The Nonlinear Eigenvalue Problem

Güttel, Stefan and Tisseur, Françoise

2017

MIMS EPrint: **2017.7**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

The Nonlinear Eigenvalue Problem

Stefan Güttel
The University of Manchester, School of Mathematics
Oxford Road, M13 9PL, UK
Email: stefan.guettel@manchester.ac.uk

Françoise Tisseur*
The University of Manchester, School of Mathematics
Oxford Road, M13 9PL, UK
Email: francoise.tisseur@manchester.ac.uk

Nonlinear eigenvalue problems arise in a variety of science and engineering applications and in the past ten years there have been numerous breakthroughs in the development of numerical methods. This article surveys nonlinear eigenvalue problems associated with matrix-valued functions which depend nonlinearly on a single scalar parameter, with a particular emphasis on their mathematical properties and available numerical solution techniques. Solvers based on Newton's method, contour integration, and sampling via rational interpolation are reviewed. Problems of selecting the appropriate parameters for each of the solver classes are discussed and illustrated with numerical examples. This survey also contains numerous MATLAB code snippets that can be used for interactive exploration of the discussed methods.

CONTENTS

* Supported by EPSRC grant EP/I005293 and by a Royal Society-Wolfson Research Merit Award.

1. Introduction

Nonlinear eigenvalue problems arise in many areas of computational science and engineering, including acoustics, control theory, fluid mechanics, and structural engineering. The fundamental formulation of such problems is given in the following definition.

Definition 1.1. Given a nonempty open set $\Omega \subseteq \mathbb{C}$ and a matrix-valued function $F : \Omega \to \mathbb{C}^{n\times n}$, the *nonlinear eigenvalue problem* (NEP) consists of finding scalars $\lambda \in \Omega$ (the *eigenvalues*) and nonzero vectors $v \in \mathbb{C}^n$ and $w \in \mathbb{C}^n$ (right and left *eigenvectors*) such that

$$F(\lambda)v = 0, \quad w^*F(\lambda) = 0^*.$$

Here 0 denotes the zero column vector of $\mathbb{C}^n$, and $(\cdot)^*$ denotes the conjugate transpose of a vector. We refer to (λ, v) as an *eigenpair* of F. The set of all eigenvalues is denoted by $\Lambda(F)$ and referred to as the *spectrum of* F, while $\Omega \setminus \Lambda(F)$ is called the *resolvent set of* F.

Clearly, the eigenvalues λ of F are the solutions of the scalar equation $f(z) = \det F(z) = 0$. The dependence of $F(z)$ on the parameter z is typically nonlinear, giving the "N" in NEP, but the eigenvectors enter the problem only linearly. Throughout this work we will denote by z the parameter of F as an independent variable, and we will use λ for the eigenvalues.

Three simple but representative examples of NEPs are listed below. We also give links to corresponding problems in the NLEVP collection by Betcke, Higham, Mehrmann, Schröder and Tisseur (2013). We will use some of these problems later on for numerical illustrations.

(a) *Stability analysis of delay differential equations* (DDEs). Consider the linear first-order homogeneous DDE

$$u'(t) + Au(t) + Bu(t-1) = 0, \quad u(t) \text{ given for } t \in [-1, 0]. \tag{1.1}$$

Solutions of the form $u(t) = \mathrm{e}^{\lambda t}v$ can be obtained from the NEP

$$F(\lambda)v = (\lambda I + A + Be^{-\lambda})v = 0,$$

and the real parts of the eigenvalues λ determine the growth of $\|u\| = e^{t\,\mathrm{Re}\lambda}\|v\|$. A problem of this type is part of the NLEVP collection under the name `time_delay`. Detailed expositions of the stability theory of time-delay systems can be found in (Gu, Kharitonov and Chen 2003, Michiels and Niculescu 2007).

(b) *Differential equations with nonlinear boundary conditions.* A minimal example from (Solov′ëv 2006) is a boundary value problem on $[0, 1]$,

$$-u''(x) = \lambda u(x), \quad u(0) = 0, \quad -u'(1) = \phi(\lambda)u(1). \tag{1.2}$$

Upon applying a finite element discretization with linear hat functions

centered at the equispaced points $x_i = i/n$ $(i = 1, 2, \ldots, n)$, (1.2) can be rewritten as an NEP

$$F(\lambda)u = (C_1 - \lambda C_2 + \phi(\lambda)C_3)u = 0 \tag{1.3}$$

with $n \times n$ matrices

$$C_1 = n \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & 2 & -1 \\ & & -1 & 1 \end{bmatrix}, \; C_2 = \frac{1}{6n} \begin{bmatrix} 4 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & 4 & 1 \\ & & 1 & 2 \end{bmatrix}, \; C_3 = e_n e_n^T,$$

and $e_n = [0, \ldots, 0, 1]^T$. With the rational function $\phi(\lambda) = \lambda/(\lambda - 1)$ this problem is contained in the NLEVP collection under the name `loaded_string`.

A property that is common to many NEPs is that the nonlinearity appears only in a few entries of $F(z)$. Indeed, the nonlinearity in this example affects only a single entry through the rank-1 matrix C_3. This *low-rank structure* is typically worth exploiting.

(c) *Transparent boundary conditions.* This is a special case of (b), where the right boundary condition is chosen so that (1.2) is satisfied on an interval larger than the discretization interval, e.g., for all $x \in [0, +\infty)$. Note that the solutions of (1.2) on $[0, +\infty)$ are of the form

$$u(x) = \alpha \sin(\sqrt{\lambda} x).$$

Therefore, in order to model a "transparent" boundary, the Dirichlet-to-Neumann map ϕ at $x = 1$ should satisfy

$$\phi(\lambda) = -\frac{\sqrt{\lambda}}{\tan(\lambda)}.$$

Problems with similar types of nonlinearities are encountered, for example, with the modeling of waveguides. The `gun` problem available in the NLEVP collection,

$$F(\lambda)v = \Big(K - \lambda M + \mathrm{i}\sqrt{\lambda - \omega_1^2}\, W_1 + \mathrm{i}\sqrt{\lambda - \omega_2^2}\, W_2\Big)v = 0, \tag{1.4}$$

models a waveguide-loaded accelerator cavity with the values of ω_j corresponding to cut-off wave numbers of modes in two waveguide ports. Here M, K, W_1, and W_2 are real symmetric 9956×9956 matrices arising from the finite element discretization of Maxwell's equation

$$\nabla \times \Big(\frac{1}{\mu} \nabla \times E\Big) - \lambda \epsilon E = 0$$

for the electric field E on some spatial domain; see (Liao, Bai, Lee and

Ko 2010) for details. Also in this example the matrices W_1 and W_2 associated with the waveguide boundaries are of low rank.

For an exposition of other interesting applications where NEPs arise we refer to the introductory chapters of the PhD theses by Effenberger (2013*a*) and Van Beeumen (2015). Note that, independent of a particular application, NEPs are of mathematical interest as systems of nonlinear equations, or scalar root-finding problems, e.g., for $f(z) = \det F(z)$.

There are also problems for which F does not depend on z but depends instead on the eigenvectors or a selection of eigenvectors, with applications in electronic structure calculations (Saad, Stathopoulos, Chelikowsky, Wu and Öğüt 1996), machine learning (Bühler and Hein 2009), and even for the ranking of football teams (Keener 1993). Formally, these eigenvalue problems consist of finding nontrivial solutions of $F(V)V = VD$, where F is an $n \times n$ matrix that depends on an $n \times k$ matrix of eigenvectors V, and D is a $k \times k$ diagonal matrix containing the corresponding eigenvalues. This type of nonlinear dependency will not be considered here. Another interesting class are multi-parameter NEPs $F(\lambda, \gamma)v = 0$ which arise, for example, in the stability analysis of parametrized nonlinear wave equations (Beyn and Thümmler 2009) and the delay-independent stability analysis of DDEs (Gu et al. 2003, Chapter 4.6). Although we will not further discuss multi-parameter NEPs in this survey, continuation methods for their solution typically require solving an NEP as in Definition 1.1 at every iteration.

Our aim is to survey NEPs through their interesting mathematical properties and numerical solution techniques. While we mention and list some applications where NEPs arise, these are not our main focus. We hope that this survey will be useful for mathematicians, computational scientists, and engineers alike. Those new to the field may want to learn first about the basic mathematical properties of NEPs, such as their solution structure and the sensitivity of eigenvalues, and Section 2 should be a good starting point for that. We also provide an extensive list of references for further exploration, although we do not claim that this is a complete list for this very rich and rapidly developing area. Practitioners who already have a concrete NEP to solve can use this survey to get an overview of available solution techniques. To facilitate choosing the right method, we highlight the guiding principles and various problems that come with each numerical method such as, for example, the selection of suitable parameters.

With practical solution methods being the main focus, and to make this survey more interactive, we include several MATLAB code snippets. Some of the snippets require MATLAB R2016b or later, because they use local functions in a script (Higham and Higham 2017). The codes are also available online at

`http://www.maths.manchester.ac.uk/nep/`.

We encourage the reader to execute them in MATLAB and play with the parameters. Most of the codes also run with no or little modification in GNU Octave (Eaton, Bateman, Hauberg, and Wehbring 2016). We emphasize that our code snippets should not be considered as full implementations of robust numerical methods! They merely serve the purpose of illustration.

In addition to the codes, the above web site also contains the bibliography of this survey, as well as an up-to-date list of links to available software related to NEPs. Alongside this survey, the reader is encouraged to consult the review by Mehrmann and Voss (2004), which discusses many NEP applications and describes some of the methods contained here (but not, for example, the contour-based and rational interpolation-based solvers developed since 2004). Quadratic eigenvalue problems are reviewed in detail by Tisseur and Meerbergen (2001). For quadratic and more generally polynomial eigenvalue problems the exploitation of special structure in the coefficient matrices is well researched; see, e.g., (Mackey, Mackey, Mehl and Mehrmann 2006*a*, Mackey, Mackey and Tisseur 2015) and the references therein. In our discussions of NEPs with special structure we will restrict our attention to the Hermitian case (see Section 3) and problems with low-rank structure. Hermitian NEPs are the most frequently encountered in practice and by far the best understood. The exploitation of low-rank structure in NEPs often leads to considerable computational savings.

The outline of this work is as follows. Section 2 provides an overview of the solution structure of NEPs, including discussions of root functions and invariant pairs, the Smith form and Keldysh's theorem, and generalized Rayleigh functionals, as well as some perturbation results. The short Section 3 is devoted to Hermitian NEPs. Sections 4, 5, and 6 are on numerical methods. Section 4 is about NEP solvers based on Newton's method, with particular attention paid to relating the various existing methods and illustrating their convergence properties with examples. Section 5 is devoted to NEP solvers using contour integrals. Here again our main focus is on summarizing and relating existing algorithms, as well as discussing their dependence on parameters. Section 6 gives an overview of solution methods based on the linearization of rational eigenvalue problems. Here our focus is on the practical and efficient construction of rational interpolants via sampling of the matrix-valued function defining an NEP. Each of the sections on numerical methods ends with a brief overview of existing software, giving links to online resources whenever possible.

Within the sections of this paper we use common counters in the numbering of Definitions, Theorems, Algorithms, Figures, etc. We believe this simplifies referencing for the reader since, for example, locating Remark 2.2 tells one that a definition with number 2.3 should follow. Subsections and equations are enumerated separately within each section.

2. Solution structure of NEPs

In this section we will shed light on the solution structure of NEPs. As the examples in this section aim to illustrate, the solution properties of general NEPs are quite different from linear eigenvalue problems $Av = \lambda v$, where $A \in \mathbb{C}^{n\times n}$ and $0 \neq v \in \mathbb{C}^n$, although such problems are special cases of NEPs with $F(z) = A - zI$. Nevertheless, we will see that with an appropriate definition of eigenvalue multiplicity and generalized eigenvectors, a rather elegant characterization of NEP solutions can be obtained. Much of the theory we review here can be found in the monograph by Mennicken and Möller (2003), therein developed in more detail and greater generality for operator functions in Banach spaces. The appendix of the monograph by Kozlov and Maz'ja (1999) also contains a good summary of parts of the theory, including a complete proof of Keldysh's theorem. A collection of mathematical facts about NEPs is given by Voss (2014).

2.1. Eigenvalues and eigenvectors

To develop our intuition about the eigenvalue structure of NEPs, it will be instructive to first think of a scalar function $F : \Omega \to \mathbb{C}$. The associated NEP $F(\lambda)v = 0$, $v \neq 0$, is just a scalar root-finding problem. It becomes immediately apparent that an NEP on $\Omega = \mathbb{C}$ may have

- no solutions at all, e.g., if $F(z) = \exp(z)$,
- finitely many solutions, e.g., if $F(z) = z^3 - 1$,
- infinitely many solutions, e.g., if $F(z) = \cos(z)$.

The fact that even an NEP of size 1×1 can have more than one eigenvalue tells us that the eigenvectors belonging to distinct eigenvalues need not be linearly independent: indeed the "vector" $v = [\,1\,]$ is an eigenvector for all eigenvalues of an 1×1 NEP. An example of size 2×2 is

$$F(z) = \begin{bmatrix} e^{\mathrm{i}z^2} & 1 \\ 1 & 1 \end{bmatrix}, \tag{2.1}$$

which is a singular matrix exactly at the points $z \in \mathbb{C}$ where $e^{\mathrm{i}z^2} = 1$. Hence the eigenvalues of F are $\lambda_k = \pm\sqrt{2\pi k}$ ($k = 0, \pm 1, \pm 2, \ldots$) and $\left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$ is a right and left eigenvector for all of them. It will be useful to keep the function F defined by (2.1) in mind as we will often refer back to it for illustration.

In the following discussions we will assume that $F : \Omega \to \mathbb{C}^{n\times n}$ is holomorphic in a connected open set $\Omega \subseteq \mathbb{C}$ (the *domain*), denoted by $F \in H(\Omega, \mathbb{C}^{n\times n})$. This assumption is satisfied for most NEPs encountered in practice, where the elements of $F(z)$ are usually polynomial, rational, algebraic, or exponential functions of $z \in \Omega$, or a combination thereof. As the definition of a matrix determinant involves only finite sums and products of

the matrix entries, $F \in H(\Omega, \mathbb{C}^{n\times n})$ also implies that $\det F(z) \in H(\Omega, \mathbb{C})$. Hence the eigenvalues of F are the roots of a scalar holomorphic function in the domain Ω. The following discreteness result is a consequence of that and can be found, e.g., in (Mennicken and Möller 2003, Theorem 1.3.1).

Theorem 2.1. Let $\Omega \subseteq \mathbb{C}$ be a domain and $F \in H(\Omega, \mathbb{C}^{n\times n})$. Then the resolvent set $\Omega \setminus \Lambda(F)$ is open. If the resolvent set is nonempty, every eigenvalue $\lambda \in \Lambda(F)$ is isolated, i.e., there exists an open neighborhood $\mathcal{U} \subset \Omega$ such that $\mathcal{U} \cap \Lambda(F) = \{\lambda\}$.

The condition that the resolvent set be nonempty is equivalent to saying that $\det F(z)$ does not vanish identically on Ω. In this case we say that F is *regular*. While a regular F can have an infinite number of eigenvalues in Ω, the set of eigenvalues $\Lambda(F)$ does not have accumulation points in Ω. An illustrating example is the scalar function $F(z) = \sin(1/z)$, which is holomorphic in $\Omega = \mathbb{C} \setminus \{0\}$ with an infinite number of roots accumulating at $0 \notin \Omega$.

The *algebraic multiplicity of an eigenvalue* λ is defined as the multiplicity of the root of $\det F(z)$ at $z = \lambda$, i.e., the smallest integer $j \geq 1$ such that

$$\frac{\mathrm{d}^j}{\mathrm{d}z^j} \det F(z)\Big|_{z=\lambda} \neq 0.$$

The algebraic multiplicity of an isolated eigenvalue must be finite, but in contrast to linear eigenvalue problems, in the nonlinear case the algebraic multiplicity of an eigenvalue is not necessarily bounded by the problem size n. This is illustrated by the scalar example $F(z) = z^{n+1}$, which has the eigenvalue $\lambda = 0$ of algebraic multiplicity $n + 1$.

By Definition 1.1, the right eigenvectors associated with an eigenvalue λ are the nonzero vectors in the null space of $F(\lambda)$, denoted by $\mathrm{null} F(\lambda)$. The dimension of this null space, $\dim(\mathrm{null} F(\lambda))$, is called the *geometric multiplicity of* λ. An eigenvalue λ is called *semisimple* if its geometric multiplicity is equal to its algebraic multiplicity, and it is called *simple* if its algebraic multiplicity equals one.

Remark 2.2. In some cases it is of interest to consider NEPs with eigenvalues at infinity, for example in the stability analysis of linear time-invariant delay differential-algebraic equations (Du, Linh, Mehrmann and Thuan 2013) or when deflating already computed eigenvalues (see Section 4.3). We say that $F : \Omega \to \mathbb{C}^{n\times n}$ defined on $\Omega \subseteq \overline{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$ with $\infty \in \Omega$ has an eigenvalue at $\lambda = \infty$ if

$$G : z \mapsto F(1/z)$$

has an eigenvalue at $z = 0$. The algebraic and geometric multiplicities of $\lambda = \infty$ in F are simply defined as the corresponding algebraic and geometric multiplicities of $z = 0$ in G. Indeed, if an NEP is considered on the extended

complex plane $\overline{\mathbb{C}}$ instead of $\mathbb{C}$, the point $z = \infty$ is not more or less distinguished than any other complex number. In particular, the point at infinity can always be mapped to a finite point via a Möbius transformation. Hence there is no need to further discuss infinite eigenvalues of NEPs separately.

Note that our definition of eigenvalues at infinity is not in contradiction to the common definition of infinite eigenvalues for matrix polynomials $P(z) = \sum_{j=0}^{\ell} z^j C_j$, $C_\ell \neq O$, as the zero eigenvalues of the reversal matrix polynomial $z^\ell P(1/z)$. When considered as an NEP in the sense used here, the function P is actually not well defined at $z = \infty$. Hence the convention is to instead consider the function $F(z) = z^{-\ell} P(z)$, which is well defined (and even holomorphic) at $z = \infty$ and so it makes sense to consider eigenvalues there. Indeed, $G(z) = F(1/z)$ is the reversal of P.

2.2. Root functions and generalized eigenvectors

An elegant definition of generalized eigenvectors is based on root functions, a concept introduced by Trofimov (1968). To motivate the idea of root functions, it is helpful to first consider a linear eigenvalue problem $Av = \lambda v$ for $A \in \mathbb{C}^{n\times n}$ and $0 \neq v \in \mathbb{C}^n$, and to recall from matrix analysis that a sequence $(v_0, v_1, \ldots, v_{m-1})$ of nonzero vectors is called a *Jordan chain for the eigenvalue* λ if

$$(A - \lambda I)v_0 = 0,\ (A - \lambda I)v_1 = v_0,\ \ldots,\ (A - \lambda I)v_{m-1} = v_{m-2}; \qquad (2.2)$$

see, e.g., the monograph by Horn and Johnson (1985, Chapter 3). The vector v_0 is a right eigenvector of A, or equivalently, a nonzero right null vector of $A - \lambda I$, and the $v_0, v_1, \ldots, v_{m-1}$ are *generalized eigenvectors of* A. Generalized eigenvectors of a matrix can easily be seen to be linearly independent. The maximal length m of a Jordan chain starting with v_0 is called the *rank of* v_0.

Interestingly, by defining the functions $v(z) = \sum_{j=0}^{m-1} (z-\lambda)^j v_j$ and $F(z) = A - zI$, the m equations in (2.2) can be rewritten as

$$\frac{\mathrm{d}^j}{\mathrm{d}z^j} F(z)v(z)\Big|_{z=\lambda} = j! \sum_{k=0}^{j} \frac{F^{(k)}(\lambda)}{k!} v_{j-k} = 0, \quad j = 0, 1, \ldots, m-1, \qquad (2.3)$$

where $F^{(k)}(z) = \frac{\mathrm{d}^k}{\mathrm{d}z^k} F(z)$. In other words, $F(z)v(z)$ has a root of multiplicity at least m at $z = \lambda$. While (2.2) is specific to linear eigenvalue problems, (2.3) straightforwardly extends to the nonlinear case. The main difference with NEPs is that generalized eigenvectors need not necessarily be linearly independent, and even the zero vector is admitted as a generalized eigenvector. We make this precise in the following definition, which has been adopted from (Mennicken and Möller 2003, Definition 1.6.1).

Definition 2.3. Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ be regular on a nonempty domain $\Omega \subseteq \mathbb{C}$, and let $\lambda \in \Omega$ be given.

(i) A holomorphic vector-valued function $v \in H(\Omega, \mathbb{C}^n)$ such that $v(\lambda) \neq 0$ and $F(\lambda)v(\lambda) = 0$ is called a *root function for F at λ*. The multiplicity of the root $z = \lambda$ of $F(z)v(z)$ is denoted by $s(v)$.

(ii) A tuple $(v_0, v_1, \ldots, v_{m-1}) \in (\mathbb{C}^n)^m$ with $m \geq 1$ and $v_0 \neq 0$ is called a *Jordan chain for F at λ* if $v(z) = \sum_{k=0}^{m-1}(z-\lambda)^k v_k$ is a root function for F at λ and $s(v) \geq m$.

(iii) For a given vector $v_0 \in \mathrm{null}F(\lambda) \setminus \{0\}$ the number

$$r(v_0) = \max\{s(v) : v \text{ is a root function for } F \text{ at } \lambda \text{ with } v(\lambda) = v_0\}$$

is called the *rank of v_0*.

(iv) A system of vectors in $\mathbb{C}^n$,

$$V = \left(v_k^j : 0 \leq k \leq m_j - 1,\ 1 \leq j \leq d\right)$$

is a *complete system of Jordan chains for F at λ* if

(a) $d = \dim(\mathrm{null}F(\lambda))$ and $\{v_0^1, v_0^2, \ldots, v_0^d\}$ is a basis of $\mathrm{null}F(\lambda)$;

(b) the tuple $(v_0^j, v_1^j, \ldots, v_{m_j-1}^j)$ is a Jordan chain for F at λ for $j = 1, 2, \ldots, d$;

(c) $m_j = \max\left\{r(v_0) : v_0 \in \mathrm{null}F(\lambda) \setminus \mathrm{span}\{v_0^\nu : 1 \leq \nu < j\}\right\}$ for $j = 1, 2, \ldots, d$.

It can be shown (Mennicken and Möller 2003, Proposition 1.6.4) that a complete system of Jordan chains always exists, which essentially requires the verification that for a regular NEP the rank $r(v_0)$ of any eigenvector v_0 is finite. The numbers m_j satisfy $m_1 \geq m_2 \geq \cdots \geq m_d$ by definition and are called the *partial multiplicities of λ*. It can be shown that $\sum_{j=1}^d m_j$ corresponds to the algebraic multiplicity of λ (Mennicken and Möller 2003, Proposition 1.8.5). The number m_1 is called the *index of λ*. If $m_1 = \cdots = m_d = 1$, then λ is a semisimple eigenvalue; if in addition $d = 1$, then λ is a simple eigenvalue.

Example 2.4. Consider the 2×2 function F defined in (2.1). The eigenvalue $\lambda_0 = 0$ has algebraic multiplicity two and geometric multiplicity one. The corresponding right eigenvector v_0 and generalized eigenvector v_1 are both multiples of $\left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, and hence are not linearly independent. The vector-valued function $v(z) = v_0 + zv_1 = \left[\begin{smallmatrix} 1+z \\ -1-z \end{smallmatrix}\right]$ is a root function for F at $\lambda_0 = 0$ and the multiplicity of λ_0 as a root of $F(z)v(z)$ can easily shown to be $s(v) = 2$: using the MATLAB Symbolic Math Toolbox we find that $\frac{\mathrm{d}^2}{\mathrm{d}z^2}F(z)v(z)$ is the smallest-order derivative which does not vanish at $z = 0$:

```
syms z; order = 2;
F = [ exp(1i*z.^2) 1; 1 1 ]; v = [1+z;1-z];
subs(diff(F*v, order), z, 0)
ans =
 2i
  0
```

Hence the rank of the right eigenvector v_0 with eigenvalue λ_0 is $r(v_0) \geq 2$, but since λ_0 has algebraic multiplicity two, $r(v_0) = 2$. The pair $\left(\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\right)$ is a Jordan chain for F at $\lambda_0 = 0$, and also a complete system of Jordan chains there.

So far we have only considered *right* eigenvectors of F, but clearly left eigenvectors of F can be obtained as the right eigenvectors of F^*. It turns out that there is a natural way to complement a complete system of Jordan chains for F at λ with a complete system of Jordan chains of F^* at λ, and the latter is uniquely determined if certain normalization conditions are enforced. The following result makes this precise (Mennicken and Möller 2003, Theorem 1.6.5).

Theorem 2.5. Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ be regular on a nonempty domain $\Omega \subseteq \mathbb{C}$, and consider an eigenvalue $\lambda \in \Lambda(F)$. Let

$$V = \left(v_k^j : 0 \leq k \leq m_j - 1,\ 1 \leq j \leq d\right)$$

be a complete system of Jordan chains for F at λ. Then there exists a unique complete system of Jordan chains

$$W = \left(w_k^j : 0 \leq k \leq m_j - 1,\ 1 \leq j \leq d\right)$$

for F^* at λ such that each eigenvector w_0^j has rank $r(w_0^j) = m_j$ and

$$\sum_{\alpha=0}^{k} \sum_{\beta=1}^{m_i} w_{k-\alpha}^{j*} \frac{F^{(\alpha+\beta)}(\lambda)}{(\alpha+\beta)!} v_{m_i-\beta}^{i} = \delta_{ij}\delta_{0k}, \quad 0 \leq k \leq m_j - 1,\ 1 \leq i, j \leq d, \tag{2.4}$$

with the Kronecker delta $\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases}$

Example 2.6. Consider the matrix-valued function F in (2.1). We found in Example 2.4 that the pair $V = (v_0, v_1) = \left(\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\right)$ is a complete system of Jordan chains for F at $\lambda = 0$. A similar construction shows that $W = (w_0, w_1) = \left(\rho_0\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right], \rho_1\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\right)$ with $\rho_0, \rho_1 \in \mathbb{C}\setminus\{0\}$ is a complete system of Jordan chains for F^* at $\lambda = 0$ satisfying $r(w_0) = 2$. The normalization

conditions (2.4) become

$$w_0^* F'(0)v_1 + w_0^* \frac{F''(0)}{2} v_0 = 1,$$
$$w_1^* F'(0)v_1 + w_1^* \frac{F''(0)}{2} v_0 + w_0^* \frac{F''(0)}{2} v_1 + w_0^* \frac{F'''(0)}{6} v_0 = 0,$$

and they are satisfied if we choose $\rho_0 = \mathrm{i}$ and $\rho_1 = -\mathrm{i}$.

While the normalization conditions in Theorem 2.5 look rather messy in general, they simplify if the eigenvalue λ is semisimple. In this case we have $m_1 = \cdots = m_d = 1$ so that (2.4) becomes

$$w_0^{j*} F'(\lambda) v_0^i = \delta_{ij}, \quad 1 \le i, j \le d, \tag{2.5}$$

i.e., the left and right eigenvectors can be chosen $F'(\lambda)$-biorthonormal.

2.3. Factorizations: Smith form

It is often useful to have a factorization of an NEP about one or multiple eigenvalues, giving rise to *local* or *global* factorizations, respectively. A very useful factorization is the so-called Smith form, which we state in its global version; see (Leiterer 1978, Gohberg and Rodman 1981) and (Kozlov and Maz'ja 1999, Chapter A.6). It uses the notion of a *unimodular* matrix-valued function $P \in H(\Omega, \mathbb{C}^{n\times n})$, which means that $\det P(z)$ is a nonzero constant over Ω.

Theorem 2.7 (Smith form). Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ be regular on a nonempty domain Ω. Let λ_i $(i = 1, 2, \ldots)$ be the distinct eigenvalues of F in Ω with partial multiplicities $m_{i,1} \ge m_{i,2} \ge \cdots \ge m_{i,d_i}$. Then there exists a *global Smith form*

$$P(z)F(z)Q(z) = D(z), \quad z \in \Omega,$$

with unimodular matrix-valued functions $P, Q \in H(\Omega, \mathbb{C}^{n\times n})$, and $D(z) = \operatorname{diag}(\delta_1(z), \delta_2(z), \ldots, \delta_n(z))$. The diagonal entries δ_j are of the form

$$\delta_j(z) = h_j(z) \prod_{i=1,2,\ldots} (z - \lambda_i)^{m_{i,j}}, \quad j = 1, \ldots, n,$$

where $m_{i,j} = 0$ if $j > d_i$ and each $h_j \in H(\Omega, \mathbb{C})$ is free of roots on Ω.

From the Smith form it is easy to identify eigenvectors of F for an eigenvalue λ_i. Let Q be partitioned columnwise as $Q(z) = [q_1(z), q_2(z), \ldots, q_n(z)]$. Then by the diagonal structure of D, the canonical unit vectors $e_1, e_2, \ldots, e_{d_i}$ form a basis of $\operatorname{null} D(\lambda_i)$, and therefore also a basis of $\operatorname{null}\big(P(\lambda_i)F(\lambda_i)Q(\lambda_i)\big)$. Hence we have found a basis of d_i right eigenvectors of F for λ, namely,

$$q_1(\lambda_i), q_2(\lambda_i), \ldots, q_{d_i}(\lambda_i).$$

By the same argument one can identify the first d_i rows of $P(z) = [p_1(z), p_2(z), \ldots, p_n(z)]^*$ with left eigenvectors of F for $z = \lambda_i$.

Another application of the Smith form is for the characterization of the singularities of the *resolvent* given by

$$F(z)^{-1} = Q(z)D(z)^{-1}P(z) = \sum_{j=1}^{n} \delta_j(z)^{-1} q_j(z) p_j(z)^*. \tag{2.6}$$

Here the diagonal entries $\delta_j(z)^{-1}$ of $D(z)^{-1}$ are scalar meromorphic functions with poles of multiplicities $m_{i,j}$ at the eigenvalues λ_i,

$$\delta_j(z)^{-1} = \frac{1}{h_j(z)} \prod_{i=1,2,\ldots} (z - \lambda_i)^{-m_{i,j}}, \quad j = 1, \ldots, n.$$

2.4. Expansions: Keldysh's theorem

Let us consider a selected eigenvalue $z = \lambda_i$ of F. Then each term $\delta_j(z)^{-1} q_j(z) p_j(z)^*$ in (2.6) can be expanded into a matrix-valued Laurent series (Ahlfors 1953, Chapter 5.1.3) that is valid in some neighborhood $\mathcal{U} \subseteq \Omega$ about λ_i,

$$\delta_j(z)^{-1} q_j(z) p_j(z)^* = \sum_{k=1}^{m_{i,j}} S_{i,j,k} (z - \lambda_i)^{-k} + R_j(z), \quad z \in \mathcal{U} \setminus \{\lambda_i\},$$

with coefficient matrices $S_{i,j,k} \in \mathbb{C}^{n \times n}$, $S_{i,j,m_{i,j}} \neq O$, and the remainder function $R_j \in H(\mathcal{U}, \mathbb{C}^{n \times n})$. Combining these series for every term in (2.6) into a single expansion at $z = \lambda_i$, we arrive at

$$F(z)^{-1} = \sum_{j=1}^{d_i} \sum_{k=1}^{m_{i,j}} S_{i,j,k} (z - \lambda_i)^{-k} + R(z), \quad z \in \mathcal{U} \setminus \{\lambda_i\}, \tag{2.7}$$

with a remainder function $R \in H(\mathcal{U}, \mathbb{C}^{n \times n})$. This shows that the *resolvent* $F(z)^{-1}$ is meromorphic in $\mathcal{U}$ with a pole of multiplicity $\max_{j=1,\ldots,d_i}\{m_{i,j}\} = m_{i,1}$ at λ_i.

Let us now assume that F has only a *finite* number of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$ in the domain Ω. Then we can apply the expansion (2.7) recursively at all the eigenvalues as follows: starting with (2.7) about $z = \lambda_1$, we can expand the remainder R into another Laurent series about $z = \lambda_2$, and so forth, giving rise to the global expansion valid on the whole of Ω,

$$F(z)^{-1} = \sum_{i=1}^{s} \sum_{j=1}^{d_i} \sum_{k=1}^{m_{i,j}} S_{i,j,k} (z - \lambda_i)^{-k} + \widetilde{R}(z), \tag{2.8}$$

with $\widetilde{R} \in H(\Omega, \mathbb{C}^{n \times n})$.

