Numerical Algorithms
https://doi.org/10.1007/s11075-025-02021-z

RESEARCH

On computing the zeros of Laguerre–Sobolev polynomials

T. Laudadio[1] · N. Mastronardi[1] · F. J. Marcellán Español[2] · N. Van Buggenhout[2] · P. Van Dooren[3]

Received: 21 November 2024 / Accepted: 29 January 2025

Abstract
In this work, we focus on the computation of the zeros of a monic Laguerre–Sobolev orthogonal polynomial of degree n. Taking into account the associated four–term recurrence relation, this problem can be formulated as a generalized eigenvalue problem, involving a lower bidiagonal matrix and a 2–banded lower Hessenberg matrix of order n. Unfortunately, the considered generalized eigenvalue problem is very ill–conditioned, and classical balancing procedures do not improve it. Therefore, customary techniques for solving the generalized eigenvalue problem, like the QZ method, yield unreliable results. Here, we propose a novel balancing procedure that drastically reduces the ill–conditioning of the eigenvalues of the involved matrix pencil. Moreover, we propose a fast and reliable algorithm, with $\mathcal{O}(n^2)$ computational complexity and $\mathcal{O}(n)$ memory, exploiting the structure of the considered matrix pencil.

Keywords Laguerre–Sobolev orthogonal polynomials · Zeros of polynomials · Generalized eigenvalue problem

Mathematics Subject Classification (2010) 33C47 · 65D32 · 65F15

1 Introduction

Given two probability measures ν_0 and ν_1 supported on the real line, the inner product

$$\langle f, g\rangle_{\mathfrak{S}} = \int f(x)g(x)d\nu_0(x) + s\int f'(x)g'(x)d\nu_1(x), \quad s > 0, \tag{1}$$

Dedicated to Michela Redivo Zaglia on the occasion of her retirement, and to Hassan Sadok for his 65th anniversary.

T. Laudadio, N. Mastronardi, F. J. Marcellán Español, N. Van Buggenhout, and P. Van Dooren contributed equally to this work.

Extended author information available on the last page of the article

Published online: 08 February 2025

defined in the linear space of polynomials, is known to be very important from theoretical and practical points of view. In the literature it is known as a weighted Sobolev inner product. By using the Gram–Schmidt orthogonalization method for the monomial basis $\{x^n\}_{n\geq 0}$, a sequence of monic orthogonal polynomials (MOPs, in short) with respect to such an inner product can be generated. This sequence of polynomials is called a Sobolev orthogonal polynomial sequence and in recent years it has received the attention of many researchers (see, the survey paper [22] and the references therein). Numerical methods for the construction of such a polynomial sequence are given in [9, 25]. These are general purpose methods and do not exploit additional properties a Sobolev inner product might have.

In [14] some particular pairs of measures are introduced in connection with the Fourier expansions of functions with respect to the above Sobolev inner product. The name coherent pairs of positive measures is introduced in [13], where the concepts of coherent and symmetric coherent pairs of measures supported on the real line are described.

A pair of positive measures $\{\nu_0, \nu_1\}$ is said to be a coherent pair of positive measures supported on the real line if the corresponding sequences of MOPs satisfy

$$P_n(\nu_1; x) = \frac{1}{n+1}\left[P'_{n+1}(\nu_0; x) - \rho_{n-1} P'_n(\nu_0; x)\right], \ \rho_{n-1} \neq 0, \ n \geq 2. \quad (2)$$

A pair of positive and symmetric measures $\{\nu_0, \nu_1\}$ is said to be a symmetric coherent pair of positive measures on the real line if the corresponding sequences of MOPs satisfy

$$P_n(\nu_1; x) = \frac{1}{n+1}\left[P'_{n+1}(\nu_0; x) - \varrho_{n-1} P'_{n-1}(\nu_0; x)\right], \ \varrho_{n-1} \neq 0, \ n \geq 2. \quad (3)$$

In case $\nu_0 = \nu_1$, the measure ν_0 is called *self–coherent* [22]. All coherent (resp. symmetric) pairs of positive measures on the real line were completely determined by H. G. Meijer in 1997 [23]. Indeed, if $\{\nu_0, \nu_1\}$ is a coherent pair of measures on the real line, then one of the measures must be classical (either Jacobi or Laguerre) and the other one is a rational perturbation of it. In particular, if $d\nu_0 = d\nu_1 = e^{-x}x^\alpha$, $\alpha > -1$, is the absolutely continuous measure associated with the gamma distribution supported in $(0, +\infty)$, the MOPs studied in [21] are obtained. In the literature, they are known as Laguerre–Sobolev orthogonal polynomials and all their zeros are real. Moreover, if $\{\nu_0, \nu_1\}$ is a symmetric coherent pair of measures on the real line, then one of the measures must be classical (either Gegenbauer or Hermite) and the other one is a rational perturbation of it. In particular, when $d\nu_0 = d\nu_1$ is the Lebesgue measure supported in $(-1, 1)$, the Althammer MOPs studied in [3] are obtained, which are also called Legendre–Sobolev orthogonal polynomials.

Sobolev orthogonal polynomials appear in a natural way in the study of spectral methods for boundary value problems for elliptic Ordinary Differential Equations when dealing with the variational formulation of such problems. The competitive advantages of using Sobolev orthogonal polynomials, instead of the standard orthogonal ones, has been pointed out in recent contributions ([2, 7, 8, 26] in the Jacobi case, and [20] in the Hermite case).

It is well known that the zeros of an orthogonal polynomial sequence with respect to a probability measure supported on an infinite subset E of the real line are real and simple, interlace and are located in the interior of the convex hull of E. As a consequence of the fact that such an orthogonal polynomial sequence satisfies a three–term recurrence relation (TTRR, in short), the zeros of the polynomial of degree n are the eigenvalues of a tridiagonal matrix of order n and, thus, they can be computed by using the classical Golub–Welsch algorithm [10]. Unfortunately, such a kind of recurrence relation does not hold for a sequence $\{S_n(x)\}_{n\geq 0}$ of monic Sobolev orthogonal polynomials, which satisfy a higher order recurrence relation $xS_n(x) = S_{n+1}(x) + \sum_{k=0}^{n} c_{n,k} S_k(x)$, $n \geq 0$. Consequently, the zeros of S_n are the eigenvalues of a lower Hessenberg matrix having the entries in the first superdiagonal equal to 1.

Very recently, the problem of computing the zeros of $S_n(x)$ has been formulated as a Hessenberg inverse eigenvalue problem [25]. Two methods are proposed to solve it. The first one is based on the Arnoldi iteration with full reorthogonalization. The other one is based on a procedure making use of plane rotations. Besides this formulation, an alternative method to compute the zeros of Legendre–Sobolev orthogonal polynomials is analyzed in [17], where the Hessenberg matrix is first transformed into a similar symmetric tridiagonal one, and then the zeros are computed as the eigenvalues of the latter tridiagonal matrix.

In this contribution, we propose an algorithm to compute the zeros of the monic Laguerre–Sobolev orthogonal polynomial of degree n as the eigenvalues of a generalized eigenvalue problem associated with the pencil $xB_n - A_n$, where B_n and A_n are lower bidiagonal and two–banded lower Hessenberg matrices of order n, respectively. The Laguerre–Sobolev inner product is a coherent pair, and therefore we can obtain the entries of these matrices from a four–term recurrence relation. Since the considered generalized eigenvalue problem is very ill–conditioned, inspired by [15, 16], a balancing technique is applied transforming the problem into a well–conditioned one.

The complexity and memory properties of the proposed algorithm are studied, and some illustrative examples are provided.

2 Notations and definitions

Matrices are denoted by upper–case letters $A, B, \ldots$; vectors with bold lower–case letters $\mathbf{x}, \mathbf{y}, \ldots, \boldsymbol{\omega}, \ldots$; scalars with lower–case letters $x, y, \ldots, \lambda, \theta, \ldots$.

