INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **103**:840–858
Published online 5 June 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4919

A Padé approximate linearization algorithm for solving the quadratic eigenvalue problem with low-rank damping

Ding Lu[1], Xin Huang[1], Zhaojun Bai[2,3,*,†] and Yangfeng Su[1]

[1]*School of Mathematical Sciences, Fudan University, Shanghai 200433, China*
[2]*Department of Computer Science, University of California, Davis, CA 95616, USA*
[3]*Department of Mathematics, University of California, Davis, CA 95616, USA*

SUMMARY

The low-rank damping term appears commonly in quadratic eigenvalue problems arising from physical simulations. To exploit the low-rank damping property, we propose a Padé approximate linearization (PAL) algorithm. The advantage of the PAL algorithm is that the dimension of the resulting linear eigenvalue problem is only $n + \ell m$, which is generally substantially smaller than the dimension $2n$ of the linear eigenvalue problem produced by a direct linearization approach, where n is the dimension of the quadratic eigenvalue problem, and ℓ and m are the rank of the damping matrix and the order of a Padé approximant, respectively. Numerical examples show that by exploiting the low-rank damping property, the PAL algorithm runs 33–47% faster than the direct linearization approach for solving modest size quadratic eigenvalue problems. Copyright © 2015 John Wiley & Sons, Ltd.

Received 1 July 2014; Revised 7 January 2015; Accepted 11 March 2015

KEY WORDS: quadratic eigenvalue problem; low-rank damping; linearization; Padé approximation

1. INTRODUCTION

We consider the quadratic eigenvalue problem (QEP)

$$\mathcal{Q}(\lambda)x \equiv (\lambda^2 M + \lambda C + K)x = 0, \tag{1.1}$$

where M, C, and K are $n \times n$ matrices, referred to as mass, damping, and stiffness matrices, respectively, in structural dynamics analysis. The low-rank damping property refers to the case where the damping matrix C is of rank ℓ, $\ell \ll n$ and admits the rank-revealing decomposition

$$C = EF^{\mathrm{T}}, \tag{1.2}$$

where E and F are $n \times \ell$ full column rank matrices.

The QEP with the low-rank damping arises frequently from analysis of structural dynamics [1, 2] and acoustic analysis [3–5]. In these applications, the damping force is typically applied to the boundary and/or a small region. The finite element discretization of the governing equations leads to a QEP with an extremely sparse and low-rank damping matrix C.

To compute the eigenvalues of the QEP (1.1) close to a point σ of interest, a standard approach is to first apply the shift spectral transformation $\mu = \lambda - \sigma$ and then solve the QEP

$$(\mu^2 M + \mu C_\sigma + K_\sigma)x = 0 \tag{1.3}$$

[*]Correspondence to: Zhaojun Bai, Department of Computer Science, University of California, Davis, CA 95616, USA.
[†]E-mail: bai@cs.ucdavis.edu

by a linearization technique, where $C_\sigma = C + 2\sigma M$ and $K_\sigma = \sigma^2 M + \sigma C + K$. For example, in the first companion form, the QEP (1.3) is equivalent to the linear eigenvalue problem (LEP)

$$\begin{bmatrix} -C_\sigma & -K_\sigma \\ I_n & 0 \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix} = \mu \begin{bmatrix} M & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix}, \tag{1.4}$$

where I_n is the $n \times n$ identity matrix. For other forms of linearization, see [6, 7] and the references therein. The task of finding eigenvalues λ of the QEP (1.1) close to the shift σ becomes one of the extracting smallest (in modulus) few eigenvalues μ of the LEP (1.4).

After linearization, a variety of subspace projection-based methods and software for the resulting LEP can be applied. However, the dimension of the LEP (1.4) is twice the dimension of the QEP (1.1), and consequently, memory and computational costs are increased substantially. The Jacobi–Davidson method [8], SOAR [9], and Q-Arnoldi [10] are memory-efficient QEP algorithms. However, none of these algorithms explicitly exploit the low-rank damping property for computational efficiency.

In this paper, we propose an algorithm to explicitly exploit the low-rank damping property for computational efficiency. The new algorithm is referred to as Padé approximate linearization, abbreviated as PAL. The dimension of the LEP produced by the PAL algorithm is $n_{\mathrm{L}} = n + \ell m$, where ℓ is the rank of C, and m is the order of Padé approximant. Because typically $\ell \ll n$ and m is a small positive integer, n_{L} is much smaller than the dimension $2n$ of the LEP derived by a direct linearization. Consequently, the PAL leads to a substantial reduction in memory and computational costs. Numerical examples show that with comparable accuracy, by exploiting the low-rank damping property, the new PAL algorithm runs 33–47% faster than the direct linearization approach for solving the QEPs of modest sizes.

The rest of this paper is organized as follows. In section 2, we introduce a spectral transformation that transforms the QEP (1.1) into a nonlinear eigenvalue problem (NEP). In Section 3, we present the PAL algorithm. In Section 4, we present a backward error analysis and a scaling scheme. In Section 5, we give some implementation details of the PAL algorithm. In Section 6, we present three numerical examples to demonstrate the accuracy and efficiency of the PAL algorithm. Concluding remarks are in Section 7.

2. SPECTRAL TRANSFORMATION

To compute eigenvalues of the QEP (1.1) close to a prescribed nonzero shift σ while preserving the low-rank damping property, let us consider the spectral transformation

$$\begin{aligned} g_\sigma : \mathbb{S}_\sigma &\longrightarrow \mathbb{C} \\ \lambda &\longmapsto \mu = \frac{\lambda^2}{\sigma^2} - 1, \end{aligned} \tag{2.1}$$

where $\mathbb{S}_\sigma$, shown in Figure 1, defines a domain of the complex plane $\mathbb{C}$

$$\mathbb{S}_\sigma \equiv \left\{ z \in \mathbb{C} \mid \arg\left(\frac{z}{\sigma}\right) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right] \right\} \cup \{0\}. \tag{2.2}$$

In addition, let us define the mapping

$$\begin{aligned} f_\sigma : \mathbb{C} &\longrightarrow \mathbb{S}_\sigma \\ \mu &\longmapsto \lambda = \sigma\sqrt{\mu + 1}, \end{aligned} \tag{2.3}$$

where $\sqrt{\cdot}$ denotes the principal square root.[‡]

[‡] Using the polar coordinate system, a complex number z can be expressed as $z = te^{i\theta}$, where $t \geqslant \mathbf{0}$ is the modulus and the distance to the origin, and $\theta \in (-\pi, \pi]$ is the angle that the line from z to the origin makes with the positive real axis. The principal square root of z is then defined by $\sqrt{z} = \sqrt{t}e^{i\theta/2}$.

DOI: 10.1002/nme

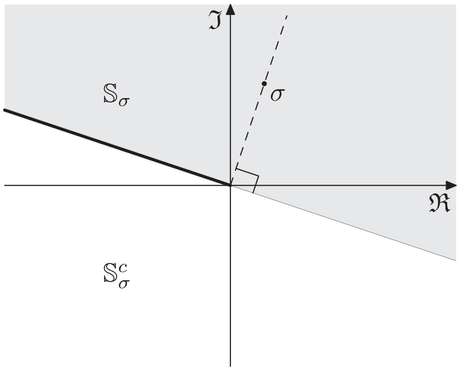

Figure 1. The domain $\mathbb{S}_\sigma$ (gray region including the solid line) and its complement $\mathbb{S}_\sigma^c$. If σ is a positive real number, then $\mathbb{S}_\sigma$ is the right half-plane including the non-negative imaginary axis, and if σ is a pure imaginary number with positive imaginary part, then $\mathbb{S}_\sigma$ is the upper half-plane including the non-positive real axis.

The following lemma characterizes the relationship between mappings g_σ and f_σ.

Lemma 1
Let $\mathbb{S}_\sigma$, f_σ and g_σ be as in (2.1)–(2.3). (a) If $\lambda \in \mathbb{S}_\sigma$ and $\mu = g_\sigma(\lambda)$, then $f_\sigma(\mu) = \lambda$. (b) If $\mu \in \mathbb{C}$ and $\lambda = f_\sigma(\mu)$, then $\lambda \in \mathbb{S}_\sigma$ and $g_\sigma(\lambda) = \mu$.

Proof
First, let us show that if $\lambda \in \mathbb{S}_\sigma$, then $\sqrt{(\lambda/\sigma)^2} = \lambda/\sigma$. In fact, by the polar coordinate $\lambda/\sigma = te^{\theta i}$, where t is the modulus and $\theta \in (-\pi/2, \pi/2]$, we have $(\lambda/\sigma)^2 = t^2 e^{2\theta i}$. Consequently, by the definition of principal square root, we have the identity $\sqrt{(\lambda/\sigma)^2} = \lambda/\sigma$.

Statement (a) is a straightforward computation.

$$f_\sigma(\mu) = \sigma\sqrt{\mu+1} = \sigma\sqrt{\frac{\lambda^2}{\sigma^2} - 1 + 1} = \sigma\sqrt{\frac{\lambda^2}{\sigma^2}} = \sigma\frac{\lambda}{\sigma} = \lambda. \tag{2.4}$$

For (b), because $\lambda/\sigma = \sqrt{\mu+1}$, and by the definition of principal square root, we have $\sqrt{\mu+1} = te^{\theta i}$ with t being the modular and $\theta \in (-\pi/2, \pi/2]$, which implies $\lambda \in \mathbb{S}_\sigma$. By (a), we have $\mu = g_\sigma(\lambda)$. □

Using the spectral transformation (2.1), the QEP (1.1) is transformed into the following NEP:

$$\mathcal{N}(\mu)x \equiv [K_\sigma - \mu M_\sigma + f_\sigma(\mu)C]\,x = 0, \tag{2.5}$$

where $K_\sigma = K + \sigma^2 M$ and $M_\sigma = -\sigma^2 M$.

The following theorem shows the relationship between the QEP (1.1) and the NEP (2.5) with respect to the domain $\mathbb{S}_\sigma$.

