Appl. Numer. Math. 207 (2025) 210–221

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

Fast and reliable algorithms for computing the zeros of Althammer polynomials

Teresa Laudadio [a,*], Nicola Mastronardi [a], Paul Van Dooren [b]

[a] Istituto per le Applicazioni del Calcolo "Mauro Picone", CNR, Via Amendola 122/D, Bari, 70126, Italy
[b] Department of Mathematical Engineering, Catholic University of Louvain, Batiment Euler (A.202), Avenue Georges Lemaitre 4, Louvain-la-Neuve, 1348, Belgium

ARTICLE INFO

MSC:
33C47
65D32
65F15

Keywords:
Sobolev orthogonal polynomials
Zeros
Hessenberg eigenvalue problem

ABSTRACT

In this manuscript, we propose a stable algorithm for computing the zeros of Althammer polynomials. These polynomials are orthogonal with respect to a Sobolev inner product, and are even if their degree is even, odd otherwise. Furthermore, their zeros are real, distinct, and located inside the interval $(-1, 1)$. The Althammer polynomial $p_n(x)$ of degree n satisfies a long recurrence relation, whose coefficients can be arranged into a Hessenberg matrix of order n, with eigenvalues equal to the zeros of the considered polynomial.
Unfortunately, the eigenvalues of this Hessenberg matrix are very ill–conditioned, and standard balancing procedures do not improve their condition numbers. Here, we introduce a novel algorithm for computing the zeros of $p_n(x)$, which first transforms the Hessenberg matrix into a similar symmetric tridiagonal one, i.e., a matrix whose eigenvalues are perfectly conditioned, and then computes the zeros of $p_n(x)$ as the eigenvalues of the latter tridiagonal matrix. Moreover, we propose a second algorithm, faster but less accurate than the former one, which computes the zeros of $p_n(x)$ as the eigenvalues of a truncated Hessenberg matrix, obtained by properly neglecting some diagonals in the upper part of the original matrix. The computational complexity of the proposed algorithms are, respectively, $\mathcal{O}(\frac{n^3}{6})$, and $\mathcal{O}(\ell^2 n)$, with $\ell \ll n$ in general.

1. Introduction

The analytic theory of orthogonal polynomials with respect to a Sobolev inner product associated with a vector of probability measures has received a lot of attention from the pioneering works by Althammer [1] and Schäfke [12], taking into account a 2–dimensional vector of measures whose components are the Lebesgue measure supported on the interval $(-1, 1)$.

From the point of view of approximation theory, in [7,9] they have been studied in a more general framework when the 2–dimensional vector of measures is a coherent pair. Therein, a simple algebraic connection formula between Sobolev orthogonal polynomials and the standard orthogonal polynomials with respect to the first measure has been stated. An overview on Sobolev orthogonal polynomials can be found in the survey [10].

In [14,4,2] the numerical aspects of Sobolev orthogonal polynomials have been studied and, more recently, in [13] the problem of generating a finite sequence of Sobolev orthogonal polynomials has been formulated as a Hessenberg inverse eigenvalue problem. The spectral information can be analyzed in terms of the Jordan matrix containing the eigenvalues of the Hessenberg matrix and the

* Corresponding author.
E-mail address: teresa.laudadio@cnr.it (T. Laudadio).

https://doi.org/10.1016/j.apnum.2024.09.004
Received 29 April 2024; Received in revised form 23 August 2024; Accepted 3 September 2024

Available online 5 September 2024

normalized first entries of its eigenvectors. This represents an extension of the well known Golub–Welsch approach [5] in the context of classical orthogonality theory.

In this work, we focus on the computation of the zeros of Althammer polynomials [1], which are orthogonal with respect to a Sobolev inner product with the Legendre weight on the interval $(-1,1)$. They are also called Legendre–Sobolev orthogonal polynomials [10].

Given an Althammer polynomial $p_n(x)$ of degree n, it is even if n is even, and it is odd otherwise. Furthermore, Althammer polynomials satisfy a higher order recurrence relation, and its zeros, real, distinct, and located inside the interval $(-1,1)$ [3,12], can be computed as the eigenvalues of a Hessenberg matrix of order n obtained by properly arranging the coefficients of the recurrence relation in matrix form.

Unfortunately, the eigenvalues of the aforementioned Hessenberg matrix are very ill–conditioned [13], and standard balancing procedures [11] do not improve their condition numbers.

In [13], the problem of computing the zeros of $p_n(x)$ was formulated as a Hessenberg inverse eigenvalue problem and two methods were proposed to solve it, one based on the Arnoldi iteration with full reorthogonalization and the other based on a procedure using plane rotations.

Here, we propose a stable algorithm, denoted by $\mathcal{A}_1$, for computing the zeros of $p_n(x)$, which first transforms the Hessenberg matrix into a similar symmetric tridiagonal one, and then computes the zeros of $p_n(x)$ as the eigenvalues of the latter tridiagonal matrix. The overall computational complexity is $\mathcal{O}(\frac{n^3}{6})$.

Moreover, observing that the absolute values of the entries different from zero, in each row of the upper triangular part of the Hessenberg matrix, are in a decreasing order and that many of them are very tiny, a faster modification of the proposed algorithm is described, denoted by $\mathcal{A}_2$. This computes the zeros of $p_n(x)$ as the eigenvalues of a truncated Hessenberg matrix, obtained by neglecting diagonals, with entries below a certain tolerance, in the upper part of the original matrix. Its computational complexity is $\mathcal{O}(\ell^2 n)$, with $\ell \ll n$ in general.

The paper is organized as follows. Notations are introduced in Section 2. The main features of Althammer polynomials are described in Section 3. The proposed algorithms are described in Section 4. Some numerical tests are reported in Section 6, followed by the conclusions.

2. Notations

Upper–case letters $A, B, \ldots$, denote matrices and $A_{m,n}$, or simply A_m if $m = n$, denotes matrices of size (m,n). The entry (i,j) of a matrix A is denoted by $a_{i,j}$.

Submatrices are denoted by the colon notation of `Matlab`, i.e., $A(i:j, k:l)$ is the submatrix of A obtained by the intersection of rows i to j and columns k to l, and $A(i:j,:)$ and $A(:,k:l)$ are the rows of A from i to j and the columns of A from k to l, respectively.

The diagonal matrix with entries $d_1, \ldots, d_n$ is denoted by $\texttt{diag}(d_1, \ldots, d_n)$.