Matrices of size (m, n) are denoted by $H_{m,n}$ or simply by H_m if $m = n$. The entry (i, j) of a matrix A is generally denoted by $a_{i,j}$ and the ith entry of a vector $\mathbf{x}$ is denoted by x_i, if not explicitly defined.

Submatrices are denoted by the colon notation of `Matlab`: $A(i : j, k : l)$ denotes the submatrix of A formed by the intersection of rows i to j and columns k to l, and $A(i : j, :)$ and $A(:, k : l)$ denote the rows of A from i to j and the columns of A and from k to l, respectively. The function `triu`$(A, -1)$ sets to zero the lower-triangular part below the first subdiagonal of the matrix A.

The symbol `Inf` denotes a real value that is too large to be represented as a floating point number.

The identity matrix of order n is denoted by I_n, and its ith column, $i = 1, \ldots, n$, i.e., the ith vector of the canonical basis of $\mathbb{R}^n$, is denoted by $\mathbf{e}_i$.

Let $M, N, \in \mathbb{R}^{n\times n}$. The square pencil $\lambda N - M$ is *regular* if $\det(\lambda N - M)$ is not identically zero for all values of λ.

Given $M, N, \in \mathbb{R}^{n\times n}$, $\lambda_i \in \mathbb{C}$, $i = 1, \ldots, n$, $\lambda_i \neq \lambda_j$, $i \neq j$, the generalized eigenvalues of the regular pencil [18] $\lambda N - M$, and $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively, the right and left eigenvectors of λ_i, the *relative eigenvalue condition number* of λ_i, $i = 1, \ldots, n$, as defined in [18], is given by

$$\kappa(\lambda_i, M, N) = \sqrt{\frac{\|M\|_2^2 + \|N\|_2^2}{|\,\xi_i\,|^2 + |\,\psi_i\,|^2}}, \tag{4}$$

where $\xi_i = \mathbf{y}_i^T M\mathbf{x}_i / (\|\mathbf{x}_i\|_2 + \|\mathbf{y}_i\|_2)$, $\psi_i = \mathbf{y}_i^T N\mathbf{x}_i / (\|\mathbf{x}_i\|_2 + \|\mathbf{y}_i\|_2)$.

The *defect from normality* of a regular pencil $\lambda N - M$, with $M, N, \in \mathbb{R}^{n\times n}$, with N nonsingular, is defined as

$$\Gamma(M, N) := \sum_{i=1}^{n} {\sigma_i^{(r)}}^2 + \sum_{i=1}^{n} {\sigma_i^{(\ell)}}^2 - 2\sum_{i=1}^{n} |\,\lambda_i\,|^2, \tag{5}$$

where

$$\sigma_i^{(r)} = \sigma_i(N^{-1}M), \quad \sigma_i^{(\ell)} = \sigma_i(MN^{-1}),$$

are the singular values of $N^{-1}M$ and MN^{-1}, respectively, and λ_i are the generalized eigenvalues of the pencil $\lambda N - M$, $i = 1, \ldots, n$.

3 Laguerre–Sobolev polynomials

Let us consider the sequence of Laguerre–Sobolev polynomials $\{q_n(x)\}_{n\geq 0}$, orthogonal with respect to the Sobolev inner product

$$(f, g)_S = \int_0^{+\infty} f(x)g(x)x^\alpha e^{-x}dx + \gamma \int_0^{+\infty} f'(x)g'(x)x^\alpha e^{-x}dx, \tag{6}$$

with $\alpha > -1$ and $\gamma \geq 0$, i.e., the n–th monic Laguerre–Sobolev polynomial $q_n(x)$ satisfies the following orthogonality conditions

$$(q_n, x^k)_S = 0, \qquad k = 0, 1, \ldots, n-1.$$

Taking into account that the pair of measures in (6) is a coherent pair, indeed a self coherent pair, the sequence of monic Laguerre–Sobolev orthogonal polynomials $\{q_n(x)\}_{n\geq 0}$ satisfies the following four–term recurrence relation

$$x\,(q_{n-1}(x) + e_n q_{n-2}(x)) = a_n q_n(x) + b_n q_{n-1}(x) + c_n q_{n-2}(x) + d_n q_{n-3}(x), \tag{7}$$

with

$$\begin{aligned} a_n &= 1 \\ b_n &= 2n + \alpha + f_n(\gamma) \\ c_n &= n(n+\alpha-1) + (2n+\alpha) f_{n-1}(\gamma) \\ d_n &= n(n+\alpha-1) f_{n-2}(\gamma) \\ e_n &= f_{n-1}(\gamma) \end{aligned} \tag{8}$$

and

$$\begin{cases} f_1(\gamma) = \frac{2(\alpha+1)}{\gamma+\alpha+1} \\ \\ f_n(\gamma) = \frac{(n+1)(n+\alpha)}{n(2+\gamma)+\alpha-f_{n-1}(\gamma)}, \quad n = 2, 3, \ldots. \end{cases}$$

As pointed out in [21], the coefficient $f_n(\gamma)$ can be expressed as

$$f_n(\gamma) = (n+1)(n+\alpha)\frac{q_{n-1}(\gamma)}{q_n(\gamma)},$$

where the polynomials $\{q_n(\gamma)\}_{n\geq 0}$ satisfy the TTRR

$$\begin{cases} q_0(\gamma) = 1, \\ q_1(\gamma) = \gamma + \alpha + 1, \\ q_n(\gamma) = (n\gamma + 2n + \alpha) q_{n-1}(\gamma) - n(n+\alpha-1) q_{n-2}(\gamma), \quad n \geq 2. \end{cases} \tag{9}$$

Denoting $r_n(\gamma) = q_n(2\gamma - 2)$, the above expression can be formulated in terms of monic polynomials as

$$\begin{cases} r_0(\gamma) = 1, \\ r_1(\gamma) = \gamma + \frac{\alpha-1}{2}, \\ r_n(\gamma) = \left(\gamma + \frac{\alpha}{2n}\right) r_{n-1}(\gamma) - \frac{n+\alpha-1}{4(n-1)}, \qquad n \geq 2. \end{cases} \tag{10}$$

If we compare the above recurrence formula with the TTRR satisfied by the monic generalized Pollaczek polynomials ([5, p. 185]), then we deduce that the polynomials $\{r_n(\gamma)\}_{n\geq 0}$ are the co–recursive polynomials with parameter $1/2$ for the generalized Pollaczek polynomials.

Written in a matrix form, (7) reads

$$\left(x \begin{bmatrix} 1 & & & & \\ e_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & e_{n-1} & 1 & \\ & & & e_n & 1 \end{bmatrix} - \begin{bmatrix} b_1 & 1 & & & \\ c_2 & b_2 & 1 & & \\ d_3 & c_3 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} & 1 \\ & & d_n & c_n & b_n \end{bmatrix} \right) \begin{bmatrix} q_0(x) \\ q_1(x) \\ q_2(x) \\ \vdots \\ q_{n-2}(x) \\ q_{n-1}(x) \end{bmatrix} = q_n(x) \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\Updownarrow$$

$$(xB_n - A_n)\mathbf{q}_n(x) = q_n(x)\mathbf{e}_n$$

$$\Updownarrow$$

$$(xB_{n,n+1} - A_{n,n+1}) \quad \begin{bmatrix} \frac{\mathbf{q}_n(x)}{q_n(x)} \end{bmatrix} = \mathbf{0},$$

where $B_{n,n+1} := [B_n \mid \mathbf{0}]$ and $A_{n,n+1} := [A_n \mid \mathbf{e}_n]$.

Therefore, $\bar{x}$ is a zero of $q_n(x)$, i.e., $q_n(\bar{x}) = 0$, if and only if $\bar{x}$ is a generalized eigenvalue of

$$(xB_n - A_n)\mathbf{q}_n(x) = \mathbf{0}, \tag{11}$$

with $\mathbf{q}_n(\bar{x})$ the corresponding eigenvector.