Theorem 1

(a) If (λ, x) is an eigenpair of the QEP (1.1) and $\lambda \in \mathbb{S}_\sigma$, then $(\mu = g_\sigma(\lambda), x)$ is an eigenpair of the NEP (2.5).
(b) If (μ, x) is an eigenpair of the NEP (2.5), then $\lambda = f_\sigma(\mu) \in \mathbb{S}_\sigma$ and (λ, x) is an eigenpair of the QEP (1.1).

DOI: 10.1002/nme

Proof

(a) By Lemma 1(a), we have $\lambda = f_\sigma(\mu)$, where $\mu = g_\sigma(\lambda)$. Because (λ, x) is an eigenpair, it follows that

$$\begin{aligned} 0 = \mathcal{Q}(\lambda)x &= (\lambda^2 M + \lambda C + K)x \\ &= \left[(\mu+1)\sigma^2 M + f_\sigma(\mu)C + K\right]x = \mathcal{N}(\mu)x. \end{aligned}$$

Therefore, (μ, x) is an eigenpair of the NEP (2.5).

(b) By Lemma 1(b), we have $\lambda \in \mathbb{S}_\sigma$, and $\mu = g_\sigma(\lambda)$. Because (μ, x) is an eigenpair of $\mathcal{N}(\mu)$, it follows that

$$\begin{aligned} 0 = \mathcal{N}(\mu)x &= [K_\sigma - \mu M_\sigma + f_\sigma(\mu)C]\,x \\ &= [K_\sigma - g_\sigma(\lambda)M_\sigma + \lambda C]\,x = \mathcal{Q}(\lambda)x. \end{aligned}$$

Hence, (λ, x) is an eigenpair of the QEP (1.1).

□

For computing the eigenvalues of the QEP in the complement of $\mathbb{S}_\sigma$, i.e., $\mathbb{S}_\sigma^c = \mathbb{S}_{-\sigma} \setminus \{0\}$, we consider the following NEP

$$\mathcal{N}^{\mathrm{c}}(\mu)x \equiv [K_\sigma - \mu M_\sigma - f_\sigma(\mu)C]\,x = 0. \tag{2.6}$$

The following theorem shows the equivalence between the QEP (1.1) and the NEP (2.6) with respect to the domain $\mathbb{S}_\sigma^c$.

Theorem 2

(a) If (λ, x) is an eigenpair of the QEP (1.1) and $\lambda \in \mathbb{S}_\sigma^c$, then $(\mu = g_\sigma(\lambda), x)$ is an eigenpair of the NEP (2.6).

(b) If (μ, x) is an eigenpair of the NEP (2.6) and $\mu \neq 0$, then $\lambda = -f_\sigma(\mu) \in \mathbb{S}_\sigma^c$ and (λ, x) is an eigenpair of the QEP (1.1).

Proof
Similar to the proof of Theorem 1. Note that $\mathbb{S}_\sigma^c = \mathbb{S}_{-\sigma} \setminus \{0\}$ and $f_{-\sigma}(\mu) = -f_\sigma(\mu)$. □

By Theorems 1 and 2, the eigenvalues of the QEP (1.1) in $\mathbb{S}_\sigma$ are transformed to the eigenvalues of the NEP (2.5), while the eigenvalues of the QEP in $\mathbb{S}_\sigma^c$ are transformed to the eigenvalues of the NEP (2.6). Because we are interested in extracting the eigenvalues of the QEP close to σ, we will focus on the NEP (2.5) in the rest of the paper.

By the spectral transformation (2.1), λ close to σ corresponds to μ close to 0. Thus, seeking eigenvalues λ of the QEP (1.1) close to the shift σ turns into seeking small (in modulus) eigenvalues μ of the NEP (2.5) in a disk $|\mu| \leqslant \rho$. Specifically, the region in $\mathbb{S}_\sigma$ corresponding to the disk $|\mu| \leqslant \rho$ is

$$\mathbb{A}_{\sigma,\rho} = \{\lambda \mid \lambda \in \mathbb{S}_\sigma \text{ and } |g_\sigma(\lambda)| \leqslant \rho\} \equiv \sigma\mathbb{B}_\rho, \tag{2.7}$$

where $\mathbb{B}_\rho = \{\lambda \mid \lambda \in \mathbb{S}_1 \text{ and } |\lambda^2 - 1| \leqslant \rho\}$. $\mathbb{A}_{\sigma,\rho}$ is the unit region $\mathbb{B}_\rho$ scaled by σ as shown in Figure 2. As we can see, $\mathbb{A}_{\sigma,\rho}$ leans toward the origin $(0, 0)$. $\mathbb{A}_{\sigma,\rho}$ can be regarded as the *domain of confidence* for the spectral transformation (2.1). This is similar to the notion for the shift spectral transformation [11]. We note that in practice, the shift $|\sigma|$ should not be chosen too small. Otherwise, the domain of confidence $\mathbb{A}_{\sigma,\rho}$ in the λ-plane corresponding to $|\mu| \leqslant \rho$ is small. In this case, the Padé approximant to be introduced in the next section will be able to approximate only a small number of eigenvalues of the QEP.

DOI: 10.1002/nme

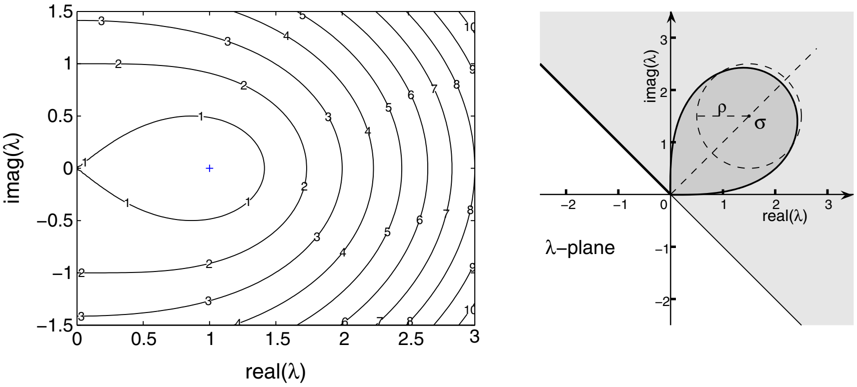

Figure 2. Left: the regions $\mathbb{B}_\rho$ with $\rho = 1, 2, \cdots, 10$. Right: the region $\mathbb{A}_{\sigma,1} = \sigma\mathbb{B}_1$ with $\sigma = 1.5 + 1.5i$ (darker gray).

3. PADÉ APPROXIMATE LINEARIZATION

In this section, we start with an approximation of the NEP (2.5) by a rational eigenvalue problem (REP) via Padé approximation. Then we apply a trimmed linearization technique to convert the REP into an LEP.

3.1. Padé approximation

To find an accurate approximation of the NEP (2.5), let us consider an order-(m, m) Padé approximation [12] of the function $\sqrt{\mu + 1}$. In matrix–vector form, it can be written as

$$r_m(\mu) = -a^{\mathrm{T}}(I_m - \mu D_m)^{-1}a + d, \tag{3.1}$$

where a is a column vector $a = \left[(\gamma_1/\xi_1)^{\frac{1}{2}},\ (\gamma_2/\xi_2)^{\frac{1}{2}},\ \ldots,\ (\gamma_m/\xi_m)^{\frac{1}{2}}\right]^{\mathrm{T}}$, D_m is a diagonal matrix $D_m = -\mathrm{diag}(\xi_1,\ \xi_2,\ \ldots,\ \xi_m)$, $d = 2m + 1$ and

$$\gamma_j = \frac{2}{2m+1}\sin^2\frac{j\pi}{2m+1} \quad \text{and} \quad \xi_j = \cos^2\frac{j\pi}{2m+1}.$$

The poles of $r_m(\mu)$ are $-1/\xi_j$ for $j = 1, 2, \ldots, m$. In [13], it is shown that the approximation error is given by

$$e(\mu) \equiv \sqrt{\mu+1} - r_m(\mu) = 2\sqrt{\mu+1}\,\frac{\theta^{2m+1}}{1+\theta^{2m+1}}, \tag{3.2}$$

where $\theta = \left(\sqrt{\mu+1} - 1\right)/\left(\sqrt{\mu+1} + 1\right)$.[§] When $|\mu|$ is sufficiently small, $|e(\mu)| = O(\mu^{2m+1})$. The Padé approximation is more accurate than the polynomial-based Taylor approximation. For example, Figure 3 is a contour plot of the error $|e(\mu)|$ of $r_5(\mu)$. For $\mu = 2$, we have $|e(2)| \approx 1.77 \times 10^{-6}$. In contrast, the error of the 10th-order Taylor approximation is about 5.9938.

By the Padé approximant (3.1), the NEP (2.5) can be written as

$$\mathcal{N}(\mu)x = [K_\sigma - \mu M_\sigma + \sigma(r_m(\mu) + e(\mu))C]\,x = 0.$$

By truncating the error $e(\mu)$, it is then turned into the following REP:

[§] In [13], this result is shown for the case when μ is real and greater than -1. However, it can be extended directly to complex μ.

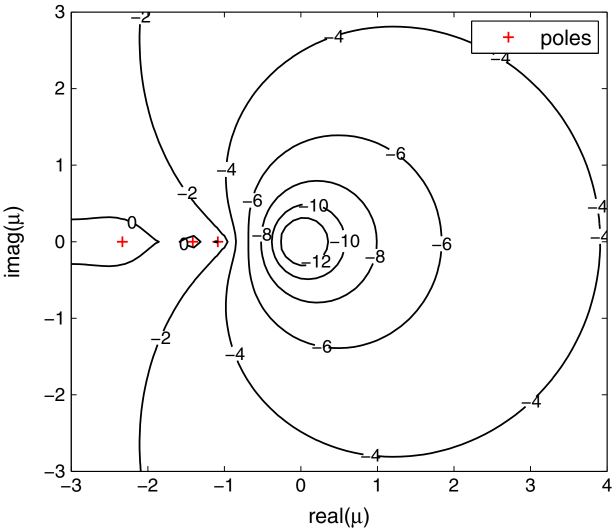

Figure 3. The contour plot of $\log_{10}|e(\mu)|$ with the order-(5,5) Padé approximant. Three marked poles on the real axis are approximately −1.0862, −1.4130 and −2.3319. The other two poles are approximately −5.7948 and −49.3742.

$$\mathcal{R}(\mu)x \equiv [K_\sigma - \mu M_\sigma + \sigma\, r_m(\mu) C]\, x = 0. \tag{3.3}$$

Note that it is an abuse of notation that we use (μ, x) to denote the eigenpairs of the NEP (2.5) and the REP (3.3). We will connect the eigenpairs of the REP (3.3) with the original QEP (1.1) directly. The eigenpairs of the NEP will no longer be referenced.