Bold lower–case letters $\boldsymbol{x}, \boldsymbol{y}, \ldots, \boldsymbol{\omega}, \ldots$, denote vectors, and x_i denotes the ith element of the vector $\boldsymbol{x}$.

Lower–case letters $x, y, \ldots, \lambda, \theta, \ldots$, denote scalars.

The notation $\lfloor y \rfloor$ stands for the largest integer not exceeding $y \in \mathbb{R}_+$.

The identity matrix of order n is denoted by $\boldsymbol{I}_n$.

The vector $\boldsymbol{e}_{k,n}$, or simply $\boldsymbol{e}_k$ if there is no ambiguity, denotes the kth vector of the canonical basis of $\mathbb{R}^n$, i.e., $\boldsymbol{e}_{k,n} = \boldsymbol{I}_n(:,k)$.

The notation $\|\cdot\|_p$ denotes any of the p–norms.

3. Althammer polynomials

Althammer polynomials $p_n(x)$, $n = 0, 1, \ldots$, are orthogonal with respect to the Sobolev inner product

$$(v,w)_S = \int_{-1}^{1} v(x)w(x)dx + \gamma \int_{-1}^{1} v'(x)w'(x)dx, \qquad \gamma > 0. \tag{1}$$

The monic Althammer polynomial (MAP) $p_n(x)$ of degree $n \geq 0$ is odd if n is odd, and it is even if n is even (odd–even property) [1]. It satisfies the recurrence relation

$$p_n(x) = xp_{n-1}(x) - \sum_{i=0}^{\lfloor \frac{n-2}{2} \rfloor} h_{2i+i_0,n} p_{2i+i_0-1}(x), \qquad i_0 = \begin{cases} 1 & \text{if } n \text{ even}, \\ 2 & \text{if } n \text{ odd}, \end{cases} \tag{2}$$

where

$$h_{i,j} = \frac{(xp_{j-1}, p_{i-1})_S}{(p_{i-1}, p_{i-1})_S}. \tag{3}$$

Observe that, due to the odd–even property of MAPs, $h_{i,j} = 0$ if $i + j$ is even. The coefficients $h_{i,j}$ in (3) can be computed by the modified Chebyshev algorithm or by the Stieltjes algorithm in an accurate way [2,4,3].

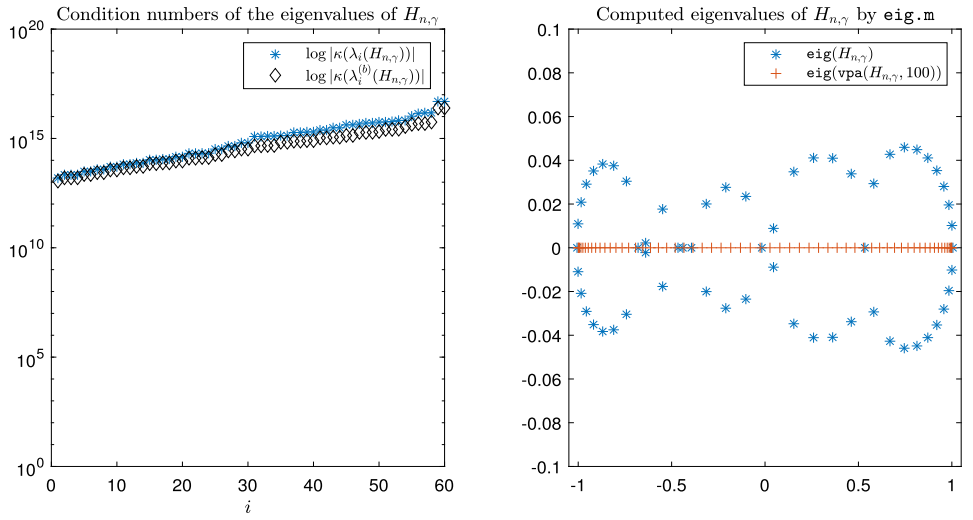

Fig. 1. Left: condition numbers of the eigenvalues λ_i of $H_{n,\gamma}$ ("$*$"), for $n = 60$, and $\gamma = 100$, and condition numbers of the eigenvalues $\lambda_i^{(b)}$, obtained applying the function `balance.m` of `Matlab` to $H_{n,\gamma}$ ("$\diamond$"), on a logarithmic scale. Right: eigenvalues of $H_{n,\gamma}$, for $n = 60$, and $\gamma = 100$, computed by `eig.m` of `Matlab` in double precision ("$*$") and variable precision with 100 digits ("$+$"). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Let H_γ be the semi–infinite matrix

$$H_\gamma = \begin{bmatrix} h_{1,2} & h_{1,4} & h_{1,6} & \\ h_{2,1} & h_{2,3} & h_{2,5} & \ddots \\ h_{3,2} & h_{3,4} & h_{3,6} & \\ & h_{4,3} & h_{4,5} & \ddots \\ & & h_{5,4} & \ddots \\ & & & \ddots \end{bmatrix}, \qquad h_{i+1,i} = 1, \quad i = 1, 2, \dots,$$

and

$$\boldsymbol{p}_n(x) = \begin{bmatrix} p_0(x), & p_1(x), & \dots & p_{n-1}(x), & p_n(x) \end{bmatrix}^T, \quad n \in \mathbb{N}.$$

Then, (2) can be written in a matrix form

$$\boldsymbol{p}_{n-1}^T(x) H_{n,\gamma} = x \boldsymbol{p}_{n-1}^T(x) - p_n(x) \boldsymbol{e}_n^T, \tag{4}$$

where $H_{n,\gamma} = H_\gamma(1:n, 1:n)$.

The zeros of $p_n(x)$ are real, distinct, and inside the interval $(-1, 1)$ [3,12]. Moreover, by (4), $\bar{x}$ is a zero of $p_n(x)$ if and only if $\bar{x}$ is an eigenvalue of $H_{n,\gamma}$. Therefore, the computation of the zeros of $p_n(x)$ reduces to that of the eigenvalues of $H_{n,\gamma}$.

Unfortunately, the eigenvalues of $H_{n,\gamma}$ are very ill–conditioned, independently of the choice of γ in (1), and the `Matlab` function `balance.m`, commonly used as a balancing technique [11], does not improve their condition numbers (see Fig. 1, left plot). As a consequence, the `Matlab` function `eig.m` yields unreliable results if used in double precision (see Fig. 1, right plot).