It is possible to characterize the matrices $S_{i,j,k}$ in (2.8) in terms of generalized eigenvectors of F, a result that is known as *Keldysh's theorem*. More precisely, for each distinct eigenvalue $\lambda_i \in \Omega$ $(i = 1, 2, \ldots, s)$, let

$$\left(v_k^{ij} : 0 \le k \le m_{ij} - 1,\ 1 \le j \le d_i\right), \quad \left(w_k^{ij} : 0 \le k \le m_{ij} - 1,\ 1 \le j \le d_i\right)$$

be pairs of complete systems of Jordan chains for F and F^* at λ_i as defined by Theorem 2.5. Then the matrices $S_{i,j,k}$ in (2.8) are given as

$$S_{i,j,k} = \sum_{\ell=0}^{m_{ij}-k} v_\ell^{ij} w_{m_{ij}-k-\ell}^{ij*}. \tag{2.9}$$

Keldysh's theorem can be found in this form, e.g., in (Mennicken and Möller 2003, Theorem 1.6.5) and (Kozlov and Maz'ja 1999, Theorem A.10.2). A more compact representation can be obtained by defining Jordan blocks

$$J_{ij} = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \in \mathbb{C}^{m_{ij} \times m_{ij}} \tag{2.10}$$

and arranging the generalized eigenvectors in columns of matrices

$$V_{ij} = \left[v_0^{ij}, v_1^{ij}, \ldots, v_{m_{ij}-1}^{ij}\right] \quad \text{and} \quad W_{ij} = \left[w_{m_{ij}-1}^{ij}, w_{m_{ij}-2}^{ij}, \ldots, w_0^{ij}\right]. \tag{2.11}$$

(Note the descending order of the left generalized eigenvectors in W_{ij}.) By the Jordan-form definition of a matrix function (Higham 2008, Definition 1.2) we have

$$(zI - J_{ij})^{-1} = \begin{bmatrix} (z-\lambda_i)^{-1} & (z-\lambda_i)^{-2} & \cdots & (z-\lambda_i)^{-m_{ij}} \\ & (z-\lambda_i)^{-1} & \ddots & \vdots \\ & & \ddots & (z-\lambda_i)^{-2} \\ & & & (z-\lambda_i)^{-1} \end{bmatrix},$$

and hence using (2.9),

$$\sum_{k=1}^{m_{i,j}} S_{i,j,k} (z - \lambda_i)^{-k} = V_{ij} (zI - J_{ij})^{-1} W_{ij}^*.$$

We can now rewrite (2.8) and state Keldysh's theorem in the following form.

Theorem 2.8 (Keldysh). Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ be regular on a nonempty domain Ω. Let $\lambda_1, \lambda_2, \ldots, \lambda_s$ be the distinct eigenvalues of F in Ω of partial multiplicities $m_{i,1} \ge m_{i,2} \ge \cdots \ge m_{i,d_i}$, and define $\overline{m} = \sum_{i=1}^{s} \sum_{j=1}^{d_i} m_{ij}$. Then there are $n \times \overline{m}$ matrices V and W whose

columns are generalized eigenvectors, and an $\overline{m} \times \overline{m}$ Jordan matrix J with eigenvalues λ_i of partial multiplicities m_{ij}, such that

$$F(z)^{-1} = V(zI - J)^{-1}W^* + \widetilde{R}(z) \tag{2.12}$$

for some $\widetilde{R} \in H(\Omega, \mathbb{C}^{n\times n})$. With the matrices defined in (2.10)–(2.11),

$$\begin{aligned} J &= \operatorname{diag}(J_1, \ldots, J_s), & J_i &= \operatorname{diag}(J_{i,1}, \ldots, J_{i,d_i}), \\ V &= [V_1, \ldots, V_s], & V_i &= [V_{i,1}, \ldots, V_{i,d_i}], \\ W &= [W_1, \ldots, W_s], & W_i &= [W_{i,1}, \ldots, W_{i,d_i}]. \end{aligned}$$

Let us briefly consider two special cases of Theorem 2.8. Assume that all eigenvalues λ_i of F are semisimple, i.e., $m_{i,1} = \cdots = m_{i,d_i} = 1$. Then the matrix J will be diagonal with each eigenvalue λ_i appearing exactly d_i times. Further, if all λ_i of F are simple (i.e., $d_1 = \cdots = d_s = 1$), then J is an $s \times s$ diagonal matrix with distinct eigenvalues and we have

$$F^{-1}(z) = \sum_{i=1}^{s} (z - \lambda_i)^{-1} v_0^i w_0^{i*} + \widetilde{R}(z),$$

where v_0^i and w_0^i are right and left eigenvectors, respectively, satisfying $w_0^{i*} F'(\lambda_i) v_0^i = 1$.

Theorem 2.8 shows that, up to a holomorphic additive term, the behavior of the resolvent $F^{-1}(z)$ at the eigenvalues λ_i is captured by the inverse of a shifted Jordan matrix. In the linear case $F(z) = zI - A$, we can set $\widetilde{R} \equiv 0$ in (2.12) and choose V as a basis of $n = \overline{m}$ generalized eigenvectors and $W^* = V^{-1}$, resulting in the standard Jordan decomposition $A = VJV^{-1}$. In the nonlinear case, however, V and W can have an arbitrary (but identical) number $\overline{m}$ of columns which need not be linearly independent.

Example 2.9. Consider again the matrix-valued function F in (2.1) and let Ω be the disc of center 0 and radius 3. Then F has eigenvalues $\{0, -\sqrt{2\pi}, \sqrt{2\pi}, -\mathrm{i}\sqrt{2\pi}, \mathrm{i}\sqrt{2\pi}\}$ inside Ω, all simple except the eigenvalue 0, which has algebraic multiplicity two and geometric multiplicity one. It follows from Examples 2.4 and 2.6 that the matrices J, V, W of Theorem 2.8 are given by

$$J = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, -\sqrt{2\pi}, \sqrt{2\pi}, -\mathrm{i}\sqrt{2\pi}, \mathrm{i}\sqrt{2\pi}\right), \tag{2.13}$$

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}, \tag{2.14}$$

$$W = \frac{1}{2}\begin{bmatrix} -2\mathrm{i} & 2\mathrm{i} & \frac{-\mathrm{i}}{\sqrt{2\pi}} & \frac{\mathrm{i}}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} & \frac{-1}{\sqrt{2\pi}} \\ 2\mathrm{i} & -2\mathrm{i} & \frac{\mathrm{i}}{\sqrt{2\pi}} & \frac{-\mathrm{i}}{\sqrt{2\pi}} & \frac{-1}{\sqrt{2\pi}} & \frac{1}{\sqrt{2\pi}} \end{bmatrix}.$$

Hence

$$F(z)^{-1} = \begin{bmatrix} -g(z) & g(z) \\ g(z) & -g(z) \end{bmatrix} + \widetilde{R}(z), \quad g(z) = \frac{3\mathrm{i}z^4 - 4\mathrm{i}\pi^2}{z^2(z^4 - 4\pi^2)}.$$

2.5. Invariant pairs

Invariant pairs play an important role in the analysis of NEPs and the derivation of algorithms. Assume again that $F \in H(\Omega, \mathbb{C}^{n\times n})$ on a nonempty domain $\Omega \subseteq \mathbb{C}$. Then by the Cauchy integral formula (Ahlfors 1953, Chapter 4.2) we have the representation

$$F^{(k)}(\mu) = \frac{k!}{2\pi\mathrm{i}} \int_\Gamma \frac{F(z)}{(z-\mu)^{k+1}}\,\mathrm{d}z,$$

where $\Gamma \subset \Omega$ is a contour containing the point μ in its interior. This formula is the basis for the concept of invariant pairs as defined by Beyn, Effenberger and Kressner (2011) and further advocated in (Effenberger 2013*a*).

Definition 2.10 (Invariant pair). A pair $(V, M) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ is called an *invariant pair for* $F \in H(\Omega, \mathbb{C}^{n\times n})$ if $\mathcal{F}(V, M) = O$, where

$$\mathcal{F}(V, M) := \frac{1}{2\pi\mathrm{i}} \int_\Gamma F(z)V(zI - M)^{-1}\mathrm{d}z \in \mathbb{C}^{n\times m}.$$

Here Γ is a contour containing the eigenvalues of M in its interior.

Note that $F \in H(\Omega, \mathbb{C}^{n\times n})$ can always be written in the "split form"

$$F(z) = f_1(z)C_1 + f_2(z)C_2 + \cdots + f_\ell(z)C_\ell \tag{2.15}$$

with at most $\ell = n^2$ coefficient matrices C_j and scalar functions $f_j \in H(\Omega, \mathbb{C})$ (one for each matrix entry). In many cases, F is naturally given in this form and $\ell \ll n^2$. Clearly, an invariant pair (V, M) for F satisfies

$$\mathcal{F}(V, M) = C_1 V f_1(M) + C_2 V f_2(M) + \cdots + C_\ell V f_\ell(M) = O, \tag{2.16}$$

where each

$$f_j(M) = \frac{1}{2\pi\mathrm{i}} \int_\Gamma f_j(z)(zI - M)^{-1}\mathrm{d}z$$

is a matrix function. Hence invariant pairs can be seen as a generalization to NEPs of invariant subspaces for a single matrix, and standard pairs for matrix polynomials (Gohberg, Lancaster and Rodman 2009). The next result shows that invariant pairs can easily be constructed from root functions; see, e.g., (Gohberg, Kaashoek and van Schagen 1993).

Lemma 2.11. Let $v(z) = \sum_{j=0}^{m-1}(z-\lambda)^j v_j$ be a root function for $F \in H(\Omega, \mathbb{C}^{n\times n})$ at λ of multiplicity larger or equal to m. Define $V =$

$[v_0, v_1, \ldots, v_{m-1}] \in \mathbb{C}^{n\times m}$ and let

$$J = \begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in \mathbb{C}^{m\times m}$$

be the $m \times m$ Jordan block for λ. Then (V, J) is an invariant pair for F.

Proof. By the Jordan-form definition of a matrix function (Higham 2008, Definition 1.2) we have $(zI - J)^{-1} = \sum_{k=0}^{m-1}(z-\lambda)^{-(k+1)}E^k$, where $J = \lambda I + E$. Therefore,

$$\begin{aligned} \mathcal{F}(V,J) &= \frac{1}{2\pi\mathrm{i}}\int_\Gamma F(z)V(zI-J)^{-1}\mathrm{d}z \\ &= \sum_{k=0}^{m-1}\left(\frac{1}{2\pi\mathrm{i}}\int_\Gamma \frac{F(z)}{(z-\lambda)^{k+1}}\mathrm{d}z\right)VE^k \\ &= \sum_{k=0}^{m-1}\frac{F^{(k)}(\lambda)}{k!}VE^k. \end{aligned}$$

For $j = 1, 2, \ldots, m$, the jth column of $\mathcal{F}(V, J)$ is $\sum_{k=0}^{j-1}\frac{F^{(k)}(\lambda)}{k!}v_{j-k}$, which is equal to the derivative $\left(\frac{\mathrm{d}^{j-1}}{\mathrm{d}z^{j-1}}F(z)v(z)|_{z=\lambda}\right)/(j-1)!$; see (2.3). By the definition of a root function this derivative is zero for $j = 1, 2, \ldots, m$, hence $\mathcal{F}(V, J) = O$. □

By the block structure of the matrices V and J defined in Theorem 2.8, we immediately find that (V, J) is an invariant pair. An invariant pair (V, M) as defined in Definition 2.10 may contain redundant information as, for example, $\big([V, V], \mathrm{diag}(M, M)\big)$ is an invariant pair of at least twice the necessary size. Also, any (V, M) with $V = O$ is a trivial invariant pair. The following concept of *minimal pairs* prevents such anomalies.

Definition 2.12 (Minimal pair, minimality index). A pair $(V, M) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ is called *minimal* if there is an integer p such that $\mathrm{rank}\mathcal{V}_p(V, M) = m$, where

$$\mathcal{V}_p(V,M) = \begin{bmatrix} V \\ VM \\ \vdots \\ VM^{p-1} \end{bmatrix}. \tag{2.17}$$

The smallest such p is called the *minimality index* of the pair (V, M).

For the matrix-valued function $F \in H(\Omega, \mathbb{C}^{2\times 2})$ in (2.1) the pair (V, J) with J, V as in (2.13)–(2.14) is a minimal invariant pair with minimality index 6. Note that the minimality index of an invariant pair (V, M) is generically equal to one when $m \leq n$. The following result gives a simple lower and upper bound on the minimality index. The lower bound appears in (Beyn 2012, Lemma 5.1) and the upper bound in (Kressner 2009).

Lemma 2.13. Let $\Omega \subseteq \mathbb{C}$ be a nonempty domain containing all distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$ of $F \in H(\Omega, \mathbb{C}^{n\times n})$, where each λ_i is of geometric multiplicity d_i and has corresponding partial multiplicities $m_{i,1} \geq m_{i,2} \geq \cdots \geq m_{i,d_i}$. In addition let $(V, M) \in \mathbb{C}^{n\times \overline{m}} \times \mathbb{C}^{\overline{m}\times \overline{m}}$ be an invariant pair with V, M, and $\overline{m}$ as defined in Theorem 2.8. Then the minimality index p of (V, M) satisfies

$$\sum_{i=1}^{s} m_{i,1} \leq p \leq \overline{m}.$$

Let (V, M) be a minimal invariant pair for F with minimality index p and write F in the form (2.15). If $M = UTU^*$ is the Schur decomposition of M then on using $f_j(M) = Uf_j(T)U^*$ we have that (2.16) is equivalent to

$$\mathcal{F}(VU, T) = \sum_{j=1}^{\ell} C_j VU f_j(T) = O,$$

i.e., (VU, T) is an invariant pair for F and it is also minimal since $\mathcal{V}_p(VU, T) = \mathcal{V}_p(V, M)U$. If we let $v = VUe_1$ then $v \neq 0$ because $\mathcal{V}_p(VU, T)e_1 = [1, t_{11}, \ldots, t_{11}^{p-1}]^T \otimes v \neq 0$ since $\mathcal{V}_p(VU, T)$ is of full column rank. Now, $t_{11} = \lambda$ is an eigenvalue of M and $VUf_j(\lambda)e_1 = f_j(\lambda)VUe_1 = f_j(\lambda)v$ so that

$$\mathcal{F}(VU, T)e_1 = \sum_{j=1}^{\ell} C_j VU f_j(T)e_1 = \sum_{j=1}^{\ell} f_j(\lambda)C_j v = 0,$$

showing that λ is also an eigenvalue of F.

Lemma 2.14. If (V, M) is a minimal pair for $F \in H(\Omega, \mathbb{C}^{n\times n})$ then the eigenvalues of M are eigenvalues of F.

The following theorem summarizes the correspondence between (minimal) invariant pairs and Jordan chains for F at λ. Results of this type have been derived by Gohberg et al. (1993, Lemma 2.1), Beyn et al. (2011, Proposition 2.4), and Effenberger (2013*a*, Theorem 3.1.13).

Theorem 2.15. Let $\lambda_1, \lambda_2, \ldots, \lambda_s$ be distinct eigenvalues of $F \in$

$H(\Omega, \mathbb{C}^{n\times n})$ and consider a matrix $V = [V_1, \ldots, V_s]$ with

$$V_i \;=\; \big[V_{i,1}, \ldots, V_{i,\widehat{d}_i}\big] \quad \text{and} \quad V_{ij} \;=\; \big[v_0^{ij}, v_1^{ij}, \ldots, v_{\widehat{m}_{ij}-1}^{ij}\big],$$

with all $v_0^{ij} \neq 0$. Then the columns of every V_{ij} form a Jordan chain for F at λ_i if and only if (V, J) is an invariant pair with a block Jordan matrix $J = \operatorname{diag}(J_1, J_2, \ldots, J_s)$, where $J_i = \operatorname{diag}(J_{i,1}, \ldots, J_{i,\widehat{d}_i})$ and every J_{ij} is an $\widehat{m}_{ij} \times \widehat{m}_{ij}$ Jordan block for λ_i. Moreover, (V, J) is minimal if and only if the vectors $v_0^{i,1}, v_0^{i,2}, \ldots, v_0^{i,\widehat{d}_i}$ are linearly independent for all $i = 1, 2, \ldots, d_i$.

Note that we have not made any explicit assumptions on the integers $\widehat{d}_i$ and $\widehat{m}_{ij}$ in Theorem 2.15. If we assume that $\widehat{d}_i$ equals the geometric multiplicity of each λ_i, then if (V, J) is a minimal pair the columns of each V_i in Theorem 2.15 form a complete system of Jordan chains. In this case the algebraic multiplicities of the eigenvalues of J coincide with the algebraic multiplicities of the corresponding eigenvalues of F. This gives rise to the following definition.

Definition 2.16 (Complete invariant pair). An invariant pair (V, M) of $F \in H(\Omega, \mathbb{C}^{n\times n})$ is called *complete* if it is minimal and the algebraic multiplicities of the eigenvalues of M are the same as the algebraic multiplicities of the corresponding eigenvalues of F.

Complete invariant pairs have been introduced by Kressner (2009) under the name *simple invariant pairs*. We prefer to use the term *complete* as it better aligns with the terminology of a complete system of Jordan chains in Definition 2.3.

2.6. Nonlinear Rayleigh functionals

Nonlinear Rayleigh functionals are the generalization of Rayleigh quotients for matrices. Duffin (1955) introduces them for symmetric overdamped quadratic eigenvalue problems. They are generalized to larger classes of Hermitian NEPs in (Rogers 1964, Hadeler 1967, Voss 2009), and to general NEPs in (Schreiber 2008, Schwetlick and Schreiber 2012).

Let λ be an eigenvalue of the matrix-valued function $F \in H(\Omega, \mathbb{C}^{n\times n})$ with corresponding right and left eigenvectors v and w. For some $\rho > 0$ and $\varepsilon < \pi/2$, let

$$\overline{\mathbb{D}}_{\lambda,\rho} = \{z \in \mathbb{C} : \; |z - \lambda| \leq \rho\},$$
$$\overline{\mathbb{K}}_{\varepsilon}(v) = \{x \in \mathbb{C}^n \setminus \{0\} : \angle(x, v) \leq \varepsilon\}$$

be neighborhoods of λ and v, respectively. Here $\angle(x, v)$ denotes the angle between the nonzero vectors x and v, that is,

$$\angle(x, v) = \cos^{-1} \frac{|v^* x|}{\|x\|_2 \|v\|_2}.$$

A functional

$$p : \overline{\mathbb{K}}_\varepsilon(v) \times \overline{\mathbb{K}}_\varepsilon(w) \ni (x, y) \mapsto p(x, y) \in \overline{\mathbb{D}}_{\lambda,\rho}$$

is a *two-sided nonlinear Rayleigh functional* if the following conditions hold:

$$p(\alpha x, \beta y) = p(x, y) \text{ for all nonzero } \alpha, \beta \in \mathbb{C}, \tag{2.18}$$

$$y^* F\big(p(x, y)\big)x = 0, \tag{2.19}$$

$$y^* F'\big(p(x, y)\big)x \neq 0. \tag{2.20}$$

The condition (2.20) restricts the functional to vectors close to eigenvectors corresponding to a simple eigenvalue.

The next result, which appears in (Schwetlick and Schreiber 2012, Theorem 5), concerns the local uniqueness of $p(x, y)$ and bounds the distance of $p(x, y)$ to the exact eigenvalue in terms of the angles between eigenvectors and approximations to eigenvectors.

Theorem 2.17. Let λ be a simple eigenvalue of $F \in H(\Omega, \mathbb{C}^{n\times n})$ with corresponding right and left eigenvectors v and w normalized to have unit 2-norms. Let $\rho > 0$ be such that $\overline{\mathbb{D}}_{\lambda,\rho} \subset \Omega$. Then there exist constants $0 < \rho_0 < \rho$, $0 < \varepsilon_0 < \varepsilon < \pi/2$ such that for all $(x, y) \in \overline{\mathbb{K}}_{\varepsilon_0}(v) \times \overline{\mathbb{K}}_{\varepsilon_0}(w)$ there exists a unique $p(x, y) \in \overline{\mathbb{D}}_{\lambda,\rho_0}$ satisfying $y^* F\big(p(x, y)\big)x = 0$ and

$$|p(x, y) - \lambda| \leq \frac{8}{3} \frac{\|F(\lambda)\|_2}{|y^* F'\big(p(x, y)\big)x|} \tan\big(\angle(x, v)\big) \tan\big(\angle(y, w)\big).$$

It then follows from Theorem 2.17 that $p(v, w) = \lambda$, i.e., $p(v, w)$ is an eigenvalue of F.

Example 2.18. Let F be the matrix-valued function defined in (2.1), and let Ω be the disk with center 0 and radius 3. If we let $x = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$ and $y = \left[\begin{smallmatrix} y_1 \\ y_2 \end{smallmatrix}\right]$ then the condition (2.20) is equivalent to $p(x, y) \neq 0$ and $\overline{y}_1 x_1 \neq 0$. Now if $\overline{y}_1 x_1 \neq 0$, (2.19) can be rewritten as

$$e^{\mathrm{i}p(x,y)^2} = -\frac{\overline{y}_1 x_2 + \overline{y}_2 x_1 + \overline{y}_2 x_2}{\overline{y}_1 x_1}$$

so that

$$\mathrm{i}p(x, y)^2 = \log\Big(-\frac{\overline{y}_1 x_2 + \overline{y}_2 x_1 + \overline{y}_2 x_2}{\overline{y}_1 x_1}\Big) + 2\pi k\mathrm{i}, \quad k = 0, \pm 1, \pm 2, \ldots$$

for any complex logarithm function log. Hence the set of two-sided nonlinear Rayleigh functionals for F is given by

$$p(x, y) = \sqrt{-\mathrm{i}\log\Big(-\frac{\overline{y}_1 x_2 + \overline{y}_2 x_1 + \overline{y}_2 x_2}{\overline{y}_1 x_1}\Big) + 2\pi k}, \quad k = 0, \pm 1, \pm 2, \ldots,$$

where for a complex number $z = \rho e^{\mathrm{i}\theta}$, $\sqrt{z} = \pm\sqrt{\rho}e^{\mathrm{i}\theta/2}$. Note that the functional $p(x, y)$ satisfies (2.18).

When $x = v$ and $y = w$ are right and left eigenvectors for F with $v = w = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, we recover the set of nonzero eigenvalues of F,

$$p(v, w) = \pm\sqrt{2\pi k}, \quad k = \pm 1, \pm 2, \ldots,$$

where we excluded $p(v, w) = 0$ corresponding to $k = 0$ since (2.20) implies that $p(v, w) \neq 0$.

Schreiber (2008) shows that the nonlinear Rayleigh functional p is stationary at the eigenvectors (v, w), that is,

$$|p(v + \Delta v, w + \Delta w) - \lambda| = O\big((\|\Delta v\|_2 + \|\Delta w\|_2)^2\big),$$

where $\lambda = p(v, w)$. In other words, the first-order terms in a perturbation expansion of p at the eigenvectors (v, w) vanish identically.

2.7. Some perturbation results

It is often of interest, for example in sensitivity analysis and also in numerical computing, to give simple regions in which the eigenvalues of F are located. Probably the most famous result for linear eigenvalue problems is Gershgorin's theorem; see (Horn and Johnson 1991, Chapter 6). Bindel and Hood (2013, Theorem 3.1) provide a generalization for NEPs.

Theorem 2.19 (Gershgorin's theorem for NEPs). Let $F(z) = D(z) + E(z)$ with $D, E \in H(\Omega, \mathbb{C}^{n\times n})$ and D diagonal. Then for any $0 \leq \alpha \leq 1$,

$$\Lambda(F) \subset \bigcup_{j=1}^{n} \mathbb{G}_j^{\alpha},$$

where $\mathbb{G}_j^{\alpha}$ is the jth generalized Gershgorin region

$$\mathbb{G}_j^{\alpha} = \big\{z \in \Omega : \ |d_{jj}(z)| \leq r_j(z)^{\alpha} c_j(z)^{1-\alpha}\big\}$$

and $r_j(z)$, $c_j(z)$ are the jth absolute row and column sums of E, that is,

$$r_j(z) = \sum_{k=1}^{n} |e_{jk}(z)|, \quad c_j(z) = \sum_{k=1}^{n} |e_{kj}(z)|.$$

Moreover, if $\mathbb{U}$ is a bounded connected component of $\bigcup_{j=1}^{n} \mathbb{G}_j^{\alpha}$ such that $\overline{\mathbb{U}} \subset \Omega$ then $\mathbb{U}$ contains the same number of eigenvalues of F and D. Furthermore, if $\mathbb{U}$ includes p connected components of the Gershgorin regions, it must contain at least p eigenvalues of F.

Also of interest is the sensitivity of an eigenvalue λ of $F \in H(\Omega, \mathbb{C}^{n\times n})$ under small perturbations $\Delta F \in H(\Omega, \mathbb{C}^{n\times n})$. To address this point, we

assume that λ is a nonzero simple eigenvalue with right eigenvector v and left eigenvector w. We consider F expressed as in (2.15) with the perturbation

$$\Delta F(z) = f_1(z)\Delta C_1 + f_2(z)\Delta C_2 + \cdots + f_\ell(z)\Delta C_\ell \in H(\Omega, \mathbb{C}^{n\times n}). \quad (2.21)$$

To measure the sensitivity of λ, we can use the *normwise condition number*

$$\kappa(\lambda, F) = \limsup_{\varepsilon\to 0} \Big\{ \frac{|\Delta\lambda|}{\varepsilon|\lambda|} : \big(F(\lambda+\Delta\lambda) + \Delta F(\lambda+\Delta\lambda)\big)(v+\Delta v) = 0, \\ \|\Delta C_j\|_2 \le \varepsilon\alpha_j,\ j = 1, \ldots, \ell \Big\}, \quad (2.22)$$

where the α_j are nonnegative parameters that allow freedom in how perturbations are measured—for example, in an absolute sense ($\alpha_j = 1$) or in a relative sense ($\alpha_j = \|C_j\|_2$). By setting $\alpha_j = 0$ we can force $\Delta C_j = 0$ and thus keep C_j unperturbed.

Theorem 2.20. The normwise condition number $\kappa(\lambda, F)$ in (2.22) is

$$\kappa(\lambda, F) = \frac{\big(\sum_{j=1}^{\ell} \alpha_j |f_j(\lambda)|\big)\|v\|_2\|w\|_2}{|\lambda||w^*F'(\lambda)v|}.$$

Proof. By expanding the first constraint in (2.22) and keeping only the first-order terms, we get

$$\Delta\lambda F'(\lambda)v + F(\lambda)\Delta v + \Delta F(\lambda)v = O(\varepsilon^2).$$

Premultiplying by w^* leads to

$$\Delta\lambda w^*F'(\lambda)v + w^*\Delta F(\lambda)v = O(\varepsilon^2).$$

Since λ is a simple eigenvalue, we have from (2.5) that $w^*F'(\lambda)v \neq 0$. Thus

$$\Delta\lambda = -\frac{w^*\Delta F(\lambda)v}{w^*F'(\lambda)v} + O(\varepsilon^2)$$

and so

$$\frac{|\Delta\lambda|}{\varepsilon|\lambda|} \le \frac{\big(\sum_{j=1}^{\ell} \alpha_j |f_j(\lambda)|\big)\|w\|_2\|v\|_2}{|\lambda||w^*F'(\lambda)v|} + O(\varepsilon).$$

Hence the expression on the right-hand side of $\kappa(\lambda, F)$ in the theorem is an upper bound for the condition number. To show that this bound can be attained we consider the matrix $H = wv^*/(\|w\|_2\|v\|_2)$, for which

$$\|H\|_2 = 1, \quad w^*Hv = \|v\|_2\|w\|_2.$$

Let

$$\Delta C_j = \varepsilon\alpha_j \frac{\overline{f_j(\lambda)}}{|f_j(\lambda)|} H, \quad j = 1, \ldots, \ell.$$

Then all the norm inequalities in (2.22) are satisfied as equalities and

$$|w^*\Delta F(\lambda)v| = \varepsilon\Big(\sum_{j=1}^{\ell}\alpha_j|f_j(\lambda)|\Big)\|w\|_2\|v\|_2.$$

Dividing by $\varepsilon|\lambda||w^*F'(\lambda)v|$ and taking the limit as $\varepsilon \to 0$ gives the desired equality. □

Pseudospectra are another established tool for gaining insight into the sensitivity of the eigenvalues of a matrix to perturbations; see Trefethen and Embree (2005) and the references therein. The *ε-pseudospectrum* of $F \in H(\Omega, \mathbb{C}^{n\times n})$ is the set

$$\Lambda_\varepsilon(F) = \bigcup_{\substack{\Delta F\in H(\Omega,\mathbb{C}^{n\times n})\\ \|\Delta F\|<\varepsilon}} \Lambda(F+\Delta F), \tag{2.23}$$

where, for example

$$\|\Delta F\| = \|\Delta F\|_\Omega := \sup_{z\in\Omega}\|\Delta F(z)\|_2. \tag{2.24}$$

If F is expressed as in (2.15) then we can consider perturbations ΔF of the form (2.21) and measure them using

$$\|\Delta F\| = \|\Delta F\|_{\max} := \max_{1\le j\le \ell}\|\Delta F_j\|_2/\alpha_j, \tag{2.25}$$

where the α_j are nonnegative parameters defined as for (2.22). Other measures of the perturbations can be used too; see for example (Tisseur and Higham 2001, Higham and Tisseur 2002, Michiels, Green, Wagenknecht and Niculescu 2006).

The following characterizations of $\Lambda_\varepsilon(F)$ (see (Bindel and Hood 2013, Proposition 4.1) and (Michiels et al. 2006, Theorem 1)) provide an easy way to check whether a point $z \in \Omega$ is in the ε-pseudospectrum of F or not.

Theorem 2.21. Let $F \in H(\Omega, \mathbb{C}^{n\times n})$. Then for $\Lambda_\varepsilon(F)$ defined by (2.23)

$$\Lambda_\varepsilon(F) = \big\{z \in \Omega :\ \|F(z)^{-1}\|_2 > \big(\varepsilon f(z)\big)^{-1}\big\},$$

where $f(z) = 1$ when the perturbations are measured using (2.24), and $f(z) = \sum_{j=1}^{\ell}\alpha_j|f_j(z)|$ when the perturbations are measured using (2.25).

The ε-pseudospectrum is connected to the *backward error* of an approximate eigenvalue $\widetilde{\lambda}$ of F defined by

$$\eta_F(\widetilde{\lambda}) = \min_{\substack{v\in\mathbb{C}^n\\ v\neq 0}}\eta_F(\widetilde{\lambda}, v), \tag{2.26}$$

where

$$\eta_F(\widetilde{\lambda}, v) = \min\{\varepsilon :\ (F(\widetilde{\lambda}) + \Delta F(\widetilde{\lambda}))v = 0,\ \|\Delta F\| \le \varepsilon\}$$

with perturbations measured using, for example, (2.24) or (2.25). Then, by comparing the definitions (2.23) and (2.26), we have that

$$\Lambda_\varepsilon(F) = \{z \in \Omega : \ \eta_F(z) < \varepsilon\}.$$

For an approximate eigenpair $(\widetilde{\lambda}, \widetilde{v})$ of F and for perturbations measured using (2.25), a straightforward generalization of (Tisseur 2000, Theorem 1 and Lemma 3) leads to explicit expressions for $\eta_F(\widetilde{\lambda}, \widetilde{v})$ and $\eta_F(\widetilde{\lambda})$,

$$\eta_F(\widetilde{\lambda}, \widetilde{v}) = \frac{\|F(\widetilde{\lambda})\widetilde{v}\|_2}{f(\widetilde{\lambda})\,\|\widetilde{v}\|_2} \quad \text{and} \quad \eta_F(\widetilde{\lambda}) = \frac{1}{f(\widetilde{\lambda})\,\|F(\widetilde{\lambda})^{-1}\|_2},$$

where $f(\widetilde{\lambda}) = \sum_{j=1}^{\ell} \alpha_j |f_j(\widetilde{\lambda})|$.

Example 2.22. Let us consider once more the function F from (2.1), now rewritten in the form

$$F(z) = e^{\mathrm{i}z^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} =: f_1(z)C_1 + f_2(z)C_2.$$

An easy computation shows that the nonzero eigenvalues of F, $\lambda_k = \pm\sqrt{2\pi k}$, $k = \pm 1, \pm 2, \ldots$, have small condition numbers

$$\kappa(\lambda_k, F) = \frac{1 + \sqrt{1 + (3 + 5^{1/2})/2}}{2\pi|k|},$$

and these condition numbers get smaller as k increases in modulus. This is confirmed by the spectral portrait in Figure 2.23, which shows that nonzero eigenvalues of F move only slightly even under very large perturbations of F. The zero eigenvalue is defective and more sensitive to perturbations.

The largest possible real part of points in the ε-pseudospectrum of F is called the *ε-pseudospectral abscissa* of F, i.e.,

$$\alpha_\varepsilon(F) = \max\{\mathrm{Re}\,(\lambda) : \ \Lambda_\varepsilon(F)\}.$$

This quantity is of particular interest for NEPs associated with systems of differential equations such as the DDE (1.1) since perturbations of size ε may cause the system to become unstable whenever $\alpha_\varepsilon(F) \geq 0$. The distance to instability of a stable system associated with F can then be defined as

$$\delta(F) = \inf\{\varepsilon : \ \alpha_\varepsilon(F) \geq 0\}.$$

Algorithms to compute $\alpha_\varepsilon(F)$ can be found in (Michiels and Guglielmi 2012, Verhees, Van Beeumen, Meerbergen, Guglielmi and Michiels 2014).