The Padé approximation induces m poles $\{-1/\xi_1, \ldots, -1/\xi_m\}$ in the REP (3.3). All these poles are real and less than −1. Large Padé error only occurs in a small region around the poles, as illustrated in Figure 3. Because the eigenvalues μ of interest of the REP (3.3) are close to 0, the presence of the poles is generally not a concern in practice.

The REP (3.3) can be interpreted as a perturbation of the original QEP (1.1). Specifically, if (μ_*, x_*) is an eigenpair of the REP (3.3), then it is easy to verify that $(\lambda_* = f_\sigma(\mu_*), x_*)$ is an eigenpair of the QEP

$$\widetilde{Q}(\lambda)x = \left[\lambda^2 M + \lambda(C + \Delta C) + K\right] x = 0, \tag{3.4}$$

where

$$\Delta C = -\frac{e(\mu_*)}{\sqrt{\mu_* + 1}}\, C\, \frac{x_* x_*^{\mathrm{H}}}{\|x_*\|^2}.$$

The QEP (3.4) is a perturbation of the original QEP (1.1). The perturbation only occurs in the damping matrix C, and the perturbed damping term is still of low rank. Furthermore, the relative perturbation error

$$\frac{\|\Delta C\|}{\|C\|} = \frac{|e(\mu_*)|}{|\sqrt{\mu_* + 1}|}\, \frac{\|C x_*\|}{\|C\|\|x_*\|}$$

is expected to be small because of small Padé approximation error $|e(\mu_*)|$, where $\|\cdot\|$ denotes the vector or matrix 2-norms. In addition, the quantity $\|Cx_*\|/(\|C\|\|x_*\|)$ is expected to be small too because of the low-rank damping property. It provides extra accuracy to the approximation. In the extreme case where $Cx_* = 0$, the REP (3.3) and the QEP (1.1) share the same eigenpair. See Example 1 in Section 6.

DOI: 10.1002/nme

The idea of approximating a NEP by a simple eigenvalue problem has been proposed repeatedly. In [14, 15], a successive linear approximation method is used to solve an NEP by successively solving a sequence of LEPs. Instead of linear approximation, high-order polynomial and rational function approximations are studied [16–21]. Although the notion of using one type of eigenvalue problem to approximate another type is widely adopted, a comprehensive error analysis of such an approach is not trivial. This is particularly true for NEPs. Recently, the eigenvalue approximation error is characterized using the first-order perturbation theory [16] and the nonlinear perturbation of LEPs [22]. Here, for the REP approximation (3.3) of the NEP (2.5), the approximation error can be interpreted as the backward error to the original QEP (1.1). We can apply the well-studied perturbation theory of the QEP; see, for example, [23]. This is another advantage of our proposed approach.

3.2. Trimmed linearization

To solve the REP (3.3), we apply the trimmed linearization technique [24]. It converts the REP (3.3) to an LEP. Specifically, by the Padé approximant (3.1) and the factorization (1.2) of the damping matrix C, the rational term of the REP (3.3) can be rewritten as

$$\begin{aligned}\sigma r_m(\mu)C &= -\sigma a^{\mathrm{T}}(I_m-\mu D_m)^{-1}a\cdot EF^{\mathrm{T}}+\sigma dC\\ &= -\sigma E\left(I_\ell\cdot a^{\mathrm{T}}(I_m-\mu D_m)^{-1}a\right)F^{\mathrm{T}}+\sigma dC\\ &= -\sigma E(I_\ell\otimes a^{\mathrm{T}})(I_\ell\otimes I_m-\mu I_\ell\otimes D_m)^{-1}(I_\ell\otimes a)F^{\mathrm{T}}+\sigma dC\\ &= -E_{\sigma_1}(I_{\ell m}-\mu I_\ell\otimes D_m)^{-1}F_{\sigma_2}^{\mathrm{T}}+\sigma dC,\end{aligned} \tag{3.5}$$

where $E_{\sigma_1}=\sigma_1 E(I_\ell\otimes a^{\mathrm{T}})$, $F_{\sigma_2}=\sigma_2 F(I_\ell\otimes a^{\mathrm{T}})$, $\otimes$ is the Kronecker product, and $\sigma=\sigma_1\sigma_2$ with σ_1 and σ_2 being two scalars.[¶] By (3.5), the REP (3.3) can be written as

$$\mathcal{R}(\mu)x=\left[K_\sigma+\sigma dC-\mu M_\sigma-E_{\sigma_1}(I_{\ell m}-\mu I_\ell\otimes D_m)^{-1}F_{\sigma_2}^{\mathrm{T}}\right]x=0. \tag{3.6}$$

Applying the trimmed linearization proposed in [24], the REP (3.6) can be recast as the LEP of dimension $n_{\mathrm{L}}=n+\ell m$

$$\mathcal{L}(\mu)x_{\mathrm{L}}\equiv(A-\mu B)x_{\mathrm{L}}=0, \tag{3.7}$$

where

$$A=\begin{bmatrix}K_\sigma+\sigma dC & E_{\sigma_1}\\ F_{\sigma_2}^{\mathrm{T}} & I_{\ell m}\end{bmatrix},\quad B=\begin{bmatrix}M_\sigma & 0\\ 0 & I_\ell\otimes D_m\end{bmatrix},$$

and

$$x_{\mathrm{L}}=Hx\quad\text{with}\quad H=\begin{bmatrix}I_n\\ -(I_{\ell m}-\mu I_\ell\otimes D_m)^{-1}F_{\sigma_2}^{\mathrm{T}}\end{bmatrix}.$$

The connection between the REP and the LEP is shown in the following theorem presented in [24]. Here, $y(i:j)$ denotes the entries i to j of a vector y.

Theorem 3

(a) If μ is an eigenvalue of the REP (3.3), then it is also an eigenvalue of the LEP (3.7). (b) If (μ, x_{L}) is an eigenpair of the LEP (3.7) and μ is not a pole of the REP (3.3) and $x_{\mathrm{L}(1n)}\neq 0$, then $(\mu, x_{\mathrm{L}}(1:n))$ is an eigenpair of the REP (3.3). Moreover, the algebraic and geometric multiplicities of μ for the REP (3.3) and the LEP (3.7) are the same.

[¶] The decomposition $\sigma=\sigma_1\sigma_2$ is not unique. A desirable choice of $\sigma=\sigma_1\sigma_2$ will be discussed in Section 5.

DOI: 10.1002/nme

Note that it is imposed that μ is not a pole of the REP (3.3). This condition can be easily verified because all poles $\{-1/\xi_1, \dots, -1/\xi_m\}$ of the REP (3.3) are known from the choice of the Padé approximant $r_m(\mu)$.

3.3. Summary

The following is a summary of the proposed algorithm for computing a few eigenpairs of the QEP (1.1) around the shift σ.

- Use the spectral transformation (2.1) to convert QEP (1.1) to NEP (2.5).
- Select a Padé approximant $r_m(\mu)$ by (3.1).
- Approximate the NEP (2.5) by the REP (3.3).
- Use the trimmed linearization to the REP (3.3) and obtain the LEP (3.7).
- Compute a few small (in modulus) eigenpairs (μ, x_{L}) of the LEP (3.7),
- Return $(\lambda, x) = (f_\sigma(\mu), x_{\mathrm{L}}(1\!:\!n))$ as approximate eigenpairs of the QEP (1.1).

We call this approach the PAL. A discussion on implementation aspects will be presented in Section 5.

4. ERROR BOUND AND SCALING

In this section, we provide a backward error analysis for the proposed PAL algorithm and discuss a scaling scheme to reduce the backward error.

4.1. Error bound

Let $(\widehat{\mu}, \widehat{x}_{\mathrm{L}})$ be a computed eigenpair of the LEP (3.7). Then by the PAL algorithm, $(\widehat{\lambda}, \widehat{x}) = (f_\sigma(\widehat{\mu}), \widehat{x}_{\mathrm{L}}(1:n))$ is an approximate eigenpair of the original QEP (1.1). By using the backward error analysis in [23], the accuracy of computed eigenpairs of the LEP (3.7) and the QEP (1.1) is measured by the backward errors

$$\eta_{\mathrm{L}}(\widehat{\mu}, \widehat{x}_{\mathrm{L}}) = \frac{\|\mathcal{L}(\widehat{\mu})\widehat{x}_{\mathrm{L}}\|}{\varphi(\widehat{\mu})\|\widehat{x}_{\mathrm{L}}\|} \quad \text{and} \quad \eta_{\mathrm{Q}}(\widehat{\lambda}, \widehat{x}) = \frac{\|\mathcal{Q}(\widehat{\lambda})\widehat{x}\|}{\rho(\widehat{\lambda})\|\widehat{x}\|}, \tag{4.1}$$

respectively, where $\varphi(\widehat{\mu}) = \|A\| + |\widehat{\mu}|\|B\|$ and $\rho(\widehat{\lambda}) = |\widehat{\lambda}|^2\|M\| + |\widehat{\lambda}|\|C\| + \|K\|$. We now derive an upper bound of $\eta_{\mathrm{Q}}(\widehat{\lambda}, \widehat{x})$ in terms of $\eta_{\mathrm{L}}(\widehat{\mu}, \widehat{x}_{\mathrm{L}})$. First, similar to the discussion in [25], the residual of the approximate eigenpair $(\widehat{\lambda}, \widehat{x})$ of the QEP is related to the residual of the approximate eigenpair $(\widehat{\mu}, \widehat{x}_{\mathrm{L}})$ of the LEP as follows:

$$\begin{aligned} \mathcal{Q}(\widehat{\lambda})\widehat{x} &= \mathcal{N}(\widehat{\mu})\widehat{x} = \mathcal{R}(\widehat{\mu})\widehat{x} + \sigma e(\widehat{\mu})C\widehat{x} \\ &= \mathcal{R}(\widehat{\mu})\left[\, I_n \;\; 0 \,\right]\widehat{x}_{\mathrm{L}} + \sigma e(\widehat{\mu})C\widehat{x} = G\mathcal{L}(\widehat{\mu})\widehat{x}_{\mathrm{L}} + \sigma e(\widehat{\mu})C\widehat{x}, \end{aligned} \tag{4.2}$$

where for the last equality, we used the identity

$$\mathcal{R}(\widehat{\mu})\left[\, I_n \;\; 0 \,\right] = G\mathcal{L}(\widehat{\mu}) \quad \text{with} \quad G = \left[\, I_n \;\; -E_{\sigma_1}(I - \widehat{\mu} I_\ell \otimes D_m)^{-1} \,\right].$$