On the other hand, for any $x \in \mathbb{R}$, $xB_{n,n+1} - A_{n,n+1}$ is full row rank and $\begin{bmatrix} \frac{\mathbf{q}_n(x)}{q_n(x)} \end{bmatrix}$ is the only vector spanning its right null–space.

4 Proposed algorithms

Here, we propose two algorithms for computing the zeros of the Laguerre–Sobolev polynomials.

4.1 Algorithm 1

The zeros of the Laguerre–Sobolev polynomials, that are the eigenvalues of $B_n^{-1}A_n$ as well as the generalized eigenvalues of (11), are real [21, Th. 6.1].

Taking into account the relative condition numbers for the eigenvalues λ_i, $i = 1, \ldots, n$, defined in Section 2, it turns out that (11) is very ill–conditioned for any value of α and γ (see, for example, $\kappa(x_i, A_n, B_n)$, denoted by "$*$" in Fig. 1, left plot), and customary methods to compute its generalized eigenvalues, such as the QZ algorithm [19], yield unreliable results (see $x_i(A_n, B_n)$, denoted by "$*$" in Fig. 1, right plot).

In order to overcome this issue, we propose a new algorithm whose first step consists of determining a diagonal matrix $\Delta_n = \mathrm{diag}(\delta_1, \ldots, \delta_n)$, balancing both matrices B_n and A_n as follows,

$$\hat{B}_n = \Delta_n^{-1} B_n \Delta_n, \qquad \hat{A}_n = \Delta_n^{-1} A_n \Delta_n, \tag{12}$$

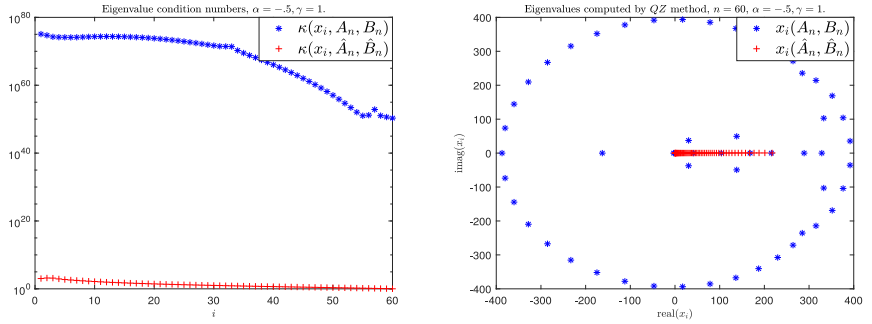

Fig. 1 Relative eigenvalue condition numbers (left) and eigenvalues computed by the QZ method (right) of problems (11) and (13), with $n = 60$, $\alpha = -.5$, $\gamma = 1$, on a logarithmic scale

such that $\mathtt{triu}(\hat{B}_n^{-1}\hat{A}_n, -1)$ is symmetric. The generalized eigenvalue problem (11) is thus transformed into the following equivalent one

$$(x\hat{B}_n - \hat{A}_n)\hat{\mathbf{q}}_n(x) = \mathbf{0}, \tag{13}$$

denoted also by

$$(\hat{A}_n, \hat{B}_n), \tag{14}$$

with $\hat{\mathbf{q}}_n(x) = \Delta_n^{-1}\mathbf{q}_n(x)$, which turns out to be well–conditioned (see $\kappa(x_i, \hat{B}_n, \hat{A}_n)$, denoted by "+" in Fig. 1, left plot). In this case the QZ algorithm provides reliable results (see $x_i(\hat{A}_n, \hat{B}_n)$, denoted by "+" in Fig. 1, right plot).

This balancing is in fact a renormalization of the polynomial sequence. The balancing matrix Δ_n can be obtained directly from the four–term recurrence relation, see the `Matlab`–like algorithm displayed in Table 1. Although the proposed balancing algorithm drastically reduces the conditioning of (11), it is worth stressing that the

Table 1 Computation of the balancing matrix Δ_n

`function` $[\Delta_n]$=`balanceD`($\mathbf{b}$, $\mathbf{c}$, $\mathbf{e}$)
% Computation of the diagonal balancing matrix Δ_n
% Input: $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, the main diagonal, the first and second subdiagonal of A_n
% $\mathbf{e}$, the first subdiagonal of B_n
% Output: Δ_n, the diagonal balancing matrix
$\delta(1) = 1;\ v(1) = 0;\ e(1) = 0;$
`for i=2:n`
$v(i) = c(i) - e(i)(b(i-1) - e(i-1));$
$\delta(i) = \delta(i-1)\sqrt{v(i)};$
`end`
$\Delta_n = \mathtt{diag}(\delta(1), \delta(2), \ldots, \delta(n));$

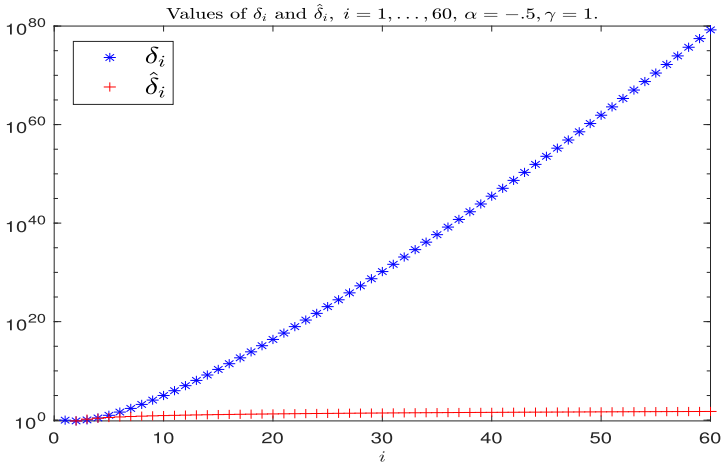

Fig. 2 Growth of the entries in the balancing matrix Δ_n, δ_i ("$*$") and $\hat{\delta}_i$, ("+") $i = 1, \ldots, 60$, $\alpha = -.5$, $\gamma = 1$, on a logarithmic scale. The values δ_i grow exponentially, leading to computational difficulties, while the values based on ratios, $\hat{\delta}_i$, remain small

entries of Δ_n, denoted by "$*$" in Fig. 2, grow exponentially, for any value of α and γ, causing overflow for $n > 200$.

In order to prevent this phenomenon, taking into account that

$$\hat{A}_n = \begin{bmatrix} b_1 & \frac{\delta_2}{\delta_1} & & & \\ c_2\frac{\delta_1}{\delta_2} & b_2 & \frac{\delta_3}{\delta_2} & & \\ d_3\frac{\delta_1}{\delta_3} & c_3\frac{\delta_2}{\delta_3} & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} & \frac{\delta_n}{\delta_{n-1}} \\ & & d_n\frac{\delta_{n-2}}{\delta_n} & c_n\frac{\delta_{n-1}}{\delta_n} & b_n \end{bmatrix} \tag{15}$$

and

$$\hat{B}_n = \begin{bmatrix} 1 & & & & \\ e_2\frac{\delta_1}{\delta_2} & 1 & & & \\ & \ddots & \ddots & & \\ & & e_{n-1}\frac{\delta_{n-2}}{\delta_{n-1}} & 1 & \\ & & & e_n\frac{\delta_{n-1}}{\delta_n} & 1 \end{bmatrix}, \tag{16}$$

imposing $\delta_0 = 1$, it is sufficient to compute the ratios

$$\hat{\delta}_i = \frac{\delta_i}{\delta_{i-1}}, \quad i = 1, \ldots, n,$$

denoted by "+" in Fig. 2, which do not exhibit an exponential growth. Therefore, matrices (15) and (16) can be written as follows, without constructing the balancing matrix Δ_n,

$$\hat{A}_n = \begin{bmatrix} b_1 & \hat{\delta}_2 & & & \\ \frac{c_2}{\hat{\delta}_2} & b_2 & \hat{\delta}_3 & & \\ \frac{d_3}{\hat{\delta}_2\hat{\delta}_3} & \frac{c_3}{\hat{\delta}_3} & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} & \hat{\delta}_n \\ & & \frac{d_n}{\hat{\delta}_{n-1}\hat{\delta}_n} & \frac{c_n}{\hat{\delta}_n} & b_n \end{bmatrix}$$

and

$$\hat{B}_n = \begin{bmatrix} 1 & & & \\ \frac{e_2}{\hat{\delta}_2} & 1 & & \\ \ddots & \ddots & & \\ & \frac{e_{n-1}}{\hat{\delta}_{n-1}} & 1 & \\ & & \frac{e_n}{\hat{\delta}_n} & 1 \end{bmatrix}.$$

The elements of the matrices $\hat{A}_n$ and $\hat{B}_n$ are computed by the `Matlab`–like algorithm displayed in Table 2.