A novel algorithm for computing the zeros of $p_n(x)$, i.e., the eigenvalues of $H_{n,\gamma}$, in an efficient and accurate way in double precision is described in the next Section.

4. Computation of the zeros of Althammer polynomials

The algorithm we propose for computing the eigenvalues of $H_{n,\gamma}$ can be divided into three steps:

Step 1. Reduction algorithm transforming $H_{n,\gamma}$ into a similar tridiagonal one $T_{n,\gamma}$.
Step 2. Transformation of $T_{n,\gamma}$ into a similar symmetric tridiagonal one $\hat{T}_{n,\gamma}$.
Step 3. Computation of the eigenvalues of $\hat{T}_{n,\gamma}$.

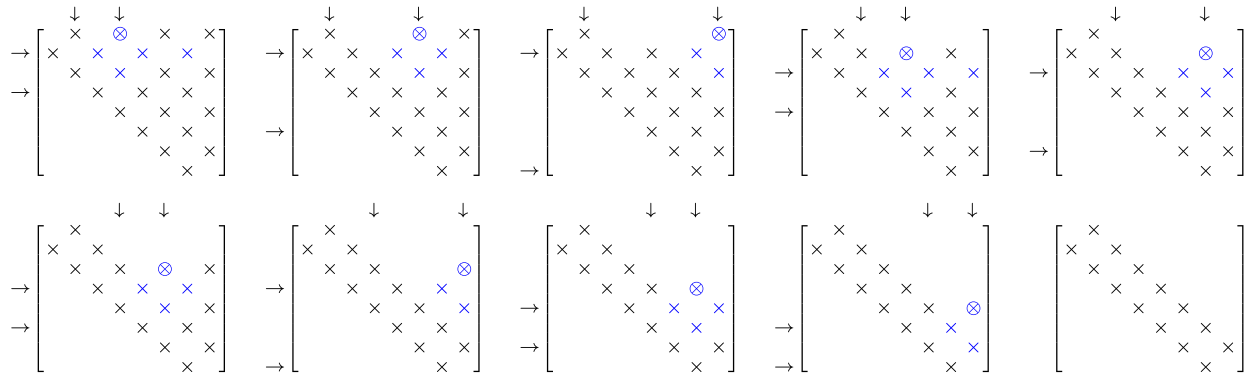

Fig. 2. Reduction of $H_{n,\gamma}$, $n = 8$, to the similar matrix $T_{n,\gamma}$ by a sequence of elementary similarity transformations. The nonzero entries of the matrix are denoted by "×". At each step, the rows and columns involved in the computation are indicated by the arrows "→" and "↓", respectively, the entry of the matrix to be eliminated is denoted by "⊗" and the modified entries are denoted by "×".

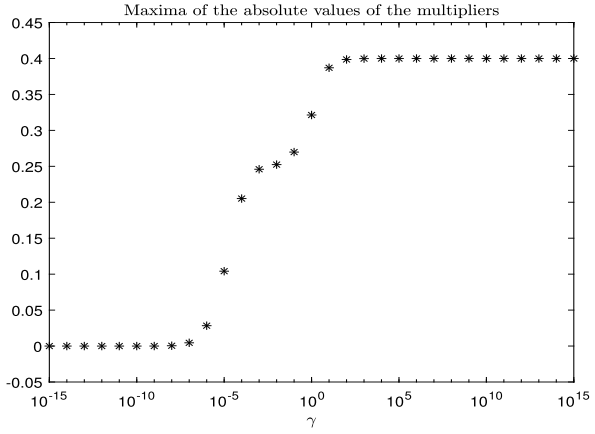

Fig. 3. Maxima of the absolute values of the multipliers involved in the reduction algorithm of the matrices $H_{n,\gamma}$, $n = 100$, $\gamma = 10^k$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$.

4.1. Step 1. Reduction algorithm transforming $H_{n,\gamma}$ into a similar tridiagonal one $T_{n,\gamma}$

This step is accomplished by applying a sequence of elementary Gaussian similarity transformations

$$M_{i,j} = I_n + \eta_{i+1,j} e_{i+1} e_j^T, \quad \begin{array}{l} i = 1, 2, \ldots, n-3, \\ j = i+3, i+5, \ldots, i+2\left\lfloor \frac{n-i-1}{2} \right\rfloor + i + 1, \end{array}$$

where $\eta_{i+1,j}$ are the so called multipliers [6, p. 112], such that

$$T_{n,\gamma} = R_{n,\gamma}^{-1} H_{n,\gamma} R_{n,\gamma}$$

is a tridiagonal matrix, with

$$R_{n,\gamma} = \prod_{i=1}^{n-3} \prod_{i+3}^{\left\lfloor \frac{n-i-1}{2} \right\rfloor + i + 1} M_{i,j}.$$

This step is depicted in Fig. 2 in a graphical fashion, and its computational complexity is $\mathcal{O}(\frac{n^3}{6})$.

The stability of this reduction depends on the size of the multipliers. In [6, p. 130] it is shown that, if the absolute values of all the multipliers are bounded by 1, $T_{n,\gamma}$ is computed in a stable way.

Example 1. The matrices $H_{n,\gamma}$ are computed by the modified Chebyshev algorithm, and the reduction algorithm is applied to them, evaluating the maximum of the multipliers in absolute value, for $n = 100$ and $\gamma = 10^k$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$. The results are displayed in Fig. 3. It can be noticed that, for all considered matrices, the maximum in absolute value of the multipliers is below 0.4. Therefore, in all considered cases, the tridiagonal matrices $T_{n,\gamma}$ are computed in a stable way.

Table 1
Matlab function `reduction.m`.

```
function[H,w,ell] = reduction(H,n)
% reduction of the upper Hessenberg matrix to a similar tridiagonal one
w=0;
for i=1:n-3
  for j=i+3:2:n
     v=-H(i,j)/H(i,i+1);
     w=max(w,abs(v));
     for k=i:2:i+2
       H(k,j)=v*H(k,i+1)+H(k,j);
     end
     for k=i+2:2:n
       H(i+1,k)=H(i+1,k)-v*H(j,k);
    end
  end
end
```

Table 2
Matlab function `DscaleT.m`.

```
function[T] = DscaleT(T,n)
% balancing the tridiagonal matrix T_{n,γ}
% input: T_{n,γ}
%        n, size of the matrix
% output: T̂_{n,γ}
for i=1:n-1
  T(i,i+1)=sqrt(T(i,i+1)*T(i+1,i));
  T(i+1,i)=T(i,i+1);
end
```

The `Matlab` function `reduction.m`, implementing this step, is described in Table 1.