Then by (4.2), we have the bound

$$\begin{aligned} \frac{\|\mathcal{Q}(\widehat{\lambda})\widehat{x}\|}{\|\widehat{x}\|} &\leqslant \|G\| \, \frac{\|\mathcal{L}(\widehat{\mu})\widehat{x}_{\mathrm{L}}\|}{\|\widehat{x}\|} + |\sigma e(\widehat{\mu})| \, \frac{\|C\widehat{x}\|}{\|\widehat{x}\|} \\ &\leqslant \|G\| \frac{\|\widehat{x}_{\mathrm{L}}\|}{\|\widehat{x}\|} \, \frac{\|\mathcal{L}(\widehat{\mu})\widehat{x}_{\mathrm{L}}\|}{\|\widehat{x}_{\mathrm{L}}\|} + |\sigma e(\widehat{\mu})| \frac{\|C\widehat{x}\|}{\|\widehat{x}\|}. \end{aligned} \tag{4.3}$$

DOI: 10.1002/nme

In terms of the backward errors $\eta_{\mathrm{Q}}(\widehat{\lambda},\widehat{x})$ and $\eta_{\mathrm{L}}(\widehat{\mu},\widehat{x}_{\mathrm{L}})$, the inequality (4.3) can be written as

$$\eta_{\mathrm{Q}}(\widehat{\lambda},\widehat{x}) \leqslant \alpha\, \eta_{\mathrm{L}}(\widehat{\mu},\widehat{x}_{\mathrm{L}}) + \beta, \tag{4.4}$$

where α and β are given by

$$\alpha = \|G\| \frac{\|\widehat{x}_{\mathrm{L}}\|}{\|\widehat{x}\|} \frac{\varphi(\widehat{\mu})}{\rho(\widehat{\lambda})} \quad \text{and} \quad \beta = \frac{|\sigma e(\widehat{\mu})|}{\rho(\widehat{\lambda})} \frac{\|C\widehat{x}\|}{\|\widehat{x}\|}. \tag{4.5}$$

The quantity α is an error growth factor from the solution of the LEP (3.7) to the solution of the QEP (1.1). Later, we will show how to reduce α via a proper scaling scheme. The quantity β is dominated by the Padé approximation error $e(\widehat{\mu})$, which is small in practice as we have discussed in Section 3.1. Another contributing factor to make the term β even smaller is the quantity $\|C\widehat{x}\|$. If $C\widehat{x} = 0$, then the approximate eigenpair $(\widehat{\lambda},\widehat{x})$ of QEP (1.1) is also an approximate eigenpair of the undamped eigenvalue problem $(\lambda^2 M + K)x = 0$. In this case, the bound (4.4) implies that there is no Padé approximation error contributing to the overall error. In summary, the upper bound (4.4) indicates that in order to have an accurate approximation of the QEP (1.1) by the LEP (3.7), the Padé approximation error β should be within the desired threshold, and the error growth factor α be bounded and small.

4.2. Scaling

It is a common practice to use a proper scaling scheme to the QEP for obtaining an LEP with a better condition number and smaller backward error [2, 26–29]. A popular scaling scheme is to scale the QEP by a pair of parameters ω and ζ such that the coefficient matrices of the following scaled QEP have nearly unit 2-norms

$$\mathcal{Q}_{\mathrm{s}}(\lambda_{\mathrm{s}})x_{\mathrm{s}} \equiv (\lambda_{\mathrm{s}}^2 M_{\mathrm{s}} + \lambda_{\mathrm{s}} C_{\mathrm{s}} + K_{\mathrm{s}})x_{\mathrm{s}} = 0, \tag{4.6}$$

where $\lambda_{\mathrm{s}} = \omega^{-1}\lambda$, $M_{\mathrm{s}} = \omega^2\zeta M$, $C_{\mathrm{s}} = \omega\zeta C$, and $K_{\mathrm{s}} = \zeta K$. If the shift σ for $\mathcal{Q}_{\mathrm{s}}$ is applied, then we should use the scaled shift $\sigma_{\mathrm{s}} = \omega^{-1}\sigma$. It is shown [2, 26, 30] that with the choice of scaling parameters

$$\omega = (\|K\|/\|M\|)^{1/2} \quad \text{and} \quad \zeta = 2(\|K\| + \omega\|C\|)^{-1}, \tag{4.7}$$

the companion form linearization of the scaled QEP (4.6) generally yields a better conditioned LEP.

Applying the PAL algorithm to the scaled QEP (4.6), we obtain the following scaled LEP

$$\mathcal{L}_{\mathrm{s}}(\mu_{\mathrm{s}})x_{\mathrm{L}_{\mathrm{s}}} \equiv (A_{\mathrm{s}} - \mu_{\mathrm{s}} B_{\mathrm{s}})x_{\mathrm{L}_{\mathrm{s}}} = 0, \tag{4.8}$$

where

$$A_{\mathrm{s}} = \begin{bmatrix} \zeta(K_\sigma + \sigma d C) & \sqrt{\zeta} E_{\sigma_1} \\ \sqrt{\zeta} F_{\sigma_2}^{\mathrm{T}} & I_{\ell m} \end{bmatrix} \quad \text{and} \quad B_{\mathrm{s}} = \begin{bmatrix} \zeta M_\sigma & \\ & I_\ell \otimes D_m \end{bmatrix}.$$

If $(\widehat{\mu}_{\mathrm{s}}, \widehat{x}_{\mathrm{L}_{\mathrm{s}}})$ is an approximate eigenpair of the LEP (4.8), then $(\widehat{\lambda}_{\mathrm{s}}, \widehat{x}_{\mathrm{s}}) = (\sigma_{\mathrm{s}}\sqrt{\widehat{\mu}_{\mathrm{s}} + 1}, \widehat{x}_{\mathrm{L}_{\mathrm{s}}}(1\!:\!n))$ is an approximate eigenpair of the scaled QEP (4.6). Subsequently,

$$\widehat{\lambda} = \omega\widehat{\lambda}_{\mathrm{s}} = \omega\sigma_{\mathrm{s}}\sqrt{\widehat{\mu}_{\mathrm{s}} + 1} = \sigma\sqrt{\widehat{\mu}_{\mathrm{s}} + 1} = f_\sigma(\widehat{\mu}_{\mathrm{s}}) \quad \text{and} \quad \widehat{x} = \widehat{x}_{\mathrm{s}} \tag{4.9}$$

are an approximate eigenpair of the original QEP (1.1).

We observe that the scaled LEP (4.8) does not depend on the scaling parameter ω and neither does the approximate eigenpair $(\widehat{\lambda},\widehat{x})$ of the QEP. Furthermore, if the eigenpair $(\widehat{\mu}_{\mathrm{s}}, \widehat{x}_{\mathrm{L}_{\mathrm{s}}})$ of the scaled LEP (4.8) is computed with the backward error `rtol`, then by (4.4), we have

$$\eta_{\mathrm{Q}}(\widehat{\lambda},\widehat{x}) \leqslant \alpha_{\mathrm{s}} \cdot \mathtt{rtol} + \beta, \tag{4.10}$$

where

$$\alpha_{\mathrm{s}} = \|G_{\mathrm{s}}\| \frac{\varphi_{\mathrm{s}}(\widehat{\mu}_{\mathrm{s}})}{\zeta\rho(\widehat{\lambda})} \frac{\|\widehat{x}_{\mathrm{L_s}}\|}{\|\widehat{x}\|}, \tag{4.11}$$

and $\varphi_{\mathrm{s}}(\widehat{\mu}_{\mathrm{s}}) = \|A_{\mathrm{s}}\| + |\widehat{\mu}_{\mathrm{s}}|\|B_{\mathrm{s}}\|$, and $G_{\mathrm{s}} = [I_n, \; -\sqrt{\zeta} E_{\sigma_1}(I_{\ell m} - \widehat{\mu}_{\mathrm{s}} I_\ell \otimes D_m)^{-1}]$.

By (4.10), we see that to reduce backward error η_{Q}, the scaling parameter ζ should be chosen to yield a small growth factor α_{s}. Toward this goal, we have the following theorem to give an upper bound of α_{s}.

Theorem 4
Let the rank-revealing decomposition $C = EF^{\mathrm{T}}$ and the shift splitting $\sigma = \sigma_1\sigma_2$ be chosen such that

$$|\sigma_1|\|E\| = |\sigma_2|\|F\| = \sqrt{|\sigma|\,\|C\|}. \tag{4.12}$$

Then with the scaling parameter

$$\zeta = \frac{1}{\max\{\|\sigma^2 M\|, 2m\|\sigma C\|, \|K\|\}}, \tag{4.13}$$

we have

$$\alpha_{\mathrm{s}} \leqslant \left(\frac{4m\tau}{\tau+2} + 2 + |\widehat{\mu}_{\mathrm{s}}|\right)\left(\frac{1+\delta^2}{1-\delta\nu}\right)\frac{\rho(\sigma)}{\rho(\widehat{\lambda})}, \tag{4.14}$$

where $\tau = \|C\|/\sqrt{\|M\|\|K\|}$, $\delta = \|(I_{\ell m} - \widehat{\mu}_{\mathrm{s}} I_\ell \otimes D_m)^{-1}\|$, and $\nu = \|\mathcal{L}_{\mathrm{s}}(\widehat{\mu}_{\mathrm{s}})\widehat{x}_{\mathrm{L_s}}\|/\|\widehat{x}_{\mathrm{L_s}}\|$.