We observe that the computation of the matrices $\hat{A}_n$ and $\hat{B}_n$ involves only multiplications and divisions by $\hat{\delta}_i$, $i \in \{1, \ldots, n\}$. Hence, they are computed with high relative accuracy.

The second step of the proposed algorithm consists in applying the QZ method to the well–conditioned generalized eigenvalue problem (13) to compute the zeros of the Laguerre–Sobolev polynomial of degree n.

Algorithm 1 can be summarized in the following steps:

Algorithm 1

```
Step 1.  construct the matrices Bn and An
Step 2.  compute the matrices B̂n and Ân as in (12)
Step 3.  solve the generalized eigenvalue problem (13)
         by the QZ algorithm (eig.m of Matlab)
```

Its complexity is $\mathcal{O}(n^3)$, requiring $\mathcal{O}(n^2)$ memory.

4.2 Algorithm 2

Here, we propose a more efficient variant of Algorithm 1, replacing the QZ method by a variant of the Ehrlich–Aberth method [1, 4, 6].

Table 2 Computation of the matrices $\hat{A}_n$ and $\hat{B}_n$

```
function [â, b̂, ĉ, d̂, ê]=balanceD1(b, c, d, e)
% Computation of entries of Â_n and B̂_n, balanced by Δ_n
% Input: b, c, d, the main diagonal, the first and second subdiagonal of A_n
%        e, the first subdiagonal of B_n
% Output: â, b̂, ĉ, d̂, the first superdiagonal, the main diagonal,
%         the first and second subdiagonal of Â_n
%         ê, the first subdiagonal of B̂_n
δ̂(1) = 0; v(1) = 0;
for i=2:n
    v(i) = c(i) − e(i)(b(i − 1) − e(i − 1));
    δ̂(i) = √v(i);
end
for i=2:n
    â(i) = δ̂(i);
    b̂(i) = b(i);
    ĉ(i) = c(i)/δ̂(i);
    ê(i) = e(i)/δ̂(i);
    if i>2
        d̂(i) = d(i)/(δ̂(i)δ̂(i − 1));
    end
end
```

Starting from a vector of initial approximations $\boldsymbol{x}^{(0)}$ of the zeros of $\hat{q}_n(x)$, its entries, at the $(i+1)$th iteration, $i = 0, 1, \ldots$, of the Ehrlich–Aberth method, are updated as follows

$$x_j^{(i+1)} = x_j^{(i)} - \frac{\dfrac{\hat{q}_n(x_j^{(i)})}{\hat{q}_n'(x_j^{(i)})}}{1 - \dfrac{\hat{q}_n(x_j^{(i)})}{\hat{q}_n'(x_j^{(i)})} \displaystyle\sum_{\substack{k=1 \\ k\neq j}}^{n} \frac{1}{x_j^{(i)} - x_k^{(i)}}}, \quad \begin{cases} j = 1, \ldots, n, \\ \\ i = 0, 1, \ldots. \end{cases} \tag{17}$$

Good choices for the initial approximation $\mathbf{x}^{(0)}$ and how we compute $\hat{q}'(x)$ is discussed below. The computed approximations $x_j^{(i)}$ converge cubically to the generalized eigenvalues of (13), or even faster if the method is implemented in a Gauss–Seidel fashion, since the zeros of $\hat{q}_n(x)$ are simple [4].

To ensure a fast convergence of the Ehrlich–Aberth method, a good set of initial approximations for the zeros and a fast and reliable computation of the Newton correction $\hat{q}_n(x)/\hat{q}_n'(x)$ are needed.

Let $\{L_i^{(\alpha)}(x)\}_{i\geq 0}$, be the Laguerre–Sonine polynomials, orthogonal with respect to the inner product

$$(f, g) = \int_0^{+\infty} f(x)g(x)x^{\alpha}e^{-x}dx, \quad \alpha > -1.$$

A good set of initial approximations $\boldsymbol{x}^{(0)}$ for the zeros of $\hat{q}_n(x)$ can be obtained by considering the following Theorem 1 [21, Th. 6.1].

Theorem 1 *Let $\gamma > 0$, then the polynomial $\hat{q}_n(x)$, $n \geq 2$, has exactly n different real zeros, and at least $n-1$ of them are positive. When $\alpha \geq 0$, all the zeros are positive. Moreover, the roots of $\hat{q}_n(x)$ separate those of $L_n^{(\alpha)}(x)$. If we denote by $x_{1,n} < x_{2,n} < \cdots < x_{n,n}$ the zeros of $L_n^{(\alpha)}(x)$ and by $\hat{x}_{1,n} < \hat{x}_{2,n} < \cdots < \hat{x}_{n,n}$ those of $\hat{q}_n(x)$, then*

$$\hat{x}_{1,n} < x_{1,n} < \hat{x}_{2,n} < x_{2,n} < \cdots < x_{n-1,n} < \hat{x}_{n,n} < x_{n,n}. \tag{18}$$

Considering the interlacing property of the zeros of $\{L_i^{(\alpha)}(x)\}_{i\geq 0}$ and (18), we propose four different choices for the vector $\boldsymbol{x}^{(0)} := [x_{1,n}^{(0)}, x_{2,n}^{(0)}, \ldots, x_{n,n}^{(0)}]^T$ of initial approximations of the zeros of $\hat{q}_n(x)$:

I_1: $\boldsymbol{x}^{(0)} := [x_{1,n}, x_{2,n}, \ldots, x_{n,n}]^T$, i.e., the zeros of $L_n^{(\alpha)}(x)$;
I_2: $\boldsymbol{x}^{(0)} := [x_{1,n+1}, x_{2,n+1}, \ldots, x_{n,n+1}]^T$, i.e., the smallest n zeros of $L_{n+1}^{(\alpha)}(x)$;
I_3:

$$\begin{cases} x_{1,n}^{(0)} = 0, & \text{if } \alpha < 0, \\ x_{1,n}^{(0)} = \dfrac{x_{1,n}}{2}, & \text{if } \alpha \geq 0, \\ x_{i,n}^{(0)} = \dfrac{x_{i-1,n} + x_{i,n}}{2}, & i = 2, \ldots, n. \end{cases}$$

The zeros of $L_n^{(\alpha)}(x)$ can be computed by the Golub–Welsch algorithm, with $\mathcal{O}(n^2)$ computational complexity and $\mathcal{O}(n)$ memory or exploiting the technique described in [12].

Observe that

$$\lim_{\gamma \to 0^+} \hat{x}_{i,n} = x_{i,n}, \qquad i = 1, \ldots, n.$$

Therefore, the initial vector $\boldsymbol{x}^{(0)}$ should be chosen equal to I_1 for small values of γ.

A comparison, in terms of efficiency, of the Ehrlich–Aberth method when adopting the different aforementioned initial vectors, is reported in Section 5.