4.2. Step 2. Transformation of $T_{n,\gamma}$ into a similar symmetric tridiagonal one $\hat{T}_{n,\gamma}$

Although the matrix $T_{n,\gamma}$ is tridiagonal with 0 on the main diagonal, the condition numbers of its eigenvalues are still comparable to those of the eigenvalues of $H_{n,\gamma}$. Since the eigenvalues of $T_{n,\gamma}$ are real, there exists a diagonal matrix D_n such that

$$\hat{T}_{n,\gamma} = D_n^{-1} T_{n,\gamma} D_n$$

is symmetric tridiagonal and, hence, its eigenvalues are perfectly conditioned.

The `Matlab` function `DscaleT.m`, implementing this step, is reported in Table 2.

4.3. Step 3. Computation of the eigenvalues of $\hat{T}_{n,\gamma}$

Let

$$\hat{T}_{n,\gamma} = \begin{bmatrix} & \hat{t}_1 & & & & & \\ \hat{t}_1 & & \hat{t}_2 & & & & \\ & \hat{t}_2 & & \ddots & & & \\ & & \ddots & & \hat{t}_{n-2} & & \\ & & & \hat{t}_{n-2} & & \hat{t}_{n-1} & \\ & & & & \hat{t}_{n-1} & & \end{bmatrix},$$

and let Z_n be the permutation matrix $Z_n = I_n(:, \boldsymbol{z}_n)$, where $\boldsymbol{z}_n = [1:2:n, 2:2:n]^T$. Then

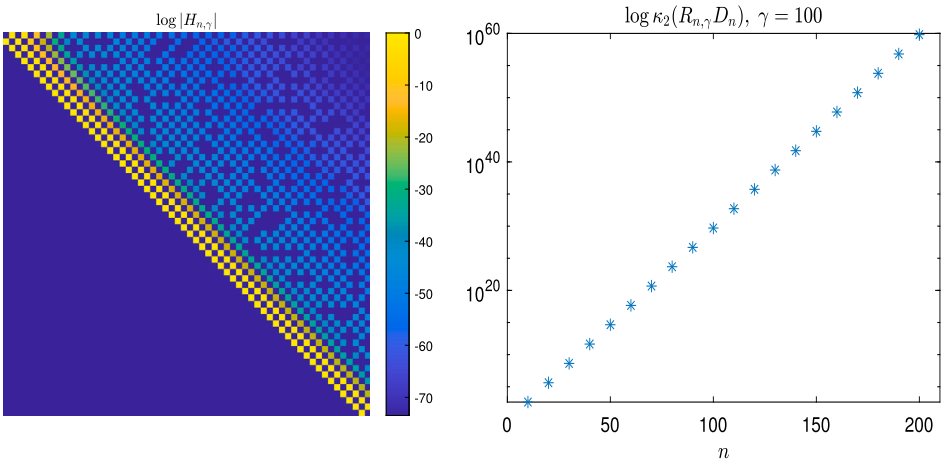

Fig. 4. Left: size of the absolute values of the entries of $H_{n,\gamma}$, for $n = 60$ and $\gamma = 100$, on a logarithmic scale. Right: condition numbers of $R_{n,\gamma}D_n$, $n = 10, 20, \ldots, 190, 200$, on a logarithmic scale.

$$Z_n^T \hat{T}_{n,\gamma} Z_n =: \left[\begin{array}{c|c} & B_{\hat{n}} \\ \hline B_{\hat{n}}^T & \end{array}\right] = \begin{cases} \left[\begin{array}{cccccc|cccccc} & & & & & & \hat{t}_1 & & & & & \\ & & & & & & \hat{t}_2 & \hat{t}_3 & & & & \\ & & & & & & & \ddots & \ddots & & & \\ & & & & & & & & & \hat{t}_{n-2} & \hat{t}_{n-1} & \\ \hline \hat{t}_1 & \hat{t}_2 & & & & & & & & & & \\ & \hat{t}_3 & \ddots & & & & & & & & & \\ & & \ddots & \hat{t}_{n-2} & & & & & & & & \\ & & & \hat{t}_{n-1} & & & & & & & & \end{array}\right], & \text{if } n \text{ even}, \\ \left[\begin{array}{ccccccc|cccccc} & & & & & & & \hat{t}_1 & & & & & \\ & & & & & & & \hat{t}_2 & \hat{t}_3 & & & & \\ & & & & & & & & \ddots & \ddots & & & \\ & & & & & & & & & \hat{t}_{n-3} & \hat{t}_{n-2} & \\ & & & & & & & & & & \hat{t}_{n-1} & \\ \hline \hat{t}_1 & \hat{t}_2 & & & & & & & & & & & \\ & \hat{t}_3 & \ddots & & & & & & & & & & \\ & & \ddots & \hat{t}_{n-3} & & & & & & & & & \\ & & & \hat{t}_{n-2} & \hat{t}_{n-1} & & & & & & & & \end{array}\right], & \text{if } n \text{ odd}. \end{cases}$$

with $B_{\hat{n}}$ bidiagonal, $B_{\hat{n}} \in \mathbb{R}^{\bar{m}\times\bar{n}}$, where $\bar{m} = \bar{n} = \lfloor \frac{n}{2} \rfloor$ if n even, $\bar{m} = \lfloor \frac{n}{2} \rfloor + 1$, $\bar{n} = \lfloor \frac{n}{2} \rfloor$ if n odd.

Let $\sigma_1 > \sigma_2 > \cdots > \sigma_{\lfloor \frac{n}{2} \rfloor}$ be the nonzero singular values of $B_{\hat{n}}$. Then, there exists an orthogonal matrix $Q_n \in \mathbb{R}^{n\times n}$ such that [6, p. 486]

$$Q_n^T \left[\begin{array}{c|c} & B_{\hat{n}} \\ \hline B_{\hat{n}}^T & \end{array}\right] Q_n = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\lfloor \frac{n}{2} \rfloor}, -\sigma_1, -\sigma_2, \ldots, -\sigma_{\lfloor \frac{n}{2} \rfloor}, \underbrace{0}_{\bar{m}-\bar{n}}).$$

Therefore, the positive eigenvalues of $H_{n,\gamma}$ can be computed as the singular values of $B_{\hat{n}}$ by the function `svd.m`, available in `Matlab`, with $\mathcal{O}(\frac{n^2}{2})$ computational complexity.