Proof
To show the upper bound (4.14), we start with the definition (4.11) of α_{s}. For the term $\|G_{\mathrm{s}}\|$ of α_{s}, we have

$$\|G_{\mathrm{s}}\|^2 \leqslant 1 + \|\sqrt{\zeta} E_{\sigma_1}\|^2 \|(I_{\ell m} - \widehat{\mu}_{\mathrm{s}} I_\ell \otimes D_m)^{-1}\|^2. \tag{4.15}$$

Because of the assumption (4.12) and the choice of the scaling ζ as in (4.13), we have

$$\|\sqrt{\zeta} E_{\sigma_1}\| = \|\sqrt{\zeta}\sigma_1 E(I_\ell \otimes a^{\mathrm{T}})\| \leqslant \sqrt{\zeta}|\sigma_1|\|E\|\|a\| = \sqrt{\zeta 2m\|\sigma C\|} < 1, \tag{4.16}$$

where we used the identity $\|I_\ell \otimes a^{\mathrm{T}}\| = \|a\| = (2m)^{1/2}$.[‖] Therefore, by (4.15) and the definition of δ, we have

$$\|G_{\mathrm{s}}\| \leqslant \sqrt{1+\delta^2}. \tag{4.17}$$

For the second quantity $\varphi_{\mathrm{s}}(\widehat{\mu}_{\mathrm{s}})/\zeta\rho(\widehat{\lambda})$ of α_{s}, let us first bound $\|A_{\mathrm{s}}\|$ and $\|B_{\mathrm{s}}\|$.

$$\begin{aligned}
\|A_{\mathrm{s}}\| &\leqslant 2\max\left\{1,\; \sqrt{\zeta}\|E_{\sigma_1}\|,\; \sqrt{\zeta}\|F_{\sigma_2}\|,\; \zeta\|K_\sigma + \sigma d C\|\right\} \\
&\leqslant 2\max\{1, \zeta\|K_\sigma + \sigma d C\|\} \\
&\leqslant 2\max\{1, \zeta\left(|\sigma|^2\|M\| + (2m+1)|\sigma|\|C\| + \|K\|\right)\} \\
&= 2\max\{1, \zeta\left(\rho(\sigma) + 2m|\sigma|\|C\|\right)\} \\
&= 2\zeta\rho(\sigma) + 4m\zeta|\sigma|\|C\|,
\end{aligned} \tag{4.18}$$

[‖] Note that the identity $\sum_{j=1}^{m} \tan^2 \frac{j\pi}{2m+1} \equiv 2m^2 + m$; see [31].

DOI: 10.1002/nme

where for the first inequality, we repeatedly apply the inequality $\|[A_1, A_2]\| \leqslant \sqrt{2}\max\{\|A_1\|, \|A_2\|\}$. For the second inequality, we use the inequalities (4.16) and $\|\sqrt{\zeta}F_{\sigma_2}\| \leqslant 1$, which is derived by using an analogous derivation of (4.16). The last equality uses the choice of scaling parameter ζ. Meanwhile, the choice of scaling ζ yields

$$\|B_{\rm s}\| \leqslant \max\left\{1,\ (|\sigma|^2\|M\|)\zeta\right\} = 1. \tag{4.19}$$

Combining (4.18) and (4.19), we have

$$\begin{aligned}\varphi_{\rm s}(\widehat{\mu}_{\rm s}) &= \|A_{\rm s}\| + |\widehat{\mu}_{\rm s}|\|B_{\rm s}\| \\ &\leqslant 2\zeta\rho(\sigma) + 4m\zeta|\sigma|\|C\| + |\widehat{\mu}_{\rm s}| \\ &\leqslant 2\zeta\rho(\sigma)\left(1 + \frac{2m\tau}{\tau+2}\right) + |\widehat{\mu}_{\rm s}|,\end{aligned} \tag{4.20}$$

where for the last inequality, we use the inequality

$$\frac{|\sigma|\|C\|}{\rho(\sigma)} = \frac{|\sigma|}{|\sigma^2|\|M\|/\|C\| + \|K\|/\|C\| + |\sigma|} \leqslant \frac{|\sigma|}{2|\sigma|/\tau + |\sigma|} = \frac{\tau}{\tau+2}.$$

Dividing the inequality (4.20) by $\zeta\rho(\widehat{\mu}_{\rm s})$ on both sides gives rise to

$$\frac{\varphi_{\rm s}(\widehat{\mu}_{\rm s})}{\zeta\rho(\widehat{\lambda})} \leqslant \left(2\left(1+\frac{2m\tau}{\tau+2}\right) + \frac{|\widehat{\mu}_{\rm s}|}{\zeta\rho(\sigma)}\right)\frac{\rho(\sigma)}{\rho(\widehat{\lambda})} \leqslant \left(2 + \frac{4m\tau}{\tau+2} + |\widehat{\mu}_{\rm s}|\right)\frac{\rho(\sigma)}{\rho(\widehat{\lambda})}, \tag{4.21}$$

where the second inequality uses the inequality $\zeta\rho(\sigma) \geqslant 1$.

Finally, we bound the quantity $\|\widehat{x}_{\rm L_s}\|/\|\widehat{x}\|$ of $\alpha_{\rm s}$. By the definition $\widehat{x} = \widehat{x}_{\rm L_s}(1\!:\!n)$, it holds that

$$\widehat{x}_{\rm L_s} = H_{\rm s}\widehat{x} + \begin{bmatrix} 0 \\ (I_{\ell m} - \widehat{\mu}_{\rm s} I_\ell \otimes D_m)^{-1}[0, I_{\ell m}]\mathcal{L}_{\rm s}(\widehat{\mu}_{\rm s})\widehat{x}_{\rm L_s}. \end{bmatrix},$$

where

$$H_{\rm s} = \begin{bmatrix} I_n \\ -\sqrt{\zeta}(I_{\ell m} - \widehat{\mu}_{\rm s} I_\ell \otimes D_m)^{-1}F_{\sigma_2}^{\rm T} \end{bmatrix}.$$

Therefore, we have

$$\frac{\|\widehat{x}_{\rm L_s}\|}{\|\widehat{x}\|} \leqslant \|H_{\rm s}\| + \|(I_{\ell m} - \widehat{\mu}_{\rm s} I_\ell \otimes D_m)^{-1}\|\frac{\|\mathcal{L}_{\rm s}(\widehat{\mu}_{\rm s})\widehat{x}_{\rm L_s}\|}{\|\widehat{x}_{\rm L_s}\|}\frac{\|\widehat{x}_{\rm L_s}\|}{\|\widehat{x}\|}, \tag{4.22}$$

which yields

$$\frac{\|\widehat{x}_{\rm L_s}\|}{\|\widehat{x}\|} \leqslant \frac{\|H_{\rm s}\|}{1-\delta\nu} \leqslant \frac{\sqrt{1+\delta^2}}{1-\delta\nu}, \tag{4.23}$$

where the bound $\|H_{\rm s}\| \leqslant \sqrt{1+\delta^2}$ can be derived similarly to the derivation for the upper bound (4.17) of $\|G_{\rm s}\|$.

Combining (4.17), (4.21), and (4.23), we have the bound (4.14). □

We note that because the eigenvalues $\widehat{\mu}_{\rm s}$ of interest of the scaled LEP (4.8) are small, i.e., $|\widehat{\mu}_{\rm s}| \approx 0$, then $\delta \approx 1$ and $\rho(\widehat{\lambda}) \approx \rho(\sigma)$. Consequently, if the scaled LEP (4.8) has been solved with the residual norm $\nu \ll 1$, then the bound (4.14) is simplified to

$$\alpha_{\mathrm{s}} \lesssim 4\left(\frac{2m\tau}{\tau+2}+1\right).$$

Moreover, if $\tau \ll 1$, known as a heavily underdamped system, then $\alpha_{\mathrm{s}} \lesssim 4$. In addition, we note that the assumption (4.12) is mild in practice and will be discussed in detail in Section 5.

5. PADÉ APPROXIMATE LINEARIZATION ALGORITHM

Algorithm 1 is a complete description of the PAL algorithm for computing a few eigenpairs of the QEP (1.1) around the prescribed shift σ.

Algorithm 1 PAL

```
Initialize
  (a) the shift $\sigma \neq 0$
  (b) $k$ for the desired number of eigenpairs, and rtol for the backward error tolerance
  (c) the order $m$ of Padé approximant $r_m(\mu)$
Compute the scaling factor $\zeta$ by (4.13)
Compute the shift splitting $\sigma = \sigma_1\sigma_2$ to satisfy the condition (4.12)
Compute the LU factorization of $\mathcal{Q}(\sigma)$
Compute the $k$ smallest (in modulus) eigenpairs $(\widehat{\mu}_{\mathrm{s}}, \widehat{x}_{\mathrm{L_s}})$ of the scaled LEP (4.8) with the
  backward errors $\eta_{\mathrm{L_s}}(\widehat{\mu}_{\mathrm{s}}, \widehat{x}_{\mathrm{L_s}}) \leqslant$ rtol
Discard those $\widehat{\mu}_{\mathrm{s}}$ which coincide with the poles of $r_m(\mu)$
Compute the approximate eigenpairs $(\widehat{\lambda}, \widehat{x}) = \left(\sigma\sqrt{\widehat{\mu}_{\mathrm{s}}+1}, \widehat{x}_{\mathrm{L_s}}(1:n)\right)$ of the QEP (1.1) and
  the corresponding backward errors $\eta_{\mathrm{Q}}(\widehat{\lambda}, \widehat{x})$
```

To apply the proposed scaling parameter ζ in (4.13), we assume that the rank-revealing decomposition (1.2) and the shift splitting $\sigma = \sigma_1\sigma_2$ are chosen to satisfy the assumption (4.12). If C is symmetric positive semi-definite, then $E = F$ in the rank-revealing factorization (1.2) of C. We can then select $\sigma_1 = \sigma_2 = \sqrt{\sigma}$. In general, given the rank-revealing decomposition (1.2), one can compute the QR factorization $E = QR$, where Q is $n \times \ell$ orthogonal and R is $\ell \times \ell$, then with an updated rank-revealing factorization of C with $E = Q$ and $F := FR^{\mathrm{T}}$, we can let $\sigma_1 = \sqrt{\sigma\|F\|}$ and $\sigma_2 = \sqrt{\sigma/\|F\|}$ to satisfy the assumption (4.12).