3. Hermitian NEPs

A matrix-valued function $F(z)$ is said to be Hermitian if $F(\bar{z}) = F(z)^*$ for all $z \in \mathbb{C}$. It is easily seen that the eigenvalues of a Hermitian F are either

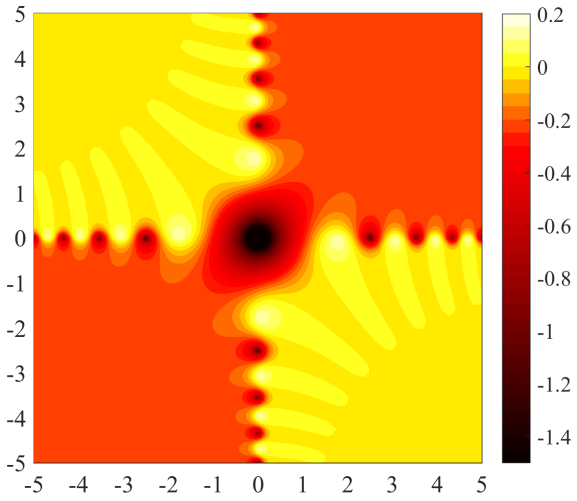


Figure 2.23: Spectral portrait of the matrix-valued function F defined in (2.1). The bar shows the mapping of colors to $\log_{10}$ of the reciprocal of resolvent norm, i.e., the mapping $z \mapsto \log_{10} \|F(z)^{-1}\|_2$.

real or they come in pairs $(\lambda, \bar{\lambda})$. If v and w are right and left eigenvectors of F for the eigenvalue $\lambda \in \mathbb{C}$, then w and v are right and left eigenvectors of F for the eigenvalue $\bar{\lambda}$, whereas if λ is a real eigenvalue of F for the right eigenvector v, then v is also a left eigenvector for the eigenvalue λ.

As for Hermitian linear eigenvalue problems, variational principles for real eigenvalues of Hermitian NEPs exist as we now discuss. For this we assume that the assumptions (A1)–(A3) below hold.

(A1) The matrix-valued function $F : \mathbb{R} \supseteq \mathbb{I} \to \mathbb{C}^{n\times n}$ is Hermitian and continuously differentiable on the open real interval $\mathbb{I}$.

(A2) For every $x \in \mathbb{C}^n \setminus \{0\}$, the real nonlinear equation

$$x^* F\big(p(x)\big)x = 0 \tag{3.1}$$

has at most one real solution $p(x) \in \mathbb{I}$.

Note that (3.1) defines implicitly the *one-sided generalized Rayleigh functional* p on some open subset $\mathbb{K}(p) \subseteq \mathbb{C}^n \setminus \{0\}$. When the Rayleigh functional p is defined on the whole space $\mathbb{C}^n \setminus \{0\}$, the NEP $F(\lambda)v = 0$ is called *overdamped*.

(A3) For every $x \in \mathbb{K}(p)$ and any $z \in \mathbb{I}$ such that $z \neq p(x)$,

$$\big(z - p(x)\big)\big(x^* F(z)x\big) > 0, \tag{3.2}$$

that is, $x^*F(z)x$ is increasing at $z = p(x)$.

When F is overdamped, assumption (A3) holds if $x^*F'(p(x))x > 0$ for all $x \in \mathbb{C}^n \setminus \{0\}$. Moreover, if F is overdamped and twice continuously differentiable, and $x^*F''(p(x))x \neq 0$ for all $x \in \mathbb{C}^n \setminus \{0\}$, then (3.2) is equivalent to $x^*F'(p(x))x > 0$ for all $x \in \mathbb{C}^n \setminus \{0\}$ (Szyld and Xue 2016, Proposition 2.4).

Let $\lambda \in \mathbb{I}$ be an eigenvalue of F, then λ is called a *kth eigenvalue of F* if $\mu = 0$ is the kth largest eigenvalue of the Hermitian matrix $F(\lambda)$. The next result, which can be found in (Hadeler 1968, Voss and Werner 1982, Voss 2009), provides a minmax characterization and a maxmin characterization of the real eigenvalues of F.

Theorem 3.1 (nonlinear variational principle). Assume that the matrix-valued function F satisfies (A1)–(A3). Then F has at most n eigenvalues in $\mathbb{I}$. Moreover, if λ_k is a kth eigenvalue of F, then

$$\lambda_k = \min_{\substack{V \in \mathbb{S}_k \\ V \cap \mathbb{K}(p) \neq \emptyset}} \max_{\substack{x \in V \cap \mathbb{K}(p) \\ x \neq 0}} p(x) \in \mathbb{I}, \tag{3.3}$$

$$\lambda_k = \max_{\substack{V \in \mathbb{S}_{k-1} \\ V^\perp \cap \mathbb{K}(p) \neq \emptyset}} \min_{\substack{x \in V^\perp \cap \mathbb{K}(p) \\ x \neq 0}} p(x) \in \mathbb{I}, \tag{3.4}$$

where $\mathbb{S}_j$ denotes the set of all subspaces of $\mathbb{C}^n$ of dimension j.

The characterization of the eigenvalues in (3.3) is a generalization of the minmax principle of Poincaré (1890) for linear eigenvalue problems and that in (3.4) is a generalization of the maxmin characterization of Courant (1920), Fischer (1905), and Weyl (1912).

When F satisfies (A1)–(A3) and is overdamped then there exist exactly n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of F in $\mathbb{I}$ and it follows from Theorem 3.1 that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, thereby offering an ordering of the real eigenvalues of F in $\mathbb{I}$. The variational principle can be used to derive an interesting property of eigenvalue approximations obtained by projecting F onto low-dimensional spaces; see (Szyld and Xue 2016, Theorem 2.7).

Theorem 3.2 (nonlinear Cauchy interlacing theorem). Assume that F satisfies (A1)–(A3) and is overdamped. Let $U \in \mathbb{C}^{n \times k}$ be of full column rank. Then the projected matrix-valued function $U^*F(z)U$ has exactly k eigenvalues satisfying the nonlinear variational principle, and if μ_j is the jth eigenvalue of $U^*F(z)U$ then $\lambda_j \leq \mu_j \leq \lambda_{n-k+j}$.

The nonlinear Cauchy interlacing theorem and the variational principle in Theorem 3.1 allow the development of special algorithms for Hermitian NEPs satisfying the assumptions (A1)–(A3) in Section 4.6.

Example 3.3. Let us consider the `loaded_string` problem with F defined

in (1.3). The associated quadratic matrix polynomial

$$Q(z) = -(z-1)F(z) = z^2C_2 - z(C_1 + C_2 + C_3) + C_1$$

is *hyperbolic* since its leading matrix coefficient C_2 is positive definite and the scalar equation $x^*Q(z)x = 0$ has two distinct real roots (Al-Ammari and Tisseur 2012, Theorem 3.4). As a result, Q has only real eigenvalues. But clearly, if λ is an eigenvalue of F, then it is an eigenvalue of Q. Hence F has only real eigenvalues. For $n = 100$ the first nine eigenvalues of F greater than 1 are given to 10 digits by

$$\begin{array}{lll} \lambda_1 = \texttt{4.482176546}, & \lambda_2 = \texttt{24.22357311}, & \lambda_3 = \texttt{63.72382114}, \\ \lambda_4 = \texttt{123.0312211}, & \lambda_5 = \texttt{202.2008991}, & \lambda_6 = \texttt{301.3101627}, \\ \lambda_7 = \texttt{420.4565631}, & \lambda_8 = \texttt{559.7575863}, & \lambda_9 = \texttt{719.3506601}, \end{array} \tag{3.5}$$

see (Solov′ëv 2006), and there is also an eigenvalue smaller than 1,

$$\lambda_0 = \texttt{0.45731848895}. \tag{3.6}$$

Compared to F, Q has $n-1$ extra eigenvalues at $z = 1$.

It is easy to check that property (A1) holds on the interval $\mathbb{I} = (1, +\infty)$. If we let

$$a(x) = x^*C_2x, \quad b(x) = -x^*(C_1 + C_2 + C_3)x, \quad c(x) = x^*C_1x,$$

then

$$p(x) = \frac{-b(x) + \sqrt{b^2(x) - 4a(x)c(x)}}{2a(x)} > 1$$

is well defined for every $x \in \mathbb{C}^n \setminus \{0\}$ since $a(x) > 0$, and it is the only root of $x^*F(p(x))x = 0$ in $\mathbb{I}$. Hence property (A2) holds and F is overdamped on $\mathbb{I}$. Now,

$$x^*F'(z)x = -x^*C_2x - \frac{1}{(z-1)^2}x^*C_3x < 0$$

for all $x \in \mathbb{C}^n \setminus \{0\}$ so that property (A3) holds for $-F(z)$. As a result, the eigenvalues of F in $\mathbb{I}$ satisfy the nonlinear variational principle in Theorem 3.1.

4. Solvers based on Newton's method

Newton's method is a natural approach to compute eigenvalues or eigenpairs of NEPs very efficiently and accurately provided that good initial guesses are available. The choice of an initial guess is typically the only crucial parameter of a Newton-type method, which is a great advantage over other NEP solution approaches. As a result, many algorithmic variants have been developed and applied over the years, including the Newton-QR iteration of Kublanovskaya (1970) and its variant in (Garrett, Bai and

Li 2016), the Newton-trace iteration of Lancaster (1966), nonlinear inverse iteration (Unger 1950), residual inverse iteration (Neumaier 1985), Rayleigh functional iterations (Lancaster 1961, Schreiber 2008), the block Newton method of Kressner (2009), and for large sparse NEPs, Jacobi–Davidson type methods (Betcke and Voss 2004, Sleijpen, Booten, Fokkema and van der Vorst 1996).

Generally speaking, there are two broad ways the NEP $F(\lambda)v = 0$ can be tackled by a Newton-type method. First, one can apply Newton's method to a scalar equation $f(z) = 0$ whose roots correspond to the wanted eigenvalues of F, or second, Newton's method can be applied directly to the vector problem $F(\lambda)v = 0$ together with some normalization condition on v. We discuss both approaches.

4.1. Newton's method for scalar functions

For a given initial guess $\lambda^{(0)}$, Newton's method for finding the roots of a scalar equation $f(z) = 0$ is given by

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{f\big(\lambda^{(k)}\big)}{f'\big(\lambda^{(k)}\big)}, \quad k = 0, 1, \ldots. \tag{4.1}$$

This iteration has local quadratic convergence to simple roots (Wilkinson 1965, Section 7.25). In order to apply it for the solution of an NEP $F(\lambda)v = 0$, we only need a scalar function f whose roots are the eigenvalues of F, and its derivative f'. Different methods result from different choices of f.

Probably the most obvious choice is

$$f(z) = \det F(z).$$

The trace theorem, e.g., (Lancaster 1966, Theorem 5.1), states that if the entries of $F(z)$ are differentiable functions of z, then for any $z \in \mathbb{C}$ for which $f(z) = \det F(z) \neq 0$ we have

$$f'(z) = \det F(z) \operatorname{trace}\big(F^{-1}(z)F'(z)\big), \tag{4.2}$$

and so the Newton iteration (4.1) can be rewritten as

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{1}{\operatorname{trace}\big(F^{-1}\big(\lambda^{(k)}\big)F'\big(\lambda^{(k)}\big)\big)} \quad k = 0, 1, \ldots. \tag{4.3}$$

We refer to (4.3) as the *Newton-trace iteration* (Lancaster 1966, Sec. 5.5); see also (Khazanov and Kublanovskaya 1988). Note that we only need the diagonal entries of $F^{-1}\big(\lambda^{(k)}\big)F'\big(\lambda^{(k)}\big)$. Nevertheless, with a straightforward implementation each iteration requires the factorization of an $n \times n$ matrix $F\big(\lambda^{(k)}\big)$ and $2n$ triangular solves, making it a rather expensive method. A basic MATLAB implementation of the Newton-trace iteration (4.2) is provided in Figure 4.1, where we use a simple stopping criterion based on the relative residual $|f\big(\lambda^{(k)}\big)|/\|F(\lambda^{(k)})\|_F$.

```
% Newton_trace
F  = @(z) [exp(1i*z.^2) 1; 1 1];
Fp = @(z) [2i*z*exp(1i*z.^2) 0; 0 0];
tol = 1e-8; maxit = 20; lam = 2.2 + 1e-4i;
for k = 0:maxit
    [L,U] = lu(F(lam));
    if abs(prod(diag(U)))/norm(F(lam),'fro')<tol, break, end
    corr = trace(U\(L\Fp(lam)));
    lam = lam - 1/corr;
end
if k < maxit, nbr_iter = k, lambda = lam, end
```

Figure 4.1: Basic MATLAB implementation of the Newton-trace iteration (4.3) for problem (2.1). The NEP parameters F and F' are specified in lines 2–3 and the method's parameters in line 4. Upon convergence, `lambda` is the eigenvalue and `nbr_iter` the number of iterations.

Kublanovskaya (1970) proposes to apply Newton's method to

$$f(z) = r_{nn}(z),$$

where $r_{nn}(z)$ is the bottom-right entry of the R factor in a rank-revealing QR decomposition of $F(z)$,

$$F(z)\Pi(z) = Q(z)R(z), \tag{4.4}$$

with $\Pi(z)$ being a permutation matrix which ensures that $r_{nn}(z)$ becomes zero before any other diagonal entry does; see also (Kublanovskaja 1969). Since $r_{nn}(z) = 0$ is equivalent to $\det F(z) = \det R(z) = 0$, we know that the roots of $f(z)$ are exactly the roots of $\det F(z) = 0$.

The permutation $\Pi(z)$ in (4.4) is not a continuous function of z, but in a small neighborhood of an approximate eigenvalue $\lambda^{(k)}$ of F the permutation can be kept constant. So if we let

$$F\big(\lambda^{(k)}\big)\Pi = Q_k R_k = Q_k \begin{bmatrix} R_{11}^{(k)} & r_{12}^{(k)} \\ 0 & r_{nn}^{(k)} \end{bmatrix}, \tag{4.5}$$

then in a small neighborhood of $\lambda_k^{(k)}$ we can use

$$F(z)\Pi = Q(z)R(z) = Q(z) \begin{bmatrix} R_{11}(z) & r_{12}(z) \\ 0 & r_{nn}(z) \end{bmatrix}$$

with $Q\big(\lambda^{(k)}\big) = Q_k$ and $R\big(\lambda^{(k)}\big) = R_k$. Garrett et al. (2016) show that

$$r_{nn}(z) = r_{nn}^{(k)} + e_n^T Q_k^* F'\big(\lambda^{(k)}\big)\Pi \begin{bmatrix} -p \\ 1 \end{bmatrix} (z - \lambda^{(k)}) + O(|z - \lambda^{(k)}|^2),$$

```
% Newton_QR
n = 2; F = @(z) [exp(1i*z.^2) 1; 1 1];
Fp = @(z) [2i*z*exp(1i*z.^2) 0; 0 0];
tol = 1e-8; maxit = 20; lam = 2.2 + 1e-4i;
for k = 0:maxit
    [Q,R,P] = qr(F(lam));
    if abs(R(n,n))/norm(F(lam),'fro') < tol, break, end
    p = R(1:n-1,1:n-1)\R(1:n-1,n);
    lam = lam - R(n,n)/(Q(:,n)'*Fp(lam)*P*[-p; 1]);
end
if k < maxit, nbr_iter = k, lambda = lam, end
```

Figure 4.2: Basic MATLAB implementation of the Newton-QR iteration for problem (2.1). The NEP parameters F and F' are specified in lines 2–3 and the method's parameters in line 4. Upon convergence, `lambda` is the eigenvalue and `nbr_iter` the number of iterations.

where p solves $R_{11}^{(k)}p = r_{12}^{(k)}$, and therefore

$$r'_{nn}\big(\lambda^{(k)}\big) = e_n^T Q_k^* F'\big(\lambda^{(k)}\big) \Pi \begin{bmatrix} -p \\ 1 \end{bmatrix}.$$

This leads to the *Newton-QR iteration* for a root of $r_{nn}(z)$,

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{r_{nn}\big(\lambda^{(k)}\big)}{r'_{nn}\big(\lambda^{(k)}\big)} = \lambda^{(k)} - \frac{r_{nn}^{(k)}}{e_n^T Q_k^* F'\big(\lambda^{(k)}\big) \Pi \begin{bmatrix} -p \\ 1 \end{bmatrix}}. \tag{4.6}$$

At convergence, we can take

$$v = \Pi \begin{bmatrix} -p \\ 1 \end{bmatrix}, \quad w = Q_k e_n$$

as approximations for the right and left eigenvectors of the converged approximate eigenvalue, respectively.

A basic MATLAB implementation of the Newton-QR iteration (4.6) is given in Figure 4.2. It requires the repeated computation of a rank-revealing QR factorization of $F(\lambda^{(k)})$ and hence is only feasible for problems of moderate size (unless F has some structure that can be exploited; see Section 4.5). However, the iteration (4.6) can be useful in the context of iterative refinement. Yang (1983) and Wobst (1987) consider an approach similar to that of Kublanovskaya by using an LU factorization with column pivoting in place of a rank-revealing QR factorization.

To avoid working with nearly singular matrices $F\big(\lambda^{(k)}\big)$, Andrew, Chu

and Lancaster (1995) propose to use instead of $F(z)$ the bordered matrix

$$G(z) = \begin{bmatrix} F(z) & b \\ c^T & 0 \end{bmatrix}, \tag{4.7}$$

where $b, c \in \mathbb{C}^n$ are such that the matrix $G(\lambda)$ is nonsingular and well-conditioned at a simple eigenvalue λ. Now if $c^T v \neq 0$ and $w^T b \neq 0$, where v and w are the right and left eigenvectors of F associated with the eigenvalue λ, then $G(\lambda)$ is nonsingular (Andrew, Chu and Lancaster 1993, Theorem 10.1). For a vector x such that $c^T x = 1$, the authors introduce the linear system

$$\begin{bmatrix} F(z) & b \\ c^T & 0 \end{bmatrix} \begin{bmatrix} x \\ f \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4.8}$$

For z near λ, the matrix $G(z)$ is nonsingular and hence $x = x(z)$ and $f = f(z)$ are smooth functions of z. By Cramer's rule we have

$$f(z) = \frac{\det F(z)}{\det G(z)} \tag{4.9}$$

and therefore $f(z) = 0$ if and only if $\det F(z) = 0$. Differentiating (4.8) with respect to z leads to

$$\begin{bmatrix} F(z) & b \\ c^T & 0 \end{bmatrix} \begin{bmatrix} x'(z) \\ f'(z) \end{bmatrix} = - \begin{bmatrix} F'(z)x(z) \\ 0 \end{bmatrix}. \tag{4.10}$$

The *BDS method* of Andrew et al. (1995) and the *implicit determinant method* of Spence and Poulton (2005) consist of applying a Newton iteration to f in (4.9). The latter method is detailed in Algorithm 4.3, with a basic MATLAB implementation given in Figure 4.4.

Algorithm 4.3: Implicit determinant method

Choose an initial approximate eigenvalue $\lambda^{(0)}$ and vectors $b, c \in \mathbb{C}^n$ such that $G(\lambda^{(0)})$ in (4.7) is nonsingular.

for $k = 0, 1, \ldots$ until convergence **do**

Solve (4.8) with $z = \lambda^{(k)}$ for $f\big(\lambda^{(k)}\big)$ and $x^{(k)}$.

Solve (4.10) with $z = \lambda^{(k)}$ for $f'\big(\lambda^{(k)}\big)$ using $x(z) = x^{(k)}$ for the right-hand side.

Perform the Newton update $\lambda^{(k+1)} = \lambda^{(k)} - f\big(\lambda^{(k)}\big)/f'\big(\lambda^{(k)}\big)$.

end

At convergence, $x^{(k)}$ provides an approximation to the right eigenvector v. Note that the factorization used to solve (4.8) can be reused to solve (4.10). Andrew et al. (1995) propose different choices for the vectors b and c leading

```
% Newton_implicit_determinant
n = 2; F = @(z) [exp(1i*z.^2) 1; 1 1];
Fp = @(z) [2i*z*exp(1i*z.^2) 0; 0 0];
tol = 1e-8; maxit = 20; b = [0; 1]; c = b; lam = 2.2 + 1e-4i;
for k = 0:maxit
    [L,U] = lu([F(lam) b; c.' 0]);
    xf = U\(L\[zeros(n,1); 1]);
    if abs(xf(n+1))/norm(F(lam),'fro') < tol, break, end
    xfp = U\(L\[-Fp(lam)*xf(1:n); 0]);
    lam = lam - xf(n+1)/xfp(n+1);
end
if k < maxit, nbr_iter = k, lambda = lam, end
```

Figure 4.4: Basic MATLAB implementation of the implicit determinant method for problem (2.1). The NEP parameters F and F' are specified in lines 2–3 and the method's parameters in line 4. Upon convergence, `lambda` is the eigenvalue and `nbr_iter` the number of iterations.

to the *row/column deletion method* ($b = e_i$, $c = e_j$), the *column substitution method* ($b \approx v$, $c = e_j$), giving the name BDS (bordered, deletion, substitution) for this class of methods.

Deflation may be necessary when computing several nearby eigenvalues in order to avoid the Newton iteration to converge to an already computed eigenvalue. Suppose we have already computed m roots λ_ℓ ($\ell = 1, \ldots, m$) of f near λ_0, each of multiplicity m_ℓ. As suggested by Wilkinson (1965, Section 7.48), we can apply Newton's method to

$$\widetilde{f}(z) = \frac{f(z)}{\prod_{\ell=1}^{m}(z-\lambda_\ell)^{m_\ell}},$$

which leads to the iteration

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{f(\lambda^{(k)})}{f'(\lambda^{(k)}) - f(\lambda^{(k)})\sum_{\ell=1}^{m}\frac{m_i}{\lambda^{(k)}-\lambda_\ell}}. \tag{4.11}$$

This strategy has been shown to work well in practice (Garrett et al. 2016).

When the derivative of F is not available, it can be replaced by a finite difference approximation, leading to a quasi-Newton method. In particular, the secant method is obtained by using the approximation

$$F'(\lambda^{(k)}) \approx \frac{F(\lambda^{(k)}) - F(\lambda^{(k-1)})}{\lambda^{(k)} - \lambda^{(k-1)}}. \tag{4.12}$$

Example 4.5 (Basins of attraction). Newton's method typically requires a good initial guess $\lambda^{(0)}$ for the eigenvalue of interest. Let us illustrate this with the matrix-valued function F defined in (2.1). Recall that F has

eigenvalues 0, $\pm\sqrt{2\pi}$, and $\pm\mathrm{i}\sqrt{2\pi}$ in the square

$$\Omega = \{z \in \mathbb{C} : -3 \le \mathrm{Re}\,(z) \le 3,\ -3 \le \mathrm{Im}\,(z) \le 3\}.$$

We generate 10^6 equidistant initial guesses in Ω and for each run the basic MATLAB codes given in Figures 4.1–4.4 with tolerance `tol = 1e-8` and `maxit = 20`. All codes use as stopping criterion

$$\frac{\left|f\left(\lambda^{(k)}\right)\right|}{\|F\left(\lambda^{(k)}\right)\|_F} \le \texttt{tol}, \tag{4.13}$$

but $f(z) = \mathrm{det}F(z)$ for the Newton-trace iteration, $f(z) = r_{nn}(z)$ for the Newton-QR iteration, and $f(z) = \mathrm{det}F(z)/\mathrm{det}G(z)$ for the implicit determinant method. The convergence basins of the Newton iteration for these three different scalar functions are illustrated in Figure 4.6. For this particular example and our choice of the parameters b and c for the bordered matrix (4.7), the convergence basins of the Newton-trace iteration and the implicit determinant method are rotated versions of each other (at least to visual accuracy). The basins of convergence for the Newton-QR iteration are quite different and somewhat smaller. The Newton-trace iteration (4.3) with $F'\left(\lambda^{(k)}\right)$ replaced by the secant approximation (4.12) produces basins of convergence similar to that of Newton's method for $\mathrm{det}F(z) = 0$ but of smaller sizes.

4.2. Newton's method for the vector equation

Newton's method can be applied directly to the NEP $F(\lambda)v = 0$ rather than to a scalar equation $f(z) = 0$ whose roots are eigenvalues of F. For this, a normalization condition on the eigenvector of the form $u^*v = 1$ for some nonzero vector u is added to $F(\lambda)v = 0$ so as to have $n+1$ equations for the $n+1$ unknowns in (λ, v). Note that $u^*v = 1$ ensures that v is not the zero vector. This approach is briefly mentioned in (Unger 1950) and discussed further in (Ruhe 1973). In order to apply Newton's method to $\mathcal{N}\left[\begin{smallmatrix} v \\ \lambda \end{smallmatrix}\right] = 0$, where

$$\mathcal{N}\begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} F(\lambda)v \\ u^*v - 1 \end{bmatrix},$$

we need the Jacobian matrix of the $(n+1)$-dimensional operator $\mathcal{N}$, which is readily calculated as

$$J_{\mathcal{N}}\begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} F(\lambda) & F'(\lambda)v \\ u^* & 0 \end{bmatrix}.$$

Newton's iteration is now given as

$$\begin{bmatrix} v^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \left(J_{\mathcal{N}}\begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix}\right)^{-1} \mathcal{N}\begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix}. \tag{4.14}$$

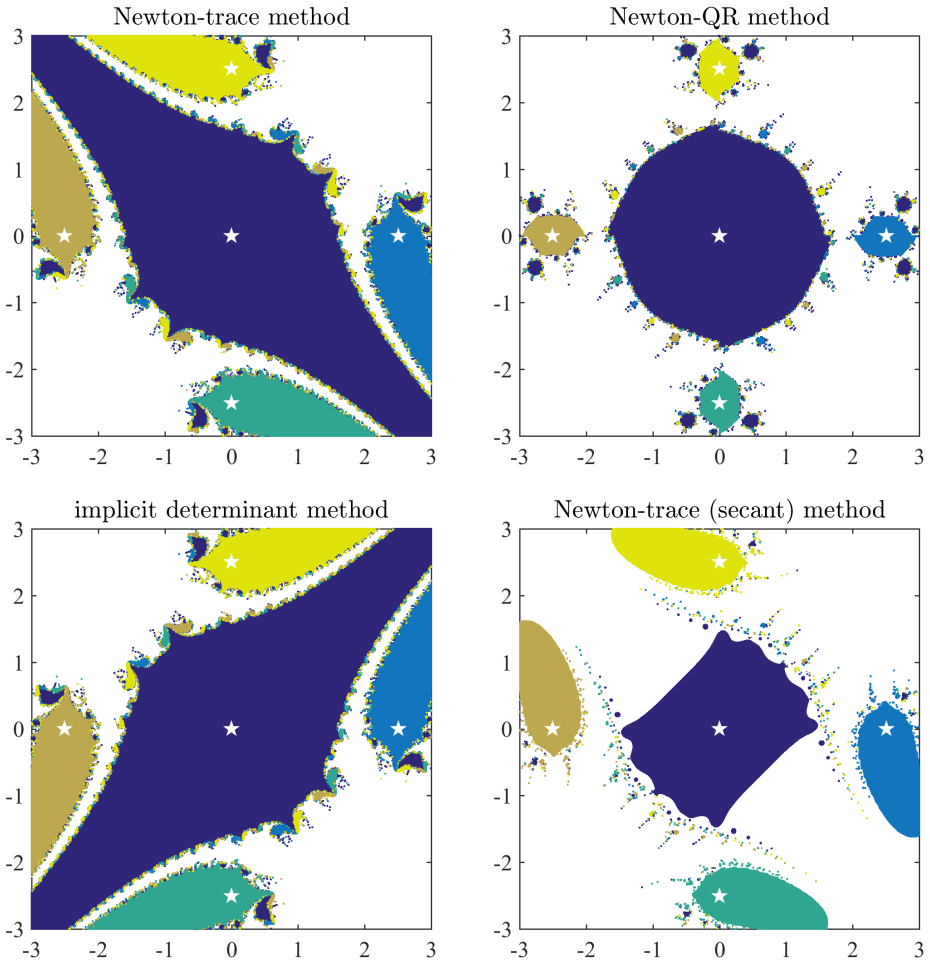


Figure 4.6: Basins of convergence for various Newton-type methods discussed in Example 4.5 for problem (2.1). Each initial point in the square region Ω is colored according to the eigenvalue a method is converging to, with the five exact eigenvalues of the NEP in Ω indicated by white $\star$ symbols. White areas indicate that no convergence is achieved within 20 iterations or the method converged to an eigenvalue outside Ω.

If we assume that the approximate eigenvector $v^{(k)}$ is normalized such that $u^*v^{(k)} = 1$ for some nonzero vector u, then (4.14) can be rewritten as

$$F\big(\lambda^{(k)}\big)v^{(k+1)} = -(\lambda^{(k+1)} - \lambda^{(k)})F'\big(\lambda^{(k)}\big)v^{(k)}, \tag{4.15}$$

$$u^*v^{(k+1)} = 1. \tag{4.16}$$

The Newton iteration (4.15)–(4.16) is equivalent to the *nonlinear inverse iteration* (Unger 1950), which is given in Algorithm 4.7 with a basic MATLAB implementation given in Figure 4.8.

Algorithm 4.7: Nonlinear inverse iteration

Choose an initial pair $(\lambda^{(0)}, v^{(0)})$ with $\|v^{(0)}\| = 1$ and a nonzero vector u.

for $k = 0, 1, \ldots$ until convergence **do**

Solve $F\big(\lambda^{(k)}\big)\widetilde{v}^{(k+1)} = F'\big(\lambda^{(k)}\big)v^{(k)}$ for $\widetilde{v}^{(k+1)}$.

Set $\lambda^{(k+1)} = \lambda^{(k)} - \dfrac{u^*v^{(k)}}{u^*\widetilde{v}^{(k+1)}}$.

Normalize $v^{(k+1)} = \widetilde{v}^{(k+1)}/\|\widetilde{v}^{(k+1)}\|$.

end

The normalization of $v^{(k+1)}$ is necessary to avoid numerical overflow or underflow and any vector norm can be used. When $\lambda^{(k)}$ is close to an eigenvalue, the matrix $F\big(\lambda^{(k)}\big)$ of the linear system to be solved is nearly singular. However, standard theory of inverse iteration (see, e.g., (Ipsen 1997, Section 6.3) or (Peters and Wilkinson 1979, Section 2)) shows that the error in the computed vector $\widetilde{v}^{(k+1)}$ will be almost parallel to $v^{(k+1)}$, that is, the inaccuracy is concentrated in the length of $\widetilde{v}^{(k+1)}$ and not its direction. As it is derived from a Newton iteration, the nonlinear inverse iteration converges locally and quadratically for any simple eigenpair. The vector u in (4.16) and in Algorithm 4.7 can be chosen in a number of ways as discussed below.

- A simple choice is to take u to be the ith unit vector e_i, which corresponds to keeping the ith entry of the vectors $v^{(k)}$ constant.
- To prevent the iteration from converging to previously computed eigenpairs, u can be chosen orthogonal to already computed eigenvectors (Anselone and Rall 1968).
- Choosing $u = F\big(\lambda^{(k)}\big)^* w^{(k)}$, where $w^{(k)}$ is an approximate left eigenvector, allows us to rewrite the eigenvalue update as

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{w^{(k)*}F\big(\lambda^{(k)}\big)v^{(k)}}{w^{(k)*}F'\big(\lambda^{(k)}\big)v^{(k)}},$$

```
% inverse_iteration
n = 2; F = @(z) [exp(1i*z.^2) 1; 1 1];
Fp = @(z) [2i*z*exp(1i*z.^2) 0; 0 0];
tol = 1e-8; maxit = 20; lam = 2.2 + 1e-4i;
v = [1; 1]; v = v/norm(v); u = [1; 0];
for k = 0:maxit-1
    if norm(F(lam)*v) < tol, break; end
    vt = F(lam)\Fp(lam)*v;
    lam = lam - u'*v/(u'*vt)
    v = vt/norm(vt);
end
if k < maxit, nbr_iter = k, lambda = lam, end
```

Figure 4.8: Basic MATLAB implementation of the nonlinear inverse iteration for problem (2.1). The NEP parameters F and F' are specified in lines 2–3 and the method's parameters in lines 4–5. Upon convergence, `lambda` is the eigenvalue and `nbr_iter` the number of iterations.

which is the *generalized Rayleigh quotient iteration* that Lancaster (1966) derived for the polynomial eigenvalue problem.

As mentioned in the third point above, the nonlinear inverse iteration can be combined with the computation of a left eigenvector, and a two-sided Rayleigh functional as defined in Section 2.6 to yield the *two-sided Rayleigh functional iteration* (Schreiber 2008), given in Algorithm 4.9.

Algorithm 4.9: Two-sided Rayleigh functional iteration

Choose an initial triple $(\lambda^{(0)}, v^{(0)}, w^{(0)})$ with $\|v^{(0)}\| = \|w^{(0)}\| = 1$.

for $k = 0, 1, \ldots$ until convergence **do**

- Solve $F\big(\lambda^{(k)}\big)\widetilde{v}^{(k+1)} = F'\big(\lambda^{(k)}\big)v^{(k)}$ for $\widetilde{v}^{(k+1)}$.
- Set $v^{(k+1)} = \widetilde{v}^{(k+1)}/\|\widetilde{v}^{(k+1)}\|$.
- Solve $F\big(\lambda^{(k)}\big)^*\widetilde{w}^{(k+1)} = F'\big(\lambda^{(k)}\big)^*w^{(k)}$ for $\widetilde{w}^{(k+1)}$.
- Set $w^{(k+1)} = \widetilde{w}^{(k+1)}/\|\widetilde{w}^{(k+1)}\|$.
- Find the root ρ of the scalar equation $w^{(k+1)*}F(\rho)v^{(k+1)} = 0$ closest to $\lambda^{(k)}$ and set $\lambda^{(k+1)} = \rho$.

end

Schreiber (2008) show that Algorithm 4.9 achieves local cubic convergence for a simple eigentriple (λ, v, w). Note that at each iteration, an LU factor-

ization of $F\big(\lambda^{(k)}\big)$ used to solve the first linear system can be reused to solve the second linear system involving $F^*\big(\lambda^{(k)}\big)$.