In order to efficiently compute (17), let

$$\begin{aligned} \hat{B}_{n,n+1} &= \Delta_n^{-1} B_{n,n+1} \Delta_{n+1}, \\ \hat{A}_{n,n+1} &= \Delta_n^{-1} A_{n,n+1} \Delta_{n+1}, \\ \hat{M}_{n,n+1}(x) &= x\hat{B}_{n,n+1} - \hat{A}_{n,n+1}, \end{aligned} \tag{19}$$

with $\Delta_{n+1} = \mathrm{diag}(\delta_1, \ldots, \delta_n, \delta_{n+1})$ such that $\mathtt{triu}(\Delta_{n+1}^{-1} B_{n+1}^{-1} A_{n+1} \Delta_{n+1}, -1)$ is a symmetric tridiagonal matrix of order $n+1$.

Let $\hat{\mathbf{q}}_{n+1}(x) := \begin{bmatrix} \hat{q}_0(x) \\ \hat{q}_1(x) \\ \vdots \\ \hat{q}_{n-1}(x) \\ \hat{q}_n(x) \end{bmatrix} = \Delta_{n+1}^{-1} \mathbf{q}_{n+1}(x)$, with $\mathbf{q}_{n+1}(x) := \begin{bmatrix} \mathbf{q}_n(x) \\ q_n(x) \end{bmatrix}$. Then

$$\hat{M}_{n,n+1}(x) \hat{\mathbf{q}}_{n+1}(x) = \mathbf{0}_{n+1} \tag{20}$$

defines a four–term recurrence relation for the sequence of polynomials $\hat{q}_i(x)$, $i = 0, 1, \ldots, n$. Since $\hat{q}_n(x) = q_n(x)/\delta_{n+1}$, the zeros of the polynomial $\hat{q}_n(x)$ are the zeros of $q_n(x)$.

Moreover, by (20), the vector $\hat{\mathbf{q}}_{n+1}(x)$ spans the right null–space of $\hat{M}_{n,n+1}(x)$, for any $x \in \mathbb{R}$.

Given a point $x \in \mathbb{R}$, the Newton correction $\hat{q}_n(x)/\hat{q}_n'(x)$ in (17) can be efficiently computed as follows.

The vector spanning the right null–space can be computed by applying a sequence of Givens rotations

$$G_i = \begin{bmatrix} I_{i-1} & & & \\ & c_i & s_i & \\ & -s_i & c_i & \\ & & & I_{n-i-1} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad i = 1, \ldots, n,$$

to the right of $\hat{M}_{n,n+1}$, chosen such that the matrix

$$\hat{M}_{n,n+1} G_1^T G_2^T \cdots G_i^T, \quad i = 1, \ldots, n,$$

has the entry $(i, i+1)$ annihilated.

This shows that the null–space is spanned by

$$\hat{\mathbf{q}}_{n+1}^T = \delta_{n+1} \hat{\mathbf{r}}_{n+1}^T,$$

where

$$\hat{\mathbf{r}}_{n+1} = \begin{bmatrix} \hat{r}_0(x) \\ \hat{r}_1(x) \\ \vdots \\ \hat{r}_{n-1}(x) \\ \hat{r}_n(x) \end{bmatrix} = G_1^T G_2^T \cdots G_{n-1}^T G_n^T \mathbf{e}_{n+1} = \begin{bmatrix} (-1)^{n+1} s_1 s_2 \cdots s_{n-1} s_n \\ (-1)^n c_1 s_2 \cdots s_{n-1} s_n \\ \vdots \\ c_{n-2} s_{n-1} s_n \\ -c_{n-1} s_n \\ c_n \end{bmatrix}$$

is the vector spanning the null–space of $\hat{M}_{n,n+1}$, and δ_{n+1} is a normalization factor.

The number of required floating point operations for computing $\hat{\mathbf{r}}_{n+1}$ is $13n$.

In order to compute $\hat{q}'_n(x)$ we consider the derivative of (20):

$$\hat{M}'_{n,n+1}(x)\hat{\mathbf{q}}_{n+1}(x) + \hat{M}_{n,n+1}(x)\hat{\mathbf{q}}'_{n+1}(x) = \mathbf{0}_n, \tag{21}$$

where $\hat{M}'_{n,n+1}(x) = \left(x\hat{B}_{n,n+1} - \hat{A}_{n,n+1}\right)' = \hat{B}_{n,n+1}$. Furthermore, since $\hat{q}'_0(x) = 0$ and $\hat{B}_{n,n+1}\mathbf{e}_{n+1} = \mathbf{0}_n$, (21) becomes:

$$\tilde{M}_n\hat{\mathbf{q}}'_{n+1}(2:n+1) = -\hat{B}_n\hat{\mathbf{q}}_{n+1}(1:n), \tag{22}$$

with $\tilde{M}_n = \hat{M}_{n,n+1}\begin{bmatrix} 0 \\ I_n \end{bmatrix}$ a four–band lower triangular matrix of order n.

Instead of solving (22), we solve

$$\tilde{M}_n\hat{\mathbf{r}}'_{n+1}(2:n+1) = -\hat{B}_n\hat{\mathbf{r}}_{n+1}(1:n), \tag{23}$$

with

$$\hat{\mathbf{r}}'_{n+1} = \begin{bmatrix} \hat{r}'_0(x) \\ \hat{r}'_1(x) \\ \vdots \\ \hat{r}'_{n-1}(x) \\ \hat{r}'_n(x) \end{bmatrix} = \frac{\hat{\mathbf{q}}'_{n+1}}{\delta_{n+1}}.$$

This is enough, since

$$\frac{\hat{q}_n(x)}{\hat{q}'_n(x)} = \frac{\delta_{n+1}\hat{r}_n(x)}{\delta_{n+1}\hat{r}'_n(x)} = \frac{\hat{r}_n(x)}{\hat{r}'_n(x)}.$$

The computation of the right–hand side of (23) requires $3n$ floating point operations.

Moreover, forward substitution [11, p. 106] can be used to solve the linear system (23), requiring $7n$ floating point operations.

Algorithm 2 can be summarized in the following steps:

Algorithm 2

```
Step 1.  construct the matrices B_n and A_n;
Step 2.  compute the matrices B̂_n and Â_n as in (12)
Step 3.  compute the zeros of q̂_n(x) by a variant
         of the Ehrlich-Aberth method [1, 4, 6].
```

Its complexity is $\mathcal{O}(n^2)$, requiring $\mathcal{O}(n)$ memory.

5 Numerical results

In this section we report some numerical tests in order to compare the performance of the algorithms described in Section 4 in terms of efficiency and accuracy. All

the computations are performed in `Matlab R2022a`, with machine precision $\epsilon \approx 2.22 \times 10^{-16}$.

In all the examples, the generalized eigenvalues of (11), for different values of n, α and γ, are computed by Algorithms 1 and 2, and are denoted, respectively, by $x_i^{(\mathcal{A}_1)}$ and $x_i^{(\mathcal{A}_2)}$, $i = 1, \ldots, n$.

The generalized eigenvalues of (11) are also computed by the Advanpix Multiprecision Computing Toolbox[1] for `Matlab` [24], with precision of 800 digits and rounded in double precision by the `Matlab` function `double.m`. These values, denoted by $x_i^{(exact)}$, $i = 1, \ldots, n$, are assumed to be the exact ones.

Example 1 The first numerical test is aimed to select the initial vector, among those listed in Subsection 4.2, for which the Ehrlich–Aberth method requires less iterations. The results are displayed in Table 3, setting $n = 100$, and considering different values of α and γ, reported in columns 1 and 2, respectively. In columns 3–5, the means of the number of iterations required by the Ehrlich–Aberth method when adopting the different initial vectors I_1, I_2 and I_3 are displayed. Finally, in columns 6 and 7, the defect from normality (5) for the generalized eigenvalue problems (11) and (13), are shown. A small defect from normality is desirable, since it implies that the sensitivity of the generalized eigenvalues is small as well [18]. In general, Algorithm 2 performs similarly for any choice of the initial vector. Nevertheless, as highlighted in Subection 4.2, Algorithm 2 performs best in terms of efficiency when choosing I_1 as initial vector, for small values of γ. Furthermore, Algorithm 2 with initial vector I_1 requires, on average, a smaller number of iteration for $\alpha \geq 10^2$.