To solve the scaled LEP (4.8) by an iterative solver, such as the Arnoldi method [32], we need to provide the product of the matrix $A_{\mathrm{s}}^{-1}B_{\mathrm{s}}$ with an arbitrary vector u, that is,

$$v = A_{\mathrm{s}}^{-1}B_{\mathrm{s}}u. \tag{5.1}$$

By exploiting the structure of A_{s}, we can implement the matrix–vector product efficiently. Specifically, we first note that the matrix A_{s} can be factorized as

$$A_{\mathrm{s}} = \begin{bmatrix} I_n & \sqrt{\zeta}E_{\sigma_1} \\ & I_{\ell m} \end{bmatrix} \begin{bmatrix} \zeta\left(K_\sigma + \sigma d C - E_{\sigma_1}F_{\sigma_2}^{\mathrm{T}}\right) & \\ & I_{\ell m} \end{bmatrix} \begin{bmatrix} I_n & \\ \sqrt{\zeta}F_{\sigma_2}^{\mathrm{T}} & I_{\ell m} \end{bmatrix}. \tag{5.2}$$

By the identity (3.5) and $r_m(0) = 1$, we have

$$\begin{aligned} K_\sigma + \sigma d C - E_{\sigma_1}F_{\sigma_2}^{\mathrm{T}} &= K_\sigma + \sigma d C - E_{\sigma_1}(I_{\ell m} - 0 \cdot I_\ell \otimes D_m)^{-1}F_{\sigma_2}^{\mathrm{T}} \\ &= K_\sigma + \sigma r_m(0)C = \sigma^2 M + \sigma C + K = \mathcal{Q}(\sigma). \end{aligned}$$

Therefore, the inverse of A_{s} is given by

$$A_{\mathrm{s}}^{-1} = \begin{bmatrix} I_n & \\ -\sqrt{\zeta}F_{\sigma_2}^{\mathrm{T}} & I_{\ell m} \end{bmatrix} \begin{bmatrix} \mathcal{Q}(\sigma)^{-1}/\zeta & \\ & I_{\ell m} \end{bmatrix} \begin{bmatrix} I_n & -\sqrt{\zeta}E_{\sigma_1} \\ & I_{\ell m} \end{bmatrix}.$$

If vectors $v = \begin{bmatrix} v_1^{\mathrm{T}} & v_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $u = \begin{bmatrix} u_1^{\mathrm{T}} & u_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ are partitioned to be conformal with the blocks of matrices A_{s} and B_{s}, then the matrix–vector product (5.1) can be computed by the following formulae:

$$\begin{aligned} v_1 &= (\mathcal{Q}(\sigma)^{-1}/\zeta)\left(\zeta M_\sigma u_1 - \sqrt{\zeta}E_{\sigma_1}(I_\ell \otimes D_m)u_2\right) \\ &= -\mathcal{Q}(\sigma)^{-1}\left(\sigma^2 M u_1 + (\sigma_1/\sqrt{\zeta})E(I_\ell \otimes a^{\mathrm{T}}D_m)u_2\right), \end{aligned} \qquad (5.3a)$$

$$\begin{aligned} v_2 &= (I_\ell \otimes D_m)u_2 - \sqrt{\zeta}F_{\sigma_2}^{\mathrm{T}}v_1 \\ &= (I_\ell \otimes D_m)u_2 - \sqrt{\zeta}\sigma_2(I_\ell \otimes a)F^{\mathrm{T}}v_1, \end{aligned} \qquad (5.3b)$$

where the identity $(A \otimes B)(C \otimes D) = AC \otimes BD$ is used in (5.3a) for matrices A, B, C, and D of sizes that the matrix products AC and BD are defined.

By (5.3), we can compute the LU factorization of $\mathcal{Q}(\sigma)$ once and then apply it for the matrix–vector multiplication with $\mathcal{Q}(\sigma)^{-1}$. Hence, the PAL algorithm takes about the same amount of work as the direct linearization in terms of the matrix–vector products in an iterative LEP solver.

6. NUMERICAL EXAMPLES

In this section, we present three numerical examples to demonstrate the accuracy and efficiency of the PAL algorithm. The accuracy of a computed eigenpair $(\widehat{\lambda}, \widehat{x})$ is measured by the QEP normwise backward error $\eta_{\mathrm{Q}}(\widehat{\lambda}, \widehat{x})$ defined in (4.1), where 1-norm $\|\cdot\|_1$ is used for computing the norms of matrices M, C, and K.

In our MATLAB implementation of the PAL algorithm, we use the functions `eig` or `eigs` to solve the LEP (4.8). The function `eigs` is an implementation of the implicitly restarted Arnoldi method (IRAM) [33]. We use the function `lu` for computing the LU factorization of $\mathcal{Q}(\sigma)$. For sparse matrices, the function `lu` is from UMFPACK [34]. The testing data are collected on a Dell computer with an Intel(R) Dual Core(TM) 2.20 GHz i7-3632QM CPU and 6-GB RAM.

We have implemented the PAL algorithm in C++. For comparison, we have also implemented a direct linearization (DLIN) algorithm in C++. The DLIN algorithm is based on the linearization (1.4) and the two-parameter scaling (4.7) computed with the matrix 1-norm. The LEPs (1.4) and (4.8) are solved by using `ARPACK++` [35], which is based on the IRAM [33]. We use the default parameters provided in `ARPACK++`. Specifically, the number of Lanczos vectors is $p = 2k + 1$ with k being the number of eigenvalues required. The residual error tolerance `rtol` is the machine precision. The sparse LU factorization of $\mathcal{Q}(\sigma)$ is computed using `SuperLU` [36] with a threshold pivoting parameter $u = 0.1$ to control numerical stability. The testing data are collected on a cluster with two Intel Xeon X5670 2.93-GHz CPUs and 94-GB RAM. No parallelization is attempted.

Example 1

In this example, we demonstrate numerical accuracy of the PAL algorithm and effectiveness of the backward error bound (4.10) with the scaling scheme (4.13). We use a QEP arising from the vibration analysis of a slender beam supported at both ends and damped at the midpoint [2, 37]. The $n \times n$ mass and stiffness matrices M and K are positive definite. The damping matrix C has only one nonzero positive entry at the center position $(n/2, n/2)$. Therefore, the rank of C is 1, $\ell = 1$, and has the decomposition $C = EE^{\mathrm{T}}$, where $E = \delta^{\frac{1}{2}}e_{n/2}$, $\delta > 0$, and $e_{n/2}$ is the unit column vector with only one entry at the position $n/2$ and zeros at the others. It is known [2] that half of the eigenvalues in this example is pure imaginary and is eigenvalues of the undamped problem $(\lambda^2 M + K)x = 0$, so the corresponding eigenvectors satisfy $Cx = 0$. PAL will introduce no truncation errors for these eigenpairs.

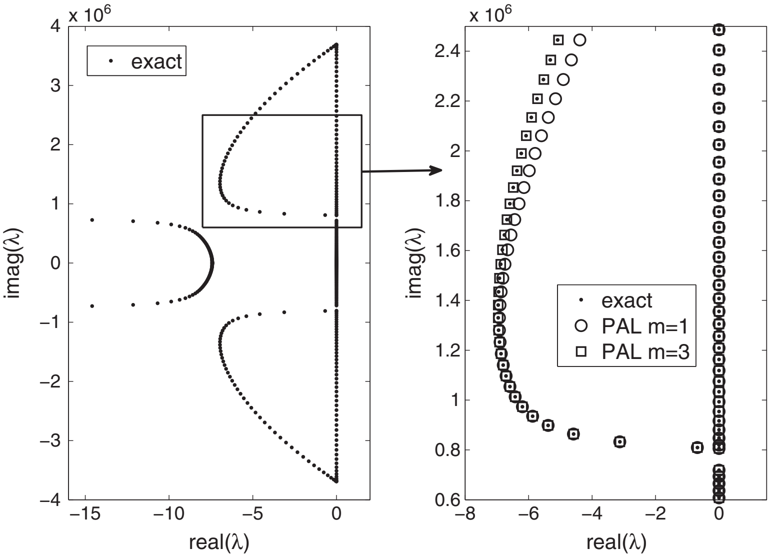

Figure 4. Left: 'exact' eigenvalues. Right: a fraction of exact eigenvalues and approximated eigenvalues by PAL.

To demonstrate the accuracy of the PAL algorithm, we take the dimension $n = 200$ and $\delta = 5$. It is an underdamped QEP with $\tau \approx 0.0153598$. By the analysis in Section 4.2, we expect the error growth factor $\alpha_{\mathrm{s}} \leqslant 4$. The left plot in Figure 4 shows all eigenvalues computed by the MATLAB function `polyeig` with the scaling strategy (4.7).