Each step of the nonlinear inverse iteration, Algorithm 4.7, requires the factorization of $F\big(\lambda^{(k)}\big)$ to solve the linear system, which cannot be reused at the next iteration since $\lambda^{(k)}$ varies. We could replace $\lambda^{(k)}$ by a fixed value σ, say $\sigma = \lambda^{(0)}$, but then the resulting iteration would converge to an eigenpair (μ, v) of the linear problem $Av = \mu Bv$, where $A = F(\sigma)$ and $B = F'(\lambda_*)$ for some $\lambda_* \in \mathbb{C}$. If $F(z)$ is linear in z so that $F'(z)$ is a constant matrix, we could easily recover an eigenpair (λ, v) of $F(\lambda)$ from (μ, v), since $F(\lambda)v = 0$ and $Av = \mu Bv$ are directly related eigenvalue problems. This is not the case if $F(z)$ is truly nonlinear in z.

Neumaier (1985) shows that this difficulty can be avoided by considering a variant of the nonlinear inverse iteration based on the use of the residual. If F is twice continuously differentiable, iteration (4.15) can be rewritten as

$$\begin{aligned} v^{(k)} - v^{(k+1)} &= v^{(k)} - (\lambda^{(k+1)} + \lambda^{(k)})F\big(\lambda^{(k)}\big)^{-1}F'\big(\lambda^{(k)}\big)v^{(k)} \\ &= F\big(\lambda^{(k)}\big)^{-1}\big(F\big(\lambda^{(k)}\big) + (\lambda^{(k+1)} - \lambda^{(k)})F'\big(\lambda^{(k)}\big)\big)v^{(k)} \\ &= F\big(\lambda^{(k)}\big)^{-1}F\big(\lambda^{(k+1)}\big)v^{(k)} + O(|\lambda^{(k+1)} - \lambda^{(k)}|^2). \end{aligned}$$

Ignoring the second-order term leads to an iteration

$$v^{(k+1)} = v^{(k)} - F\big(\lambda^{(k)}\big)^{-1}F\big(\lambda^{(k+1)}\big)v^{(k)}, \tag{4.17}$$

where the new approximant $\lambda^{(k+1)}$ for the eigenvalue λ needs to be determined beforehand. Neumaier (1985) shows that replacing $\lambda^{(k)}$ by a fixed shift σ in (4.17) does not destroy the convergence of the iteration to the wanted eigenpair. This leads to the *residual inverse iteration*, the pseudocode of which is given in Algorithm 4.10.

Algorithm 4.10: Residual inverse iteration

Choose an initial pair $(\lambda^{(0)}, v^{(0)})$ with $u^*v^{(0)} = 1$ for some nonzero vector u. Set $\sigma = \lambda^{(0)}$.

for $k = 0, 1, \ldots$ until convergence **do**

Solve for z the scalar equation

$$u^*F(\sigma)^{-1}F'(z)v^{(k)} = 0, \tag{4.18}$$

and accept as $\lambda^{(k+1)}$ the root z closest to $\lambda^{(k)}$.

Solve $F(\sigma)x^{(k)} = F\big(\lambda^{(k+1)}\big)v^{(k)}$ for $x^{(k)}$.

Set $v^{(k+1)} = \widetilde{v}^{(k+1)}/\big(u^*\widetilde{v}^{(k+1)}\big)$, where $\widetilde{v}^{(k+1)} = v^{(k)} - x^{(k)}$.

end

The initial vector $v^{(0)}$ can be computed by solving

$$F(\sigma)\widetilde{v}^{(0)} = b, \quad v^{(0)} = \widetilde{v}^{(0)}/\big(u^*\widetilde{v}^{(0)}\big)$$

for some nonzero vector b. As suggested by Wilkinson (1965) for the standard inverse iteration, we can choose $u = e$, where e is the vector of all ones, and $b = Le$, where $F(\sigma) = LU$ is an LU factorization. Then $\widetilde{v}^{(0)} = U^{-1}e$. Note that compared with the nonlinear inverse iteration, Algorithm 4.7, in the residual inverse iteration the eigenvector is updated by solving a linear system whose right-hand side is the NEP residual, giving its name to the iteration. Due to the fixed shift σ, the residual inverse iteration converges only linearly, and in practice, it is advisable to update the shift every now and then, in particular when the convergence is slow.

4.3. Deflation of computed eigenvalues and eigenpairs

The methods we described so far are directed towards computing one eigenvalue or one eigenpair only. In principle, several runs of the same iteration with different initial guesses could return several eigenpairs, but special care needs to be taken to prevent the iteration from converging to already computed eigenpairs. This is what *deflation* aims to achieve.

A standard deflation technique consists of applying a nonequivalence transformation to the matrix-valued function F that maps the already computed eigenvalues to infinity. This can be done as follows. Suppose we have computed ℓ simple eigenvalues of F, $\lambda_1, \dots, \lambda_\ell$ and let $x_i, y_i \in \mathbb{C}^n$ be such that $y_i^* x_i = 1$, $i = 1, \dots, \ell$. Consider

$$\widetilde{F}(z) = F(z)\prod_{i=1}^{\ell}\Big(I - \frac{z-\lambda_i-1}{z-\lambda_i}y_i x_i^*\Big). \tag{4.19}$$

Then it is not difficult to see that

$$\Lambda(\widetilde{F}) = \Lambda(F) \setminus \{\lambda_1, \dots, \lambda_\ell\} \cup \{\infty\}.$$

Indeed, since $y_i^* x_i = 1$,

$$\det\widetilde{F}(z) = \det F(z)\prod_{i=1}^{\ell}\Big(\frac{1}{\lambda_i - z}\Big)$$

so that λ_i is not a root of $\det\widetilde{F}(z)$ since it is a simple root of $\det F(z)$. With the exception of the eigenvalues λ_i, $i = 1, \dots, \ell$, $\widetilde{F}$ and F have the same finite eigenvalues. Now suppose that F does not have an eigenvalue at infinity, that is, $G(z) = F(1/z)$ does not have an eigenvalue at zero (see

Remark 2.2), and let $\widetilde{G}(z) = \widetilde{F}(1/z)$. Then from

$$\det\widetilde{G}(z) = \det G(z) \prod_{i=1}^{\ell} \Big(\frac{z}{1 - z\lambda_i}\Big)$$

we see that $\widetilde{G}$ has ℓ eigenvalues at zero, and therefore $\widetilde{F}$ has ℓ eigenvalues at infinity. It follows from (4.19) that if $\widetilde{v}$ is an eigenvector of $\widetilde{F}$ with eigenvalue λ then

$$v = \prod_{i=1}^{\ell} \Big(I - \frac{z - \lambda_i - 1}{z - \lambda_i} y_i x_i^*\Big)\widetilde{v} \tag{4.20}$$

is an eigenvector of F associated with the eigenvalue λ. To compute the next eigenpair of F, we can apply any Newton-based method to $\widetilde{F}(z)$ and recover the eigenvector using (4.20).

The choice of the vectors x_i and y_i affects the basins of convergence of Newton's method. A choice that seems to work well in practice is to set x_i and y_i to be (approximate) right and left eigenvectors corresponding to λ_i. Note that similar deflation strategies have been used in (Ferng, Lin, Pierce and Wang 2001) and (Huang, Lin and Mehrmann 2016).

Example 4.11. Using the basic MATLAB implementation of the nonlinear inverse iteration given in Figure 4.8, we obtain an approximation $\widehat{\lambda} =$`lam` to the eigenvalue $\lambda = \sqrt{2\pi}$ of the matrix-valued function F in (2.1), which we move to infinity by constructing

$$\widetilde{F}(z) = \begin{bmatrix} e^{\mathrm{i}z^2} & 1 \\ 1 & 1 \end{bmatrix} \Big(I - \frac{z - \widehat{\lambda} - 1}{z - \widehat{\lambda}} y x^*\Big). \tag{4.21}$$

We consider two choices for x and y:

(a) $x = y = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, i.e., x and y are right and left eigenvectors of F at λ,
(b) $x = y = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$.

As in Example 4.5 we generate 10^6 equidistant initial guesses $\lambda^{(0)}$ in

$$\Omega = \{z \in \mathbb{C} : -3 \le \mathrm{Re}\,(z) \le 3,\ -3 \le \mathrm{Im}\,(z) \le 3\},$$

and for each $\lambda^{(0)}$ and $v^{(0)} = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, we run the nonlinear inverse iteration as implemented in Figure 4.8 but with $\widetilde{F}$ in place of F. The basins of convergence are shown in Figure 4.12. The plot on the left corresponds to the above choice (a) for x and y. It shows that the deflation is successful as there is no basin of convergence to the point $\sqrt{2\pi}$. The right plot is obtained with the above choice (b). In this case the point $\sqrt{2\pi}$ still has a basin of convergence even if it is no longer an eigenvalue of $\widetilde{F}$. This unwanted behavior is likely caused by numerical ill-conditioning of the matrix $\widetilde{F}(z)$

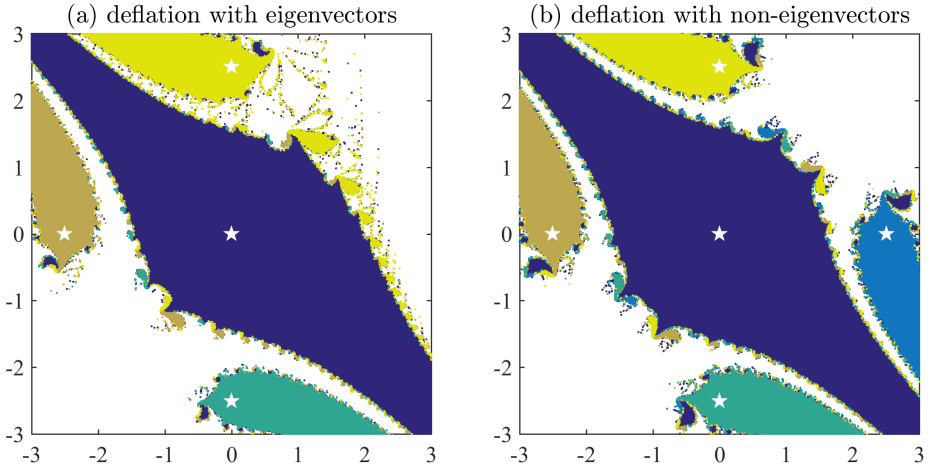


Figure 4.12: Basins of convergence for the nonlinear inverse iteration applied to $\widetilde{F}$ in (4.21) for two different choices, (a) and (b), of x and y discussed in Example 4.11. Each initial point in the square region Ω is colored according to the eigenvalue the method is converging to, with the exact eigenvalues of the NEP in Ω indicated by white $\star$ symbols. White areas indicate that no convergence is achieved within 20 iterations or the method converged to an eigenvalue outside Ω.

near $\sqrt{2\pi}$. Indeed, even if an eigenvalue λ is exactly deflated from F to obtain $\widetilde{F}$ and hence $\det\big(\widetilde{F}(z)\big)$ is nonzero and holomorphic in a punctured neighborhood of λ, the matrix $\widetilde{F}(z)$ can have two eigenvalues which converge to zero and infinity, respectively, as $z \to \lambda$. Special care has to be taken when working with such ill-conditioned matrices.

Another approach proposed by Effenberger (2013*b*) makes use of minimal invariant pairs as defined in Section 2.5. Effenberger's deflation strategy works as follows. Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ and suppose that we have already computed a minimal invariant pair $(V, M) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ for F with minimality index p. We want to extend (V, M) into another minimal invariant pair

$$(\widehat{V}, \widehat{M}) = \left(\begin{bmatrix} V & x \end{bmatrix}, \begin{bmatrix} M & b \\ 0 & \lambda \end{bmatrix} \right) \in \mathbb{C}^{n\times(m+1)} \times \mathbb{C}^{(m+1)\times(m+1)}$$

of one size larger. By Definition 2.10, the pair $(\widehat{V}, \widehat{M})$ is invariant if and only if $\mathcal{F}(\widehat{V}, \widehat{M}) = O$. Since

$$(zI - \widehat{M})^{-1} = \begin{bmatrix} (zI - M)^{-1} & (zI - M)^{-1} b (z-\lambda)^{-1} \\ 0 & (z-\lambda)^{-1} \end{bmatrix},$$

we have

$$\mathcal{F}(\widehat{V},\widehat{M}) = \begin{bmatrix}\mathcal{F}(V,M) & F(\lambda)x + U(\lambda)b\end{bmatrix}$$

with

$$U(\lambda) = \frac{1}{2\pi \mathrm{i}} \int_{\Gamma} F(z)V(zI - M)^{-1}(z-\lambda)^{-1}\mathrm{d}z \tag{4.22}$$

and Γ a contour enclosing the eigenvalues of M and λ. Since the pair (V, M) is invariant, $\mathcal{F}(V, M) = O$, and so $(\widehat{V}, \widehat{M})$ is an invariant pair if and only if

$$F(\lambda)x + U(\lambda)b = 0. \tag{4.23}$$

The condition that $(\widehat{V}, \widehat{M})$ be minimal is more involved. Effenberger (2013*b*, Lemma 3.4) shows that $(\widehat{V}, \widehat{M})$ is minimal with minimality index not exceeding $p + 1$ if

$$A(\lambda)x + B(\lambda)b = 0, \tag{4.24}$$

where the $m \times n$ matrix-valued function $A(\lambda)$ and the $m \times m$ matrix-valued function $B(\lambda)$ are given by

$$A(\lambda) = \sum_{j=0}^{p} \lambda^j (VM^j)^*, \quad B(\lambda) = \sum_{j=1}^{p} (VM^j)^* V q_j(\lambda)$$

with $q_j(\lambda) = \sum_{k=0}^{j-1} \lambda^k M^{j-k-1}$. It then follows from (4.23) and (4.24) that $\left(\lambda, \left[\begin{smallmatrix} x \\ b \end{smallmatrix}\right]\right)$ is an eigenpair of the $(n+m) \times (n+m)$ NEP

$$\widetilde{F}(\lambda)\widetilde{v} = 0, \tag{4.25}$$

where

$$\widetilde{F}(\lambda) = \begin{bmatrix} F(\lambda) & U(\lambda) \\ A(\lambda) & B(\lambda) \end{bmatrix}, \quad \widetilde{v} = \begin{bmatrix} x \\ b \end{bmatrix} \neq 0.$$

The matrix-valued function $\widetilde{F}(\lambda)$ is holomorphic since $U(\lambda)$ is holomorphic (Effenberger 2013*b*, Lemma 3.2), and $A(\lambda)$ and $B(\lambda)$ are matrix polynomials. If $F(\lambda)$ is regular, then so is $\widetilde{F}(\lambda)$. Moreover, if $\left(\lambda, \left[\begin{smallmatrix} x \\ b \end{smallmatrix}\right]\right)$ is an eigenpair of $\widetilde{F}(\lambda)$, then $\left([Vx], \left[\begin{smallmatrix} M & b \\ 0 & \lambda \end{smallmatrix}\right]\right)$ is a minimal invariant pair for $F(\lambda)$ (Effenberger 2013*b*, Theorem 3.6). This shows that the pair (V, M) is deflated from the computation when solving the NEP (4.25) in place of $F(\lambda)v = 0$. An eigenpair for the latter can be computed using any of the methods described in Sections 4.1–4.2. But for these, we need to evaluate $U(\lambda)$ in (4.22) and its derivative at a scalar λ. This can be done via numerical integration techniques similar to those described in Section 5.2. Note also that, by using $\lambda I - M = (zI - M) - (z - \lambda)I$, (4.22), and the definition of invariant

pairs, we have

$$\begin{aligned} U(\lambda)(\lambda I - M) &= \frac{1}{2\pi \mathrm{i}} \int_\Gamma F(z)V(z-\lambda)^{-1}\mathrm{d}z - \frac{1}{2\pi \mathrm{i}} \int_\Gamma F(z)V(zI-M)^{-1}\mathrm{d}z \\ &= F(\lambda)V + \mathcal{F}(V,M) = F(\lambda)V; \end{aligned}$$

see (Effenberger 2013*b*, Lemma 4.2). Hence, if λ is not an eigenvalue of M, then $U(\lambda) = F(\lambda)V(\lambda I - M)^{-1}$. For large NEPs, Effenberger (2013*b*) combines this deflation strategy with a Jacobi–Davidson method (see Section 4.5) and uses a contour integration method as described in Section 5 to solve the the projected NEP.

4.4. Block Newton method

Kressner (2009) presents a block analog of the Newton iteration on vectors which computes a complete invariant pair $(V, M) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ for $F \in H(\Omega, \mathbb{C}^{n\times n})$ with minimality index p. Recall from Definition 2.16 that an invariant pair (V, M) is complete if it is minimal and the algebraic multiplicities of the eigenvalues of M are the same as the algebraic multiplicities of the eigenvalues of F.

Let (V, M) be an invariant pair for F so that

$$\mathcal{F}(V, M) = O_{n\times m}, \tag{4.26}$$

where $\mathcal{F}$ is as in Definition 2.10, and assume that (V, M) satisfies the normalization condition

$$\mathcal{N}(V, M) = O_{m\times m}, \tag{4.27}$$

where $\mathcal{N}(V, M) = U^*\mathcal{V}_p(V, M) - I_m$ with $U \in \mathbb{C}^{pn\times m}$ a fixed matrix of full column rank and $\mathcal{V}_p(V, M)$ is as in (2.17). To apply Newton's method to (4.26)–(4.27), we need the Fréchet derivatives of $\mathcal{F}$ and $\mathcal{N}$ at (V, M), which are given by

$$\begin{aligned} L_{\mathcal{F}}(\Delta V, \Delta M) &= \mathcal{F}(\Delta V, M) + \frac{1}{2\pi \mathrm{i}} \int_\Gamma F(z)V(zI-M)^{-1}\Delta M(zI-M)^{-1}\mathrm{d}z, \\ L_{\mathcal{N}}(\Delta V, \Delta M) &= U^*\mathcal{V}_p(\Delta V, M) + \sum_{j=1}^{p-1} U_j^* V\Big(\sum_{i=0}^{j} M^i \Delta M M^{j-i-1}\Big), \end{aligned}$$

where $U^* = [U_0^*, \ldots, U_{p-1}^*]$ with $U_j \in \mathbb{C}^{n\times m}$. Kressner (2009, Theorem 10) shows that the invariant pair (V, M) is complete if and only if the linear matrix operator

$$\begin{aligned} \mathcal{M} : \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m} &\to \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m} \\ (\Delta V, \Delta M) &\mapsto \big(L_{\mathcal{F}}(\Delta V, \Delta M), L_{\mathcal{N}}(\Delta V, \Delta M)\big) \end{aligned}$$

corresponding to the Jacobian of (4.26)–(4.27) at (V, M) is invertible. Then,

by the implicit function theorem for holomorphic functions (Krantz 1982), we have that the entries of a complete invariant pair (V, M) vary analytically under analytic changes of F. Hence, complete invariant pairs are well-posed and Newton's method for solving (4.26)–(4.27) converges locally quadratically to a complete invariant pair. Now assuming that $U^*\mathcal{V}_p(V, M) = I_m$, the Newton correction $(\Delta V, \Delta M)$ satisfies

$$\mathcal{M}(\Delta V, \Delta M) = -\big(\mathcal{F}\big(V, M\big), O_{m\times m}\big).$$

The *block Newton method* for computing an invariant pair $(V, M) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ of F is given by the pseudocode in Algorithm 4.13.

Algorithm 4.13: Block Newton method for computing an invariant pair

Choose an initial pair $(\widetilde{V}^{(0)}, \widetilde{M}^{(0)}) \in \mathbb{C}^{n\times m} \times \mathbb{C}^{m\times m}$ with minimality index p.

for $k = 0, 1, \ldots$ until convergence **do**

Compute the compact QR factorization $\mathcal{V}_p(\widetilde{V}^{(k)}, \widetilde{M}^{(k)}\big) = QR$ and let $V^{(k)} = \widetilde{V}^{(k)}R^{-1}$, $M^{(k)} = R\widetilde{M}^{(k)}R^{-1}$.

Solve the linear matrix equation

$$\mathcal{M}(\Delta V, \Delta M) = \big(\mathcal{F}\big(V^{(k)}, M^{(k)}\big), O_{m\times m}\big) \tag{4.28}$$

for $(\Delta V, \Delta M)$.

Update $\widetilde{V}^{(k+1)} = V^{(k)} - \Delta V$ and $\widetilde{M}^{(k+1)} = M^{(k)} - \Delta M$.

end

The initial pair $(\widetilde{V}^{(0)}, \widetilde{M}^{(0)})$ can be constructed with a block variant of inverse iteration (Kressner 2009, Algorithm 2). The solution to the linear matrix equation (4.28) is the most expensive part of the iteration. Kressner (2009) shows how to do that efficiently using ideas from Beyn and Thümmler (2009), and provides a MATLAB function `nlevp_newtonstep` that implements these ideas and returns the pair $\big(\Delta V, \Delta M\big)$.

```
% block_Newton
A(:,:,1) = [0 1; 1 1]; A(:,:,2) = [1 0; 0 0];
n = 2; k = 4; ell = k; maxit = 30; tol = n*eps;
% construct initial pair (V,M)
a = sqrt(2*pi); % eigenvalues are [a, -a, 1i*a, -1i*a]
d = [a -a a*1i -a*1i]+1e-2*(randn(1,k)+1i*randn(1,k)); M = diag(d);
V = diag([1 -1])*ones(n,k) + 1e-2*(randn(n,k)+1i*randn(n,k));
for iter = 0:maxit
  Z = zeros(n*ell,k); Z(1:n,:) = V;
  for j = 2:ell, Z((j-1)*n+1:j*n,:) = Z((j-2)*n+1:(j-1)*n,:)*M; end
  [Q,R] = qr(Z); R = R(1:k,1:k); V = V/R; M = R*(M/R);
  W(:,:,1) = V; for j = 2:ell, W(:,:,j) = W(:,:,j-1)*M; end
  Res = A(:,:,1)*V*f(1,M) + A(:,:,2)*V*f(2,M);
  if norm(Res,'fro') < tol, break, end
  [DV,DM] = nlevp_newtonstep(A,@f,V,M,W,Res,zeros(k));
  V = V - DV; M = M - DM;
end
if k < maxit, nbr_iter = k, evs = eig(M), end

function X = f(j,M)
if j == 1, X = eye(size(M)); end
if j == 2, X = expm(1i*M*M); end
```

Figure 4.14: Basic MATLAB implementation of the block Newton method for problem (2.1) The lines preceding the outer `for` loop specify the problem and the method's parameters. When the iteration converges, it returns the number of iterations and the eigenvalues associated with the computed invariant pair (V, M).

Example 4.15. Let us consider the matrix-valued function F in (2.1),

$$F(z) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} + e^{\mathrm{i}z^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

and aim to compute approximations for the eigenvalues $\pm\sqrt{2\pi}$ and $\pm\mathrm{i}\sqrt{2\pi}$. As initial pair we use a random perturbation of the exact invariant pair

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}, \quad M = \operatorname{diag}(\sqrt{2\pi}, -\sqrt{2\pi}, \mathrm{i}\sqrt{2\pi}, -\mathrm{i}\sqrt{2\pi}).$$

Using the MATLAB code given in Figure 4.14, we consistently obtain all four eigenvalues with a relative error of order 10^{-15} or less. No convergence takes place if we perturb V and M with random perturbations of order 10^{-1}.

We ran the same code but with lines 2–6 replaced by

```
n = 2; k = 6; ell = k; maxit = 100; tol = n*eps;
% construct initial pair (V,M)
a = sqrt(2*pi); % eigenvalues are [a, -a, 1i*a, -1i*a, 0, 0]
```

```
d = [a -a a*1i -a*1i 0 0]+1e-10*(randn(1,k)+1i*randn(1,k));
M = diag(d);
V = diag([1 -1])*ones(n,k)+1e-10*(randn(n,k)+1i*randn(n,k));
```

so as to find an invariant pair containing all the eigenvalues inside the circle $\{z \in \mathbb{C} : |z| = 3\}$. The result varies from run to run due to the random perturbations of V and M, but most often after `maxit` iterations we obtain the simple eigenvalues to about 10 digits and the double eigenvalue $\lambda = 0$ to about 7 digits, which shows that the defective eigenvalue 0 affects the accuracy of the other nondefective eigenvalues. We refer to (Szyld and Xue 2013*b*) for a discussion of the sensitivity of invariant pairs. The iterations fail to converge if we apply larger perturbations to V and M when constructing the initial pair.

4.5. Large sparse NEPs

Most of the methods we described so far require the solution of linear systems. For large sparse matrices, there are efficient direct methods implementing Gaussian elimination with some form of pivoting that make clever use of the sparsity structure to avoid fill-in and save storage. These include HSL (2016), MUMPS (2016), PARDISO (Schenk and Gärtner 2004), and UMFPACK (Davis 2004). In place of direct methods to solve the sparse linear systems in Algorithms 4.7–4.10, we can also use iterative methods, e.g., Krylov subspace methods such as the generalized minimal residual method (GMRES) of Saad and Schultz (1986), the biconjugate gradient method in its stabilized form (BICGSTAB) by van der Vorst (1992), or the quasi-minimal residual method (QMR) of Freund and Nachtigal (1996). Solving the linear systems iteratively leads to inexact versions of nonlinear inverse iteration, residual inverse iteration, and Rayleigh functional iteration. Let us consider in particular the inexact nonlinear inverse iteration, Algorithm 4.7, with the exact solve replaced by an inexact solve. Let $\tau^{(k)}$ be a tolerance such that the approximate solution $\widetilde{v}^{(k+1)}$ returned by the iterative solver at step k satisfies

$$\text{Res} = \|F'\big(\lambda^{(k)}\big)v^{(k)} - F\big(\lambda^{(k)}\big)\widetilde{v}^{(k+1)}\| \le \tau^{(k)}\|F'\big(\lambda^{(k)}\big)v^{(k)}\|,$$

and let

$$e^{(k)} = \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \begin{bmatrix} v \\ \lambda \end{bmatrix}$$

denote the error at iteration k. Szyld and Xue (2013*a*, Theorem 6) show that if at each iteration

$$\tau^{(k)} \le c\, \|e^{(k)}\|$$

is ensured with a constant c independent of k, then the inexact nonlinear inverse iteration converges at least quadratically. In a similar way, they show

that for inexact versions of the residual inverse iteration and the two-sided Rayleigh functional iteration, the same order of convergence as for the exact iterations can be achieved if an appropriate sequence of tolerances is used for the inner solves.

Garrett et al. (2016) propose a variation of Kublanovskaya's Newton-QR method for either large banded NEPs with narrow bandwidths relative to the size n of the NEP, or large sparse NEPs that can be reduced to such banded NEPs. They make use of a special data structure for storing the matrices to keep memory and computational costs low. They replace the rank-revealing QR factorization of $F(\lambda^{(k)})$ in (4.5) by a banded QR factorization followed by a rank-revealing QR factorization of the R factor. MATLAB and C++ implementations of this approach including the deflation strategy (4.11) are publicly available (Garrett and Li 2013).

Example 4.16. Let us consider the `loaded_string` problem defined by (1.3). This NEP has real eigenvalues and we are interested in the five eigenvalues in the interval $[4, 296]$, all of which are given in (3.5). The MATLAB code in Figure 4.17 calls the function `NQR4UB` from Garrett and Li (2013) implementing the Newton-QR method (i.e., Kublanovskaya's method) for unstructurally banded matrix-valued functions. With initial guess `lambda0 = 4.0`, it returns

```
evs =
    4.4822   63.7238  123.0312  202.2009  719.3507
```

which, compared with (3.5), are excellent approximations to the eigenvalues $\lambda_1, \lambda_3, \lambda_4, \lambda_5$, and to another eigenvalue λ_9 outside the interval of interest. Note that the eigenvalue λ_2 is missed. The parameter `h` returned by `NQR4UB` is a cell array containing the values of the residual in (4.13) at each iteration (these are used as a stopping criterion). The left and right normalized residuals,

$$\eta_F(\lambda, v) = \frac{\|F(\lambda)v\|_2}{\|F(\lambda)\|_F \, \|v\|_2} \quad \text{and} \quad \eta_F(\lambda, w^*) = \frac{\|w^*F(\lambda)\|_2}{\|F(\lambda)\|_F \, \|w\|_2}, \tag{4.29}$$

are reported in Figure 4.18. The eigenvalue λ_2 is found by the code in Figure 4.17 if we replace `lambda0 = 4.0` by `lambda0 = 20.0`, but this is the only eigenvalue of F found with this choice of initial guess.

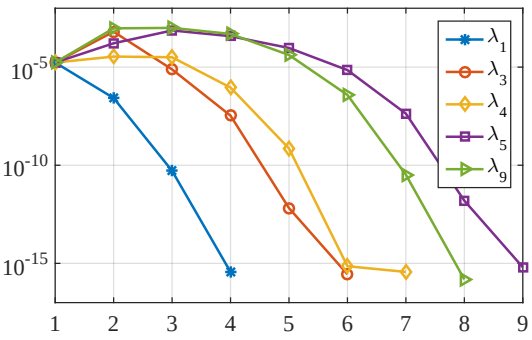
For large sparse NEPs, the iterations described in Sections 4.1–4.2 can be used as part of an iterative projection method in the following way. Suppose we are given a matrix $U \in \mathbb{C}^{n\times k}$ with $k \ll n$ having orthonormal columns that span a subspace (the *search space*) containing an eigenvector of interest. Then instead of solving $F(\lambda)v = 0$ we can solve the $k \times k$ projected NEP

$$Q^*F(\vartheta)Ux = 0, \tag{4.30}$$

```
% Newton_QR_banded
n = 100; lambda0 = 4.0;
ind(1,:) = [1 (1:n-1)]; ind(2,:) = [(2:n) n]; ind(3,:) = ind(2,:);
opts.tol = 5.0e-10; opts.maxitn = 20; opts.rr = 0; opts.nevs = 0;
fun = @(z) f_loaded_string(z,n);
for k = 1:5
    [lam,v,w,h] = NQR4UB(n,fun,ind,lambda0,opts);
    opts.evs(k) = lam; opts.nevs = opts.nevs+1;
end
evs = opts.evs

function [F Fp] = f_loaded_string(z,n)
% Return F(z) and derivative F'(z) for the loaded_string problem.
% Both F(z) and F'(z) are compactly stored as required by NQR4UB.
F(1,:) = [2*n-2*z/3/n (-n-z/6/n)*ones(1,n-1)];
F(2,:) = [-n-z/(6*n)  (2*n-2*z/3/n)*ones(1,n-2) n-z/3/n+z/(z-1)];
F(3,:) = [0 (-n-z/6/n)*ones(1,n-2) 0];
Fp(1,:) = [-2/3/n (-1/6/n)*ones(1,n-1)];
Fp(2,:) = [-1/(6*n)  (-2/3/n)*ones(1,n-2) -1/3/n-1/((z-1)^2)];
Fp(3,:) = [0 (-1/6/n)*ones(1,n-2) 0];
F = sparse(F); Fp = sparse(Fp);
```

Figure 4.17: Basic MATLAB M-file calling the Newton-QR method implemented in the Unstructurally Banded Nonlinear Eigenvalue Software as `NQR4UB`. Lines 2–5 define the input parameters for `NQR4UB`. The M-file requires the function `f_loaded_string`.



i	λ_i	$\eta_F(\lambda_i, v_i)$	$\eta_F(\lambda_i, w_i^*)$
1	4.482	3.5e-17	3.7e-16
3	63.72	3.1e-17	2.1e-16
4	123.0	2.7e-17	9.2e-17
5	202.2	4.9e-17	5.4e-16
9	719.4	3.5e-17	1.8e-16

Figure 4.18: Eigenvalue computation for the `loaded_string` problem using basic MATLAB calls to `NQR4UB` as in Figure 4.17. Left: Residuals $|r_{nn}(\lambda)|/||F(\lambda)||_F$ at each iteration for the found eigenvalues λ. Right: Final residuals as defined in (4.29).

where $Q \in \mathbb{C}^{n\times k}$ is some matrix with orthonormal columns spanning the *test space*. Now let (ϑ, x) be an eigenpair of the projected problem (4.30), selected so that ϑ is as close as possible to the target eigenvalue. The pair (ϑ, v) with $v = Ux$ is the corresponding *Ritz eigenpair* for F and if the residual $\|F(\vartheta)v\|$ is small enough, we can accept (ϑ, v) as an approximate eigenpair for F. If the residual $\|F(\vartheta)v\|$ is too large, then the search space $\operatorname{span}\{U\}$ can be extended by one step of the Newton iteration with initial guess (ϑ, v) and v normalized to have unit 2-norm. The Newton step in (4.14) is rewritten as

$$\left\{\begin{array}{rcl} F(\vartheta)\Delta v &=& -F(\vartheta)v - \Delta\vartheta\, F'(\vartheta)v, \\ v^*\Delta v &=& 0, \end{array}\right. \tag{4.31}$$

where $\Delta\vartheta$ and Δv define the Newton correction (we omit the index k). Since we only need Δv to extend the search space, we premultiply the first equation in (4.31) by the oblique projector $I_n - F'(\vartheta)vq^*$, where $q \in \mathbb{C}^n$ is such that $q^*F(\vartheta)v = 0$ and normalized such that $q^*F'(\vartheta)v = 1$. This yields

$$\left\{\begin{array}{rcl} (I_n - F'(\vartheta)vq^*)F(\vartheta)\Delta v &=& -F(\vartheta)v, \\ v^*\Delta v &=& 0. \end{array}\right. \tag{4.32}$$

Because of the orthogonality condition $v^*\Delta v = 0$, we can rewrite the first equation in (4.32) as

$$(I_n - F'(\vartheta)vq^*)F(\vartheta)(I_n - vv^*)\Delta v = -F(\vartheta)v, \tag{4.33}$$

which has the form of a *Jacobi–Davidson correction equation*. We can use $\operatorname{span}\{U, \Delta v\}$ as the new search space and $\operatorname{span}\{Q, F(\vartheta)v\}$ as the new test space. This process is repeated until we obtain a Ritz eigenpair with a small residual. We can expect quadratic convergence of this process if the correction equation (4.33) is solved exactly. As for linear eigenproblems (Sleijpen and van der Vorst 1996), the correction equation does not need to be solved accurately to preserve convergence.