In the other considered cases, Algorithm 2 with initial vector I_3, turns out to be slightly faster.

Concerning the defect from normality, as reported in [21], fixed $n \in \mathbb{N}$,

$$\lim_{\gamma \to \infty} f_n(\gamma) = 0,$$

and, hence, by (8), also

$$\lim_{\gamma \to \infty} d_n = 0, \quad \lim_{\gamma \to \infty} e_n = 0,$$

from which it follows that

$$\lim_{\gamma \to \infty} B_n = I_n, \quad \lim_{\gamma \to \infty} A_n = \begin{bmatrix} b_1 & 1 & & & \\ c_2 & b_2 & 1 & & \\ & c_3 & \ddots & \ddots & \\ & & \ddots & b_{n-1} & 1 \\ & & & c_n & b_n \end{bmatrix} =: \tilde{A}_n.$$

Therefore, for $\gamma \to \infty$, the defect from normality of the regular pencil $\lambda B_n - A_n$ reduces to the defect from normality of the pencil $\lambda I_n - \tilde{A}_n$, with $\tilde{A}_n$ a tridiagonal

[1] We preferred to consider the eigenvalues computed by Advanpix Multiprecision Computing Toolbox as the exact ones since the eigenvalues computed by using variable precision arithmetic of `Matlab` where not reliable for matrices of size equal to 200.

Table 3 Means of the numbers of iterations required by the Ehrlich–Aberth method when considering I_1, I_2 and I_3 as initial vector (columns $3-5$), for $n = 100$ and different values of α (column 1) and γ (column 2); defect from normality (5) of $\Gamma(A_n, B_n)$ and $\Gamma(\hat{A}_n, \hat{B}_n)$ (columns $6-7$)

		Mean			Defect from normality	
α	γ	$it(I_1)$	$it(I_2)$	$it(I_3)$	$\Gamma(A_n, B_n)$	$\Gamma(\hat{A}_n, \hat{B}_n)$
$-.99$	10^{-6}	3.03	4.62	6.74	Inf	4.42×10^{7}
$-.99$	10^{-3}	3.30	4.72	6.12	Inf	1.29×10^{5}
$-.99$	10^{0}	4.48	4.84	4.46	1.51×10^{231}	4.03×10^{2}
$-.99$	10^{1}	4.75	4.62	4.17	4.14×10^{108}	1.05×10^{2}
$-.99$	10^{2}	4.82	4.87	4.09	3.71×10^{9}	1.02×10^{2}
$-.99$	10^{3}	4.82	4.67	4.09	3.71×10^{9}	1.02×10^{2}
$-.99$	10^{9}	4.83	4.65	4.09	3.71×10^{9}	1.02×10^{2}
0	10^{-6}	2.66	4.79	6.84	Inf	9.84×10^{5}
0	10^{-3}	3.25	4.78	6.16	Inf	1.47×10^{5}
0	10^{0}	4.43	4.82	4.40	1.18×10^{234}	3.11×10^{2}
0	10^{1}	4.72	4.63	4.10	2.32×10^{110}	6.29×10^{0}
0	10^{2}	4.77	4.61	4.03	3.80×10^{9}	2.96×10^{0}
0	10^{3}	4.78	4.66	4.02	3.80×10^{9}	2.99×10^{0}
0	10^{9}	4.78	4.55	4.02	3.80×10^{9}	3.00×10^{0}
10^{0}	10^{-6}	2.28	4.65	6.90	Inf	4.94×10^{5}
10^{0}	10^{-3}	3.18	4.61	6.30	Inf	1.51×10^{5}
10^{0}	10^{0}	4.42	4.81	4.43	4.21×10^{236}	3.16×10^{2}
10^{0}	10^{1}	4.71	4.46	4.12	1.37×10^{112}	5.85×10^{0}
10^{0}	10^{2}	4.77	4.84	4.05	3.90×10^{9}	2.42×10^{0}
10^{0}	10^{3}	4.77	4.83	4.05	3.90×10^{9}	2.49×10^{0}
10^{0}	10^{9}	4.78	4.65	4.05	3.90×10^{9}	2.50×10^{0}
10^{2}	10^{-6}	2.00	4.85	7.04	Inf	9.93×10^{3}
10^{2}	10^{-3}	2.99	4.71	7.20	Inf	9.88×10^{3}
10^{2}	10^{0}	3.99	4.62	5.14	8.85×10^{287}	7.85×10^{2}
10^{2}	10^{1}	3.99	4.38	5.00	6.07×10^{197}	1.28×10^{1}
10^{2}	10^{2}	3.99	4.59	5.00	1.89×10^{36}	1.24×10^{0}
10^{2}	10^{3}	3.99	4.42	5.00	2.00×10^{10}	1.68×10^{0}
10^{2}	10^{9}	3.99	4.40	5.00	2.00×10^{10}	2.01×10^{0}
10^{3}	10^{-6}	3.16	6.09	8.14	Inf	1.01×10^{3}
10^{3}	10^{-3}	3.16	5.95	7.98	Inf	1.01×10^{3}
10^{3}	10^{0}	4.96	6.67	8.89	Inf	6.92×10^{2}
10^{3}	10^{1}	5.59	6.72	7.48	5.30×10^{283}	7.79×10^{1}
10^{3}	10^{2}	5.33	6.18	6.53	1.33×10^{180}	3.58×10^{0}
10^{3}	10^{3}	5.33	6.22	6.50	2.39×10^{12}	1.01×10^{0}
10^{3}	10^{9}	6.41	7.26	7.52	7.57×10^{11}	2.00×10^{0}

Table 4 Absolute errors of the generalized eigenvalues computed by Algorithm 1 for different values of α (column 1), γ (column 2) and n (column $3-5$)

		$\max_i \mid x_i^{(\mathcal{A}_1)} - x_i^{(exact)} \mid$		
α	γ	$n = 100$	$n = 200$	$n = 300$
$-.99$	10^{-6}	4.4062×10^{-12}	8.25×10^{-12}	1.18×10^{-11}
$-.99$	10^{-3}	9.7078×10^{-13}	3.41×10^{-12}	1.25×10^{-10}
$-.99$	10^{0}	1.0072×10^{-11}	1.82×10^{-12}	4.07×10^{-11}
$-.99$	10^{1}	7.3896×10^{-13}	3.67×10^{-10}	5.00×10^{-12}
$-.99$	10^{2}	1.2506×10^{-12}	4.61×10^{-10}	4.34×10^{-12}
$-.99$	10^{3}	6.8212×10^{-13}	5.70×10^{-10}	6.40×10^{-12}
$-.99$	10^{9}	4.6043×10^{-12}	5.82×10^{-10}	2.98×10^{-11}
0	10^{-6}	1.0232×10^{-12}	3.84×10^{-12}	7.73×10^{-12}
0	10^{-3}	1.5348×10^{-12}	1.09×10^{-11}	1.33×10^{-11}
0	10^{0}	1.0800×10^{-12}	3.52×10^{-12}	5.00×10^{-12}
0	10^{1}	1.0232×10^{-12}	2.27×10^{-12}	5.23×10^{-12}
0	10^{2}	5.6843×10^{-13}	1.82×10^{-12}	2.96×10^{-12}
0	10^{3}	6.2528×10^{-13}	3.30×10^{-12}	2.50×10^{-12}
0	10^{9}	4.3201×10^{-12}	1.53×10^{-11}	3.12×10^{-11}
10^{0}	10^{-6}	1.3642×10^{-12}	2.96×10^{-12}	7.50×10^{-12}
10^{0}	10^{-3}	1.6094×10^{-12}	2.61×10^{-12}	7.05×10^{-12}
10^{0}	10^{0}	8.5265×10^{-13}	2.16×10^{-12}	4.09×10^{-12}
10^{0}	10^{1}	7.9581×10^{-13}	4.21×10^{-12}	8.64×10^{-12}
10^{0}	10^{2}	1.1369×10^{-12}	2.16×10^{-12}	5.46×10^{-12}
10^{0}	10^{3}	7.9581×10^{-13}	1.93×10^{-12}	4.21×10^{-12}
10^{0}	10^{9}	4.3769×10^{-12}	1.49×10^{-11}	2.96×10^{-11}
10^{2}	10^{-6}	9.0949×10^{-13}	5.23×10^{-12}	8.64×10^{-12}
10^{2}	10^{-3}	1.5916×10^{-12}	5.34×10^{-12}	9.55×10^{-12}
10^{2}	10^{0}	1.5916×10^{-12}	4.43×10^{-12}	4.32×10^{-12}
10^{2}	10^{1}	1.5916×10^{-12}	2.39×10^{-12}	9.32×10^{-12}
10^{2}	10^{2}	1.1369×10^{-12}	2.73×10^{-12}	4.55×10^{-12}
10^{2}	10^{3}	1.4779×10^{-12}	4.09×10^{-12}	4.55×10^{-12}
10^{2}	10^{9}	5.7412×10^{-12}	1.83×10^{-11}	3.46×10^{-11}
10^{3}	10^{-6}	5.0022×10^{-12}	1.41×10^{-11}	1.82×10^{-11}
10^{3}	10^{-3}	4.7748×10^{-12}	8.41×10^{-12}	1.86×10^{-11}
10^{3}	10^{0}	4.0927×10^{-12}	5.91×10^{-12}	8.19×10^{-12}
10^{3}	10^{1}	4.3201×10^{-12}	5.91×10^{-12}	8.64×10^{-12}
10^{3}	10^{2}	4.0927×10^{-12}	9.09×10^{-12}	8.64×10^{-12}
10^{3}	10^{3}	3.4106×10^{-12}	7.28×10^{-12}	8.19×10^{-12}
10^{3}	10^{9}	1.9099×10^{-11}	4.68×10^{-11}	7.82×10^{-11}