Remark 1. The size of the absolute values of the entries of $H_{n,\gamma}$, for $n = 60$ and $\gamma = 100$, computed by the modified Chebyshev algorithm [2,4,3], are displayed in Fig. 4 (left) on a logarithmic scale. It can be observed that

$$|h_{i,j}| \geq |h_{i,j+2}|, \qquad i = 1, \ldots, n-1,\ \ j = i+1, \ldots, n-2,\ \ i+j \text{ odd},$$

i.e., the absolute values of the entries different from zero, in each row of the upper triangular part of $H_{n,\gamma}$, are in a decreasing order. Moreover, many of them are very tiny.

Table 3
Matlab function `reduction1.m`.

```
function[H,w,ell] = reduction1(H,n,tol)
% reduction of the truncated upper Hessenberg matrix to a similar tridiagonal one
ell=0;
j=4;
while ell==0 & j < n
   if abs(H(1,j))>tol
      j=j+2;
   else
      ell=j;
   end
end
w=0;
for i=1:n-3
   for j=i+3:2:min(i+ell-1,n)
      v=-H(i,j)/H(i,i+1);
      w=max(w,abs(v));
      for k=i:2:i+2
         H(k,j)=v*H(k,i+1)+H(k,j);
      end
      for k=i+2:2:min(i+ell-1,n)
         H(i+1,k)=H(i+1,k)-v*H(j,k);
      end
   end
end
```

Let $E_{n,\gamma,\tau}$ be the matrix whose elements are those of $H_{n,\gamma}$ that are, in absolute value, below a certain tolerance τ. Then, by the Bauer–Fike theorem [6, p. 357], if λ is an eigenvalue of $H_{n,\gamma}$ and μ is an eigenvalue of $H_{n,\gamma} - E_{n,\gamma,\tau}$,

$$|\lambda - \mu| \le \kappa_2(X_n)\|E_{n,\gamma,\tau}\|_2,$$

where $X_n = R_{n,\gamma} D_n Z_n Q_n$.

In Fig. 4 (right), the condition numbers $\kappa_2(R_{n,\gamma} D_n)$, for $n = 10, 20, \ldots, 200$, and $\gamma = 100$, are displayed on a logarithmic scale. We point out that $\kappa_2(R_{n,\gamma} D_n) = \kappa_2(X_n)$.

It can be noticed that $\kappa_2(R_{n,\gamma} D_n)$ grows as $10^{\frac{3}{10}n-1}$. Extensive tests of the same type were carried out by considering different values of γ and the condition numbers of $R_{n,\gamma} D_n$ exhibited a similar behavior.

Let $\tilde{H}_{n,\gamma,\tau} := H_{n,\gamma} - E_{n,\gamma,\tau}$. Hence, the eigenvalues of $\tilde{H}_{n,\gamma,\tau}$ can differ from those of $H_{n,\gamma}$ by a quantity that depends on $\kappa_2(R_{n,\gamma} D_n)$, although this can be an overestimate.

Therefore, neglecting entries of $H_{n,\gamma}$ that are below a certain tolerance τ in absolute value, would reduce the overall complexity of the numerical method, but it could affect the accuracy of the eigenvalues.

Nevertheless, here we propose a simple criterion to determine which entries of $H_{n,\gamma}$ can be discarded, given a certain tolerance τ, by just looking for the column index ℓ in the first row corresponding to the first nonzero entry, below τ in absolute value, i.e.,

$$|h_{1,\ell-2}| \ge \tau \quad \text{and} \quad |h_{1,\ell}| < \tau.$$

Then, the reduction algorithm is applied to the banded Hessenberg matrix $\texttt{triu}(H_{n,\ell-2})$. The corresponding `Matlab` function `reduction1.m`, requiring $\mathcal{O}(\ell^2 n)$ computational complexity, is described in Table 3.

5. On the stability of the proposed algorithm

If an algorithm for computing the eigenvalues of an $n \times n$ diagonalizable matrix H produces the matrix $\hat{\Lambda} := \text{diag}(\hat{\lambda}_1, ..., \hat{\lambda}_n)$ of (appropriately ordered) approximate eigenvalues $\hat{\lambda}_i$, such that

$$(H + \Delta)\hat{X} = \hat{X}\hat{\Lambda}, \quad \text{where} \quad \hat{X}^{-1}\hat{X} = I_n \quad \text{and} \quad \|\Delta\|_2 \le \mathcal{O}(\epsilon)\|H\|_2, \tag{5}$$

then the algorithm is said to be *backward stable*, because the computed eigenvalues $\hat{\lambda}_i$ are the exact eigenvalues of a slightly perturbed matrix $H + \Delta$, where Δ is known as the *backward error*.

Let us consider the following modification of the Eckhart–Young Theorem [6, p. 79].

Lemma 1. *Let the smallest singular value of the $n \times n$ matrix $H - \hat{\lambda}_i I_n$ be equal to σ_n, then the smallest perturbation Δ to make the matrix $H + \Delta - \hat{\lambda}_i I_n$ singular has norm $\|\Delta\|_2$ equal to σ_n, and is given by $\Delta = u_n \sigma_n v_n^T$, where u_n and v_n are the singular vectors of $H - \lambda_i I_n$, corresponding to σ_n.*

We can not prove the backward stability for the proposed algorithm, but the next best property shows that for each computed eigenvalue $\hat{\lambda}_i$, there exists a backward error Δ_i such that

$$(H+\Delta_i)\hat{x}_i = \hat{x}_i\hat{\lambda}_i, \quad \text{where} \quad \hat{x}_i^T\hat{x}_i = 1 \quad \text{and} \quad \|\Delta_i\|_2 \le \mathcal{O}(\epsilon)\|H\|_2. \tag{6}$$

According to Lemma 1, this property is guaranteed if for each eigenvalue $\hat{\lambda}_i$, $\sigma_n(H-\hat{\lambda}_i I_n) = \epsilon_i\|H\|_2$, with $\epsilon_i = \mathcal{O}(\epsilon)$, since, in this case, we can set $\Delta_i := \hat{x}_i\epsilon_i y_i^T\|H\|_2$, where $\hat{x}_i$ and $\hat{y}_i$ are the normalized singular vectors associated with the singular value $\epsilon_i\|H\|_2$ of $(H-\hat{\lambda}_i I_n)$. The property (5) follows from property (6) if and only if there exists a common perturbation matrix Δ for all eigenvalues $\hat{\lambda}_i$ in (6). Furthermore, from Lemma 1 it follows that

$$\|\Delta\|_2 \ge \max_i \sigma_n(H-\hat{\lambda}_i I_n).$$

Therefore, the corresponding algorithm will be backward stable only if

$$\max_i \sigma_n(H-\hat{\lambda}_i I_n) = \mathcal{O}(\epsilon). \tag{7}$$

It is worth stressing that test (7) is only necessary for the backward stability of the algorithm, but it is not sufficient. Nevertheless, it is a reasonable indication that the algorithm might be backward stable. In Section 6, test (7) is performed successfully for all the considered matrices.

