center 250^2 and radius $300^2 - 200^2$; see Figure 2(a). The singularity set $\Xi = (-\infty, \sigma_2^2]$ corresponds to the union of branch cuts of the square roots. In our algorithm we have discretized Ξ by logarithmically spaced points $(\sigma_2^2 - 10^{-8+16j/10^4})$ with $j = 0, 1, \ldots, 10^4$. Thanks to the automatic scaling strategy described in section 6.1, we can solve the NLEP (7.1) directly with Algorithm 1, instead of first transforming Σ to roughly the upper half of the unit disk as in [21, 36]. We have also exploited the low-rank structure of the coefficient matrices W_1 and W_2 as explained in section 6.4.

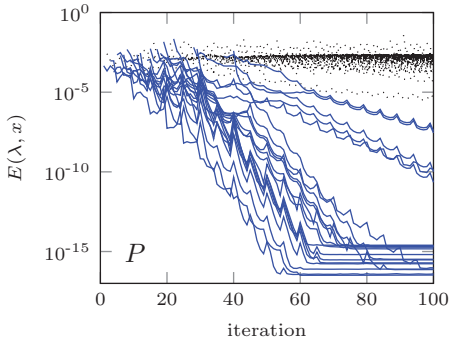
7.1.1. Polynomial (P) versus rational (R_1) interpolation. In a first experiment we compare *Variant* P and *Variant* R_1. In both variants, we chose 5 cyclically repeated interpolation nodes in Σ, indicated by "$\times$" in Figure 2(a). The corresponding Leja–Bagby poles, selected in *Variant* R_1, are indicated by "$\bullet$."

The convergence history of the eigenpairs computed with *Variant* P and *Variant* R_1 are given in Figures 2(b) and 2(c), respectively. Note that in these figures the solid and dotted lines correspond to eigenvalues lying inside and outside the target set Σ, respectively. From these figures, we can see that the eigenvalues computed with *Variant* R_1 converge much faster than these computed with *Variant* P and this with similar total computational cost. Hence, we conclude that NLEIGS can be significantly faster than the Newton rational Krylov method introduced in [36].

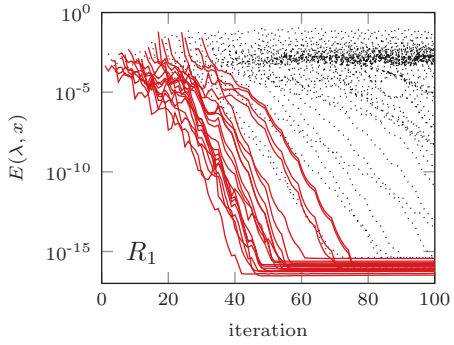
We will now look at the corresponding convergence of the approximations of $A(\lambda)$ by polynomial interpolants $P_j(\lambda)$ and rational interpolants $Q_j(\lambda)$. Figure 3 shows the maxima of the scalar (generalized) divided differences in every iteration j of Algorithm 1. We see that in *Variant* P (solid line) there is no convergence for $j \to \infty$. Thus, it is possible to miss some eigenvalues since in this case the underlying linearized polynomial eigenvalue problem does not approximate the original NLEP



(a) *Variant* R_1: nodes, poles, and eigenvalues



(b) *Variant* P: convergence of eigenpairs



(c) *Variant* R_1: convergence of eigenpairs

FIG. 2. *Results for the* **'gun'** *problem:* (a) *Approximate eigenvalues for the original NLEP* (7.1) *obtained with Variant* R_1, (b) *convergence history for Variant* P, *and* (c) *convergence history for Variant* R_1.

accurately in the whole target set Σ. See also Table 1 below, which shows that only 17 of 21 eigenvalues are found with *Variant* P.