Table 5 Absolute errors of the generalized eigenvalues computed by Algorithm 2 for different values of α (column 1), γ (column 2) and n (column $3-5$)

		$\max_i \mid x_i^{(\mathcal{A}_2)} - x_i^{(exact)} \mid$		
α	γ	$n = 100$	$n = 200$	$n = 300$
$-.99$	10^{-6}	8.0908×10^{-14}	4.36×10^{-13}	6.35×10^{-13}
$-.99$	10^{-3}	6.2630×10^{-14}	2.18×10^{-13}	2.59×10^{-13}
$-.99$	10^{0}	2.8422×10^{-14}	1.17×10^{-13}	1.14×10^{-13}
$-.99$	10^{1}	1.4211×10^{-14}	5.68×10^{-14}	2.27×10^{-13}
$-.99$	10^{2}	5.6843×10^{-14}	1.14×10^{-13}	1.14×10^{-13}
$-.99$	10^{3}	2.8422×10^{-14}	1.14×10^{-13}	1.14×10^{-13}
$-.99$	10^{9}	1.4211×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
0	10^{-6}	1.1034×10^{-13}	5.80×10^{-13}	6.86×10^{-13}
0	10^{-3}	4.9627×10^{-14}	1.76×10^{-13}	2.91×10^{-13}
0	10^{0}	5.6843×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
0	10^{1}	2.8422×10^{-14}	5.68×10^{-14}	1.14×10^{-13}
0	10^{2}	5.6843×10^{-14}	1.14×10^{-13}	1.14×10^{-13}
0	10^{3}	5.6843×10^{-14}	5.68×10^{-14}	5.68×10^{-14}
0	10^{9}	5.6843×10^{-14}	1.14×10^{-13}	1.14×10^{-13}
10^{0}	10^{-6}	1.0242×10^{-13}	3.87×10^{-13}	8.43×10^{-13}
10^{0}	10^{-3}	1.3869×10^{-13}	1.14×10^{-13}	3.66×10^{-13}
10^{0}	10^{0}	2.8422×10^{-14}	5.68×10^{-14}	2.27×10^{-13}
10^{0}	10^{1}	2.8422×10^{-14}	1.17×10^{-13}	2.27×10^{-13}
10^{0}	10^{2}	2.8422×10^{-14}	5.68×10^{-14}	2.27×10^{-13}
10^{0}	10^{3}	1.4211×10^{-14}	5.68×10^{-14}	1.14×10^{-13}
10^{0}	10^{9}	3.5527×10^{-11}	5.68×10^{-14}	1.14×10^{-13}
10^{2}	10^{-6}	5.6843×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
10^{2}	10^{-3}	5.6843×10^{-14}	1.60×10^{-13}	2.63×10^{-13}
10^{2}	10^{0}	5.6843×10^{-14}	1.14×10^{-13}	1.10×10^{-13}
10^{2}	10^{1}	5.6843×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
10^{2}	10^{2}	5.6843×10^{-14}	5.68×10^{-14}	2.27×10^{-13}
10^{2}	10^{3}	5.6843×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
10^{2}	10^{9}	5.6843×10^{-14}	1.14×10^{-13}	2.27×10^{-13}
10^{3}	10^{-6}	1.1369×10^{-13}	2.27×10^{-13}	4.55×10^{-13}
10^{3}	10^{-3}	1.1369×10^{-13}	2.27×10^{-13}	2.27×10^{-13}
10^{3}	10^{0}	2.2737×10^{-13}	2.27×10^{-13}	4.55×10^{-13}
10^{3}	10^{1}	1.1369×10^{-13}	2.27×10^{-13}	2.27×10^{-13}
10^{3}	10^{2}	1.1369×10^{-13}	4.55×10^{-13}	1.14×10^{-13}
10^{3}	10^{3}	2.2737×10^{-13}	2.27×10^{-13}	4.55×10^{-13}
10^{3}	10^{9}	2.2737×10^{-13}	4.55×10^{-13}	2.27×10^{-13}

matrix. By (8), while the entries of the first superdiagonal of $\tilde{A}_n$ are equal to 1, the entries of the main diagonal and those on the first subdiagonal grow linearly and quadratically, respectively.

The matrix pencil $\lambda I_n - \bar{A}_n$, obtained by applying the function `balanceD1` in Section 4 to $\lambda I_n - \tilde{A}_n$, has, hence, defect from normality equal to 0.

Summarizing, for a fixed $n \in \mathbb{N}$, the defect from normality of the regular pencil $\lambda B_n - A_n$ is a decreasing function of γ and it tends to be a constant of order 10^9, as $\gamma \to \infty$.

We observe that $\Gamma(A_n, B_n)$ is approximately a decreasing function of γ and it is huge for $0 < \gamma < 100$, while $\Gamma(\hat{A}_n, \hat{B}_n)$ is bounded by 10^3.

Example 2 In this example we compute the absolute errors of the generalized eigenvalues of (11) obtained when applying Algorithms 1 and 2 with initial vector I_3, for $n = 100, 200$, and 300, and the same values of α and γ considered in Example 1.

The results are reported in Table 4 for Algorithm 1 and in Table 5 for Algorithm 2.

We observe that Algorithm 2 performs best in terms of accuracy.

6 Conclusions

The computation of the zeros of the Laguerre–Sobolev polynomial of degree n can be formulated as a generalized eigenvalue problem, involving a lower bidiagonal matrix and a 2–banded lower Hessenberg matrix of order n.

Customary techniques for solving such a problem, as the QZ method, yield unreliable results since the generalized eigenvalues are very ill-conditioned.

Here, two algorithms are proposed based on a novel balancing procedure that drastically reduces the ill–conditioning of the eigenvalues. The first algorithm computes the generalized eigenvalues of the balanced problem by the QZ method, while the second one is based on a variant of the Ehrlich–Aberth method.