If an algorithm for computing the eigenvalues of an $n\times n$ diagonalizable matrix H produces the matrix $\hat{\Lambda}$ of (appropriately ordered) approximate eigenvalues $\hat{\lambda}_i$, such that

$$\max_i(|\lambda_i - \hat{\lambda}_i|) := \|\Lambda - \hat{\Lambda}\|_2 = \mathcal{O}(\epsilon)\|H\|_2,$$

then the algorithm is forward stable in an absolute sense. If, on the other hand,

$$\max_i \left|\frac{\lambda_i - \hat{\lambda}_i}{\lambda_i}\right| := \frac{\|(\Lambda-\hat{\Lambda})\|_2}{\|\Lambda\|_2} = \mathcal{O}(\epsilon),$$

then the algorithm is forward stable in a relative sense. In Section 6 we analyze this for the different algorithms described in this paper.

6. Numerical examples

The proposed numerical method, and its variant based on computing the eigenvalues of $\tilde{H}_{n,\gamma,\tau}$, denoted respectively by $\mathcal{A}_1$ and $\mathcal{A}_2$, are compared with the two algorithms proposed in [13], denoted respectively by $\mathcal{A}_3$ and $\mathcal{A}_4$, in terms of efficiency and accuracy.

In these first set of tests we analyze the absolute and relative forward stability of the method, since we look at the absolute and relative forward error in the computed eigenvalues[1]

The entries of $H_{n,\gamma}$ are computed by the modified Chebyshev algorithm [2,4,3]. Its eigenvalues, computed by the `Advanpix Multiprecision Computing Toolbox`[2] for `Matlab` [8] with precision of 200 digits and rounded in double precision by the `Matlab` function `double.m`, are considered as the exact ones and denoted by $\lambda_i^{(\mathcal{A}_0)}$, $i=1,\dots,n$.

These eigenvalues are compared with those computed by $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$, denoted respectively by $\lambda_i^{(\mathcal{A}_1)}, \lambda_i^{(\mathcal{A}_2)}, \lambda_i^{(\mathcal{A}_3)}$, and $\lambda_i^{(\mathcal{A}_4)}$, $i=1\dots,n$, such that

$$\lambda_i^{(\mathcal{A}_k)} < \lambda_{i+1}^{(\mathcal{A}_k)}, \qquad i=1,\dots,n-1, \quad k=0,\dots,4.$$

The considered matrices are $H_{n,\gamma}$, with $\gamma = 10^k$, $k=-15,-14,\dots,-1,0,1,\dots,14,15$, and $n=100,200,300$. Their norm $\|H_{n,\gamma}\|_2$ is always of the order of 1.

The accuracy and the efficiency of the considered methods, for $n=100,200,300$, are reported in Fig. 5, Fig. 6, and Fig. 7, respectively. For each considered matrix, the maxima of the absolute errors of the computed eigenvalues applying the considered methods

$$\max_{i=1,\dots,n} |(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|, \qquad i=1,\dots,n, \quad i=1,\dots,4,$$

and the execution times are displayed in the figures (left, and right, respectively).

Concerning the algorithm $\mathcal{A}_2$, the tolerance is fixed to the machine precision $\epsilon \approx 2.22\times 10^{-16}$. The number ℓ of considered diagonals are reported in Fig. 8.

We can observe that, although $\mathcal{A}_2$ is the fastest algorithm, the computed eigenvalues are not accurate if $\gamma \le 0$, while the algorithm $\mathcal{A}_1$ is the most accurate and slightly slower than $\mathcal{A}_2$.

In these experiments, the algorithm $\mathcal{A}_1$ appears to be forward stable, both in an absolute and relative sense. But this of course is not a proof that the algorithm is forward stable.

[1] We consider $H_{n,\gamma}$ matrices with n even, since 0 is an eigenvalue of $H_{n,\gamma}$ for n odd, and the relative forward error cannot be computed. On the other hand, for the proposed algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, the eigenvalue 0 is exactly detected when n is odd.

[2] 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.

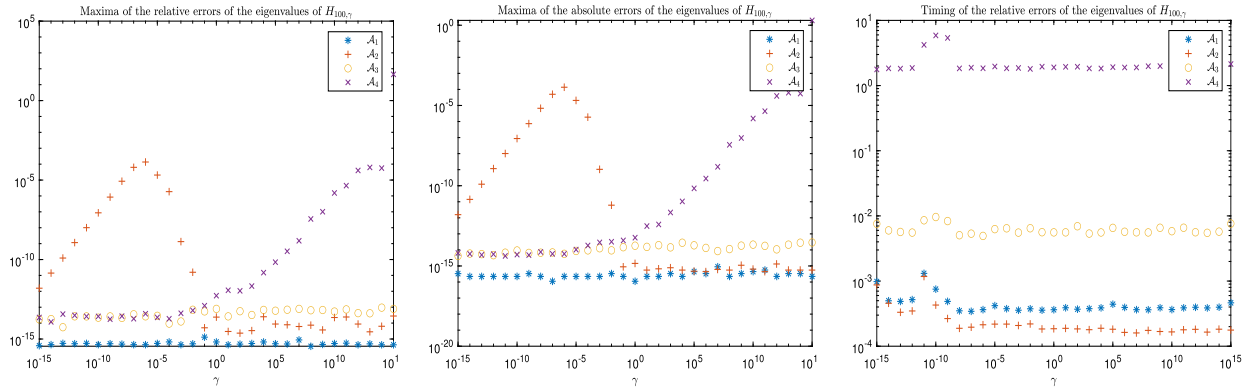

Fig. 5. Left and center: maxima of $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|/|(\lambda_i^{(\mathcal{A}_0)}|$ and $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|$, respectively, $k = 1, \ldots, 100$, $i = 1, \ldots, 4$, of $H_{100,\gamma}$ computed by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, for $\gamma = 10^k$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$, on a logarithmic scale. Right: execution time in seconds for computing the whole spectrum of $H_{100,\gamma}$ by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, on a logarithmic scale.