On the other hand, in *Variant* R_1 (dashed line), the approximations $Q_j(\lambda)$ of $A(\lambda)$ converges slowly as j increases. In order to get faster convergence of the rational approximations of $A(\lambda)$, we can use Leja–Bagby interpolation nodes and poles. In Figure 3 we see that the error of the rational interpolants with Leja–Bagby points in *Variant* R_2 (dotted line) decreases much faster than that of the rational interpolants with cyclically repeated nodes in *Variant* R_1 (dashed line).

7.1.2. Dynamic (R_2) versus static (S) interpolation. In this experiment we compare *Variant* R_2 and *Variant* S. In both variants we chose Leja–Bagby interpolation nodes and poles, indicated in Figure 4(a) by "×" and "•," respectively. We also set the tolerance for the relative residual norm $E(\lambda)$ to $\texttt{tol} = 10^{-10}$.

Although Leja–Bagby points are near optimal for uniform convergence of the rational expansions $Q_j(\lambda)$, their use as shifts of the rational Krylov space may not be advantageous for quickly finding eigenvalues inside Σ. Hence, upon convergence of $Q_j(\lambda)$, we recommend to either switch to interpolation nodes in the interior of Σ and apply the truncation strategy explained in section 6.2, or to use the static variant.

In *Variant* R_2 we used Leja–Bagby points only for the first $j = 35$ iterations, until the approximations $Q_j(\lambda)$ have converged. Next, we applied the truncation strategy

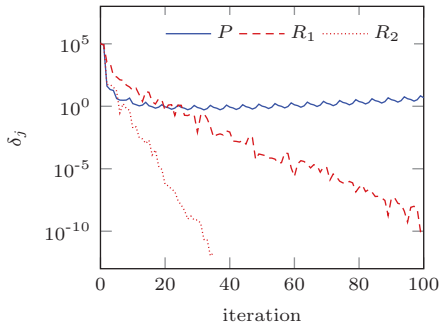


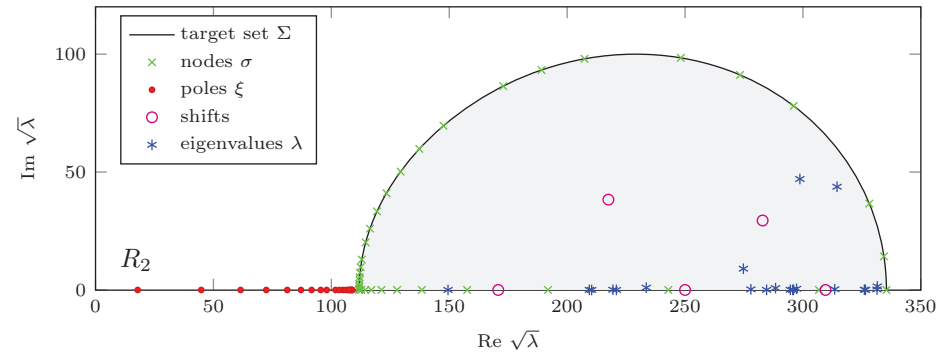
FIG. 3. *Convergence of the approximations of $A(\lambda)$ for the* `'gun'` *problem: Variant P (solid line), Variant R_1 (dashed line), and Variant R_2 (dotted line).*

and froze the linearization. As a result the rational Krylov vectors do not grow any more, resulting in lower memory consumption and a cheaper orthogonalization process. We then switched to cyclically repeated interpolation nodes, indicated by "$\circ$" in Figure 4(a), in order to obtain faster convergence to eigenvalues located in the interior of Σ. The resulting convergence history is shown in Figure 4(b). In this figure we see that during the expansion phase, i.e., when the shifts of the rational Krylov space are still chosen on $\partial\Sigma$, there is very slow convergence for some eigenvalues close to $\partial\Sigma$ and where the density of interpolation nodes is high. From iteration $j = 36$ onwards, that is when we use cyclically repeated interpolation nodes in the interior of Σ, we notice in Figure 4(b) a fast and very regular convergence for all eigenvalues in Σ.