The numerical experiments show that the second algorithm outperforms the first one in terms of accuracy. Moreover, its complexity and required memory are of the order of $\mathcal{O}(n^2)$ and $\mathcal{O}(n)$, respectively, rather than $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$, characterizing the first algorithm.

Acknowledgements This work was supported partly by Gruppo Nazionale Calcolo Scientifico (GNCS) of Istituto Nazionale di Alta Matematica (INdDAM) and partly by MIUR, PROGETTO DI RICERCA DI RILEVANTE INTERESSE NAZIONALE (PRIN) 20227PCCKZ "Low-rank Structures and Numerical Methods in Matrix and Tensor Computations and their Application", Università degli Studi di BOLOGNA CUP J53D23003620006.

Teresa Laudadio and Nicola Mastronardi are members of the Gruppo Nazionale Calcolo Scientifico–Istituto Nazionale di Alta Matematica (GNCS–INdAM). The work of Nicola Mastronardi was partly supported by MIUR, PROGETTO DI RICERCA DI RILEVANTE INTERESSE NAZIONALE (PRIN) 20227PCCKZ "Low–rank Structures and Numerical Methods in Matrix and Tensor Computations and their Application", Università degli Studi di BOLOGNA CUP J53D23003620006. The work of Francisco Marcellán was partially supported by the research project PID2021-122154NBI00 *Ortogonalidad y Aproximación con Aplicaciones en Machine Learning y Teoría de Probabilidad* funded by MCIU-AEI-10.13039-501100011033 and by *ERDF A Way of Making Europe*. The work of Paul Van Dooren was partly supported by Consiglio Nazionale delle Ricerche, under the Short Term Mobility program.

The authors would like to thank the anonymous Reviewer for the valuable comments and suggestions, which helped in improving the quality of the manuscript.

Author Contributions All authors equally contributed to the manuscript.

Funding Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement. This work was partly supported by Gruppo Nazionale Calcolo Scientifico (GNCS) of Istituto Nazionale di Alta Matematica (INdDAM).

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Code Availability The codes described in the manuscript are available from the authors, upon request.

Declarations

Competing Interests The authors declare no competing interests.

Ethics Approval Not applicable.

References

1. Aberth, O.: Iteration methods for finding all zeros of a polynomial simultaneously. Math. Comput. **27**, 339–344 (1973)
2. Ai, Q., Li, H., Wang, Z.: Diagonalized Legendre spectral methods using Sobolev orthogonal polynomals for elliptic boundary value problems. Appl. Numer. Math. **127**, 196–210 (2018)
3. Althammer, P.: Eine erweiterung des orthogonalitätsbegriffes bei polynomen und deren anwendung auf die beste approximation. J. Reine Angew. Math. **211**, 192–204 (1962). (**In German**)
4. Bini, D.A., Gemignani, L., Tisseur, F.: The Ehrlich-Aberth method for the nonsymmetric tridiagonal eigenvalue problem. SIAM J. Matrix Anal. Appl. **27**, 153–175 (2005)
5. Chihara, T.S.: An Introduction to Orthogonal Polynomials. Gordon and Breach, New York (1978)
6. Ehrlich, L.W.: A modified Newton method for polynomials. Commun. ACM **10**, 107–108 (1967)
7. Fernández, L., Marcellán, F., Pérez, T.E., Piñar, M.: Sobolev orthogonal polynomials and spectral methods in boundary value problems. Appl. Numer. Math. **200**, 254–272 (2024)
8. Fernández, L., Marcellán, F., Pérez, T.E., Piñar, M.: Sobolev orthogonal polynomials for solving the Schrödinger equation with potentials $V(x) = x^{2k}, k \geqslant 1$. In: Castillo, K., Durán, A.J. (eds.) Orthogonal Polynomials and Special Functions: In Memory of José Carlos Petronilho. Coimbra Mathematical Texts, vol. 3, pp. 81–106. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-69646-6_4
9. Gautschi, W., Zhang, M.: Computing orthogonal polynomials in Sobolev spaces. Numer. Math. **71**(2), 159–183 (1995)
10. Golub, G.H., Welsch, J.H.: Calculation of Gauss quadrature rules. Math. Comp. **23**, 221–230 (1969)
11. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
12. Huybrechs, D., Opsomer, P.: Construction and implementation of asymptotic expansions for Laguerre-type orthogonal polynomials. IMA J. Numer. Anal. **38**(3), 1085–1118 (2018)
13. Iserles, A., Koch, P.E., Nørsett, S.P., Sanz-Serna, J.M.: Orthogonality and approximation in a Sobolev space. In: Algorithms for approximation, II (Shrivenham, 1988), pp. 117–124. Chapman and Hall, London (1990)

14. Iserles, A., Koch, P.E., Nørsett, S.P., Sanz-Serna, J.M.: On polynomials orthogonal with respect to certain Sobolev inner products. J. Approx. Theory **65**(2), 151–175 (1991)
15. Laudadio, T., Mastronardi, N., Van Dooren, P.: Computational aspects of simultaneous Gaussian quadrature. Numer. Algor. (2024). https://doi.org/10.1007/s11075-024-01785-0
16. Laudadio, T., Mastronardi, N., Van Assche, W., Van Dooren, P.: A Matlab package computing simultaneous Gaussian quadrature rules for multiple orthogonal polynomials. J. Comput. Appl. Math. **451**(1), 116109 (2024). https://doi.org/10.1016/j.cam.2024.116109
17. Laudadio, T., Mastronardi, N., Van Dooren, P.: Fast and reliable algorithms for computing the zeros of Althammer polynomials. Appl. Numer. Math. **207**, 210–221 (2025)
18. Lemonnier, D., Van Dooren, P.: Balancing regular matrix pencils. SIAM J. Matrix Anal. Appl. **28**(1), 253–263 (2006)
19. Moler, C.B., Stewart, G.W.: An algorithm for generalized matrix eigenvalue problems. SIAM J. Numer. Anal. **10**(2), 241–256 (1973)
20. Liu, Y., Yu, X.-H., Wang, Z.-Q., Li, H.-Y.: Hermite-Sobolev orthogonal functions and spectral methods for second and fourth-order problems on unbounded domains. Int. J. Comput. Math. **96**(5), 950–970 (2019)
21. Marcellán, F., Pérez, T.E., Piñar, M.A.: Laguerre-Sobolev orthogonal polynomials. J. Comput. Appl. Math. **71**, 245–265 (1996)
22. Marcellán, F., Xu, Y.: On Sobolev orthogonal polynomials. Expo. Math. **33**(3), 308–352 (2015)
23. Meijer, H.G.: Determination of all coherent pairs. J. Approx. Theory **89**(3), 321–343 (1997)
24. Advanpix LLC.: Multiprecision Computing Toolbox for MATLAB. Yokohama, Japan (2015)
25. Van Buggenhout, N.: On generating Sobolev orthogonal polynomials. Numer. Math. **155**(3), 415–443 (2023)
26. Yu, X., Wang, Z., Li, H.: Jacobi Sobolev orthogonal polynomials and spectral methods for elliptic boundary value problems. Comm. Appl. Math. Comput. **1**, 283–308 (2019)

Authors and Affiliations

T. Laudadio[1] · N. Mastronardi[1] · F. J. Marcellán Español[2] · N. Van Buggenhout[2] · P. Van Dooren[3]

✉ T. Laudadio
teresa.laudadio@cnr.it

N. Mastronardi
nicola.mastronardi@cnr.it

F. J. Marcellán Español
pacomarc@ing.uc3m.es

N. Van Buggenhout
nvanbugg@math.uc3m.es

P. Van Dooren
paul.vandooren@uclouvain.be

[1] Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, via Amendola 122/D, Bari, Italy

[2] Department of Mathematics, University Carlos III de Madrid, Avenida de la Universidad, 30, Leganés, Madrid, Spain

[3] Department of Mathematical Engineering, Catholic University of Louvain, Avenue Georges Lemaitre 4, Louvain-la-Neuve, Belgium