Let us compute a few eigenvalues of the QEP around the shift $\sigma = 10^6 i$. With the order $m = 1$ of the Padé approximant $r_m(\mu)$ and scaling parameters $\sigma_1 = \sigma_2 = \sqrt{\sigma}$, the PAL leads to the LEP (4.8) of dimension $n_{\mathrm{L}} = n + \ell m = 201$. The right plot of Figure 4 shows some of the computed eigenvalues by the PAL. The following table is a profile of six selected eigenvalues and the corresponding backward errors of the LEP and the QEP:

No.	$\mathrm{Re}(\widehat{\lambda})$	$\mathrm{Im}(\widehat{\lambda}/10^6)$	$\eta_{\mathrm{L_s}}(\widehat{\mu}_{\mathrm{s}}, \widehat{x}_{\mathrm{L_s}})$	$\eta_{\mathrm{Q}}(\widehat{\lambda}, \widehat{x})$
1	$+4.787700 \times 10^{-7}$	0.993105	7.87×10^{-16}	6.44×10^{-16}
2	$+2.828370 \times 10^{-7}$	1.573793	5.68×10^{-16}	5.21×10^{-16}
3	-9.193417×10^{-6}	2.097337	5.53×10^{-16}	5.27×10^{-16}
4	-6.423440	1.013141	1.26×10^{-15}	8.55×10^{-14}
5	-6.745303	1.545041	6.22×10^{-16}	1.71×10^{-9}
6	-5.595220	2.060988	5.80×10^{-16}	4.06×10^{-9}

Furthermore, the following table shows the corresponding error bound (4.10):

No.	α_{s}	$\lvert e(\widehat{\mu}_{\mathrm{s}})\rvert$	$\lVert C\widehat{x}\rVert / \lVert\widehat{x}\rVert$	β	$\alpha_{\mathrm{s}} \cdot \eta_{\mathrm{L_s}} + \beta_{\mathrm{s}}$
1	0.844	8.22×10^{-8}	1.19×10^{-13}	1.16×10^{-24}	6.65×10^{-16}
2	0.919	3.45×10^{-2}	1.49×10^{-13}	2.78×10^{-19}	5.23×10^{-16}
3	0.952	1.79×10^{-1}	2.97×10^{-13}	1.69×10^{-18}	5.29×10^{-16}
4	0.828	5.64×10^{-7}	1.32×10^{-3}	8.55×10^{-14}	8.67×10^{-14}
5	0.916	3.01×10^{-2}	1.02×10^{-3}	1.71×10^{-9}	1.71×10^{-9}
6	0.951	1.65×10^{-1}	7.49×10^{-4}	4.06×10^{-9}	4.06×10^{-9}

where the α_{s} values are computed by the definition (4.11).

DOI: 10.1002/nme

We observe that the first three approximate the pure imaginary eigenvalues of the original QEP. Here, the PAL algorithm introduces nearly no error as shown by β and η_Q values. In particular, note that the second and third eigenvalues, although the Padé errors $|e(\widehat{\mu})|$ are not small, $\|C\widehat{x}\|/\|\widehat{x}\|$ are small. Furthermore, we observe that for all six eigenvalues, $\eta_{\mathrm{Q}} \approx \alpha_{\mathrm{s}} \cdot \eta_{\mathrm{L_s}} + \beta$, which suggests that the error bound in (4.10) is tight. In particular, for the last three eigenvalues, we actually have $\eta_Q \approx \beta$. The errors of these computed eigenvalues are dominated by the Padé approximation errors. To improve the accuracy, we use a higher Padé order $m = 9$. It leads to an LEP of dimension $n_{\mathrm{L}} = 209$. Consequently, η_Q for the last three approximate eigenvalues are all reduced to the machine precision, namely, about 10^{-16}, although $\|C\widehat{x}\|/\|\widehat{x}\|$ remains unchanged.

Example 2

This example shows the computational efficiency of the PAL algorithm. We consider an acoustic wave problem to model acoustic pressure in a two-dimensional bounded domain with boundary conditions that are partly pressure release and partly impedance [4]. By the finite element discretization of the wave equation on the unit square $[0, 1] \times [0, 1]$ with mesh size h, it leads to the QEP (1.1) of dimension $n = q(q-1)$, where $q = 1/h$. The mass and stiffness matrices M and K are both symmetric positive definite. The rank of the damping matrix $C = EE^{\mathrm{T}}$ is $q - 1$, where $E = (h/\xi)^{\frac{1}{2}} I_{q-1} \otimes e_q$, and ξ is an impedance parameter. This example is available in the NLEVP collection [37] labeled as `acoustic_wave_2d`.

To show the computational efficiency of PAL, we consider $h = 1/500$ and impedance $\xi = 1$. Consequently, the QEP has the dimension $n = q(q-1) = 249500$, and the damping matrix C has the rank $\ell = 499$. We compute $k = 300$ eigenvalues close to the shift $\sigma = 2\sqrt{2}q\mathtt{i}$. The order of the Padé approximant $r_m(\mu)$ is chosen to be $m = 3$. Consequently, with the scaling parameters $\sigma_1 = \sigma_2 = \sqrt{\sigma}$, the LEP (4.8) has the dimension $n_{\mathrm{L}} = n + \ell m = 250997$, which is only slightly larger than n but much smaller than the dimension $2n = 499000$ of the LEP (1.4) produced by the direct linearization.

The IRAM of ARPACK++ takes three update iterations (or two restarts) to converge for the LEPs (1.4) and (4.8). The computed eigenvalues and their corresponding backward errors are shown in Figure 5. As we can see, there are high agreements between the two linearizations in terms of computed eigenvalues and backward errors. The computational costs of ARPACK++ are dominated by four parts, namely, (1) sparse matrix–vector multiplications (SpMVs); (2) Gram–Schmidt process to maintain the orthogonality of basis vectors of the projection subspaces; (3) the eigenvector computation; and (4) costs of updating, such as restarting processes and solving small Hessenberg eigenvalue subproblems. The following table profiles the CPU time for each of these four parts:

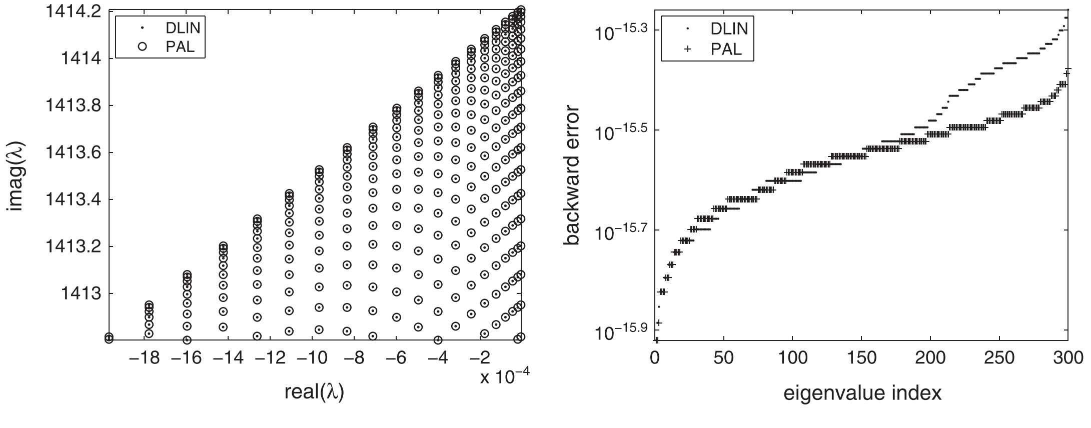

Figure 5. Left: computed eigenvalues. Right: backward errors $\eta_Q(\widehat{\lambda}, \widehat{x})$.

DOI: 10.1002/nme

	SpMVs	GS	EvComp	Updating	Subtotal
DLIN	168.95	1130.44	314.65	304.00	1918.04
PAL	162.26	562.37	137.66	156.10	1018.39

From the aforementioned table, we see that the SpMV costs for the two linearizations are almost the same, which confirm the discussion in Section 5. The bulk of computational time lies in the Gram–Schmidt orthogonalization process, where PAL reduces the cost by almost half. By adding 6.41 s for the LU factorization of $\mathcal{Q}(\sigma)$ and other setting up costs, the total CPU elapsed time of the DLIN method is 1931.89 s. On the other hand, the PAL algorithm is 1028.54 s. PAL runs 47% faster than DLIN.

Example 3

This is a modest industrial size QEP arising from the automobile industry to analyze the modal frequency responses of a car body [38]. The QEP size is $n = 655812$. Matrices M, C, and K are real symmetric with the numbers of nonzero elements being 394,508, 294, and 31,775,679, respectively. The damping matrix C is extremely sparse. Indeed, the nonzero elements of C form a 144-by-144 principal symmetric positive semi-definite submatrix. We first compute the eigenvalue decomposition of this submatrix and then truncate these eigenvalues whose magnitude is less than $10^{-16} \times \lambda_{\max}$ to obtain the rank-revealing factorization $C = EE^{\mathrm{T}}$, where the rank of E is $\ell = 126$, and $\lambda_{\max}$ is the largest eigenvalue of the submatrix.

We compute 300 eigenvalues near the shift $\sigma = 200\pi\,\mathtt{i}$. The order of the Padé approximant $r_m(\mu)$ is chosen to be $m = 3$. The scaling parameters $\sigma_1 = \sigma_2 = \sqrt{\sigma}$. The PAL algorithm leads to the LEP (4.8) of size $n_{\mathrm{L}} = n + \ell m = 656,190$. In contrast, the size of the LEP (1.4) by the direct linearization is $2n = 1,311,624$.

The IRAM takes two iterations to converge for the LEPs (1.4) and (4.8). The computed eigenvalues are shown in Figure 6. Most of the computed eigenvalues by the two linearizations are close to the imaginary axis. Zooming into this part, we can see that the PAL computes few more eigenvalues of small modulus, while DLIN computes more of large modulus. The backward errors of the eigenpairs are shown in the left plot of Figure 6. PAL has better numerical stability properties than DLIN.

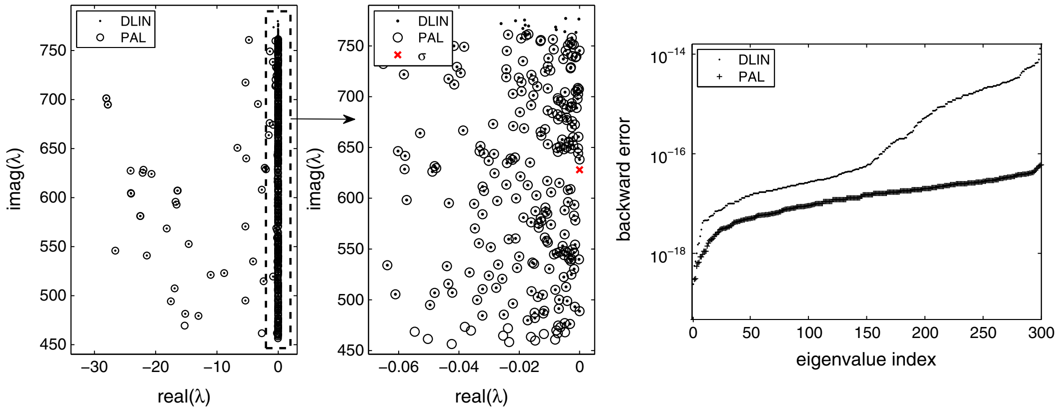

Figure 6. Left: computed eigenvalues. Right: backward errors $\eta_{\mathcal{Q}}(\widehat{\lambda}, \widehat{x})$ of the eigenpairs.