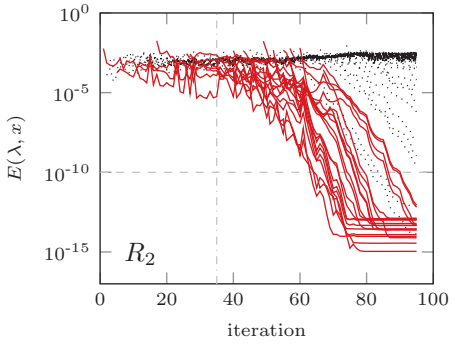
During the expansion phase of *Variant R_2* we again observe slow convergence for some eigenvalues. As before this is because the shifts of the rational Krylov space are chosen equal to the Leja–Bagby interpolation nodes in order to make the algorithm dynamic. Therefore, in *Variant S* we first determine the rational approximation $Q_N(\lambda)$ and then freeze the linearization. Next, the generalized eigenvalue problem is solved with the standard rational Krylov method with cyclically repeated shifts, indicated by "$\circ$" in Figure 4(a). The corresponding convergence history of the eigenpairs computed with *Variant S* is given in Figure 4(c). Note that *Variant S* only requires 70 iterations compared to 95 in *Variant R_2* for computing all eigenvalue inside Σ.

7.1.3. Timings and memory usage. We now compare timings and memory usage of the four different variants of Algorithm 1 for solving the `'gun'` problem. In all experiments, we used the reorthogonalization strategy of [23] and exploited the low-rank structure of the nonlinear part of (7.1) as explained in section 6.4. Hence, the orthogonalization cost is not dominant compared to the one for the system solves. In case of cyclically repeated shifts $\sigma_j \in \Omega_{\text{cycl}}$ of the rational Krylov space, we reused the LU factors of $A(\sigma_j)$.

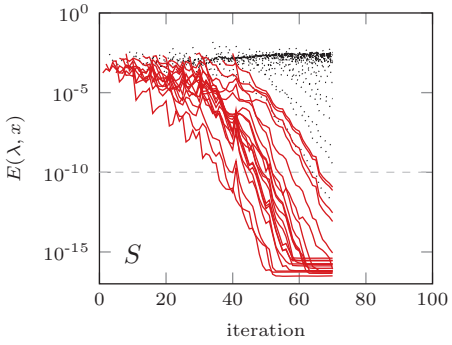
A comparison for solving the `'gun'` problem with *Variant P*, *Variant R_1*, *Variant R_2*, and *Variant S* is given in Table 1, where the memory usage also includes the storage of the LU factors ($\sim$ 75 MB each). From this table, we see that the computational cost and memory usage of *Variant P* and *Variant R_1* are very similar: both variants require the same number of system solves and LU factorizations, and the rational Krylov vectors are of the same length. There is only a difference in computational cost of the orthogonalization process, due to different amounts of reorthogonalization: only 35% of the iterations of *Variant P* require reorthogonalization, whereas in *Variant R_1* reorthogonalization is required in 89% of the iterations.



(a) *Variant* R_2: nodes, poles, shifts, and eigenvalues



(b) *Variant* R_2: convergence of eigenpairs



(c) *Variant* S: convergence of eigenpairs

FIG. 4. *Results for the 'gun' problem:* (a) *Approximate eigenvalues for the original NLEP* (7.1) *obtained with Variant* R_2, (b) *convergence history for Variant* R_2, *and* (c) *convergence history for Variant* S.

Variant R_2 requires more computation time, due to the higher number of LU factorizations, but in this variant the approximation of $A(\lambda)$ converges. Therefore *Variant* R_2 is more robust than *Variant* P and *Variant* R_1. The slightly lower memory usage in *Variant* R_2 is due to the stopping of this method after 95 iterations and also the rational Krylov vectors do not grow any more after the linearization is frozen.