A pseudocode is presented in Algorithm 4.19. Usually, only a few steps of a preconditioned Krylov subspace method such as GMRES (Saad and Schultz 1986) are necessary to solve the correction equation. If P is a preconditioner for $F(\vartheta)$ so that $Py = r$ is easy to solve, the preconditioner P should be modified to

$$\widetilde{P} = (I_n - F'(\vartheta)vq^*)P(I_n - vv^*)$$

for the correction equation (4.33). Davidson (1975) proposes to use $P = \operatorname{diag}F(\vartheta)$ for linear problems.

Variations of the Jacobi–Davidson approach are proposed in (Sleijpen et al. 1996) for polynomial eigenvalue problems, in (Betcke and Voss 2004) for symmetric NEPs, and in (Voss 2007, Effenberger 2013*b*) for nonsymmetric NEPs. Variants of a two-sided Jacobi–Davidson method are discussed

Algorithm 4.19: Nonlinear two-sided Jacobi–Davidson method

Choose initial bases U_0 and Q_0 such that $U_0^*U_0 = Q_0^*Q_0 = I$.

for $k = 0, 1, \ldots$ until convergence **do**

Compute the eigenpair (ϑ, x) of the projected matrix-valued function $Q_k^*F(z)U_k$ with ϑ closest to the wanted eigenvalue and $\|x\|_2 = 1$.

Compute the Ritz vector $v = U_k x$ and the residual $r = F(\vartheta)v$.

if $\|r\|_2 \leq$ tol, accept (ϑ, v) as approximate eigenpair, **stop**

Approximately solve the correction equation

$$(I_n - F'(\vartheta)vq^*)F(\vartheta)(I_n - vv^*)\Delta v = -r$$

for Δv orthogonal to v.

Orthogonalize Δv against U_k, normalize $\Delta v \leftarrow \frac{\Delta v}{\|\Delta v\|}$ and expand search space $U_{k+1} = [U_k, \Delta v]$.

Orthogonalize r against Q_k, normalize $r \leftarrow \frac{r}{\|r\|}$ and expand test space $Q_{k+1} = [Q_k, r]$.

end

in (Hochstenbach and Sleijpen 2003). The nonlinear Arnoldi method of Voss (2004*a*) corresponds to using one step of residual inverse iteration in place of the Jacobi–Davidson correction equation, i.e., (4.33) is replaced by $F(\sigma)\Delta v = F(\lambda)v$. The linear system is then solved approximately, $\Delta v \approx MF(\lambda)v$, using an approximation M to $F(\sigma)^{-1}$.

4.6. Hermitian NEPs

Assume that the matrix-valued function $F(z)$ is Hermitian, that is, $F(\bar{z}) = F(z)^*$ for all $z \in \mathbb{C}$. We can take advantage of this property in the algorithms described in Sections 4.1–4.2 when factorizing the Hermitian matrix $F\big(\lambda^{(k)}\big)$ using a block LDL* factorization. Also, when λ is real, the solution to the nonlinear scalar equation in (4.18) in the first step of the residual inverse iteration can be replaced by finding z (i.e., the Rayleigh functional) such that $v^{(k)*}F(z)v^{(k)} = 0$. The resulting residual inverse iteration converges quadratically (Neumaier 1985).

Special algorithms can be designed when the Hermitian matrix-valued function F satisfies properties (A1)–(A3) in Section 3 on some real interval $\mathbb{I}$. It follows from Theorem 3.1 that the eigenvalues of F can be characterized as minmax and maxmin values of the Rayleigh functional $p(x)$, which is the only root of $x^*F(p(x))x = 0$ in $\mathbb{I}$. Indeed, Theorem 3.1 asserts that if λ_k is a kth eigenvalue of $F(z)$ (i.e., $\mu = 0$ is the kth largest eigenvalue of the

Hermitian matrix $F(\lambda_k)$), then

$$\lambda_k = \min_{\substack{V\in\mathbb{S}_k \\ V\cap\mathbb{K}(p)\neq\emptyset}} \max_{\substack{x\in V\cap\mathbb{K}(p) \\ x\neq 0}} p(x) \in \mathbb{I}, \tag{4.34}$$

where $\mathbb{S}_j$ denotes the set of all subspaces of $\mathbb{C}^n$ of dimension j, and $\mathbb{K}_p$ is a subspace of $\mathbb{C}^n$ onto which the Rayleigh functional p is defined. The minimum in (4.34) is attained for an invariant subspace of the Hermitian matrix $F(\lambda_k)$ corresponding to its k largest eigenvalues and the maximum is attained for some $x \in \operatorname{null}\big(F(\lambda_k)\big)$. This suggests the method in Algorithm 4.20 for computing the jth eigenvalue of F, called *safeguarded iteration* (Werner 1970).

Algorithm 4.20: Safeguarded iteration for the jth eigenvalue of F

Choose an initial approximation $\lambda^{(0)}$ to the jth eigenvalue of F.

for $k = 0, 1, \ldots$ until convergence **do**

Compute an eigenvector $x^{(k)}$ corresponding to the jth largest eigenvalue of the Hermitian matrix $F\big(\lambda^{(k)}\big)$.

Compute the real root ρ of $x^{(k)*}F(\rho)x^{(k)} = 0$ closest to $\lambda^{(k)}$ and set $\lambda^{(k+1)} = \rho$.

end

Niendorf and Voss (2010) show that if λ_j is a simple eigenvalue of F, then the safeguarded iteration converges locally and quadratically to λ_j. For large sparse problems, the safeguarded iteration can be embedded into an iterative projection method such as the Jacobi–Davidson scheme or the nonlinear Arnoldi scheme described in Section 4.5; see (Voss 2004*b*, Niendorf and Voss 2010).

For large sparse NEPs satisfying properties (A1)–(A3) defined in Section 3, Szyld and Xue (2015) describe several variants of a preconditioned conjugate gradient method, which make use of the nonlinear variational principle (Theorem 3.1) and the nonlinear Cauchy interlacing theorem (Theorem 3.2).

```
% safeguarded_iteration
n = 100; C1 = n*gallery('tridiag',n); C1(end) = C1(end)/2;
C2 = (abs(gallery('tridiag',n)) + 2*speye(n))/(6*n);
C2(end) = C2(end)/2; C3 = sparse(n,n); C3(n,n) = 1;
F = @(z) C1 - z*C2 + C3*z/(z-1);
lam = 1.1; maxit = 5; nb_evs = 5; tol = n*eps;
fprintf('eigenvalue  #iter   residual\n')
for j = 1:nb_evs
    for k = 0:maxit
        [X,E] = eigs(F(lam),nb_evs+0,'sa'); v = X(:,j);
        res = norm(F(lam)*v)/norm(F(lam),'fro');
        if res < tol, break, end
        c1 = v'*C1*v; c2 = v'*C2*v; c3 = v'*C3*v;
        f = @(z) c1-c2*z+c3*z/(z-1); lam = fzero(f,lam);
    end
    fprintf('%9.2e %5.0f %12.2e\n',lam,k,res)
end
```

Figure 4.21: Basic MATLAB implementation of the safeguarded iteration to compute the five smallest eigenvalues of the `loaded_string` problem (1.3). Lines 2–5 define the NEP F and line 6 specifies the parameters for the iteration.

Example 4.22. As shown in Example 3.3, the `loaded_string` NEP defined with F in (1.3) satisfies assumptions (A1)–(A3) in Section 3 on the open real interval $(1, +\infty)$. Hence we can use the safeguarded iteration to compute the five smallest eigenvalues of F in that interval. A basic MATLAB implementation is given in Figure 4.21. The inner loop implements the safeguarded iteration. Once an eigenpair is found, we increase the index `j` of the eigenvalue and restart the safeguarded iteration with the previous converged eigenvalue as initial guess. Using this code we obtain

```
eigenvalue  #iter   residual
 4.48e+00     4     7.43e-17
 2.42e+01     2     1.02e-16
 6.37e+01     2     8.28e-17
 1.23e+02     2     8.44e-17
 2.02e+02     2     9.80e-17
```

Note that only a small number of iterations per eigenvalue are required for this problem.

4.7. Additional comments and software

While Newton's method is widely used, e.g., as a final refinement step in the Chebyshev interpolation-based method of Effenberger and Kressner (2012) (MATLAB code available at `http://anchp.epfl.ch/files/content/sites/anchp/files/software/chebapprox.tar.gz`), there appears to be very little software beyond mere research code available. Gander, Gander and Kwok (2014, Section 8.3.4) provide MATLAB codes for Newton's method to compute the eigenvalues of a quadratic eigenvalue problem. They show how to use algorithmic differentiation of $f(z) = \det F(z)$ to obtain the Newton update. Indeed, algorithmic differentiation is a powerful tool to compute derivative information when these are not available explicitly (Arbenz and Gander 1986, Griewank and Walther 2008).

A MATLAB implementation of a nonlinear Jacobi–Davidson algorithm with deflation developed in (Effenberger 2013*b*) is available at `http://anchp.epfl.ch/nonlinjd`. As stated by the author, this is research code and not intended for production use.

The most comprehensive suite of high-performance solvers for large-scale linear and nonlinear eigenvalue problems is SLEPc (Hernandez, Roman and Vidal 2005), available at `http://slepc.upv.es/`. SLEPc builds on top of PETSc (Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Eijkhout, Gropp, Kaushik, Knepley, McInnes, Rupp, Smith, Zampini, Zhang and Zhang 2016), which means that PETSc must be installed first in order to use SLEPc. SLEPc provides several Newton-based solvers for NEPs, namely the residual inverse iteration of Neumaier (1985) (Algorithm 4.10), the method of successive linear problems by Ruhe (1973), and the nonlinear Arnoldi method of Voss (2004*a*). All of these methods are implemented without deflation, i.e., for computing a single eigenpair near an initial guess. However, SLEPc implements Effenberger's method in Section 4.3 for the *iterative refinement* of approximations computed by some other method. The implementation is described in (Campos and Roman 2016*a*) for polynomial eigenvalue problems, but it can be used for NEPs as well.

5. Solvers based on contour integration

Keldysh's theorem (Theorem 2.8) provides the main tool for a class of NEP solvers based on contour integration. These methods have been developed in a series of works by Asakura, Sakurai, Tadano, Ikegami and Kimura (2009), Beyn (2012), and Yokota and Sakurai (2013), and they have become popular due to their relatively straightforward implementation and great potential for parallelization.

Let $F \in H(\Omega, \mathbb{C}^{n\times n})$ be a regular matrix-valued function with finitely many eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$ in the domain Ω. Further, let $\Gamma \subset \Omega$ be a contour which encloses all eigenvalues and let $f \in H(\Omega, \mathbb{C})$ be a scalar holo-

morphic function. Then by Theorem 2.8 and the Cauchy-integral definition of a matrix function (Higham 2008, Definition 1.11) we have

$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma f(z) F(z)^{-1} \, \mathrm{d}z = V f(J) W^*. \tag{5.1}$$

Here, following Theorem 2.8, V and W are $n \times \overline{m}$ matrices whose columns are right and left generalized eigenvectors, respectively, and J is an $\overline{m} \times \overline{m}$ block-diagonal Jordan matrix. The idea common to many contour-based methods is to "probe" the matrix decomposition (5.1) from the right (and left) to infer information about J and V (and W).

5.1. Beyn's "Integral algorithm 1"

Let us start with the simple case in which there are only a small number $\overline{m} \leq n$ of generalized eigenvectors and they are all linearly independent, i.e., both V and W are of full column rank $\overline{m}$. In this case it suffices to use polynomials f of degree zero and one in (5.1), also called the zeroth and first-order *moments*. For simplicity we choose the monomials 1 and z for f, but the derivation extends to any choice of two linearly independent polynomials of degree at most one.

Let $R \in \mathbb{C}^{n \times r}$ ($\overline{m} \leq r \leq n$) be a probing matrix (typically chosen at random) such that W^*R is of maximal rank $\overline{m}$. Then the pair of $n \times r$ matrices

$$A_0 = \frac{1}{2\pi \mathrm{i}} \int_\Gamma F(z)^{-1} R \, \mathrm{d}z = V W^* R, \tag{5.2}$$

$$A_1 = \frac{1}{2\pi \mathrm{i}} \int_\Gamma z F(z)^{-1} R \, \mathrm{d}z = V J W^* R \tag{5.3}$$

can be manipulated to infer J and V as follows:

1. Compute an economy-size singular value decomposition (SVD) of
$$A_0 = V_0 \Sigma_0 W_0^*, \tag{5.4}$$
where $V_0 \in \mathbb{C}^{n \times \overline{m}}$ and $W_0 \in \mathbb{C}^{r \times \overline{m}}$ have orthonormal columns, and $\Sigma_0 = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\overline{m}})$ is invertible. This SVD exists because $\mathrm{rank}(A_0) = \overline{m}$ by the above rank conditions on V, W, and R.
2. Since $\mathrm{range}(V) = \mathrm{range}(V_0)$, there exists an invertible matrix $X \in \mathbb{C}^{\overline{m} \times \overline{m}}$ such that $V = V_0 X$. From (5.2) and (5.4) we find
$$W^* R = X^{-1} \Sigma_0 W_0^*.$$
This relation can be used to eliminate W^*R from $A_1 = V J W^* R$,
$$A_1 = (V_0 X) J X^{-1} \Sigma_0 W_0^*.$$
3. We have thus arrived at
$$M := V_0^* A_1 W_0 \Sigma_0^{-1} = X J X^{-1},$$

showing that XJX^{-1} is a Jordan decomposition form of the matrix M. Hence the eigenvalues of the NEP can be obtained from the eigenvalues of M, and the columns of $V = V_0X$ contain the corresponding right generalized eigenvectors of the NEP.

By this simple three-step procedure we have effectively reduced an NEP with $\overline{m}$ eigenvalues inside Γ to a linear eigenproblem of size $\overline{m} \times \overline{m}$. By construction, this procedure returns all the eigenvalues λ_i of F inside Γ, together with corresponding right generalized eigenvectors v_i which we assumed as linearly independent; see also (Beyn 2012, Theorem 3.3). However, we need a number of further ingredients to make this method practical.

First of all, we should be aware that this method effectively amounts to the computation of a Jordan decomposition of the matrix M, which is known to be highly sensitive to perturbations. Indeed, (Beyn 2012) first derives his method for the case that all eigenvalues inside Γ are simple. In this case, and even in the semisimple case, M will be diagonalizable and a full set of linearly independent eigenvectors of M can be computed reliably by the QR algorithm.

Second, we typically do not know the number $\overline{m}$ of linearly independent eigenvectors in advance. However, as long as r is chosen large enough (that is, $r \geq \overline{m}$), we can detect $\overline{m}$ as a *numerical rank of* A_0, e.g., by counting the number of singular values of A_0 that are above a user-defined tolerance `tol`.

The third ingredient, the numerical approximation of the contour integrals in (5.2) and (5.3) by appropriate quadrature rules, will be discussed in the following section.

5.2. Quadrature rules and rational filters

Let us assume that Γ is a piecewise regular contour with parameterization $\gamma : [0, 2\pi] \to \Gamma$. Then by substituting $z = \gamma(t)$ in (5.1) we have

$$A_f := \frac{1}{2\pi \mathrm{i}} \int_\Gamma f(z)F(z)^{-1}R \,\mathrm{d}z = \frac{1}{2\pi \mathrm{i}} \int_0^{2\pi} f(\gamma(t))\gamma'(t)F(\gamma(t))^{-1}R \,\mathrm{d}t. \quad (5.5)$$

The next step is to apply a quadrature rule with, say, $n_c \geq 1$ points to the integral on the right.

To first consider the simplest case, assume that $\gamma(t) = e^{\mathrm{i}t}$ parameterizes the unit circle $\Gamma = \{z \in \mathbb{C} : |z| = 1\}$. Then the most natural quadrature rule, the *trapezoid rule*, amounts to using equispaced points $t_\ell = 2\pi\ell/n_c \in [0, 2\pi]$ for $\ell = 1, 2, \ldots, n_c$, resulting in the approximation

$$A_{f,n_c} := \sum_{\ell=1}^{n_c} \omega_\ell F(z_\ell)^{-1}R \approx A_f, \quad (5.6)$$

with quadrature weights and nodes

$$\omega_\ell = \frac{f(z_\ell)z_\ell}{n_c}, \quad z_\ell = e^{2\pi \mathrm{i}\ell/n_c}.$$

Note that only the weights ω_ℓ depend on f, hence the quadrature of both A_0 and A_1 in (5.2)–(5.3) requires only n_c evaluations of $F(z_\ell)^{-1}R$, not $2n_c$. A further reduction in the number of evaluations is possible if the $F(z_\ell)^{-1}R$ in (5.6) appear in complex conjugate pairs, as is the case when F is a Hermitian matrix-valued function and R is a matrix with real entries. The evaluations of $F(z_\ell)^{-1}R$ are typically the computationally most expensive part of a contour-based eigensolver, but they are completely decoupled and can hence be assigned to different processing units on a parallel computer. This potential for coarse-grain parallelism is one reason for the popularity of these methods.

Beyn (2012) studies the quality of the quadrature approximation (5.6). Assume that each element of $f(z)F^{-1}$ is holomorphic and bounded on an annulus of modulus ρ,

$$\mathbb{A}_\rho = \left\{ z \in \mathbb{C} : \rho^{-1} < |z| < \rho \right\}, \quad \rho > 1.$$

Then each element of $f(z)F(z)^{-1}R$ is a bounded holomorphic function on that same annulus and we can apply standard results on the convergence of the trapezoidal rule for holomorphic functions elementwise; see, e.g., (Trefethen and Weideman 2014, Theorem 2.2). This results in

$$\|A_f - A_{f,n_c}\| = O(\rho^{-n_c}),$$

i.e., the trapezoid rule yields exponential convergence in the number of quadrature nodes. The convergence factor ρ^{-1} is determined by a singularity of $F(z)^{-1}$ (an eigenvalue of F) closest to the unit circle (where "closeness" is measured in terms of the level lines $\partial\mathbb{A}_\rho$).

Care has to be taken with contours that are not circles, as the exponential accuracy of the trapezoidal rule may deteriorate or be completely lost if the rule is not transplanted conformally. As an example, consider the square contour

$$\Gamma := \left\{ z \in \mathbb{C} : \max\{|\mathrm{Re}(z)|, |\mathrm{Im}(z)|\} = 1 \right\}.$$

It seems tempting to simply apply the trapzoidal rule on each of the four sides of the square separately and, assuming that n_c is divisible by four, discretizing each side by $n_c/4 + 1$ equispaced quadrature nodes including the endpoints. A schematic view is given in Figure 5.1 on the left, with the red pluses corresponding to the trapezoid quadrature nodes on the square. As the function to be integrated, $(2\pi\mathrm{i})^{-1}f(z)F(z)^{-1}$, is generally not periodic on each of the four sides, this approach only gives convergence of order 2, $\|A_f - A_{f,n_c}\| = O(n_c^{-2})$, in line with the standard error bounds

for the trapezoidal rule for nonperiodic functions (with continuous second derivative); see, e.g., (Davis and Rabinowitz 2007, equation (2.1.11)).

A better approach to obtain a quadrature rule on a non-circular contour Γ is to use a conformal map Ψ from the annulus $\mathbb{A}_\rho$ (the w-domain) onto a doubly connected domain Ω with continuous inner and an outer boundary (the z-domain), and then to define the contour Γ as the image of the unit circle under Ψ, $\Gamma = \{z = \Psi(w) : |w| = 1\}$. (Conversely, by the Riemann mapping theorem (Ahlfors 1953, p. 255), any bounded domain Ω whose boundary consists of two disjoint continua can be identified conformally with an annulus.) Let $\gamma(t) = \Psi(e^{\mathrm{i}t})$, $t \in [0, 2\pi]$, be a parameterization of Γ. Then by applying the trapezoid rule to (5.5), we obtain the quadrature approximation

$$A_{f,n_c} := \sum_{\ell=1}^{n_c} \omega_\ell F(z_\ell)^{-1} R \approx A_f,$$

with quadrature weights and nodes

$$\omega_\ell = \frac{f(z_\ell)\Psi'(e^{2\pi\mathrm{i}\ell/n_c})e^{2\pi\mathrm{i}\ell/n_c}}{n_c}, \quad z_\ell = \Psi(e^{2\pi\mathrm{i}\ell/n_c}).$$

We can interpret $F(z)$ as a function $F(\Psi(w))$ which is holomorphic and bounded on the annulus $\mathbb{A}_\rho$, hence this quadrature rule will again converge exponentially with convergence factor ρ^{-1}. This is illustrated in Figure 5.1.

For comparison we show the quadrature error obtained when applying Gauss–Legendre quadrature with $n_c/4$ nodes on each of the four sides of the square. Such an approach will also lead to exponential convergence, with the convergence factor determined by the largest Bernstein ellipse enclosing each of the four sides such that $F(z)^{-1}$ is holomorphic in its interior. More precisely, the *Bernstein ellipse region of elliptical radius* $\rho > 1$ *associated with the interval* $[-1, 1]$ is

$$\mathbb{E}_\rho = \{z = \cos(t + \mathrm{i}\ln r) : t \in [0, 2\pi],\ r \in [1, \rho)\}, \tag{5.7}$$

and the corresponding Bernstein ellipse region associated with a (possibly complex) interval $[a, b]$ is the image $\varphi_{[a,b]}(\mathbb{E}_\rho)$ with $\varphi_{[a,b]}(z) = (a - b)z/2 + (a + b)/2$. The boundary $\partial\mathbb{E}_\rho$ is referred to as the *Bernstein ellipse*. For our example in Figure 5.1, the most restricted Bernstein ellipse is the one of radius $\rho = 1.105$ associated with the upper side $[a, b] = [-1 + \mathrm{i}, 1 + \mathrm{i}]$ of the square. The Gauss–Legendre rule associated with this side has $n_c/4$ nodes, hence the quadrature error reduces with a converge factor of $\rho^{-2n_c/4}$. The overall quadrature error behaves like $\|A_f - A_{f,n}\| = O(\rho^{-n_c/2})$, which in this example is worse than the convergence achieved with the trapezoid rule.

Conformally mapped quadrature rules have been applied, e.g., to the problem of matrix function approximation (Hale, Higham and Trefethen 2008). An alternative interpretation of quadrature rules in terms of filter functions

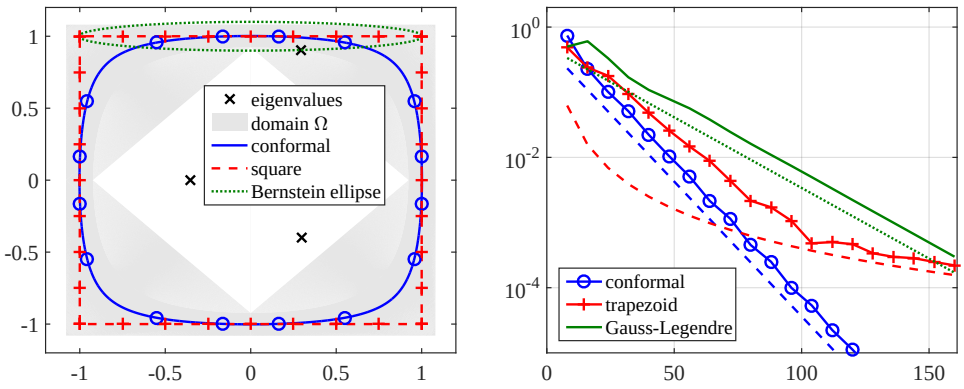


Figure 5.1: Comparison of three quadrature rules for approximating the integral in (5.1). Left: A portion of the complex plane, showing three eigenvalues of a 3×3 NEP $F(\lambda)v = 0$ (black $\times$). The doubly-connected region Ω (shown in grey) is the conformal image of an annulus with modulus $\rho = 1.1$, and the blue contour is the image of the unit circle with mapped quadrature nodes shown as blue circles. The red crosses on the square contour correspond to the quadrature nodes of trapezoidal rules applied to each side. The green curve is the largest Bernstein ellipse associated with the upper side of the square in which $F(z)^{-1}$ is analytic. Right: Quadrature errors $\|A_f - A_{f,n_c}\|$ as the number of quadrature nodes n_c increases (solid curves), together with their predicted error behaviors as discussed in Section 5.2.

has been discussed by Van Barel and Kravanja (2016). This interpretation also motivated the numerical approach to designing a rational filter for linear and nonlinear eigenvalue problems presented in (Van Barel 2016).

5.3. Higher-order moments and the Sakurai–Sugiura method

Let us now consider the general case where the number $\overline{m}$ of generalized eigenvectors is not necessarily smaller than or equal to the problem size n, and that the generalized eigenvectors are not necessarily linearly independent. In this case we need to use higher-order moments in (5.1), i.e., employ polynomials f of degree larger than one. Eigenvalue techniques based on higher-order moments have been applied successfully by Sakurai and coauthors in a number of papers on the generalized eigenvalue problem, and for the nonlinear case by Asakura et al. (2009) and Yokota and Sakurai (2013). Beyn (2012) also considers higher-order moments in his "Integral algorithm 2". In the following we will use a general formulation of these methods and then discuss how the methods available in the literature relate to this formulation.

Let $L \in \mathbb{C}^{n\times\ell}$ and $R \in \mathbb{C}^{n\times r}$ be given left and right probing matrices, respectively, and let $\overline{p} \geq 0$ be a given integer. For any $p = 0, 1, \ldots, \overline{p}$ we

define

$$A_p = \frac{1}{2\pi \mathrm{i}} \int_\Gamma z^p L^* F(z)^{-1} R \mathrm{d}z = L^* V J^p W^* R \in \mathbb{C}^{\ell \times r} \tag{5.8}$$

and arrange these matrices in $\overline{p}\ell \times \overline{p}r$ block-Hankel matrices as follows:

$$B_0 = \begin{bmatrix} A_0 & \cdots & A_{\overline{p}-1} \\ \vdots & & \vdots \\ A_{\overline{p}-1} & \cdots & A_{2\overline{p}-2} \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} A_1 & \cdots & A_{\overline{p}} \\ \vdots & & \vdots \\ A_{\overline{p}} & \cdots & A_{2\overline{p}-1} \end{bmatrix}. \tag{5.9}$$

Further, let us define the matrices

$$V_{[\overline{p}]} = \begin{bmatrix} L^*V \\ \vdots \\ L^*VJ^{\overline{p}-1} \end{bmatrix} \in \mathbb{C}^{\overline{p}\ell \times \overline{m}} \quad \text{and} \quad W^*_{[\overline{p}]} = \begin{bmatrix} W^*R, \ldots, J^{\overline{p}-1}W^*R \end{bmatrix} \in \mathbb{C}^{\overline{m} \times \overline{p}r}. \tag{5.10}$$

Then by (5.8) we have factorizations

$$B_0 = V_{[\overline{p}]} W^*_{[\overline{p}]} \quad \text{and} \quad B_1 = V_{[\overline{p}]} J W^*_{[\overline{p}]}.$$

Let us assume that L, R and $\overline{p}$ have been chosen so that $V_{[\overline{p}]}$ and $W^*_{[\overline{p}]}$ are of maximal rank, i.e.,

$$\operatorname{rank}\big(V_{[\overline{p}]}\big) = \operatorname{rank}\big(W^*_{[\overline{p}]}\big) = \overline{m}, \tag{5.11}$$

then using the same derivation as in the three-step procedure given in Section 5.1 it is again possible to infer J from the pair (B_0, B_1). (A sufficient condition for (5.11) is given by Lemma 2.13, namely $\overline{p} \geq \sum_{i=1}^{s} m_{i,1}$.) The following theorem, given by Beyn (2012, Theorem 5.2) for the case $L = I_n$, makes this precise.

Theorem 5.2. Suppose that $F \in H(\Omega, \mathbb{C}^{n\times n})$ has no eigenvalues on the contour $\Gamma \subset \Omega$ and a finite number of pairwise distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$ inside Γ. For each eigenvalue λ_i let d_i denote its geometric multiplicity with partial multiplicities $m_{i,1} \geq \cdots \geq m_{i,d_s}$. Further assume that the matrices $V_{[\overline{p}]}$ and $W^*_{[\overline{p}]}$ defined in (5.10) satisfy the rank condition (5.11). Let $V_{[\overline{p}]} W^*_{[\overline{p}]} = V_0 \Sigma_0 W_0^*$ be an economy-size SVD, where $V_0 \in \mathbb{C}^{\overline{p}\ell \times \overline{m}}$ and $W_0 \in \mathbb{C}^{\overline{p}r \times \overline{m}}$ have orthonormal columns and $\Sigma_0 = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\overline{m}})$ is invertible. Then the matrix $X = V_0^* V_{[\overline{p}]}$ is nonsingular and the matrix

$$M := V_0^* B_1 W_0 \Sigma_0^{-1} = XJX^{-1},$$

has a Jordan normal form J and hence the same eigenvalues λ_i with identical partial multiplicities $m_{i,j}$ as F.

In conjunction with numerical quadrature to approximate the matrices A_p defined (5.8) the above theorem translates directly into a numerical method;

see Figure 5.3 for a basic MATLAB implementation. Let us discuss some special cases of this method:

(a) If $\ell = r$, the numerical method indicated by Theorem 5.2 corresponds to the "block-SS method" described by Asakura et al. (2009). This method has been used successfully in applications, e.g., for solving NEPs arising in acoustics (Leblanc and Lavie 2013).
(b) If $\ell = r = 1$ and $F(\lambda) = A - \lambda B$ with an $n \times n$ matrix pencil (A, B), we recover the method of Sakurai and Sugiura (2003) for generalized eigenvalue problems, which is sometimes referred to as the "SS-method". In this case the block matrices B_0, B_1 defined in (5.9) reduce to standard square Hankel matrices of size $(2\overline{p}-1) \times (2\overline{p}-1)$.
(c) If $\overline{p} = 1$, $\overline{m} \leq r \leq n$, and $L = I_n$, the method reduces to the "Integral algorithm 1" of Beyn (2012) described in Section 5.1. This method has been applied successfully, e.g., for the solution of NEPs arising from resonance problems related to fluid-solid interaction (Kimeswenger, Steinbach and Unger 2014) and Maxwell eigenvalue problems (Wieners and Xin 2013). The latter reference also contains numerical comparisons of the integral approach with a Newton method.
(d) For $\overline{p} \geq 1$ and $L = I_n$ we obtain the "Integral algorithm 2" of Beyn (2012).
(e) Another, but closely related projection method has been presented by Yokota and Sakurai (2013). Therein the idea is to not use the matrix M defined in Theorem 5.2 to extract eigenvalues and eigenvectors, but to use the matrices A_p defined in (5.8) with $L = I_n$ to compute an orthonormal basis V_0 of $\mathrm{span}(V)$ from an economy-size SVD of $N := [A_0, A_1, \ldots, A_{\overline{p}-1}] = V_0 \Sigma_0 W_0$. Note that by (5.8) we have

$$N = V[J^0 W^* R, J^1 W^* R, \ldots, J^{\overline{p}-1} W^* R]$$

and hence $\mathrm{span}(N) = \mathrm{span}(V)$ is guaranteed by (5.11). The method of Yokota and Sakurai (2013) then amounts to solving the projected NEP $\widehat{F}(\lambda)\widehat{x} = 0$ with $\widehat{F}(\lambda) = V_0^* F(\lambda) V_0$. The authors recommend using one of Beyn's integral methods for the solution of this lower-dimensional NEP. The reported results indicate that this explicit projection approach may give better accuracy than the methods of Asakura et al. (2009) and Beyn (2012) directly applied to the original problem.