DOI: 10.1002/nme

The following table profiles the CPU timing of the four parts of ARPACK++ for solving the LEPs.

	SpMV	GS	EvComp	Update	Subtotal
DLIN	1305.94	2277.01	791.25	393.81	4768.01
PAL	1297.92	1146.89	394.37	202.43	3459.99

The SpMV cost is high in this example but is almost the same for PAL and DLIN. The bulk of computational time still lies in the Gram–Schmidt orthogonalization process and PAL reduces it by almost half. By adding 408.59 s for computing the LU factorization of $\mathcal{Q}(\sigma)$, and other setting up costs, the total CPU elapsed time of DLIN is 5196.12 s. On the other hand, PAL is 3459.99 s. PAL runs 33.4% faster than DLIN.

7. CONCLUDING REMARKS

We presented the PAL algorithm to solve the QEP with low-rank damping. The PAL algorithm combines Padé approximation and the trimmed linearization and produces an LEP with slightly larger dimension than the original QEP. Numerical experiments have demonstrated the accuracy and saving in memory and computational time comparing with the direct linearization. One interesting future research problem is to determine the Padé approximation order m adaptively based on the desired accuracy.

It is still an open problem how to efficiently exploit the low-rank property in eigenvalue computation. Recently, in [39], an algorithm was proposed to compute all eigenpairs of the QEP with low-rank damping. However, because of the computational complexity, it is not designed for solving large scale problems.

The PAL algorithm proposed in this paper can be naturally extended to computing NEPs of the form

$$\left[K - \lambda M + \sum_{\ell=1}^{\mathrm{L}} f_\ell(\lambda) C_\ell\right] x = 0,$$

where $f_\ell(\lambda)$ are nonlinear functions in λ, C_ℓ are low-rank matrices. Such NEPs are found, for example, in the cavity design of a linear accelerator [40]. To solve this problem, one can first generate an approximate REP by replacing $f_\ell(\lambda)$ with properly chosen Padé approximants then apply the trimmed linearization. This would fall in the same idea as recently proposed algorithm in [21]. It is a subject of further study for theoretical and numerical comparisons of these two approaches.

ACKNOWLEDGEMENTS

Part of this work was carried out while Xing Huang and Ding Lu were visiting at the University of California, Davis, supported by China Scholarship Council. Bai is supported in part by the National Science Foundation grant DMS-1115817. Su is supported in part by the Innovation Program of Shanghai Municipal Education Commission 13zz007, E-Institutes of Shanghai Municipal Education Commission N.E303004, and NSFC key project 91330201.

We would like to thank Dr. Benshan Liao of SIMENS for the QEP data of Example 3. We are grateful to the anonymous referees for their careful review of the manuscript and valuable comments.

REFERENCES

1. Feriani A, Perotti F, Simoncini V. Iterative system solvers for the frequency analysis of linear mechanical systems. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**(13):1719–1739.
2. Higham NJ, Mackey DS, Tisseur F, Garvey SD. Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems. *International Journal for Numerical Methods in Engineering* 2008; **73**(3):344–360.
3. Bermúdez A, Durán RG, Rodríguez R, Solomin J. Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics. *SIAM Journal on Numerical Analysis* 2000; **38**(1):267–291.
4. Chaitin-Chatelin F, Van Gijzen MB. Analysis of parameterized quadratic eigenvalue problems in computational acoustics with homotopic deviation theory. *Numerical Linear Algebra with Applications* 2006; **13**(6):487–512.

DOI: 10.1002/nme

5. Puri RS. Krylov subspace based direct projection techniques for low frequency, fully coupled, structural acoustic analysis and optimization. *Ph.D. Thesis*, Oxford Brookes Universiy, Oxford OX3 0BP, United Kingdom, 2008.
6. Gohberg I, Lancaster P, Rodman L. *Matrix Polynomials*, Vol. 58. SIAM: Philadelphia, PA, USA, 2009.
7. Mackey DS, Mackey N, Mehl C, Mehrmann V. Vector spaces of linearizations for matrix polynomials. *SIAM Journal on Matrix Analysis and Applications* 2006; **28**(4):971–1004.
8. Sleijpen GLG, Booten AGL, Fokkema DR, Van der Vorst HA. Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT Numerical Mathematics* 1996; **36**(3):595–633.
9. Bai Z, Su Y. SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications* 2005; **26**(3):640–659.
10. Meerbergen K. The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**(4):1463–1482.
11. Meerbergen K, Spence A, Roose D. Shift-invert and Cayley transforms for detection of rightmost eigenvalues of nonsymmetric matrices. *BIT Numerical Mathematics* 1994; **34**(3):409–423.
12. Baker GA, Graves-Morris PR. *Padé Approximants* (2nd edn). Encyclopedia of Mathematics and its Applications. Cambridge University Press: Cambridge, CB2 8BS, UK, 1996.
13. Lu YY. A Padé approximation method for square roots of symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications* 1998; **19**(3):833–845.
14. Ruhe A. Algorithms for the nonlinear eigenvalue problem. *SIAM Journal on Numerical Analysis* 1973; **10**(4): 674–689.
15. Mehrmann V, Voss H. Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods. *GAMM Mitteilungen Gesellschaft fur Angewandte und Mechanik* 2005; **27**:121–151.
16. Effenberger C, Kressner D. Chebyshev interpolation for nonlinear eigenvalue problems. *BIT Numerical Mathematics* 2012; **52**(4):933–951.
17. Van Beeumen R, Meerbergen K, Michiels W. A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems. *SIAM Journal on Scientific Computing* 2013; **35**(1):A327–A350.
18. Jarlebring E, Michiels W, Meerbergen K. A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numerische Mathematik* 2012; **122**(1):169–195.
19. Jarlebring E, Meerbergen K, Michiels W. Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method. *SIAM Journal on Matrix Analysis and Applications* 2014; **35**(2):411–436.
20. Jarlebring E, Güttel S. A spatially adaptive iterative method for a class of nonlinear operator eigenproblems. *Electronic Transactions on Numerical Analysis* 2014; **41**:21–41.
21. Güttel S, Van Beeumen R, Meerbergen K, Michiels W. NLEIGS: a class of robust fully rational Krylov methods for nonlinear eigenvalue problems. *Technical Report*, School of Mathematics, The University of Manchester, Manchester M13 9PL, UK. MIMS Preprint: 2013.49, (Available from: http://eprints.ma.man.ac.uk/2019/01/covered/MIMS_ep2013_49.pdf) [Accessed on 20 May 2014].
22. Bindel D, Hood A. Localization theorems for nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications* 2013; **34**(4):1728–1749.
23. Tisseur F. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra and its Applications* 2000; **309**(1):339–361.
24. Su Y, Bai Z. Solving rational eigenvalue problems via linearization. *SIAM Journal on Matrix Analysis and Applications* 2011; **32**(1):201–216.
25. Grammont L, Higham NJ, Tisseur F. A framework for analyzing nonlinear eigenproblems and parameterized linear systems. *Linear Algebra and its Applications* 2011; **435**(3):626–640.
26. Fan HY, Lin WW, Van Dooren P. Normwise scaling of second order polynomial matrices. *SIAM Journal on Matrix Analysis and Applications* 2004; **26**(1):252–256.
27. Gaubert S, Sharify M. Tropical scaling of polynomial matrices. In *Positive Systems*. Springer: Berlin Heidelberg, 2009; 291–303.
28. Hammarling S, Munro CJ, Tisseur F. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)* 2013; **39**(3):18:1–18:19.
29. Zeng L, Su Y. A backward stable algorithm for quadratic eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications* 2014; **35**(2):499–516.
30. Higham NJ, Li RC, Tisseur F. Backward error of polynomial eigenproblems solved by linearization. *SIAM Journal on Matrix Analysis and Applications* 2007; **29**(4):1218–1241.
31. Wolfram Research Inc. *Tangent*. (Available from: http://functions.wolfram.com/01.08.23.0007.01/) [Accessed on 25 February 2014].
32. Golub GH, Van Loan CF. *Matrix Computations*. Johns Hopkins University, Press: Baltimore, MD, USA, 1996.
33. Lehoucq RB, Sorensen DC, Yang C. *Arpack Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Vol. 6. SIAM: Philadelphia, 1998.
34. Davis TA. Algorithm 832: UMFPACK v4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)* 2004; **30**(2):196–199.
35. Gomes FM, Sorensen DC. *ARPACK++: an object-oriented version of arpack eigenvalue package, version 1.2*, 2000. (Available from: http://www.ime.unicamp.br/~chico/arpack++/) [Accessed on February 25, 2014].
36. Li XS. An overview of SuperLU: algorithms, implementation, and user interface. *ACM Transactions on Mathematical Software (TOMS)* 2005; **31**(3):302–325.

37. Betcke T, Higham NJ, Mehrmann V, Schröder C, Tisseur F. NLEVP: a collection of nonlinear eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)* 2013; **39**(2):7:1–7:28.
38. Louis K. *What Every Engineer Should Know about Computational Techniques of Finite Element Analysis*. CRC Press: Boca Raton, Florida, USA, 2005.
39. Taslaman L. An algorithm for quadratic eigenproblems with low rank damping. *Technical Report*, School of Mathematics, The University of Manchester, Manchester M13 9PL, UK. MIMS Preprint: 2014.21, (Available from: http://eprints.ma.man.ac.uk/2132/01/covered/MIMS_ep2014_21.pdf) [Accessed on 20 May 2014].
40. Liao BS, Bai Z, Lee LQ, Ko K. Nonlinear Rayleigh–Ritz iterative method for solving large scale nonlinear eigenvalue problems. *Taiwanese Journal of Mathematics* 2010; **14**(3A):869–883.

DOI: 10.1002/nme