Table 1 also shows that *Variant* S is the most efficient variant for solving the 'gun' problem. First, this variant requires only 70 iterations, since the rational Krylov process only starts after the approximation has converged. Together with the reuse of LU factors, this results in a low system solving cost. Second, although the rational Krylov vectors are of the same length as in *Variant* R_2, the memory usage in *Variant* S is lower since fewer iterations are needed to compute all the eigenvalues

TABLE 1
Timings and memory usage for the 'gun' problem.

	# Iter.	# Conv. λ	System solves	Orthog.	Total CPU time	Memory usage
Variant P	100	17	7.1 s	2.1 s	9.6 s	$\sim$ 460 MB
Variant R_1	100	21	7.1 s	2.9 s	10.7 s	$\sim$ 460 MB
Variant R_2	95	21	26.5 s	2.7 s	30.2 s	$\sim$ 449 MB
Variant S	70	21	6.0 s	1.1 s	8.1 s	$\sim$ 435 MB

in Σ. On the other hand, compared to the other variants, *Variant S* does no longer have the property of being dynamic.

7.2. Particle in a canyon problem. We consider the Schrödinger equation for a particle in a potential well attached to a number of contacts. In this example, the particle has mass $0.2\,\mathrm{m_e}$ and the two-dimensional potential has a canyon-like shape with a canyon length, width, and depth of 2.2 nm, 4 nm, and 3 eV, respectively, while the width and depth of the valley in the contacts is 2 nm and 3 eV, respectively. The Schrödinger equation is discretized on a $4 \times 10\text{-nm}^2$ grid. For more information about the physics we refer to [37].

The corresponding nonlinear eigenvalue problem is

$$A(\lambda)x = \Big(H - \lambda I - \textstyle\sum_{k=1}^{n_z} e^{i\sqrt{m(\lambda-\alpha_k)}} L_k U_k^*\Big)x = 0, \tag{7.2}$$

where $H \in \mathbb{R}^{16281\times 16281}$ is symmetric, $L_k, U_k \in \mathbb{R}^{16281\times 2}$, $m = 0.2$, and $n_z = 81$. The branch points are defined by $\alpha_k \in \mathbb{R}$ and sorted in ascending order. We take the interval between the first and second branch point as target set $\Sigma = [\alpha_1 + \varepsilon, \alpha_2 - \varepsilon]$, with $\alpha_1 \approx -0.198$, $\alpha_2 \approx -0.132$, and $\varepsilon = 10^{-4}$. In order to make Σ branch cut free, we define the branch cut corresponding to the first nonlinear term in (7.2) as $(-\infty, \alpha_1]$, whereas all other branch cuts are defined as $[\alpha_k, +\infty)$ for $k = 2, 3, \ldots, n_z$. The singularity set $\Xi = (-\infty, \alpha_1] \cup [\alpha_2, +\infty)$ is the union of all branch cuts. We have discretized Ξ by the union of 10^4 logarithmically spaced points on $[-10^6, \alpha_1]$ and $[\alpha_2, 10^6]$. In all experiments we used again the reorthogonalization strategy of [23] and exploited the low-rank structure of the nonlinear part of (7.2) as explained in section 6.4.