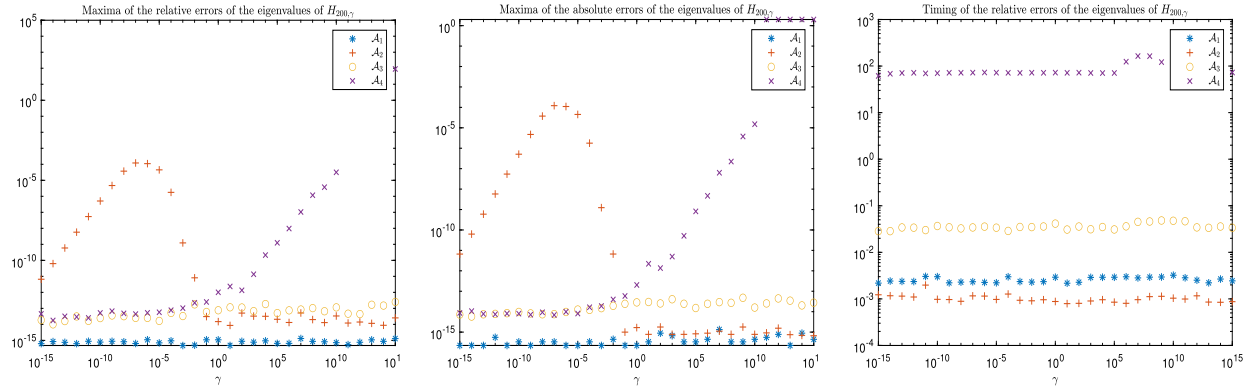

Fig. 6. Left and center: maxima of $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|/|(\lambda_i^{(\mathcal{A}_0)}|$ and $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|$, respectively, $k = 1, \ldots, 200$, $i = 1, \ldots, 4$, of $H_{200,\gamma}$ computed by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, for $\gamma = 10^k$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$, on a logarithmic scale. Right: execution time in seconds for computing the whole spectrum of $H_{200,\gamma}$ by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, on a logarithmic scale.

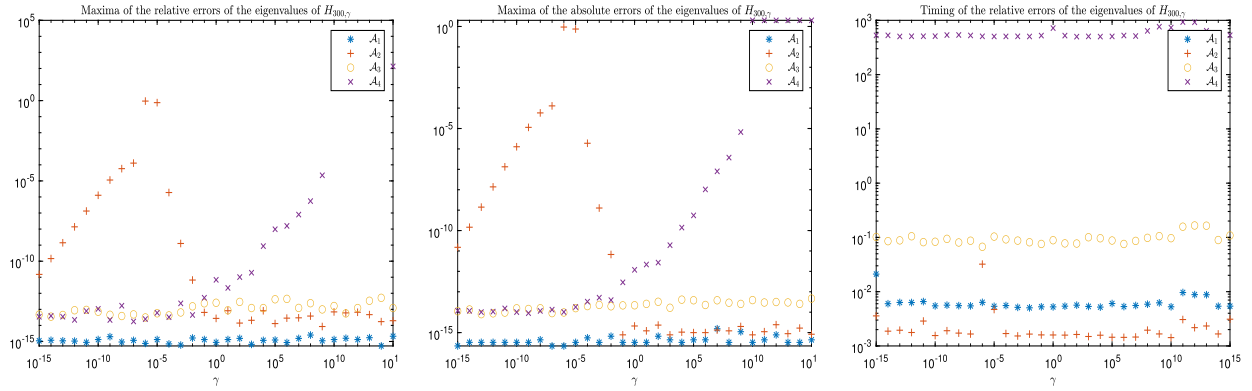

Fig. 7. Left and center: maxima of $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|/|(\lambda_i^{(\mathcal{A}_0)}|$ and $|(\lambda_i^{(\mathcal{A}_0)} - \lambda_i^{(\mathcal{A}_k)})|$, respectively, $k = 1, \ldots, 300$, $i = 1, \ldots, 4$, of $H_{300,\gamma}$ computed by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, for $\gamma = 10^k$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$, on a logarithmic scale. Right: execution time in seconds for computing the whole spectrum of $H_{300,\gamma}$ by the methods $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$, on a logarithmic scale.

To check the backward stability of the proposed method $\mathcal{A}_1$, for each matrix $H_{n,\gamma}$, $n = 100$, 200, and 300, $\gamma = 10^{-k}$, $k = -15, -14, \ldots, -1, 0, 1, \ldots, 14, 15$, we perform test (7).

For each computed eigenvalue $\lambda_i^{(\mathcal{A}_1)}$, $i = 1, \ldots, n$, the smallest singular value of $H_{n,\gamma} - \lambda_i^{(\mathcal{A}_1)} I_n$ is computed and then the maximum $r_{\gamma,n}$ is taken, i.e.,

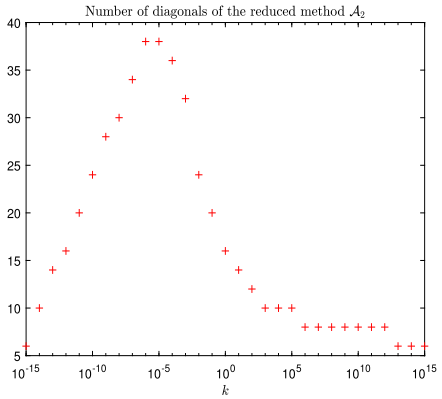

Fig. 8. Number of diagonals selected by $\mathcal{A}_2$ in the matrices $H_{n,\gamma}$, $n = 100, 200, 300$, $\gamma = 10^{-k}$, $k = -15, -14, \dots, -1, 0, 1, \dots, 14, 15$.

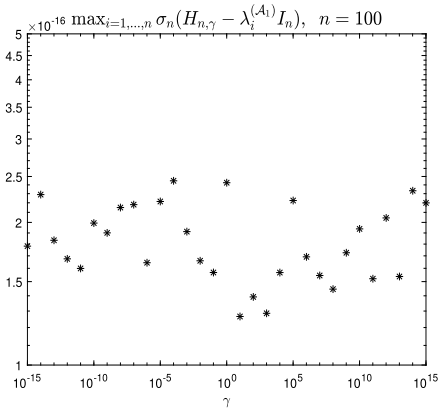

Fig. 9. $\max_{i=1,\dots,n} \sigma_n(H_{n,\gamma} - \lambda_i^{(\mathcal{A}_1)} I_n)$, $\gamma = 10^{-k}$, $k = -15, -14, \dots, -1, 0, 1, \dots, 14, 15.$, $n = 100$.