5.4. Numerical illustration

Let us consider the matrix-valued function F defined in (2.1). We choose L and R to be random vectors ($\ell = r = 1$) and look for the six eigenvalues of F inside the circle $\Gamma = \{z \in \mathbb{C} : |z| = 3\}$. In line with Lemma 2.13 we choose $\overline{p} = 6$. A MATLAB code is given in Figure 5.3. For numerical

```
% contour_eigensolver
n = 2; F = @(z) [exp(1i*z.^2), 1; 1, 1];
ell = 1; r = 1; L = rand(n,ell); R = rand(n,r); % probing matrices
gam = 0; rad = 3; nc = 200; % circle center & radius, nbr of nodes
w = exp(2i*pi*(1:nc)/nc); z = gam+rad*w; % unit roots and quad pts
pbar = 6; A = zeros(ell,r,2*pbar);       % matrices of moments
for k = 1:nc
    Fz = L'*(F(z(k))\R);
    for j = 0:2*pbar-1
        A(:,:,j+1) = A(:,:,j+1) + (w(k)^j*rad*w(k)/nc)*Fz;
    end
end
A = A(:,:); B0 = zeros(pbar*ell,pbar*r); B1 = B0;
for j = 0:pbar-1
    B0(1+j*ell:(j+1)*ell,:) = A(:,1+j*r:pbar*r+j*r);
    B1(1+j*ell:(j+1)*ell,:) = A(:,1+(j+1)*r:pbar*r+(j+1)*r);
end
[V,Sig,W] = svd(B0); mbar = find(diag(Sig)/Sig(1)>1e-12,1,'last');
V0 = V(:,1:mbar); Sig0 = Sig(1:mbar,1:mbar); W0 = W(:,1:mbar);
M = (V0'*B1*W0)/Sig0; evs = eig(M); evs = gam+rad*evs(abs(evs)<1)
```

Figure 5.3: Basic MATLAB implementation of an integral method for NEPs with a circular contour. The method's parameters are specified in lines 2–4, and the variable `evs` computed in the final line contains the eigenvalue approximations.

stability we have scaled the moment functions z^p to have unit norm on Γ. Using a trapezoidal rule with $n_c = 200$ nodes, we obtain the eigenvalue approximations

```
evs =
 -2.506628274630944 - 0.000000000000013i
 -0.000000000000028 + 2.506628274631054i
  0.000000000000045 - 2.506628274631075i
  2.506628274630940 - 0.000000000000047i
 -0.000000202602049 + 0.000001229470211i
  0.000000202602004 - 0.000001229470420i
```

The correct digits are underlined. The result will vary slightly from run to run if another random initialization of the probing matrices L and R is used, however, we observe consistently that the simple eigenvalues are accurate to about 12 digits, whereas the double eigenvalue $\lambda = 0$ is only accurate to about 6 digits. This accuracy reduction is not surprising given the higher sensitivity of $\lambda = 0$; see in particular the spectral portrait in Figure 2.23.

5.5. Eigenvalue localization

Theorem 5.2 effectively provides a method for counting the number of eigenvalues $\overline{m}$ of F inside the contour Γ by computing the (numerical) rank of the matrix B_0 defined in (5.9). In the MATLAB code in Figure 5.3, $\overline{m}$ is provided in the variable `mbar`. A more direct approach for counting the eigenvalues follows from the argument principle (Ahlfors 1953, Section 5.2), by which the number $N_0(f,\Gamma)$ defined by

$$N_p(f,\Gamma) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma z^p \frac{f'(z)}{f(z)}\,\mathrm{d}z \tag{5.12}$$

corresponds to the number of roots (counting multiplicity) of a holomorphic function f enclosed by Γ. By choosing $f(z) = \mathrm{det}F(z)$ and using the trace relation (4.2) we have

$$N_0(\mathrm{det}F,\Gamma) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \mathrm{trace}\big(F^{-1}(z)F'(z)\big)\,\mathrm{d}z. \tag{5.13}$$

We can apply the same quadrature techniques as explained in Section 5.2 to approximate this integral. The domain Ω on which to count the eigenvalues of F need not be simply connected. For example, if Γ_1 is a contour inside another contour Γ_2, then F has

$$N_0(\mathrm{det}F,\Gamma_2) - N_0(\mathrm{det}F,\Gamma_1)$$

eigenvalues in the doubly connected region bounded by Γ_1 and Γ_2.

Example 5.4. Let us determine the number of eigenvalues of F defined in (2.1) inside the annulus Ω bounded by the contours $\Gamma_j = \{z \in \mathbb{C} : |z| = \rho_j\}$, $j = 1, 2$, where $\rho_1 = 4$ and $\rho_2 = 5.25$. For this we apply the trapezoidal rule to the integral (5.13). This is implemented in the basic MATLAB code in Figure 5.5 with the parameters set up to compute an approximation to $N_0(\mathrm{det}F,\Gamma_1)$ using $n_c = 90$ quadrature nodes. The code returns

```
nbr_evs =
   9.999998431959657 - 0.000000000000001i
```

suggesting so that $N_0(\mathrm{det}F,\Gamma_1) = 10$. Then replacing `rad = 4` with `rad = 5.25` produces

```
nbr_evs =
  18.000958027557175 - 0.000000000000001i
```

suggesting that $N_0(\mathrm{det}F,\Gamma_2) = 18$. Indeed, F has exactly $18 - 10 = 8$ eigenvalues in Ω.

```
% contour_count
F = @(z) [exp(1i*z.^2) 1; 1 1];
Fp = @(z) [2i*z*exp(1i*z.^2) 0; 0 0];
gam = 0; rad = 4; nc = 90; % center, radius, nr of nodes on circle
w = exp(2i*pi*(1:nc)/nc); z = gam+rad*w; % unit roots and quad pts
nr = 0;
for k = 1:nc
    nr = nr + z(k)*trace(F(z(k))\Fp(z(k)));
end
nbr_evs = nr/nc % number of eigenvalues
```

Figure 5.5: Basic MATLAB code implementing the trapezoidal rule approximation of (5.13) for computing the number of eigenvalues of F inside the disk with center `gam` and radius `rad`. The NEP parameters are specified in lines 2–3.

5.6. Further remarks and available software

Contour-based methods for linear and nonlinear eigenproblems are closely related to earlier techniques developed for finding roots of holomorphic functions f; see the review by Ioakimidis (1987). The pioneering work by Delves and Lyness (1967) uses the contour integrals (5.12) for $p \geq 0$, which can be related to Newton sums of the unknown roots of f enclosed by Γ. However, this relation is prone to ill conditioning, in particular when the number of unknown roots surrounded by Γ is large. In order to address this issue, Delves and Lyness (1967) propose a subdivision technique. A FORTRAN 77 implementation of their method is provided in (Botten, Craig and McPhedran 1983).

Kravanja, Sakurai and Van Barel (1999*a*) present an alternative root-finding approach with better numerical stability based on formal orthogonal polynomials and the solution of a generalized eigenvalue problem. A Fortran 90 implementation is given in (Kravanja, Van Barel, Ragos, Vrahatis and Zafiropoulos 2000). Similar ideas can be used to compute roots and poles of a meromorphic function f inside Γ (Kravanja, Van Barel and Haegemans 1999*b*); see also (Austin, Kravanja and Trefethen 2014). The latter paper discusses connections of contour integration with (rational) interpolation and provides several MATLAB code snippets. An NEP solver based on rational interpolation and resolvent sampling is presented and applied in (Xiao, Zhang, Huang and Sakurai 2016*a*, Xiao, Zhou, Zhang and Xu 2016*b*).

The function f in (5.1) can be interpreted as a filter acting on the eigenvalues of F; see, e.g., (Van Barel and Kravanja 2016). While we have chosen monomials z^p in (5.8) (for notational simplicity) and scaled-and-shifted monomials $\left(\frac{z-\sigma}{\rho}\right)^p$ in Figure 5.3 (for numerical stability), one is free to use other sets of linearly independent filter functions f which are in a linear

relation to each other. The linearity is required in order to derive a factorization of the form (5.10) from which J can be inferred. Interestingly, if F is a meromorphic function with a finite number of poles in Ω, represented as $F(z) = \frac{G(z)}{q(z)}$ with $G \in H(\Omega, \mathbb{C}^{n\times n})$ and some polynomial q, then $F(z)^{-1} = q(z)G(z)^{-1}$ and (5.1) can be rewritten as

$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma f(z)F(z)^{-1}\,\mathrm{d}z = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \widetilde{f}(z)G(z)^{-1}\,\mathrm{d}z$$

with $\widetilde{f} = fq$. Therefore any contour-based method applied to the meromorphic function F with filter functions f is equivalent to applying the same method to the holomorphic function G with modified filters $\widetilde{f}$. This means that contour-based methods should be able to handle meromorphic NEPs without modification.[1] Indeed, the reader can easily test numerically that the algorithm in Figure 5.3 also works fine if the second line is replaced with the definition of the `loaded_string` problem (using lines 2–5 in Figure 4.21). This is a meromorphic (in fact, rational) NEP but the contour solver in Figure 5.3 still computes good eigenvalue approximations even when the contour Γ contains the pole $z = 1$ in its interior.

A high-performance implementation of a contour-based eigensolver is the Contour Integral Spectrum Slicing (CISS) method in SLEPc (Maeda, Sakurai and Roman 2016): `http://slepc.upv.es/`. CISS is based on the method of Sakurai and Sugiura (2003) and capable of solving generalized linear and nonlinear eigenvalue problems; see also the web page `http://zpares.cs.tsukuba.ac.jp/?page_id=56`.

6. Methods based on linearization

Instead of solving an NEP with $F \in H(\Omega, \mathbb{C}^{n\times n})$ directly, we might first replace it with a "proxy" NEP of a simpler structure. In this section we shall focus on proxies that can be obtained via polynomial or, more generally, rational approximation of F. More specifically, we will consider NEPs $R_m(\lambda)v = 0$ with

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m, \tag{6.1}$$

where the $D_j \in \mathbb{C}^{n\times n}$ are constant coefficient matrices and the b_j are rational functions of type (m, m), i.e., quotients of polynomials of degree at most m. Note that polynomial eigenvalue problems are special cases of (6.1) when all the functions b_j are polynomials.

It is crucial that $R_m \approx F$ in some sense, for otherwise the eigenpairs of F and R_m are not necessarily related to each other. For example, if $\Sigma \subset \Omega$ is

[1] This fact has been pointed out to us by Wolf-Jürgen Beyn in private communication.

a compact set, we may want to impose that

$$\|F - R_m\|_\Sigma := \max_{z\in\Sigma} \|F(z) - R_m(z)\|_2 \le \varepsilon,$$

because then the eigenvalues of R_m in Σ can be guaranteed to be approximations to some of the eigenvalues of F. This can be seen as follows: assume that (λ, v) with $\lambda \in \Sigma$ and $\|v\|_2 = 1$ is an eigenpair of R_m, i.e., $R_m(\lambda)v = 0$. Then from

$$\|F(\lambda)v\|_2 = \|\big(F(\lambda) - R_m(\lambda)\big)v\|_2 \le \|F(\lambda) - R_m(\lambda)\|_2 \le \varepsilon$$

we find that (λ, v) has a bounded residual for the original NEP $F(\lambda)v = 0$. Conversely, if $\mu \in \Sigma$ is *not* an eigenvalue of F, i.e., $F(\mu)$ is nonsingular, then a sufficiently accurate approximant R_m is also nonsingular at μ, which can be argued as follows: assume that $\|F(\mu) - R_m(\mu)\|_2 < \|F(\mu)^{-1}\|_2^{-1}$, then

$$\|I - F(\mu)^{-1}R_m(\mu)\|_2 \le \|F(\mu)^{-1}\|_2\|F(\mu) - R_m(\mu)\|_2 < 1.$$

Hence all eigenvalues of $I - F(\mu)^{-1}R_m(\mu)$ are strictly smaller in modulus than 1. As a consequence, $F(\mu)^{-1}R_m(\mu)$ and hence $R_m(\mu)$ are nonsingular. Ideally, R_m does not have any eigenvalues in Σ which are in the resolvent set of F. In this case we say that R_m is free of *spurious* eigenvalues on Σ.

In the following Section 6.1 we discuss how to obtain an effective approximant R_m via interpolation of F and how to quantify its approximation error. In Section 6.2 we explain and demonstrate some practical approaches for the construction of R_m. If the functions b_j in (6.1) satisfy a linear recurrence relation, then we can linearize R_m to obtain an equivalent generalized eigenvalue problem. This linearization technique will be the subject of Section 6.3. In Section 6.4 we review some solution techniques for the linear problem, and Section 6.5 lists related work and available software.

6.1. Polynomial and linear rational interpolation

Assume that we are given a function $F \in H(\Omega, \mathbb{C}^{n\times n})$ on a domain $\Omega \subseteq \mathbb{C}$, a compact and connected *target set* $\Sigma \subset \Omega$, and two disjoint sequences of *interpolation nodes* $(\sigma_j)_{j=0}^m \subset \Sigma$ and *poles* $(\xi_j)_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Omega$. Let the associated *nodal* (*rational*) *function* be defined as

$$s_m(z) = \prod_{j=0}^{m}(z - \sigma_j) \Big/ \prod_{\substack{j=1\\ \xi_j\neq\infty}}^{m} (z - \xi_j). \tag{6.2}$$

Further let $\Gamma \subset \Omega$ be a contour which encloses Σ, and hence contains all nodes σ_j in its interior. Then by the Walsh–Hermite integral formula (see, e.g., (Walsh 1935, Chapter VIII, Theorem 2)),

$$R_m(z) = \frac{1}{2\pi\mathrm{i}} \int_\Gamma \left(1 - \frac{s_m(z)}{s_m(\zeta)}\right) \frac{F(\zeta)}{\zeta - z}\,\mathrm{d}\zeta$$

is the unique matrix-valued rational function of type (m, m) with preassigned poles ξ_j that interpolates F in the Hermite sense (that is, counting multiplicities) at the nodes σ_j. Note that the pole parameters ξ_j are fixed and hence we are dealing here with *linearized rational interpolants*, which in contrast to nonlinear rational interpolants with free poles have existence and uniqueness properties very similar to polynomial interpolants. We refer to Stahl (1996) for an overview. As a consequence, if F itself is a rational matrix-valued function of type (m, m) with poles $\xi_1, \ldots, \xi_m$, then the interpolant R_m with these poles will be exact for any choice of sampling points σ_j away from the poles, i.e., $R_m \equiv F$.

Different choices of the nodal function in (6.2) give rise to different types of interpolants. For example, if we set all poles $\xi_j = \infty$, then R_m reduces to a matrix polynomial of degree m. Two popular instances of polynomial interpolants are:

(i) If all $\xi_j = \infty$ and all $\sigma_j = \sigma$ are identical, then $s_m(z) = (z - \sigma)^{m+1}$ and R_m is the degree m truncated Taylor expansion of F at $z = \sigma$,

$$R_m(z) = F(\sigma) + F'(\sigma)(z - \sigma) + \cdots + F^{(m)}(\sigma)\frac{(z - \sigma)^m}{m!}. \tag{6.3}$$

As $m \to \infty$, this Taylor series converges geometrically in concentric disks about the expansion point σ in which F is holomorphic. More precisely, let $\mathbb{D}_{\sigma,\rho} = \{z \in \mathbb{C} : |z - \sigma| < \rho\}$ with $\rho > 0$ be a disk in which F is holomorphic and bounded (by which we mean that $\|F(z)\|_2$ is bounded for all $z \in \mathbb{D}_{\sigma,\rho}$). Further, let $\Sigma = \overline{\mathbb{D}}_{\sigma,\rho_0}$ be a smaller closed disk with $0 < \rho_0 < \rho$. Then there exists a constant c depending only on F, σ, ρ, ρ_0 such that

$$\|F - R_m\|_\Sigma \le c\left(\frac{\rho_0}{\rho}\right)^m. \tag{6.4}$$

Clearly, this interpolation type is appropriate if the wanted eigenvalues of F lie inside a circular region about the target point σ.

(ii) If all $\xi_j = \infty$, $\Sigma = [-1, 1]$, and the σ_j are chosen as Chebyshev points of the first kind

$$\sigma_j = \cos\left(\frac{j + 1/2}{m + 1}\pi\right), \quad j = 0, 1, \ldots, m,$$

then $s_m(z) = T_{m+1}(z)$ is a Chebyshev polynomial defined by the recursion

$$T_0(z) = 1, \quad T_1(z) = z, \quad T_{m+1}(z) = 2zT_m(z) - T_{m-1}(z). \tag{6.5}$$

For $m \to \infty$, the polynomial interpolant R_m converges inside Bernstein ellipse regions (5.7) in which F is holomorphic. More precisely, let $\mathbb{E}_\rho$ with $\rho > 1$ be a Bernstein ellipse region in which F is holomorphic and

bounded, and let $\Sigma = \overline{\mathbb{E}}_{\rho_0}$ be a smaller closed Bernstein ellipse region with $1 < \rho_0 < \rho$. Then there exists a constant c depending only on F, ρ, ρ_0 such that

$$\|F - R_m\|_\Sigma \le c \left(\frac{\rho_0}{\rho}\right)^m;$$

see (Effenberger and Kressner 2012, Proposition 3.1). The same convergence rate is achieved with interpolation at Chebyshev points of the second kind

$$\sigma_j = \cos(j\pi/m), \quad j = 0, 1, \ldots, m.$$

Chebyshev interpolation is most appropriate if the wanted eigenvalues of F lie in or nearby the interval $[-1, 1]$. Via linear mapping it is possible to interpolate F on (complex) intervals $[a, b]$ other than $[-1, 1]$, and via parameterization $t \mapsto F(\gamma(t))$ on arbitrary smooth curves in the complex plane.

These two examples indicate that a "good" choice of the nodes σ_j for polynomial interpolation is dictated by the target set Σ in which the wanted eigenvalues of F are located, and that the rate of convergence is dictated by the location of the singularities of F relative to Σ.

More generally, in linear rational interpolation we also have the freedom to select the pole parameters ξ_j. An informed choice of these parameters can be made by inspecting the Walsh–Hermite formula for the error

$$F(z) - R_m(z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{s_m(z)}{s_m(\zeta)} \frac{F(\zeta)}{\zeta - z} \, \mathrm{d}\zeta.$$

By standard estimation of integrals we have

$$\|F(z) - R_m(z)\|_2 \le c \frac{|s_m(z)|}{\min_{\zeta \in \Gamma} |s_m(\zeta)|} \quad \text{for all } z \in \Sigma, \tag{6.6}$$

with a constant c that only depends on F, Σ, and Γ. The pair (Σ, Γ) is called a *condenser* (Bagby 1967, Gonchar 1969) and in view of (6.6) our goal must be to construct a rational function s_m which is uniformly small on Σ and large on Γ. A greedy approach to selecting parameters $\sigma_j \in \Sigma$ and $\xi_j \in \Gamma$ that achieve this goal is 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:

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

Any resulting sequence of points are called *Leja–Bagby points* for (Σ, Γ) (Walsh 1932, Bagby 1969). One can show (see, e.g., (Levin and Saff 2006, Theorem 3.5)) that there exists a number $\operatorname{cap}(\Sigma, \Gamma) > 0$, called the *condenser capacity*, such that any sequence of rational functions (s_m) con-

structed with the Leja–Bagby procedure satisfies

$$\lim_{m\to\infty}\left(\frac{\max_{z\in\Sigma}|s_m(z)|}{\min_{z\in\Gamma}|s_m(z)|}\right)^{1/m} = \exp\big(-1/\mathrm{cap}(\Sigma,\Gamma)\big). \tag{6.7}$$

By the maximum modulus principle for holomorphic 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 condenser capacity, i.e., $\mathrm{cap}(\Sigma,\Gamma) = \mathrm{cap}(\Sigma,\Xi)$. Combining the inequality (6.6) and (6.7), we arrive at the asymptotic convergence result

$$\limsup_{m\to\infty}\|F-R_m\|_\Sigma^{1/m} \le \exp\big(-1/\mathrm{cap}(\Sigma,\Xi)\big)$$

for rational interpolation at Leja–Bagby points. The convergence is thus asymptotically exponential with a rate depending on the target set Σ and the poles on Ξ, which should stay a positive distance away from Σ.

The determination of the numerical value $\mathrm{cap}(\Sigma,\Xi)$ is difficult for general condensers (Σ,Ξ). However, in some cases there are known closed formulas derived from conformal maps of doubly connected domains; see Nehari (1975, Chapter VII). For example, if $\Xi = (-\infty,\alpha]$ is a real interval and $\Sigma = \overline{\mathbb{D}}_{\sigma,\rho}$ is the closed disk of radius $\rho>0$ centered at $\sigma > \alpha+\rho$, then one can show that

$$\mathrm{cap}(\Sigma,\Xi) = \frac{4}{\pi}\,\frac{K(\kappa)}{K(\sqrt{1-\kappa^2})}, \qquad \kappa = \left(\frac{\sigma-\alpha}{\rho} - \sqrt{\left(\frac{\sigma-\alpha}{\rho}\right)^2-1}\right)^2, \tag{6.8}$$

where

$$K(\kappa) = \int_0^1 \frac{1}{\sqrt{(1-t^2)(1-\kappa^2t^2)}}\,\mathrm{d}t$$

is the complete elliptic integral of the first kind[2]; see (Nehari 1975, pp. 293–294). For a list of some other special condensers and formulas for their capacities see (Güttel 2013) and the references therein.

6.2. Sampling approaches

To make the interpolation process described in the previous section constructive, we need to agree on some basis functions b_j for (6.1). For practical purposes it is usually advantageous to choose basis functions that are in a linear relation with one another. Let us list some examples.

[2] The definition of $K(\kappa)$ varies in the literature. We stick to the definition used by Nehari (1975, Chapter VI). In MATLAB the value $K(\kappa)$ is obtained with `ellipke(kappa^2)`.

- Shifted and scaled monomials satisfy the linear relation

$$b_0(z) = \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z-\sigma}{\beta_{j+1}} b_j(z) \tag{6.9}$$

  with some nonzero scaling factors β_j. An interpolant R_m with this basis corresponds to the degree m truncated Taylor expansion of F at $z = \sigma$ given in (6.3).

- Orthogonal polynomials satisfy a three-term recurrence

$$b_0(z) = \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z b_j(z) + \alpha_j b_j(z) + \gamma_j b_{j-1}(z)}{\beta_{j+1}}. \tag{6.10}$$

  Both the monomials (6.9) and the Chebyshev polynomials $b_j = T_j$ in (6.5) are special cases of orthogonal polynomial sequences.

- Given nodes $\sigma_0, \sigma_1, \ldots, \sigma_m$, the (scaled) Newton polynomials are defined by the linear relation

$$b_0(z) = \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z-\sigma_j}{\beta_{j+1}} b_j(z). \tag{6.11}$$

- Given distinct nodes $\sigma_0, \sigma_1, \ldots, \sigma_m$, the (scaled) Lagrange polynomials

$$b_j(z) = \frac{1}{\beta_j} \prod_{\substack{i=0 \\ i \neq j}}^{m} (z - \sigma_i) \tag{6.12}$$

  satisfy the linear relation

$$\beta_j (z - \sigma_j) b_j(z) = \beta_k (z - \sigma_k) b_k(z).$$

  In contrast to the above listed polynomials b_j whose degrees coincide with their indices j, the Lagrange polynomials are not degree-graded.

From an approximation point of view it does not matter in which basis an interpolant $R_m \approx F$ is represented, and indeed the results in Section 6.1 are independent of the particular representation of R_m. For practical computations, however, the choice of basis functions b_j is important. In this section we will focus on degree-graded *rational Newton basis functions*, which give rise to rational interpolants that can be constructed in a greedy fashion, that is, one (distinct) interpolation node at a time, and by only knowing the values of F at these nodes.

Given sequences of interpolation nodes $(\sigma_j)_{j=0}^m \in \Sigma$ and nonzero[3] poles $(\xi_j)_{j=1}^m \subset \overline{\mathbb{C}} \setminus \Sigma$, we define rational basis functions by the recursion

$$b_0(z) = \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z - \sigma_j}{\beta_{j+1}(1 - z/\xi_{j+1})} b_j(z). \tag{6.13}$$

Each rational function b_{j+1} has roots at the interpolation nodes $\sigma_0, \sigma_1, \ldots, \sigma_j$ and poles at $\xi_1, \ldots, \xi_j$. If all poles ξ_j are chosen at infinity, (6.13) reduces to the polynomial Newton recursion (6.11).

If all nodes $\sigma_0, \sigma_1, \ldots, \sigma_m$ are pairwise distinct, we can compute the coefficient matrices D_j of the interpolant (6.1) in a straightforward manner: by the interpolation condition $F(\sigma_0) = R_m(\sigma_0)$ and the formula for b_0 we have $D_0 = \beta_0 F(\sigma_0)$. From the interpolation conditions $F(\sigma_j) = R_m(\sigma_j)$ we then find recursively

$$D_j = \frac{F(\sigma_j) - b_0(\sigma_j)D_0 - \cdots - b_{j-1}(\sigma_j)D_{j-1}}{b_j(\sigma_j)}, \quad j = 1, \ldots, m. \tag{6.14}$$

The matrix-valued numerator of D_j can be evaluated via the Horner scheme starting with the coefficient matrix D_{j-1}, using the fact that each b_{j-1} divides b_j. Computing the matrices D_j this way is mathematically equivalent to computing the diagonal entries of a divided-difference tableau with matrix entries; see also (Güttel, Van Beeumen, Meerbergen and Michiels 2014).

In the confluent case, where some of the interpolation nodes coincide, derivatives of F will enter the formulas. If F is given in the form

$$F(z) = f_1(z)C_1 + f_2(z)C_2 + \cdots + f_\ell(z)C_\ell \tag{6.15}$$

with constant coefficient matrices $C_j \in \mathbb{C}^{n\times n}$, we can compute

$$F^{(k)}(z) = f_1^{(k)}(z)C_1 + f_2^{(k)}(z)C_2 + \cdots + f_\ell^{(k)}(z)C_\ell$$

simply by calculating derivatives of scalar functions and use a straightforwardly modified version of the sampling formula (6.14). There is, however, a more convenient approach to computing the interpolant R_m which does not require any sampling at all. To explain this, let us consider again the basis recursion (6.13) and write it in linearized form as

$$\beta_{j+1}(1 - z/\xi_{j+1})b_{j+1}(z) = (z - \sigma_j)b_j(z), \quad j = 0, 1, \ldots, m-1.$$

With the help of a z-dependent vector $b(z) = [b_0(z), b_1(z), \ldots, b_m(z)]^T$ we can combine these relations in matrix form as

$$zb^T(z)\underline{K_m} = b^T(z)\underline{H_m}, \tag{6.16}$$

[3] The exclusion of poles at the origin is merely for ease of notation and can easily be remedied by replacing all ξ_j by $\xi_j + \tau$ and $F(z)$ by $F(z + \tau)$. Alternatively, one can use the more general recursion $b_{j+1}(z) = \frac{z-\sigma_j}{\beta_{j+1}(\nu_{j+1}z - \mu_{j+1})} b_j(z)$ with $\xi_{j+1} = \mu_{j+1}/\nu_{j+1}$ at the cost of dealing with two parameters (μ_{j+1}, ν_{j+1}) instead of ξ_{j+1} only.

where

$$\underline{K_m} = \left[\begin{array}{ccccc} 1 & & & & \\ \frac{\beta_1}{\xi_1} & 1 & & & \\ & \ddots & \ddots & & \\ & & \frac{\beta_{m-1}}{\xi_{m-1}} & 1 & \\ \hline & & & & \frac{\beta_m}{\xi_m} \end{array}\right], \quad \underline{H_m} = \left[\begin{array}{ccccc} \sigma_0 & & & & \\ \beta_1 & \sigma_1 & & & \\ & \ddots & \ddots & & \\ & & & \beta_{m-1} & \sigma_{m-1} \\ \hline & & & & \beta_m \end{array}\right] \tag{6.17}$$

are $(m+1) \times m$-matrices. The underscores in our notation of $\underline{K_m}$ and $\underline{H_m}$ indicate the additional rows below the horizontal line, and we will denote by (H_m, K_m) the $m \times m$ matrix pencil obtained by omitting this row.

Linear relations of the form (6.16) also exist for Taylor, orthogonal polynomial, and Lagrange basis functions. It is not difficult to construct the corresponding matrices $\underline{K_m}$ and $\underline{H_m}$ using the basis recursions (6.9)–(6.12).

The decomposition (6.16) is called *rational Krylov decomposition* and there is a close connection between matrix functions associated with the pencil (H_m, K_m) and rational divided differences. This connection was first observed by Opitz (1964) for the case of polynomial interpolation and later extended to the rational case; see, e.g., (Güttel 2013, Section 3.4). The following theorem summarizes this approach; cf. (Güttel et al. 2014, Theorem 2.1).

Theorem 6.1. Given a matrix-valued function F in the split form (6.15) and rational basis functions b_j as in (6.13), define the matrices $D_j = \sum_{i=1}^{\ell} d_{i,j} C_i$ for $j = 0, 1, \ldots, m$, where

$$\begin{bmatrix} d_{i,0} \\ d_{i,1} \\ \vdots \\ d_{i,m} \end{bmatrix} = \beta_0 f_i\big(H_{m+1} K_{m+1}^{-1}\big) e_1, \quad i = 1, 2, \ldots, \ell.$$

with K_{m+1} and H_{m+1} as in (6.17) (with m replaced by $m+1$), and $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{m+1}$. Then

$$R_m(z) = b_0(z) D_0 + b_1(z) D_1 + \cdots + b_m(z) D_m$$

is the rational matrix-valued function of type (m, m) with poles at the points $\xi_1, \ldots, \xi_m$ that interpolates F in the Hermite sense at the nodes $\sigma_0, \sigma_1, \ldots, \sigma_m$.

The following two examples compare the performance of polynomial versus rational interpolation, with the first example also demonstrating that if F itself is a rational matrix-valued function of type (m, m) with poles $\xi_1, \ldots, \xi_m$, then its rational interpolant R_m with these poles will be exact.

```
% NLEIGS_sampling
n = 100; C1 = n*gallery('tridiag',n); C1(end) = C1(end)/2;
C2 = (abs(gallery('tridiag',n)) + 2*speye(n))/(6*n);
C2(end) = C2(end)/2; C3 = sparse(n,n); C3(n,n) = 1;
F = @(z) C1 - z*C2 + C3*z/(z-1); tol = 0; mmax = 50;
Sigma = 150+146*exp(2i*pi*(0:99)/100); Xi = inf; % Leja on circle
Sigma = [4,296]; Xi = inf;                       % Leja on interval
Sigma = [4,296]; Xi = [1,inf]; mmax = 2;         % Leja-Bagby
Rm = util_nleigs(F, Sigma, Xi, tol, mmax);       % NLEIGS sampling
Lm = linearize(Rm); [Am,Bm] = Lm.get_matrices(); % linearization
```

Figure 6.2: Basic MATLAB calls of the Leja–Bagby sampling routine implemented in the RKToolbox. The first four lines define the `loaded_string` problem. The three lines starting with "`Sigma = ...`" correspond to the three different choices of Leja–Bagby points discussed in Example 6.3. The interpolant R_m computed by `util_nleigs` is represented by an object `Rm` and can be evaluated at any $z \in \mathbb{C}$ by typing `Rm(z)`. The last line generates a matrix pencil $\boldsymbol{L}_m(z) = \boldsymbol{A}_m - z\boldsymbol{B}_m$ which corresponds to a linearization of R_m. This will be discussed in Section 6.3.

Example 6.3. Let us reconsider the `loaded_string` NEP (1.3) of size $n = 100$. Recall that this is in fact a rational NEP of type $(2, 1)$ which could be transformed into a polynomial one via multiplication by $z - 1$; however, this would introduce a spurious eigenvalue $\lambda = 1$ of multiplicity $n-1$ and we prefer not to do this here. We are interested in the eigenvalues of F located in the interval $[4, 296]$ and use the sampling routine implemented in the MATLAB Rational Krylov Toolbox (RKToolbox) by Berljafa and Güttel (2014). A code example is shown in Figure 6.2. The function `util_nleigs` returns a so-called RKFUN object `Rm` which represents the rational interpolant R_m. This object can be evaluated at any point $z \in \mathbb{C}$ by typing `Rm(z)` and its poles are listed with the command `poles(Rm)`; see (Berljafa and Güttel 2015) for implementation details.

The uniform interpolation errors

$$\|F - R_m\|_{[4,296]} = \max_{z\in[4,296]} \|F(z) - R_m(z)\|_2$$

resulting from NLEIGS sampling with three different choices for the sampling nodes σ_j and the poles ξ_j are plotted as functions of the degree m on the top left of Figure 6.5, together with the predicted convergence factors shown as dashed lines. The bottom left plot shows the eigenvalues of the different interpolants R_m.

(a) Choosing the sampling points σ_j on a circle of radius $\rho_0 = 146$ centered about the point $z = 150$, while keeping all $\xi_j = \infty$, results in a sequence of Leja interpolants (the polynomial special case of Leja–Bagby interpolants) whose errors over $[4, 296]$ tend to reduce by a fac-

tor $\rho_0/\rho = 146/149$ as the degree m increases; see the grey curve with pluses and the dashed line on the top left of Figure 6.5. Here, $\rho = 149$ is the radius of the largest possible disk centered at $z = 150$ in which F is holomorphic. Recall from (6.4) that this is exactly the same convergence factor that would be achieved by truncated Taylor expansions of F about $z = 150$, however, the "sweeping out" (or, in the language of potential theory, *balayage*) of the interpolation nodes to the boundary of a disk centered at $z = 150$ allows for derivative-free sampling at the same convergence rate.