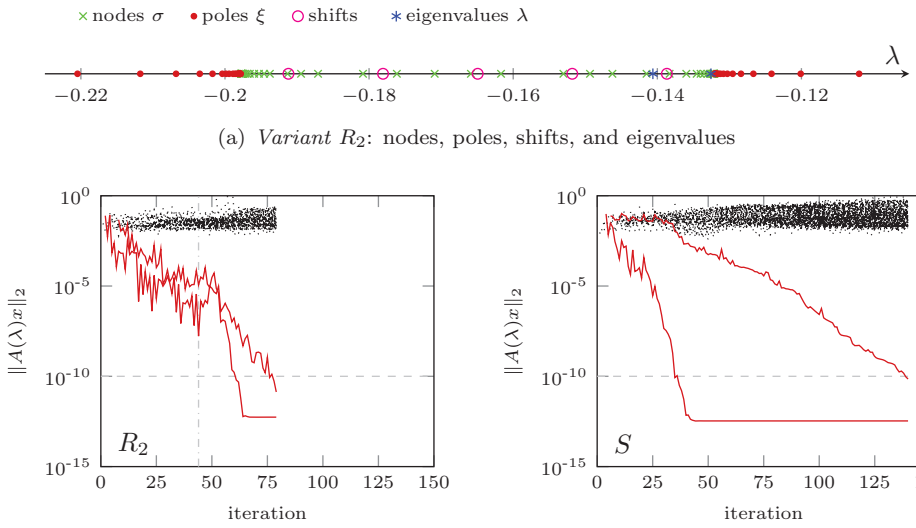
7.2.1. Dynamic ($\mathbf{R_2}$) versus static (S) interpolation. For the NLEP (7.2) we only compare *Variant* R_2 and *Variant S* of Algorithm 1. The Leja–Bagby interpolation nodes and poles, used in both experiments, are indicated in Figure 5(a) by "$\times$" and "$\bullet$," respectively. The cyclically repeated shifts are indicated by "$\circ$" in Figure 5(a). We also set the tolerance for the residual norm $\|A(\lambda)x\|_2$ to $\texttt{tol} = 10^{-10}$. The convergence histories of the eigenpairs computed with *Variant* R_2 and *Variant S* are given in Figures 5(b) and 5(c), respectively.

7.2.2. Timings and memory usage. We compare timings and memory usage of *Variant* R_2 and *Variant S* for solving the `'particle in a canyon.'`

A comparison of the computation time and memory usage is shown in Table 2, where the memory usage also includes the storage of the LU factors ($\sim 22\,\mathrm{MB}$ each). Table 2 shows that *Variant* R_2 is the most efficient, both in terms of computation time and memory usage. In contrast to the previous example, the Leja–Bagby nodes in this example lie *inside* the region of interest Σ (an interval) and hence are good choices for the shifts of the rational Krylov space. As a consequence, the Ritz pairs start converging already during the expansion phase of the algorithm (see Figure 5(b)), resulting in almost half the number of iterations required by *Variant S*. Note also

TABLE 2
Timings and memory usage for the 'particle in a canyon' problem.

	# Iter.	# Conv. λ	System solves	Orthog.	Total CPU time	Memory usage
Variant R_2	79	2	12.2 s	3.5 s	18.5 s	$\sim 234\,\mathrm{MB}$
Variant S	140	2	6.5 s	11.8 s	21.8 s	$\sim 286\,\mathrm{MB}$



(a) *Variant* R_2: nodes, poles, shifts, and eigenvalues

(b) *Variant* R_2: convergence of eigenpairs

(c) *Variant* S: convergence of eigenpairs

FIG. 5. *Results for the* `'particle in a canyon'` *problem:* (a) *approximate eigenvalues for the NLEP* (7.2) *obtained with Variant* R_2, (b) *convergence history for Variant* R_2, *and* (c) *convergence history for Variant* S.

that due to the larger number of iterations in *Variant* S, the orthogonalization process becomes the dominant computational cost.

8. Conclusions and future work. In this paper we have introduced a new linearization for nonlinear eigenvalue problems based on linear rational interpolation. We have shown how the involved divided differences can be computed using matrix functions, and how the linearization can be efficiently intertwined with a rational Krylov method for finding its eigenpairs. The resulting method is called NLEIGS. Truncation and scaling strategies were proposed to make NLEIGS computationally efficient.

Our numerical experiments have revealed that several variants of NLEIGS are viable. We found that, for all examples, NLEIGS largely outperforms the Newton rational Krylov method, both in speed and reliability. We expect two variants of NLEIGS to be useful for applications. Both first build a rational approximation of $A(\lambda)$ using Leja–Bagby nodes and poles. For the dynamic variant, the truncation strategy and switching to shifts σ_j in the interior of Σ may be important for fast rational Krylov convergence. The resulting method is by far more reliable than the Newton rational Krylov method, although not always faster in terms of computation time. We also proposed a static variant of the method where the linearization is determined first and then rational Krylov is applied to the linearized problem. For problems with eigenvalues on an interval nearby singularities, as in the canyon problem, the dynamic version purely based on Leja–Bagby points appears to be the most efficient and reliable method in terms of number of iterations. For problems with eigenvalues on a two-dimensional target set, as in the `'gun'` problem, the static version appears to be both fast and reliable. We believe that our method can be seen as a promising and robust approach for solving nonlinear eigenvalue problems.

In future work we wish to explore the use of rational interpolation with prescribed interpolation nodes σ_j in the *interior* of Σ, instead of Leja–Bagby nodes discussed

in section 5 which are typically on $\partial\Sigma$. The poles of interpolants optimal in a least-squares sense can be found easily if Σ is the unit disk (see [38]) or on more general sets when an external conformal map is available; see, e.g., [12, 4]. Other ideas include the spectrally adaptive choice of poles and interpolation nodes using Ritz value information, similarly to what has been done in [17], or the automated detection of the singularity set Ξ via rational interpolation with free poles, e.g., using the SVD-based Padé approximants from [14]. An interesting algebraic question is whether or not Theorem 3.2 provides a strong linearization in the sense that it preserves the multiplicities of eigenvalues; cf. [1]. As an algorithmic enhancement we are currently developing a restarted version of NLEIGS.

Acknowledgments. Roel Van Beeumen wishes to thank the Mathematics department at the University of Manchester for their hospitality. The authors are grateful to William Vandenberghe for providing the matrices of the particle in a canyon problem used in section 7.2. We also thank Cedric Effenberger, Elias Jarlebring, Daniel Kressner, Françoise Tisseur, and Laurent Sorber for useful discussions.

REFERENCES

[1] A. Amiraslani, R. M. Corless, and P. Lancaster, *Linearization of matrix polynomials expressed in polynomial bases*, IMA J. Numer. Anal., 29 (2009), pp. 141–157.
[2] T. Bagby, *The modulus of a plane condenser*, J. Math. Mech., 17 (1967), pp. 315–329.
[3] T. Bagby, *On interpolation by rational functions*, Duke Math. J., 36 (1969), pp. 95–104.
[4] B. Beckermann and L. Reichel, *Error estimates and evaluation of matrix functions via the Faber transform*, SIAM J. Numer. Anal., 47 (2009), pp. 3849–3883.
[5] 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. 7–28.
[6] T. Betcke and H. Voss, *A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems*, Future Gener. Comp. Systems, 20 (2004), pp. 363–372.
[7] W.-J. Beyn, *An integral method for solving nonlinear eigenvalue problems*, Linear Algebra Appl., 436 (2012), pp. 3839–3863.
[8] W.-J. Beyn, C. Effenberger, and D. Kressner, *Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems*, Numer. Math., 119 (2011), pp. 489–516.
[9] J. P. Boyd, *Finding the zeros of a univariate equation: Proxy rootfinders, Chebyshev interpolation, and the companion matrix*, SIAM Rev., 55 (2013), pp. 375–396.
[10] C. Effenberger, *Robust successive computation of eigenpairs for nonlinear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1231–1256.
[11] C. Effenberger and D. Kressner, *Chebyshev interpolation for nonlinear eigenvalue problems*, BIT, 52 (2012), pp. 933–951.
[12] D. Gaier and R. McLaughlin, *Lectures on Complex Approximation*, Birkhäuser. Boston, 1987.
[13] A. A. Gonchar, *Zolotarev problems connected with rational functions*, Math. USSR Sb., 7 (1969), pp. 623–635.
[14] P. Gonnet, S. Güttel, and L. N. Trefethen, *Robust Padé approximation via SVD*, SIAM Rev., 55 (2013), pp. 101–117.
[15] S. Güttel, *Rational Krylov Methods for Operator Functions*, Ph.D. thesis, TU Bergakademie Freiberg, Freiberg, Germany, 2010.
[16] S. Güttel, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM Mitt. Ges. Angew. Math. Mech., 36 (2013), pp. 8–31.
[17] S. Güttel and L. Knizhnerman, *A black-box rational Arnoldi variant for Cauchy Stieltjes matrix functions*, BIT, 53 (2013), pp. 595–616.
[18] E. Jarlebring and S. Güttel, *A spatially adaptive iterative method for a class of nonlinear operator eigenproblems*, Electron. Trans. Numer. Anal., 41 (2014), pp. 21–41.
[19] E. Jarlebring, K. Meerbergen, and W. Michiels, *A Krylov method for the delay eigenvalue problem*, SIAM J. Sci. Comput., 32 (2010), pp. 3278–3300.
[20] E. Jarlebring, K. Meerbergen, and W. Michiels, *Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 411–436.