The bottom left plot in Figure 6.5 shows some of the eigenvalues of the matrix polynomial R_{50} as grey pluses. Here are the first five eigenvalues of R_{50} whose imaginary part is smallest in modulus, printed in order of increasing real part:

```
  4.5086126318959 + 0.00078529409700i
123.0253244697823 + 0.00061650205038i
202.1974982022052 + 0.00019912027330i
301.3093923260613 - 0.0166126006570i
356.3668306213224 - 0.0114103053079i
```

The first three of these eigenvalues are inside the convergence disk and seem to approach some of the exact eigenvalues of F given in (3.5). We have underlined the correct digits of these three eigenvalues of R_{50} as approximations to their closest counterparts of F. The other eigenvalues of R_{50}, however, cannot be trusted as they are not inside the convergence disk. Interestingly, many of these "spurious" eigenvalues tend to align on the boundary of that disk (see again the grey pluses in the bottom left plot in Figure 6.5). This phenomenon is observed frequently in the literature; see, e.g., Jarlebring and Güttel (2014, Section 4.3). While there appears to be no analysis of this effect, it may plausibly be related to a similar behavior of roots of scalar truncated Taylor expansions. In particular, Jentzsch (1916) shows that every point of the circle of convergence for a scalar power series is an accumulation point for the roots of its partial sums. More general results of this type can be made for polynomial best approximants on compact sets; see, e.g., (Blatt, Saff and Simkani 1988). With $\mathrm{Taylor}_{\sigma,m}[\,\cdot\,]$ being the operator that returns the degree m truncated Taylor expansion about σ of its argument, it is easy to verify that

$$\det(\mathrm{Taylor}_{\sigma,m}[F(z)]) = \mathrm{Taylor}_{\sigma,m}[\det F(z)] + O((z-\sigma)^{m+1}).$$

This may be a possible starting point for an analysis of the limiting behavior of spurious eigenvalues in NEP linearizations.

(b) Choosing the sampling points σ_j on the interval $[4, 296]$, while keeping all $\xi_j = \infty$, results in a sequence of Leja interpolants whose errors over

$[4, 296]$ tend to reduce by a factor $\rho_0^{-1} = 0.817$ as the degree m increases. Here, ρ_0 is the elliptical radius of the largest Bernstein ellipse region associated with $[4, 296]$ in which F is holomorphic. This is exactly the same convergence factor that would be obtained with interpolation at Chebyshev points on the interval Σ; however, the greedy choice of Leja–Bagby nodes allows for the degree m to be adaptively increased while keeping previously chosen nodes σ_j fixed. Computationally this is often advantageous as the degree m required for a desired sampling accuracy is typically unknown in advance.
The small blue dots in the bottom left plot in Figure 6.5 correspond to the eigenvalues of the matrix polynomial R_{50}. Among them are some very good approximations to the eigenvalues of F in the interval $[4, 296]$ (correct digits are underlined):

```
  4.4821710017932 + 0.0000000000000000i
 24.2235759507902 + 0.0000000000000000i
 63.7238218842619 + 0.0000000000000000i
123.0312214021378 + 0.0000000000000000i
202.2009012596192 + 0.0000000000000000i
```

Again some of the spurious eigenvalues align on the boundary of the convergence region, which in this case is a Bernstein ellipse.

(c) Finally, we choose the "exact" poles $\xi_1 = 1$, $\xi_2 = \infty$ of F, and arbitrary sampling points away from the singularity $z = 1$. As expected, the resulting type $(2, 1)$ rational interpolant R_2 coincides exactly with F, which is indicated by the sharp drop in the interpolation error; see the solid red curve with circles in Figure 6.5 (top left). Note that, in contrast to the previous two polynomial interpolants, $R_2 \equiv F$ is exact not only in the sampling region, but throughout the complex plane. As a consequence, R_2 also discovers the eigenvalue `0.45731848895` of F; see the red circles in the bottom left of Figure 6.5.

Example 6.4 (gun). We consider the NEP (1.4). The complex square root $\sqrt{\cdot}$ corresponds to the principal branch and the parameters are chosen as $\omega_1 = 0$ and $\omega_2 = 108.8774$. The target set Σ is the closed disk $\overline{\mathbb{D}}_{\sigma,\rho_0}$ of radius $\rho_0 = 5 \times 10^4$ centered at $\sigma = 250^2$. There is numerical evidence that F has 21 eigenvalues in Σ. The singularity set $\Xi = (-\infty, \omega_2^2]$ corresponds to the union of branch cuts of the square roots in (1.4).

For this example we try two different sets of Leja–Bagby points and plot the convergence of the resulting interpolants together with their predicted convergence factors (plotted as dashed lines) on the top right of Figure 6.5, with the eigenvalues of the interpolants shown on the bottom right.

(a) Choosing Leja points σ_j on the boundary of Σ while keeping all $\xi_j = \infty$ results in a sequence of polynomial interpolants R_m whose errors over

Σ reduce with a factor $\rho_0/\rho \approx 0.987$ as m increases; see the grey curve with pluses and the dashed line in the top right of Figure 6.5. Here, $\rho = \sigma - \omega_2^2$ is the radius of the largest possible disk centered at σ so that F is holomorphic in its interior. This is the same convergence factor that would be obtained with a truncated Taylor expansion of F about σ, but choosing the nodes σ_j on the boundary of Σ avoids the need of derivatives of F.

On the bottom right of Figure 6.5 we show the eigenvalues λ of R_{100} with a relative residual $\|F(\lambda)v\|_2/\|v\|_2$ below 10^{-3} as grey pluses. Inside the target set Σ there are 13 such eigenvalues.

(b) We now choose Leja–Bagby nodes on the condenser (Σ, Ξ), resulting in a rational interpolant with a geometric convergence factor $\exp(-1/\mathrm{cap}(\Sigma, \Xi)) \approx 0.465$, where $\mathrm{cap}(\Sigma, \Xi)$ is given by (6.8). This convergence is significantly faster than for the polynomial interpolant. Some of the poles of R_{38} are shown as magenta dots in the bottom right of Figure 6.5, and some of the eigenvalues of R_{38} are shown as red circles. All 21 eigenvalues of F in Σ are very well approximated by eigenvalues of R_{38}. Moreover, R_{38} appears to be free of spurious eigenvalues on Σ, and there are even some eigenvalues outside Σ which have a small relative residual $\|F(\lambda)v\|_2/\|v\|_2$.

Looking at the examples in Figure 6.5 we conclude that rational interpolation can significantly outperform polynomial interpolation. This is in particular the case when F has singularities nearby the target set Σ.

It remains to discuss the choice of scaling parameters β_j for the rational functions b_j defined in (6.13). Perhaps the most natural approach, also advocated in Güttel et al. (2014, Section 5.1), is to select each β_j such that $\|b_j\|_\Sigma = 1$. In this case we have

$$\|F - R_m\|_\Sigma \le \|D_{m+1}\|_2 + \|D_{m+2}\|_2 + \cdots ,$$

hence if the norms of the coefficient matrices D_j decay quickly we can use $\|D_{m+1}\|_2$ (or $\|D_{m+1}\|_F$) as a cheap upper estimate for the interpolation error $\|F - R_m\|_\Sigma$. (As will be seen in the following section, this scaling also leads to well-balanced block entries in the eigenvectors of the linearization pencil $\boldsymbol{L}_m(z)$ of $R_m(z)$.) Practically, the normalization of each b_j can be achieved by setting $\beta_j = 1/\max_{\partial\Sigma_p} |\widehat{b}_j(z)|$, where $\partial\Sigma_p$ is a fine discretization of the boundary of Σ with p points (typically, $p = 1000$ is enough) and $\widehat{b}_j$ is the basis function prior to scaling. As this normalization procedure only involves evaluations of scalar functions, its computational cost is typically negligible.

While we have focused in this section on rational Newton interpolation for F due to its greedy approach, the approximation R_m in (6.1) via Lagrange interpolation is easy to construct since $D_j = F(\sigma_j)$. For Chebyshev interpolation, Effenberger and Kressner (2012) propose to use a sequence

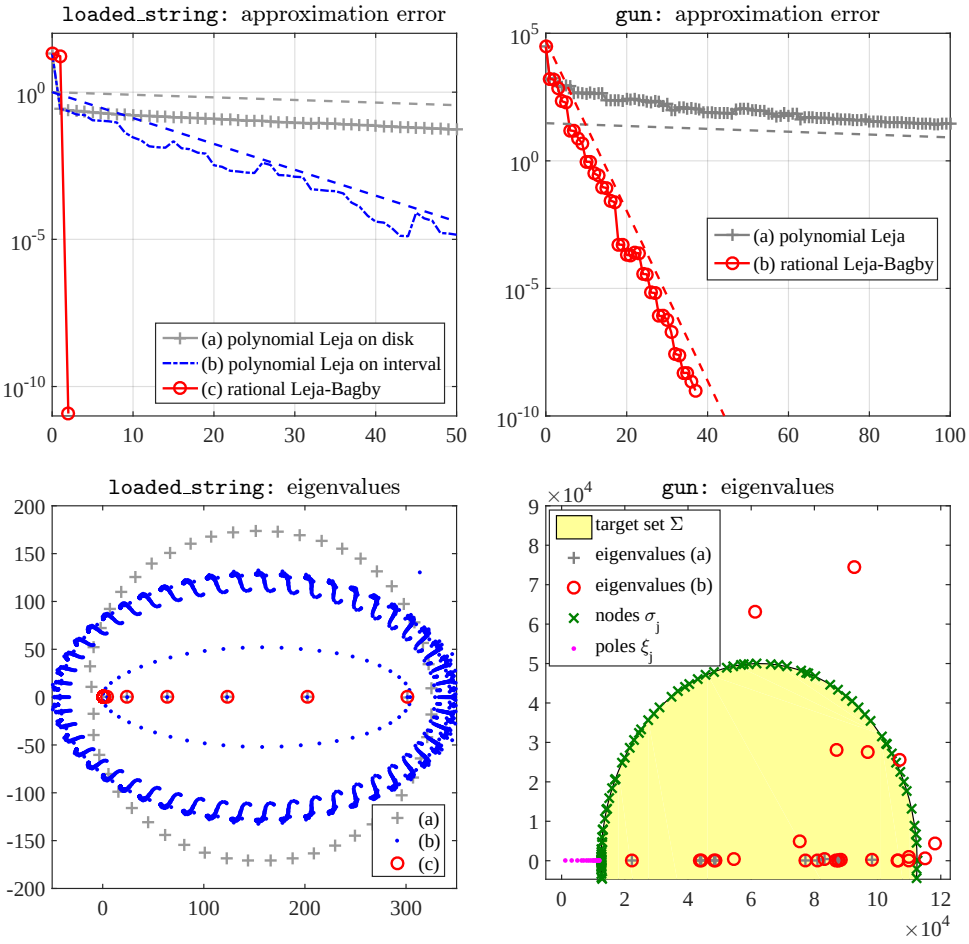


Figure 6.5: Top: Approximation errors $\|F - R_m\|_\Sigma$ of Leja–Babgy interpolants R_m for different choices of Leja–Bagby points as the degree $m \geq 0$ increases, together with the predicted error decay rates (dashed lines). Top left: Sampling of the `loaded_string` problem as discussed in Example 6.3 with $\Sigma = [4, 296]$. Top right: polynomial versus rational sampling of the `gun` problem as discussed in Example 6.4 with Σ chosen to be the closed disk of radius 5×10^4 centered at 6.25×10^4. Bottom: Eigenvalues of the different Leja–Bagby interpolants R_m for both the `loaded_string` problem (left) and the `gun` problem (right). For the `gun` problem in the case of rational Leja–Bagby interpolation we also show the interpolation nodes σ_j and poles ξ_j. A part of the lower half of the target set Σ is outside the visible region, but this part is free of eigenvalues anyway.

of inverse discrete cosine transforms whose type depends on the choice of interpolation nodes.

6.3. Linearization

The polynomial or rational eigenvalue problem $R_m(\lambda)v = 0$ with R_m of the form (6.1) can easily be converted into a linear eigenvalue problem when the basis functions b_j are in a linear relation with one another, that is, when there exist matrices $\underline{K_m}, \underline{H_m} \in \mathbb{C}^{(m+1)\times m}$ satisfying a relation of the form (6.16). Using the linear relation between b_m and b_{m-1}, $R_m(z)$ is rewritten in the form

$$g(z)R_m(z) = \sum_{j=0}^{m-1} b_j(z)(A_j - zB_j) \tag{6.18}$$

for some function $g(z)$. For example, using the Newton-type basis functions defined in (6.13), we get

$$R_m(z) = b_0(z)D_0 + \cdots + b_{m-1}(z)D_{m-1} + \underbrace{\frac{z-\sigma_{m-1}}{\beta_m(1-z/\xi_m)}b_{m-1}(z)}_{b_m(z)} D_m,$$

and by multiplying the above equation by $g(z) = (1 - z/\xi_m)$, (6.18) is obtained with

$$\begin{aligned} A_j = D_j, \quad B_j = D_j/\xi_m, \quad j = 0, 1, \ldots, m-2, \\ A_{m-1} = D_{m-1} - \frac{\sigma_{m-1}}{\beta_m}D_m, \quad B_{m-1} = \frac{D_{m-1}}{\xi_m} - \frac{D_m}{\beta_m}. \end{aligned}$$

If R_m is represented in the Taylor, orthogonal polynomial, or Lagrange basis then R_m in (6.1) is a matrix polynomial and $g(z) = 1$ in (6.18). For sake of completeness, the conversion rules for the coefficient matrices D_j used in the form (6.18) into the coefficients matrices (A_j, B_j) used in (6.18) are provided in Table 6.6; see also (Van Beeumen, Meerbergen and Michiels 2015*a*, Table 1).

Now (6.18) and the first m equations in (6.16) can be rewritten as

$$\boldsymbol{L}_m(z)\big(b(z) \otimes I_n\big) = \big(g(z)e_1 \otimes I_n\big)R_m(z), \tag{6.19}$$

where $\boldsymbol{L}_m(z) = \boldsymbol{A}_m - z\boldsymbol{B}_m$ is an $mn \times nm$ pencil with

$$\boldsymbol{A}_m = \left[\begin{array}{cccc} A_0 & A_1 & \cdots & A_{m-1} \\ \hline \multicolumn{4}{c}{\underline{H_{m-1}}^T \otimes I_n} \end{array}\right], \quad \boldsymbol{B}_m = \left[\begin{array}{cccc} B_0 & B_1 & \cdots & B_{m-1} \\ \hline \multicolumn{4}{c}{\underline{K_{m-1}}^T \otimes I_n} \end{array}\right] \tag{6.20}$$

and

$$b(z) = \begin{bmatrix} b_0(z) & b_1(z) & \cdots & b_{m-1}(z) \end{bmatrix}^T.$$

Table 6.6: Conversion of R_m in the form (6.1) into the form (6.18). Here, the basis functions c_j are chosen identically to the functions b_j for $j = 0, 1, \ldots, m-1$. The ranges of the variable j indicated in the column for B_j are the same for A_j along each row.

Basis $b_j = c_j$	A_j	B_j
Taylor (6.9)	$\begin{cases} D_j \\ D_{m-1} - \frac{\sigma}{\beta_m} D_m \end{cases}$	$\begin{cases} O, & j < m-1 \\ -D_m/\beta_m, & j = m-1 \end{cases}$
Orthogonal (6.10)	$\begin{cases} D_j \\ D_{m-2} + \frac{\gamma_{m-1}}{\beta_m} D_m \\ D_{m-1} + \frac{\alpha_{m-1}}{\beta_m} D_m \end{cases}$	$\begin{cases} O, & j < m-2 \\ O, & j = m-2 \\ -D_m/\beta_m, & j = m-1 \end{cases}$
Lagrange (6.12)	$\begin{cases} \sigma_m D_j \\ \sigma_m D_{m-1} + \frac{\beta_{m-1}\sigma_m}{\beta_m} D_m \end{cases}$	$\begin{cases} D_j, & j < m-1 \\ D_{m-1} + \frac{\beta_{m-1}}{\beta_m} D_m, & j = m-1 \end{cases}$
RatNewton (6.13)	$\begin{cases} D_j \\ D_{m-1} - \frac{\sigma_{m-1}}{\beta_m} D_m \end{cases}$	$\begin{cases} O, & j < m-1 \\ -D_m/\beta_m, & j = m-1 \end{cases}$

Using a block (permuted) UL decomposition of $\boldsymbol{L}_m(z_0)$ for $z_0 \in \mathbb{C}$, Van Beeumen et al. (2015*a*, Corollary 2.4) show that if $\overline{H_{m-1}}^T - zK_{m-1}^T$ is of rank $m-1$ for all z, then R_m regular implies that $\boldsymbol{L}_m$ is regular. While the spectra of R_m and $\boldsymbol{L}_m$ are not necessarily identical when R_m is a rational matrix-valued function, Grammont, Higham and Tisseur (2011, Theorem 3.1) show that the one-sided factorization in (6.19) implies useful relations between the eigensystem of R_m and that of $\boldsymbol{L}_m$.

Theorem 6.7. Let $R_m(\lambda)$ and $\boldsymbol{L}_m(z)$ be matrix functions of dimensions $n \times n$ and $mn \times mn$, respectively. Assume that (6.19) holds at $z = \lambda \in \mathbb{C}$ with $g(\lambda) \neq 0$ and $b(\lambda) \neq 0$. Then

(i) $b(\lambda) \otimes v$ is a right eigenvector of $\boldsymbol{L}_m$ with eigenvalue λ if and only if v is a right eigenvector of R_m with eigenvalue λ.
(ii) If $\boldsymbol{w} \in \mathbb{C}^{mn}$ is a left eigenvector of $\boldsymbol{L}_m$ with eigenvalue λ then $\big(g(z)e_1 \otimes I_n\big)^* \boldsymbol{w}$ is a left eigenvector of R_m with eigenvalue λ provided that it is nonzero.

The pencil $\boldsymbol{A}_m - z\boldsymbol{B}_m$ with $\boldsymbol{A}_m, \boldsymbol{B}_m$ of the form (6.20) is referred to as *CORK linearization*. Such pencils have been used in (Güttel et al. 2014, Van Beeumen et al. 2015*a*). They are a generalization of the

polynomial Newton-type linearizations in (Van Beeumen, Meerbergen and Michiels 2013). While these linearizations are most naturally constructed by interpolatory sampling as explained in the previous section, they may also result from non-interpolatory approximation procedures. The following example illustrates this.

Example 6.8. Consider again the `loaded_string` problem (1.3)

$$F(z) = C_1 - zC_2 + \frac{z}{z-1}C_3.$$

Using the basis functions $b_0(z) = 1$, $b_1(z) = z/(1-z)$, and $b_2(z) = -z$ we can rewrite F in the form (6.1), namely

$$R_2(z) = b_0(z)D_0 + b_1(z)D_1 + b_2(z)D_2$$

with $D_0 = C_1$, $D_1 = -C_3$, $D_2 = C_2$. By choosing the basis functions b_j we have implicitly specified the parameters $\sigma_0 = 0$, $\sigma_1 = 1$, $\xi_1 = 1$, $\xi_2 = \infty$, and $\beta_0 = \beta_1 = \beta_2 = 1$. Note that $\sigma_1 = 1$ would be an invalid choice as a sampling point as F has a pole there. Nevertheless, F and R_2 are identical. By the rational Krylov decomposition

$$z[b_0(z), b_1(z), b_2(z)] \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}}_{\underline{K_2}} = [b_0(z), b_1(z), b_2(z)] \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\underline{H_2}},$$

Table 6.6, and (6.20), we arrive at the $2n \times 2n$ CORK linearization

$$\begin{aligned} \boldsymbol{L}_m(z) &= \left[\begin{array}{cc} D_0 & D_1 - D_2 \\ \hline \multicolumn{2}{c}{\underline{H_1}^T \otimes I_n} \end{array}\right] - z \left[\begin{array}{cc} O & -D_2 \\ \hline \multicolumn{2}{c}{\underline{K_1}^T \otimes I_n} \end{array}\right] \\ &= \left[\begin{array}{cc} D_0 & D_1 - D_2 \\ \hline O & I_n \end{array}\right] - z \left[\begin{array}{cc} O & -D_2 \\ \hline I_n & I_n \end{array}\right]. \end{aligned}$$

If some of the matrix pairs (A_j, B_j) in (6.18) are of low rank, one can further reduce the size of the linearization. In the following example we illustrate this so-called *trimmed linearization* approach.

Example 6.9. In the `loaded_string` problem (1.3)

$$F(z) = C_1 - zC_2 + \frac{z}{z-1}C_3,$$

the matrix $C_3 = e_n e_n^T$ is of rank 1. Using the basis functions $b_0(z) = 1$ and $b_1(z) = 1/(1-z)$, we can rewrite F in the form (6.18) as

$$R_2(z) = b_0(z)(A_0 - zB_0) + b_1(z)(A_1 - zB_1)$$

with $A_0 = C_1$, $B_0 = C_2$, $A_1 = O$, and $B_1 = C_3$. Note that $b_1(z)$ is now a rational function of subdiagonal type $(0,1)$, referred to as "proper form" in Su and Bai (2011). The $2n \times 2n$ CORK linearization is

$$\boldsymbol{L}_2(z) \;=\; \left[\begin{array}{cc} C_1 & O \\ \hline -I_n & I_n \end{array}\right] - z \left[\begin{array}{cc} C_2 & e_n e_n^T \\ \hline O & I_n \end{array}\right].$$

Considering a structure eigenvector $\boldsymbol{v} = [b_0(\lambda)v^T, b_1(\lambda)v^T]^T$ we find that $\boldsymbol{L}_2(\lambda)\boldsymbol{v} = \boldsymbol{0}$ is equivalent to the equations

$$\begin{aligned} C_1 b_0(\lambda)v - \lambda C_2 b_0(\lambda)v - \lambda e_n e_n^T b_1(\lambda)v &= 0, \\ -b_0(\lambda)v + b_1(\lambda)v - \lambda b_1(\lambda)v &= 0. \end{aligned}$$

The last of the two equations specifies the linear relation between $b_0(z)$ and $b_1(z)$, and it does so even if we multiply it from the left by e_n^T (as long as $e_n^T v \neq 0$). Hence we can rewrite the reduced relations as a trimmed $(n+1) \times (n+1)$ linearization $\widehat{\boldsymbol{L}}_2(\lambda)\widehat{\boldsymbol{v}} = \boldsymbol{0}$, where

$$\widehat{\boldsymbol{L}}_2(z) = \left[\begin{array}{cc} C_1 & 0 \\ \hline -e_n^T & 1 \end{array}\right] - z \left[\begin{array}{cc} C_2 & e_n \\ \hline 0^T & 1 \end{array}\right], \quad \widehat{\boldsymbol{v}} = \left[\begin{array}{c} b_0(z)v \\ b_1(z)e_n^T v \end{array}\right].$$

Trimmed linearizations appeared in the context of polynomial eigenvalue problems with singular coefficient matrices in (Byers, Mehrmann and Xu 2008) and for rational eigenvalue problems in (Su and Bai 2011, Alam and Behera 2016). The approach has been extended to NEPs involving several low-rank terms in a series of works like, e.g., (Van Beeumen et al. 2013, Güttel et al. 2014, Van Beeumen, Jarlebring and Michiels 2016*a*, Lu, Huang, Bai and Su 2015).

6.4. Solving the linearized problem

Theorem 6.7 establishes a one-to-one correspondence between right eigenpairs of R_m and structured right eigenpairs of the linear $mn \times mn$ pencil $\boldsymbol{L}_m(z) = \boldsymbol{A}_m - z\boldsymbol{B}_m$, and all that remains is to solve this generalized eigenvalue problem. If mn is moderate, all eigenpairs of the linearization can be found via the QZ algorithm in $O((mn)^3)$ floating point operations. For many problems arising in applications, however, mn is too large for QZ and iterative techniques for large-scale eigenproblems are required.

In principle, any available iterative algorithm for generalized eigenproblems can be used; see, e.g., (Saad 2011, Bai, Demmel, Dongarra, Ruhe and van der Vorst 2000) for comprehensive overviews and (Hernandez, Roman, Tomas and Vidal 2009) for a survey of available software. In the context of NEPs, one of the most popular approaches to find a few eigenpairs of the pencil $\boldsymbol{L}_m$ uses the rational Arnoldi algorithm by Ruhe (1998).

Given a nonzero starting vector $\boldsymbol{v} \in \mathbb{C}^{mn}$ and a sequence of shift parameters $\tau_1, \ldots, \tau_k \in \mathbb{C} \setminus \Lambda(\boldsymbol{L}_m)$, this algorithm attempts to compute an orthonormal basis $\boldsymbol{V}_{k+1} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k+1}] \in \mathbb{C}^{mn \times (k+1)}$ of a *rational Krylov space* defined by Algorithm 6.10.

Algorithm 6.10: Rational Arnoldi algorithm

Given $\{\boldsymbol{A}_m, \boldsymbol{B}_m\} \subset \mathbb{C}^{mn \times mn}$, $\boldsymbol{v} \in \mathbb{C}^{mm} \setminus \{\boldsymbol{0}\}$, shifts $(\tau_j)_{j=1}^k \subset \mathbb{C}$.
Set $\boldsymbol{v}_1 := \boldsymbol{v}/\|\boldsymbol{v}\|_2$.
for $j = 1, 2, \ldots, k$ **do**
  Compute $\boldsymbol{w} := (\boldsymbol{A}_m - \tau_j \boldsymbol{B}_m)^{-1} \boldsymbol{B}_m \boldsymbol{v}_j$.
  Orthogonalize $\widehat{\boldsymbol{w}} := \boldsymbol{w} - \sum_{i=1}^{j} \mu_{i,j} \boldsymbol{v}_i$, where $\mu_{i,j} = \boldsymbol{v}_i^* \boldsymbol{w}$.
  Set $\mu_{j+1,j} = \|\widehat{\boldsymbol{w}}\|_2$ and normalize $\boldsymbol{v}_{j+1} := \boldsymbol{w}/\mu_{j+1,j}$.
end

If this algorithm completes without early termination, which is the generic case when all $\mu_{j+1,j} \neq 0$, the computed quantities can be combined into a rational Arnoldi decomposition

$$\boldsymbol{A}_m \boldsymbol{V}_{k+1} \underline{M_k} = \boldsymbol{B}_m \boldsymbol{V}_{k+1} \underline{N_k}, \tag{6.21}$$

where $(\underline{N_k}, \underline{M_k})$ is an $(k+1) \times k$ upper Hessenberg pencil with $\underline{M_k} = [\mu_{i,j}]$ and $\underline{N_k} = \underline{I_k} + \underline{M_k}\mathrm{diag}(\tau_1, \ldots, \tau_k)$. The rational Arnoldi decomposition (6.21) can then be used to extract Ritz pairs $(\vartheta_i, \boldsymbol{w}_i = \boldsymbol{V}_{k+1} \underline{N_k} s_i)$, where (ϑ_i, s_i) are solutions of the generalized eigenproblem

$$N_k s_i = \vartheta_i M_k s_i, \quad s_i \neq 0,$$

involving the upper $k \times k$ part of the pencil $(\underline{N_k}, \underline{M_k})$. Typically, the Ritz pairs are expected to be good approximations to some of the eigenpairs of the linearization $\boldsymbol{L}_m$, in particular, close to the shifts τ_j. In practice, one may distribute the shift parameters inside the target region Σ, run a few rational Arnoldi iterations k, and then compute the residuals of the extracted Ritz pairs. If the residuals are not satisfactory, one can extend the rational Arnoldi decomposition to a larger value of k by continuing the rational Arnoldi iteration.

The most expensive part in Algorithm 6.10 is the solution of a shifted linear system

$$\boldsymbol{L}_m(\tau_j)\boldsymbol{w} = (\boldsymbol{A}_m - \tau_j \boldsymbol{B}_m)\boldsymbol{w} = \boldsymbol{B}_m \boldsymbol{v}_j$$

for $\boldsymbol{w}$, involving the $mn \times mn$ matrices $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$. It turns out that these systems can be solved very efficiently if the underlying Kronecker structure in the lower block part of $\boldsymbol{A}_m - \tau_j \boldsymbol{B}_m$ is exploited; see Theorem 6.7. Let us illustrate this pictorially at a degree $m = 4$ linearization arising from

sampling with a (rational) Newton basis. In this case $\boldsymbol{A}_4 - \tau_j \boldsymbol{B}_4$ has the block sparsity pattern shown on the left of the following equation:

$$\left[\begin{array}{cccc} \times & \times & \times & \times \\ \hline + & + & & \\ & + & + & \\ & & + & + \end{array}\right] = \left[\begin{array}{cccc} \otimes & \times & \times & \times \\ \hline & + & & \\ & & + & \\ & & & + \end{array}\right] \left[\begin{array}{cccc} I & & & \\ \hline + & I & & \\ & + & I & \\ & & + & I \end{array}\right].$$

Here, the first block row contains coefficient matrices of an NEP associated with the Newton basis (symbolized by $\times$), while all blocks below the horizontal line are multiples of the identity matrix (symbolized by $+$). Provided that the block diagonal entries of the matrix on the left are nonzero (which is guaranteed as long as τ_j does not coincide with any of the poles of R_m), we can compute a block UL decomposition as shown on the right of the equation. Hence a linear system solve reduces to a forward and backsubstitution with the matrix factors on the right-hand side of the equation. Luckily, the only matrix to be inverted is the upper left block entry in the left factor of size $n \times n$, symbolized by $\otimes$. Hence, for each distinct shift τ_j, only a single factorization of an $n \times n$ matrix is required. The exploitation of this structure in $\boldsymbol{L}_m(\tau_j)$ is crucial for the efficient solution of the linearized problem, and it has been done, e.g., in (Effenberger and Kressner 2012, Jarlebring, Meerbergen and Michiels 2012*b*, Van Beeumen et al. 2013, Güttel et al. 2014).

Further considerable savings in arithmetic operations for the orthogonalization and storage of the orthonormal rational Krylov basis are possible by exploiting a *compact representation* of the basis vectors

$$\boldsymbol{V}_{k+1} = (I_m \otimes Q)\, \boldsymbol{U}, \tag{6.22}$$

where $Q \in \mathbb{C}^{n \times r}$ and $\boldsymbol{U} \in \mathbb{C}^{mr \times (k+1)}$ have orthonormal columns. The rank r is bounded by $m + k + 1$ and typically much smaller than $m(k+1)$. All operations required in the rational Arnoldi algorithm (matrix-vector products, linear system solves with $\boldsymbol{L}_m(\tau_j)$, and inner products) can be implemented efficiently using this representation. The stability of the compact representation within the two-level orthogonal Arnoldi procedure (TOAR) for quadratic eigenvalue problems is investigated in (Lu, Su and Bai 2016). TOAR has been extended to (polynomial) NEPs in a number of works like, e.g., (Kressner and Roman 2014, Van Beeumen et al. 2015*a*).

Finally, let us clarify that some algorithmic details are still missing in our simplistic presentation starting from Algorithm 6.10. However, most of these details appear in identical form with the Arnoldi solution of linear eigenvalue problems, including the use of reorthogonalization to avoid the loss of orthogonality in the rational Krylov basis $\boldsymbol{V}_{k+1}$, the use of inexact solves (Lehoucq and Meerbergen 1998), and Krylov–Schur restarting to further reduce storage and orthogonalization costs (Stewart 2002). For

further reading we refer to the ARPACK users' guide (Lehoucq, Sorensen and Yang 1998) and to (Ruhe 1998).

6.5. Related work and software

Solution approaches based on approximation or interpolation of the NEP are frequently employed in the literature, in particular by the boundary element method (BEM) community; see (Kamiya, Andoh and Nogae 1993) for a review. Theoretical aspects of such approximations are investigated in (Karma 1996*a*, Karma 1996*b*).

For a given matrix polynomial or rational matrix-valued function R_m there are many (in fact, infinitely many) possible linearizations (see Mackey et al. (2015) and references therein). Our aim in this Section 6.3 was to present a unified and practical approach. For other examples of linearizations based on degree-graded polynomial bases, Bernstein, Lagrange, and Hermite bases, see, e.g., (Corless 2004, Mackey, Mackey, Mehl and Mehrmann 2006*b*, Higham, Mackey, Mackey and Tisseur 2006, Amiraslani, Corless and Lancaster 2009, Van Beeumen, Michiels and Meerbergen 2015*b*, Mackey and Perović 2016, Noferini and Pérez 2016). For constructions of linearizations of rational matrix-valued functions, we refer to (Su and Bai 2011, Alam and Behera 2016).

Van Beeumen et al. (2013, Section 4.5) make a connection of rational Krylov techniques for solving the linearized problem with Newton's method, using a relation between the rational Arnoldi and Jacobi–Davidson algorithms pointed out by Ruhe (1998), and the interpretation of a Jacobi–Davidson iteration as a Newton update (Sleijpen and van der Vorst 1996). A connection between rational Krylov methods for (nonlinear) eigenvalue problems and contour-based methods is discussed in (Van Beeumen, Meerbergen and Michiels 2016*b*).