[21] E. Jarlebring, W. Michiels, and K. Meerbergen, *A linear eigenvalue algorithm for the nonlinear eigenvalue problem*, Numer. Math., 122 (2012), pp. 169–195.
[22] D. Kressner, *A block Newton method for nonlinear eigenvalue problems*, Numer. Math., 114 (2009), pp. 355–372.
[23] R. B. Lehoucq, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration*, Ph.D. thesis, Rice University, Houston, TX, 1995.
[24] A. L. Levin and E. B. Saff, *Optimal ray sequences of rational functions connected with the Zolotarev problem*, Constr. Approx., 10 (1994), pp. 235–273.
[25] E. Levin and E. B. Saff, *Potential theoretic tools in polynomial and rational approximation*, in Harmonic Analysis and Rational Approximation, J.-D. Fournier et al., ed., Lecture Notes in Control and Inform. Sci. 237, Springer, Berlin, 2006, pp. 71–94.
[26] B.-S. Liao, *Subspace Projection Methods for Model Order Reduction and Nonlinear Eigenvalue Computation*, Ph.D. thesis, Department of Mathematics, University of California at Davis, Davis, CA, 2007.
[27] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko, *Nonlinear Rayleigh–Ritz iterative method for solving large scale nonlinear eigenvalue problems*, Taiwanese J. Math., 14 (2010), pp. 869–883.
[28] 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.
[29] A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal., 22 (1985), pp. 914–923.
[30] G. Opitz, *Steigungsmatrizen*, ZAMM Z. Angew. Math. Mech., 44 (1964), pp. T52–T54.
[31] L. Reichel, *Newton interpolation at Leja points*, BIT, 30 (1990), pp. 332–346.
[32] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.
[33] A. Ruhe, *Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.
[34] E. B. Saff and V. Totik, *Logarithmic Potentials with External Fields*, Springer, Berlin, 1997.
[35] Y. Su and Z. Bai, *Solving rational eigenvalue problems via linearization*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 201–216.
[36] 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.
[37] W. G. Vandenberghe, M. V. Fischetti, R. Van Beeumen, K. Meerbergen, W. Michiels, and C. Effenberger, *Determining bound states in a semiconductor device with contacts using a nonlinear eigenvalue solver*, J. Comput. Electron., 13 (2014), pp. 753–762.
[38] J. L. Walsh, *On interpolation and approximation by rational functions with preassigned poles*, Trans. Amer. Math. Soc., 34 (1932), pp. 22–74.
[39] J. L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain*, 5th ed., AMS, Providence, RI, 1969.
[40] J. L. Walsh and H. G. Russell, *Hyperbolic capacity and interpolating rational functions* II, Duke Math. J., 33 (1966), pp. 275–279.
[41] Z. Wu and W. Michiels, *Reliably computing all characteristic roots of delay differential equations in a given right half plane*, J. Comput. Appl. Math., 236 (2012), pp. 2499–2514.