A practical advancement of linearization-based methods is the so-called *infinite Arnoldi method*, which in its original form by Jarlebring et al. (2012*b*) uses Taylor approximation and has been applied successfully to various NEPs, e.g., those associated with delay differential equations; see also (Jarlebring and Güttel 2014, Jarlebring, Meerbergen and Michiels 2012*a*, Jarlebring, Meerbergen and Michiels 2014). The main idea of the infinite Arnoldi method is to increase the degree m of the linearization $\boldsymbol{L}_m$ dynamically with every outer rational Krylov iteration k for finding its eigenvalues; i.e., $k = m$. This is possible if the starting vector $\boldsymbol{v}$ in Algorithm 6.10 has a particular structure and the interpolation nodes σ_j are chosen identically to the shifts τ_j for all j. The advantage of this approach over the first-sample-then-solve approach is that the degree m of the linearization does not need to be determined in advance. However, the restriction on the sampling points $\sigma_j = \tau_j$ may enforce the use of suboptimal interpolants for the linearization,

and the accuracy of this linearization (degree m) and the convergence of the outer rational Krylov iteration for finding its eigenvalues (index k) are not necessarily synchronous.

Several MATLAB implementations of the infinite Arnoldi method are available online, for example:

- a rank-exploiting variant described in (Van Beeumen et al. 2016*a*): `https://people.kth.se/~eliasj/src/lowranknep`;
- a tensor-version applied to a waveguide eigenvalue problem described in (Jarlebring, Mele and Runborg 2015): `http://www.math.kth.se/~gmele/waveguide`; and
- the bi-Lanczos method of (Gaaf and Jarlebring 2016): `http://www.math.kth.se/~eliasj/src/infbilanczos/`.

Other NEP solvers based on polynomial interpolation are

- a method using empirical interpolation to solve nonlinear Helmholtz eigenvalue problems in (Botchev, Sleijpen and Sopaheluwakan 2009);
- the Chebyshev interpolation approach in (Effenberger and Kressner 2012), which focuses on problems arising from the BEM discretization of 3D elliptic PDE eigenvalue problems and comes with a MATLAB code available at `http://anchp.epfl.ch/files/content/sites/anchp/files/software/chebapprox.tar.gz`;
- a linearization approach of Lagrange and Hermite interpolating matrix polynomials described in (Van Beeumen et al. 2015*b*): `http://twr.cs.kuleuven.be/research/software/nleps/lin-lagr.php`.

MATLAB codes making use of rational Leja–Bagby sampling combined with a Krylov solution of the linearization are

- the NLEIGS method described in (Güttel et al. 2014), which also supports exploitation of low-rank structure in the NEP: `http://twr.cs.kuleuven.be/research/software/nleps/nleigs.php`;
- the CORK method described in (Van Beeumen et al. 2015*a*), which implements the compact representation of Krylov basis vectors (6.22), exploitation of low-rank terms, and implicit restarting: `http://twr.cs.kuleuven.be/research/software/nleps/cork.php`;
- the `util_nleigs` function in the Rational Krylov Toolbox demonstrated in Figure 6.2, as well as the `rat_krylov` function which implements the (parallel) rational Arnoldi algorithm described in (Berljafa and Güttel 2016): `http://rktoolbox.org/`.

The `NEP` module of the SLEPc package (Hernandez et al. 2005) provides various linearization-based solvers for NEPs: `http://slepc.upv.es/`. (There is also a specialized module `PEP` for polynomial eigenvalue problems.) In particular, it implements Chebyshev interpolation on an interval and the

NLEIGS method using rational Leja–Bagby sampling. The eigenpairs of the resulting linearization are computed by a Krylov–Schur implementation that fully exploits the compact representation (6.22); see (Campos and Roman 2016*b*) for implementation details in the case of matrix polynomials.

SLEPc also provides matrix function routines to compute rational interpolants R_m of NEPs given in split form, as suggested by Theorem 6.1.

Acknowledgements. We are grateful to Wolf-Jürgen Beyn, Nick Higham, Daniel Kressner, Volker Mehrmann, and Jose E. Roman for their insightful comments and further references that significantly improved this review. We also thank Mary Aprahamian, Steven Elsworth, and Jennifer Lau for a number of useful remarks. We are thankful to Arieh Iserles for his patience with handling our manuscript, and to Glennis Starling for the editing.

REFERENCES

L. V. Ahlfors (1953), *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*, McGraw-Hill, New York.

M. Al-Ammari and F. Tisseur (2012), 'Hermitian matrix polynomials with real eigenvalues of definite type. Part I: Classification', *Linear Algebra Appl.* **436**(10), 3954–3973.

R. Alam and N. Behera (2016), 'Linearizations for rational matrix functions and Rosenbrock system polynomials', *SIAM J. Matrix Anal. Appl.* **37**(1), 354–380.

A. Amiraslani, R. M. Corless and P. Lancaster (2009), 'Linearization of matrix polynomials expressed in polynomial bases', *IMA J. Numer. Anal.* **29**, 141–157.

A. L. Andrew, K. E. Chu and P. Lancaster (1993), 'Derivatives of eigenvalues and eigenvectors of matrix functions', *SIAM J. Matrix Anal. Appl.* **14**(4), 903–926.

A. L. Andrew, K. E. Chu and P. Lancaster (1995), 'On the numerical solution of nonlinear eigenvalue problems', *Computing* **55**, 91–111.

P. M. Anselone and L. B. Rall (1968), 'The solution of characteristic value-vector problems by Newton's method', *Numer. Math.* **11**(1), 38–45.

P. Arbenz and W. Gander (1986), 'Solving nonlinear eigenvalue problems by algorithmic differentiation', *Computing* **36**, 205–215.

J. Asakura, T. Sakurai, H. Tadano, T. Ikegami and K. Kimura (2009), 'A numerical method for nonlinear eigenvalue problems using contour integral', *JSIAM Letters* **1**, 52–55.

A. P. Austin, P. Kravanja and L. N. Trefethen (2014), 'Numerical algorithms based on analytic function values at roots of unity', *SIAM J. Numer. Anal.* **52**(4), 1795–1821.

T. Bagby (1967), 'The modulus of a plane condenser', *J. Math. Mech.* **17**, 315–329.

T. Bagby (1969), 'On interpolation by rational functions', *Duke Math. J.* **36**, 95–104.

Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe and H. A. van der Vorst, eds (2000), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang and H. Zhang (2016), PETSc users manual, Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory.

M. Berljafa and S. Güttel (2014), A Rational Krylov Toolbox for MATLAB, MIMS EPrint 2014.56, Manchester Institute for Mathematical Sciences, The University of Manchester, UK. Available for download at `http://rktoolbox.org/`.

M. Berljafa and S. Güttel (2015), 'Generalized rational Krylov decompositions with an application to rational approximation', *SIAM J. Matrix Anal. Appl.* **36**(2), 894–916.

M. Berljafa and S. Güttel (2016), Parallelization of the rational Arnoldi algorithm, MIMS EPrint 2016.32, Manchester Institute for Mathematical Sciences, The University of Manchester, UK. To appear in SIAM J. Sci. Comput.

T. Betcke and H. Voss (2004), 'A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems', *Future Generation Computer Systems* **20**(3), 363–372.

T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder and F. Tisseur (2013), 'NLEVP: A collection of nonlinear eigenvalue problems', *ACM Trans. Math. Software* **39**(2), 7:1–7:28.

W.-J. Beyn (2012), 'An integral method for solving nonlinear eigenvalue problems', *Linear Algebra Appl.* **436**(10), 3839–3863.

W.-J. Beyn and V. Thümmler (2009), 'Continuation of invariant subspaces for parameterized quadratic eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **31**(3), 1361–1381.

W.-J. Beyn, C. Effenberger and D. Kressner (2011), 'Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems', *Numer. Math.* **119**(3), 489–516.

D. Bindel and A. Hood (2013), 'Localization theorems for nonlinear eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **34**(4), 1728–1749.

H.-P. Blatt, E. Saff and M. Simkani (1988), 'Jentzsch–Szegö type theorems for the zeros of best approximants', *J. London Math. Soc.* **2**(2), 307–316.

M. Botchev, G. Sleijpen and A. Sopaheluwakan (2009), 'An SVD-approach to Jacobi–Davidson solution of nonlinear Helmholtz eigenvalue problems', *Linear Algebra Appl.* **431**(3), 427–440.

L. Botten, M. Craig and R. McPhedran (1983), 'Complex zeros of analytic functions', *Comput. Phys. Commun.* **29**(3), 245–259.

T. Bühler and M. Hein (2009), Spectral clustering based on the graph p-Laplacian, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, pp. 81–88.

R. Byers, V. Mehrmann and H. Xu (2008), 'Trimmed linearizations for structured matrix polynomials', *Linear Algebra Appl.* **429**(10), 2373–2400.

C. Campos and J. E. Roman (2016*a*), 'Parallel iterative refinement in polynomial eigenvalue problems', *Numer. Linear Algebra Appl.* **23**(4), 730–745.

C. Campos and J. E. Roman (2016*b*), 'Parallel Krylov solvers for the polynomial eigenvalue problem in SLEPc', *SIAM J. Sci. Comput.* **38**(5), S385–S411.

R. M. Corless (2004), Generalized companion matrices in the Lagrange basis, in *Proceedings EACA* (L. Gonzalez-Vega and T. Recio, eds), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 317–322.

R. Courant (1920), 'Über die Eigenwerte bei den Differentialgleichungen der mathematischen Physik', *Math. Z.* **7**, 1–57.

E. R. Davidson (1975), 'The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices', *J. Comput. Phys.* **17**, 87–94.

P. J. Davis and P. Rabinowitz (2007), *Methods of Numerical Integration*, Courier Corporation.

T. A. Davis (2004), 'Algorithm 832: UMFPACK V4.3—An unsymmetric-pattern multifrontal method', *ACM Trans. Math. Software* **30**(2), 196–199.

L. Delves and J. Lyness (1967), 'A numerical method for locating the zeros of an analytic function', *Math. Comp.* **21**(100), 543–560.

N. H. Du, V. H. Linh, V. Mehrmann and D. D. Thuan (2013), 'Stability and robust stability of linear time-invariant delay differential-algebraic equations', *SIAM J. Matrix Anal. Appl.* **34**(4), 1631–1654.

R. J. Duffin (1955), 'A minimax theory for overdamped networks', *J. Rat. Mech. Anal.* **4**, 221–233.

J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring (2016), *GNU Octave version 4.2.0 manual: a high-level interactive language for numerical computations*.

C. Effenberger (2013*a*), Robust solution methods for nonlinear eigenvalue problems, PhD thesis, EPFL, Lausanne, Switzerland.

C. Effenberger (2013*b*), 'Robust successive computation of eigenpairs for nonlinear eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **34**(3), 1231–1256.

C. Effenberger and D. Kressner (2012), 'Chebyshev interpolation for nonlinear eigenvalue problems', *BIT* **52**(4), 933–951.

W. R. Ferng, W.-W. Lin, D. J. Pierce and C.-S. Wang (2001), 'Nonequivalence transformation of λ-matrices eigenproblems and model embedding approach to model tuning', *Numer. Linear Algebra Appl.* **8**(1), 53–70.

E. Fischer (1905), 'Über quadratische Formen mit reellen Koeffizienten', *Monatshefte für Mathematik und Physik* **16**, 234–249.

R. W. Freund and N. M. Nachtigal (1996), 'QMRPACK: A package of QMR algorithms', *ACM Trans. Math. Software* **22**(1), 46–77.

S. W. Gaaf and E. Jarlebring (2016), 'The infinite bi-Lanczos method for nonlinear eigenvalue problems', *arXiv preprint arXiv:1607.03454*.

W. Gander, M. J. Gander and F. Kwok (2014), *Scientific Computing – An Introduction using Maple and MATLAB*, Springer-Verlag, New York.

C. K. Garrett and R.-C. Li (2013), 'Unstructurally banded nonlinear. eigenvalue solver software'. `http://www.csm.ornl.gov/newsite/software.html`.

C. K. Garrett, Z. Bai and R.-C. Li (2016), 'A nonlinear QR algorithm for banded nonlinear eigenvalue problems', *ACM Trans. Math. Software* **43**(1), 4:1–4:19.

I. Gohberg and L. Rodman (1981), 'Analytic matrix functions with prescribed local data', *J. Anal. Math.* **40**(1), 90–128.

I. Gohberg, M. Kaashoek and F. van Schagen (1993), 'On the local theory of regular analytic matrix functions', *Linear Algebra Appl.* **182**, 9–25.

I. Gohberg, P. Lancaster and L. Rodman (2009), *Matrix Polynomials*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. Unabridged republication of book first published by Academic Press in 1982.

A. A. Gonchar (1969), 'Zolotarev problems connected with rational functions', *Math. USSR Sb.* **7**, 623–635.

L. Grammont, N. J. Higham and F. Tisseur (2011), 'A framework for analyzing nonlinear eigenproblems and parametrized linear systems', *Linear Algebra Appl.* **435**(3), 623–640.

A. Griewank and A. Walther (2008), *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

K. Gu, V. L. Kharitonov and J. Chen (2003), *Stability of Time-Delay Systems*, Springer Science & Business Media, New-York.

S. Güttel (2013), 'Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection', *GAMM-Mitt.* **36**(1), 8–31.

S. Güttel, R. Van Beeumen, K. Meerbergen and W. Michiels (2014), 'NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems', *SIAM J. Sci. Comput.* **36**(6), A2842–A2864.

K. P. Hadeler (1967), 'Mehrparametrige und nichtlineare Eigenwertaufgaben', *Arch. Rational Mech. Anal.* **27**(4), 306–328.

K. P. Hadeler (1968), 'Variationsprinzipien bei nichtlinearen Eigenwertaufgaben', *Arch. Rational Mech. Anal.* **30**, 297–307.

N. Hale, N. J. Higham and L. N. Trefethen (2008), 'Computing A^α, $\log(A)$, and related matrix functions by contour integrals', *SIAM J. Numer. Anal.* **46**(5), 2505–2523.

V. Hernandez, J. E. Roman and V. Vidal (2005), 'SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems', *ACM Transactions on Mathematical Software (TOMS)* **31**(3), 351–362.

V. Hernandez, J. Roman, A. Tomas and V. Vidal (2009), A survey of software for sparse eigenvalue problems, Technical Report SLEPc Technical Report STR-6, Universidad Politecnica de Valencia, Valencia, Spain.

D. J. Higham and N. J. Higham (2017), *MATLAB Guide*, third edn, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

N. J. Higham (2008), *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

N. J. Higham and F. Tisseur (2002), 'More on pseudospectra for polynomial eigenvalue problems and applications in control theory', *Linear Algebra Appl.* **351–352**, 435–453.

N. J. Higham, D. S. Mackey, N. Mackey and F. Tisseur (2006), 'Symmetric linearizations for matrix polynomials', *SIAM J. Matrix Anal. Appl.* **29**(1), 143–159.

M. E. Hochstenbach and G. L. Sleijpen (2003), 'Two-sided and alternating Jacobi–Davidson', *Linear Algebra Appl.* **358**(1), 145–172.

R. A. Horn and C. R. Johnson (1985), *Matrix Analysis*, Cambridge University Press, Cambridge, UK.

R. A. Horn and C. R. Johnson (1991), *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK.

HSL (2016), 'A collection of Fortran codes for large-scale scientific computation'. http://www.hsl.rl.ac.uk/.

T.-M. Huang, W.-W. Lin and V. Mehrmann (2016), 'A Newton-type method with nonequivalence deflation for nonlinear eigenvalue problems arising in photonic crystal modeling', *SIAM J. Sci. Comput.* **38**(2), B191–B218.

N. I. Ioakimidis (1987), Quadrature methods for the determination of zeros of transcendental functions—a review, in *Numerical Integration: Recent Developments, Software and Applications*, Springer-Verlag, Dordrecht, pp. 61–82.

I. C. F. Ipsen (1997), 'Computing an eigenvector with inverse iteration', *SIAM Rev.* **39**(2), 254–291.

E. Jarlebring and S. Güttel (2014), 'A spatially adaptive iterative method for a class of nonlinear operator eigenproblems', *Electron. Trans. Numer. Anal.* **41**, 21–41.

E. Jarlebring, K. Meerbergen and W. Michiels (2012*a*), The infinite Arnoldi method and an application to time-delay systems with distributed delay, in *Time Delay Systems: Methods, Applications and New Trends* (R. Sipahi, T. Vyhlídal, S.-I. Niculescu and P. Pepe, eds), Vol. 423 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, Berlin, pp. 229–239.

E. Jarlebring, K. Meerbergen and W. Michiels (2012*b*), 'A linear eigenvalue algorithm for the nonlinear eigenvalue problem', *Numer. Math.* **122**(1), 169–195.

E. Jarlebring, K. Meerbergen and W. Michiels (2014), 'Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method', *SIAM J. Matrix Anal. Appl.* **35**(2), 411–436.

E. Jarlebring, G. Mele and O. Runborg (2015), The waveguide eigenvalue problem and the tensor infinite Arnoldi method, Technical Report arXiv:1503.02096v2, KTH Stockholm.

R. Jentzsch (1916), 'Untersuchungen zur Theorie der Folgen analytischer Funktionen', *Acta Math.* **41**(1), 219–251.

N. Kamiya, E. Andoh and K. Nogae (1993), 'Eigenvalue analysis by the boundary element method: new developments', *Eng. Anal. Bound. Elem.* **12**(3), 151–162.

O. Karma (1996*a*), 'Approximation in eigenvalue problems for holomorphic Fredholm operator functions I', *Numer. Funct. Anal. Optim.* **17**(3-4), 365–387.

O. Karma (1996*b*), 'Approximation in eigenvalue problems for holomorphic Fredholm operator functions II (convergence rate)', *Numer. Funct. Anal. Optim.* **17**(3-4), 389–408.

J. P. Keener (1993), 'The Perron–Frobenius theorem and the ranking of football teams', *SIAM Rev.* **35**(1), 80–93.

V. B. Khazanov and V. N. Kublanovskaya (1988), 'Spectral problems for matrix pencils: methods and algorithms. II', *Sov. J. Numer. Anal. Math. Modelling* **3**, 467–485.

A. Kimeswenger, O. Steinbach and G. Unger (2014), 'Coupled finite and boundary element methods for fluid-solid interaction eigenvalue problems', *SIAM J. Numer. Anal.* **52**(5), 2400–2414.

V. Kozlov and V. G. Maz'ja (1999), *Differential Equations with Operator Coefficients*, Springer Monographs in Mathematics, Springer, Berlin.

S. G. Krantz (1982), *Function Theory of Several Complex Variables*, Wiley, New York.

P. Kravanja, T. Sakurai and M. Van Barel (1999*a*), 'On locating clusters of zeros of analytic functions', *BIT* **39**(4), 646–682.

P. Kravanja, M. Van Barel and A. Haegemans (1999*b*), 'On computing zeros and poles of meromorphic functions', *Series in Approximations and Decompositions* **11**, 359–370.

P. Kravanja, M. Van Barel, O. Ragos, M. Vrahatis and F. Zafiropoulos (2000), 'ZEAL: A mathematical software package for computing zeros of analytic functions', *Comput. Phys. Commun.* **124**(2), 212–232.

D. Kressner (2009), 'A block Newton method for nonlinear eigenvalue problems', *Numer. Math.* **114**(2), 355–372.

D. Kressner and J. E. Roman (2014), 'Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis', *Numer. Linear Algebra Appl.* **21**(4), 569–588.

V. N. Kublanovskaja (1969), 'On an application of Newton's method to the determination of eigenvalues of λ-matrices', *Soviet Math. Dokl.* **10**, 1240–1241.

V. N. Kublanovskaya (1970), 'On an approach to the solution of the generalized latent value problem for λ-matrices', *SIAM J. Numer. Anal.* **7**, 532–537.

P. Lancaster (1961), 'A generalised Rayleigh quotient iteration for lambda-matrices', *Arch. Rational Mech. Anal.* **8**(1), 309–322.

P. Lancaster (1966), *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford. Reprinted by Dover, New York, 2002.

A. Leblanc and A. Lavie (2013), 'Solving acoustic nonlinear eigenvalue problems with a contour integral method', *Engineering Analysis with Boundary Elements* **37**(1), 162–166.

R. Lehoucq and K. Meerbergen (1998), 'Using generalized Cayley transformations within an inexact rational Krylov sequence method', *SIAM J. Matrix Anal. Appl.* **20**(1), 131–148.

R. B. Lehoucq, D. C. Sorensen and C. Yang (1998), *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

J. Leiterer (1978), 'Local and global equivalence of meromorphic operator functions part II', *Mathematische Nachrichten* **84**(1), 145–170.

E. Levin and E. B. Saff (2006), Potential theoretic tools in polynomial and rational approximation, in *Harmonic Analysis and Rational Approximation* (J.-D. Fournier et al., ed.), Vol. 327 of *Lecture Notes in Control and Information Sciences*, Springer, Berlin, pp. 71–94.

B.-S. Liao, Z. Bai, L.-Q. Lee and K. Ko (2010), 'Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems', *Taiwanese J. Math.* **14**, 869–883.

D. Lu, X. Huang, Z. Bai and Y. Su (2015), 'A Padé approximate linearization algorithm for solving the quadratic eigenvalue problem with low-rank damping', *Internat. J. Numer. Methods Eng.* **103**(11), 840–858.

D. Lu, Y. Su and Z. Bai (2016), 'Stability analysis of the two-level orthogonal Arnoldi procedure', *SIAM J. Matrix Anal. Appl.* **37**(1), 195–214.

D. S. Mackey and V. Perović (2016), 'Linearizations of matrix polynomials in Bernstein bases', *Linear Algebra Appl.* **501**, 162–197.

D. S. Mackey, N. Mackey and F. Tisseur (2015), Polynomial eigenvalue problems: Theory, computation, and structure, in *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory* (P. Benner, M. Bollhöfer, D. Kressner, C. Mehl and T. Stykel, eds), Springer-Verlag, New York, pp. 319–348.

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

D. S. Mackey, N. Mackey, C. Mehl and V. Mehrmann (2006*b*), 'Vector spaces of linearizations for matrix polynomials', *SIAM J. Matrix Anal. Appl.* **28**(4), 971–1004.

Y. Maeda, T. Sakurai and J. Roman (2016), Contour integral spectrum slicing method in SLEPc, Technical Report SLEPc Technical Report STR-11, Universidad Politecnica de Valencia, Valencia, Spain.

V. Mehrmann and H. Voss (2004), 'Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods', *GAMM-Mitt.* **27**, 121–152.

R. Mennicken and M. Möller (2003), *Non-Self-Adjoint Boundary Eigenvalue Problems*, Vol. 192, Elsevier Science B. V., Amsterdam, The Netherlands.

W. Michiels and N. Guglielmi (2012), 'An iterative method for computing the pseudospectral abscissa for a class of nonlinear eigenvalue problems', *SIAM J. Sci. Comput.* **34**(4), A2366–A2393.

W. Michiels and S.-I. Niculescu (2007), *Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

W. Michiels, K. Green, T. Wagenknecht and S.-I. Niculescu (2006), 'Pseudospectra and stability radii for analytic matrix functions with application to time-delay systems', *Linear Algebra Appl.* **418**(1), 315–335.

MUMPS (2016), 'MUltifrontal Massively Parallel Solver: Users' guide'. `http://mumps.enseeiht.fr/`.

Z. Nehari (1975), *Conformal Mapping*, Dover, New York. Unabridged and unaltered republication of the work originally published by the McGraw-Hill Book Company, Inc. in 1952.

A. Neumaier (1985), 'Residual inverse iteration for the nonlinear eigenvalue problem', *SIAM J. Numer. Anal.* **22**(5), 914–923.

V. Niendorf and H. Voss (2010), 'Detecting hyperbolic and definite matrix polynomials', *Linear Algebra Appl.* **432**, 1017–1035.

V. Noferini and J. Pérez (2016), 'Fiedler-comrade and Fiedler–Chebyshev pencils', *SIAM J. Matrix Anal. Appl.* **37**(4), 1600–1624.

G. Opitz (1964), 'Steigungsmatrizen', *Z. Angew. Math. Mech.* **44**, T52–T54.

G. Peters and J. H. Wilkinson (1979), 'Inverse iteration, ill-conditioned equations and Newton's method', *SIAM Rev.* **21**, 339–360.

H. Poincaré (1890), 'Sur les équations aux dérivées partielles de la physique mathématique', *Amer. J. Math.* pp. 211–294.

E. H. Rogers (1964), 'A minimax theory for overdamped systems', *Arch. Rational Mech. Anal.* **16**, 89–96.

A. Ruhe (1973), 'Algorithms for the nonlinear eigenvalue problem', *SIAM J. Numer. Anal.* **10**, 674–689.

A. Ruhe (1998), 'Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils', *SIAM J. Sci. Comput.* **19**(5), 1535–1551.

Y. Saad (2011), *Numerical Methods for Large Eigenvalue Problems*, revised edn, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Y. Saad and M. H. Schultz (1986), 'GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Statist. Comput.* **7**(3), 856–869.

Y. Saad, A. Stathopoulos, J. Chelikowsky, K. Wu and S. Öğüt (1996), 'Solution of large eigenvalue problems in electronic structure calculations', *BIT* **36**(3), 563–578.

T. Sakurai and H. Sugiura (2003), 'A projection method for generalized eigenvalue problems using numerical integration', *J. Comput. Appl. Math.* **159**(1), 119–128.

O. Schenk and K. Gärtner (2004), 'Solving unsymmetric sparse systems of linear equations with PARDISO', *Future Gener. Comput. Syst.* **20**(3), 475–487. `http://www.pardiso-project.org/`.

K. Schreiber (2008), Nonlinear Eigenvalue Problems: Newton-type methods and Nonlinear Rayleigh Functionals, PhD thesis, Technischen Universitat Berlin, Germany. Avalaible at `http://www.math.tu-berlin.de/~schreibe/`.

H. Schwetlick and K. Schreiber (2012), 'Nonlinear Rayleigh functionals', *Linear Algebra Appl.* **436**(10), 3991–4016.

G. L. G. Sleijpen and H. A. van der Vorst (1996), 'A Jacobi–Davidson iteration method for linear eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **17**(2), 401–425.

G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema and H. A. van der Vorst (1996), 'Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems', *BIT* **36**(3), 595–633.

S. I. Solov′ëv (2006), 'Preconditioned iterative methods for a class of nonlinear eigenvalue problems', *Linear Algebra Appl.* **415**, 210–229.

A. Spence and C. Poulton (2005), 'Photonic band structure calculations using nonlinear eigenvalue techniques', *J. Comput. Phys.* **204**(1), 65–81.

H. Stahl (1996), 'Convergence of rational interpolants', *Bull. Belg. Math. Soc. Simon Stevin* **3**(5), 11–32.

G. W. Stewart (2002), 'A Krylov–Schur algorithm for large eigenproblems', *SIAM J. Matrix Anal. Appl.* **23**(3), 601–614.

Y. Su and Z. Bai (2011), 'Solving rational eigenvalue problems via linearization', *SIAM J. Matrix Anal. Appl.* **32**(1), 201–216.

D. B. Szyld and F. Xue (2013*a*), 'Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems', *Numer. Math.* **123**, 333–362.

D. B. Szyld and F. Xue (2013*b*), 'Several properties of invariant pairs of nonlinear algebraic eigenvalue problems', *IMA J. Numer. Anal.* **34**(3), 921–954.

D. B. Szyld and F. Xue (2015), 'Local convergence of Newton-like methods for degenerate eigenvalues of nonlinear eigenproblems. I. classical algorithms', *Numer. Math.* **129**(2), 353–381.

D. B. Szyld and F. Xue (2016), 'Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. I. Extreme eigenvalues', *Math. Comp.* **85**(302), 2887–2918.

F. Tisseur (2000), 'Backward error and condition of polynomial eigenvalue problems', *Linear Algebra Appl.* **309**, 339–361.

F. Tisseur and N. J. Higham (2001), 'Structured pseudospectra for polynomial eigenvalue problems, with applications', *SIAM J. Matrix Anal. Appl.* **23**(1), 187–208.

F. Tisseur and K. Meerbergen (2001), 'The quadratic eigenvalue problem', *SIAM Rev.* **43**(2), 235–286.

L. N. Trefethen and M. Embree (2005), *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, USA.

L. N. Trefethen and J. Weideman (2014), 'The exponentially convergent trapezoidal rule', *SIAM Rev.* **56**(3), 385–458.

V. Trofimov (1968), 'The root subspaces of operators that depend analytically on a parameter', *Mat. Issled* **3**(9), 117–125.

H. Unger (1950), 'Nichtlineare Behandlung von Eigenwertaufgaben', *ZAMM Z. Angew. Math. Mech.* **30**(8-9), 281–282.

M. Van Barel (2016), 'Designing rational filter functions for solving eigenvalue problems by contour integration', *Linear Algebra Appl.* **502**, 346–365.

M. Van Barel and P. Kravanja (2016), 'Nonlinear eigenvalue problems and contour integrals', *J. Comput. Appl. Math.* **292**, 526–540.

R. Van Beeumen (2015), Rational Krylov Methods for Nonlinear Eigenvalue Problems, PhD thesis, KU Leuven, Leuven, Belgium.

R. Van Beeumen, E. Jarlebring and W. Michiels (2016*a*), 'A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems', *Numer. Linear Algebra Appl.* **23**(4), 607–628.

R. Van Beeumen, K. Meerbergen and W. Michiels (2013), 'A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems', *SIAM J. Sci. Comput.* **35**(1), A327–A350.

R. Van Beeumen, K. Meerbergen and W. Michiels (2015*a*), 'Compact rational Krylov methods for nonlinear eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **36**(2), 820–838.

R. Van Beeumen, K. Meerbergen and W. Michiels (2016*b*), Connections between contour integration and rational Krylov methods for eigenvalue problems, Technical Report TW673, Department of Computer Science, KU Leuven, Belgium.

R. Van Beeumen, W. Michiels and K. Meerbergen (2015*b*), 'Linearization of Lagrange and Hermite interpolating matrix polynomials', *IMA J. Numer. Anal.* **35**(2), 909–930.

H. A. van der Vorst (1992), 'Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems', *SIAM J. Sci. Statist. Comput.* **13**(2), 631–644.

D. Verhees, R. Van Beeumen, K. Meerbergen, N. Guglielmi and W. Michiels (2014),

'Fast algorithms for computing the distance to instability of nonlinear eigenvalue problems, with application to time-delay systems', *Int. J. Dynam. Control* **2**(2), 133–142.

H. Voss (2004*a*), 'An Arnoldi method for nonlinear eigenvalue problems', *BIT* **44**, 387–401.

H. Voss (2004*b*), Eigenvibrations of a plate with elastically attached loads, in *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskylä, Finland* (P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux and D. Knörzer, eds). `http://www.mit.jyu.fi/eccomas2004/proceedings/proceed.html`.

H. Voss (2007), 'A Jacobi–Davidson method for nonlinear and nonsymmetric eigenproblems', *Computers and Structures* **85**, 1284–1292.

H. Voss (2009), 'A minmax principle for nonlinear eigenproblems depending continuously on the eigenparameter', *Numer. Linear Algebra Appl.* **16**, 899–913.

H. Voss (2014), Nonlinear eigenvalue problems, in *Handbook of Linear Algebra* (L. Hogben, ed.), second edn, Chapman and Hall/CRC, Boca Raton, FL, USA, pp. 115:1–115:24.

H. Voss and B. Werner (1982), 'A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems', *Math. Meth. Appl. Sci.* **4**, 415–424.

J. L. Walsh (1932), 'On interpolation and approximation by rational functions with preassigned poles', *Trans. Amer. Math. Soc.* **34**(1), 22–74.

J. L. Walsh (1935), *Interpolation and Approximation by Rational Functions in the Complex Domain*, Vol. 20, American Mathematical Society Colloquium Publications.

B. Werner (1970), Das Spektrum von Operatorenscharen mit verallgemeinerten Rayleighquotienten, PhD thesis, Universität Hamburg, Hamburg, Germany.

H. Weyl (1912), 'Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung)', *Math. Ann.* **71**, 441–479.

C. Wieners and J. Xin (2013), 'Boundary element approximation for Maxwell's eigenvalue problem', *Math. Methods Appl. Sci.* **36**(18), 2524–2539.

J. H. Wilkinson (1965), *The Algebraic Eigenvalue Problem*, Oxford University Press.

R. Wobst (1987), 'The generalized eigenvalue problem and acoustic surface wave computations', *Computing* **39**, 57–69.

J. Xiao, C. Zhang, T.-M. Huang and T. Sakurai (2016*a*), 'Solving large-scale nonlinear eigenvalue problems by rational interpolation and resolvent sampling based Rayleigh–Ritz method', *To appear in Internat. J. Numer. Methods Eng.*

J. Xiao, H. Zhou, C. Zhang and C. Xu (2016*b*), 'Solving large-scale finite element nonlinear eigenvalue problems by resolvent sampling based Rayleigh–Ritz method', *Comput. Mech.* pp. 1–18.

W. H. Yang (1983), 'A method for eigenvalues of sparse λ-matrices', *Internat. J. Numer. Methods Eng.* **19**, 943–948.

S. Yokota and T. Sakurai (2013), 'A projection method for nonlinear eigenvalue problems using contour integrals', *JSIAM Letters* **5**, 41–44.