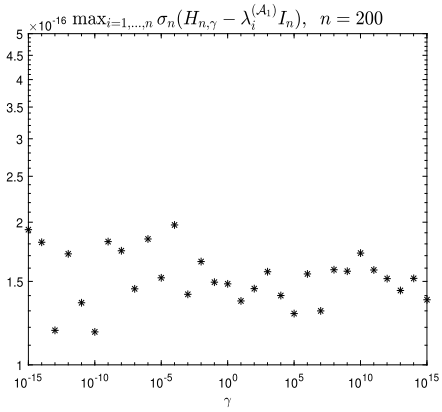

Fig. 10. $\max_{i=1,\dots,n} \sigma_n(H_{n,\gamma} - \lambda_i^{(\mathcal{A}_1)} I_n)$, $\gamma = 10^{-k}$, $k = -15, -14, \dots, -1, 0, 1, \dots, 14, 15.$, $n = 200$.

$$r_{\gamma,n} = \max_{i=1,\dots,n} \sigma_n(H_{n,\gamma} - \lambda_i^{(\mathcal{A}_1)} I_n), \quad \gamma = 10^{-k},\ k = -15, -14, \dots, -1, 0, 1, \dots, 14, 15.$$

The results are depicted in Figs. 9, 10, and 11, for $n = 100$, 200, and 300, respectively.
In all considered cases, the value of $r_{\gamma,n}$ is of the order of the machine precision.

7. Conclusions

A novel algorithm for computing the zeros of Althammer polynomials has been proposed. It relies on the computation of the eigenvalues of the associated Hessenberg matrix, which are very ill–conditioned.

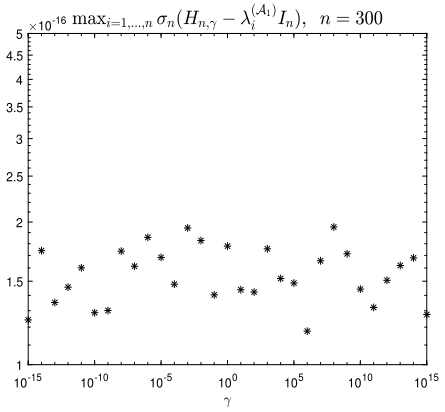

Fig. 11. $\max_{i=1,\dots,n} \sigma_n(H_{n,\gamma} - \lambda_i^{(\mathcal{A}_1)} I_n)$, $\gamma = 10^{-k}$, $k = -15, -14, \dots, -1, 0, 1, \dots, 14, 15.$, $n = 300$.

This problem is overcome transforming the given Hessenberg matrix into a similar symmetric tridiagonal one. The proposed algorithm has $\mathcal{O}(\frac{n^3}{6})$ computational complexity.

The performance of the novel algorithm is compared with that of other algorithms recently proposed in the literature, in terms of accuracy and efficiency.

Extensive tests show that the former method outperforms the latter ones.

A more efficient but less accurate variant of the novel algorithm is also proposed, consisting in neglecting some diagonals of the Hessenberg matrix, with $\mathcal{O}(\ell^2 n)$, $\ell \ll n$, computational complexity, where ℓ denotes the number of kept diagonals in the Hessenberg matrix.

Finally, it is worth stressing that the proposed algorithm is tailored to compute the eigenvalues of the Hessenberg matrices $H_{n,\gamma}$, since all the involved elementary matrices constructed in Step 1 of the algorithm have the absolute values of the multipliers bounded by 1. For other cases of Sobolev orthogonal polynomials having only real zeros, the key point is to apply a similarity transformation to the involved Hessenberg matrix, in order to transform its main tridiagonal part into a symmetric tridiagonal matrix, i.e., Step 2 of the proposed algorithm should be performed first. This step should reduce the condition numbers of the eigenvalues of the Hessenberg matrix. Then, the eigenvalues of the transformed matrix can be computed, for instance, by the QR method.

CRediT authorship contribution statement

Teresa Laudadio: Writing – review & editing, Writing – original draft, Validation, Software, Methodology. **Nicola Mastronardi:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology. **Paul Van Dooren:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology.

Acknowledgements

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 authors thank the anonymous Reviewers for their valuable comments that helped to improve the content of the paper.

References

[1] P. Althammer, Eine Erweiterung des Orthogonalitätsbegriffes bei Polynomen und deren Anwendung auf die beste Approximation, J. Reine Angew. Math. 211 (1962) 192–204.
[2] W. Gautschi, M. Zhang, Computing orthogonal polynomials in Sobolev spaces, Numer. Math. 71 (2) (1995) 159–183.
[3] W. Gautschi, Orthogonal polynomials (in Matlab), J. Comput. Appl. Math. 178 (1–2) (2005) 215–234.
[4] W. Gautschi, Orthogonal polynomials, in: Computation and Approximation, in: Numerical Mathematics and Scientific Computation Series, Oxford University Press, New York, 2004.
[5] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, Math. Comput. 23 (1969) 221–230.
[6] G.H. Golub, C.F. Van Loan, Matrix Computations, 4th edition, Johns Hopkins University Press, Baltimore, 2013.
[7] A. Iserles, J.M. Sanz–Serna, P.E. Koch, S.P. Nørsett, Orthogonality and approximation in a Sobolev space, in: J.C. Mason, M.G. Cox (Eds.), Algorithms for Approximation, Chapman and Hall, London, 1990, pp. 117–124.
[8] Multiprecision Computing Toolbox for MATLAB, Advanpix LLC., Yokohama, Japan, 2015.
[9] A. Iserles, P.E. Koch, S.P. Nørsett, J.M. Sanz–Serna, On polynomials orthogonal with respect to certain Sobolev inner products, J. Approx. Theory 65 (1991) 151–175.
[10] F. Marcellán, Y. Xu, On Sobolev orthogonal polynomials, Expo. Math. 33 (2015) 308–352.

[11] B.N. Parlett, C. Reinsch, Balancing a matrix for calculation of eigenvalues and eigenvectors, Numer. Math. 13 (1969) 293–304.
[12] F.W. Schäfke, Zu den Orthogonalpolynomen von Althammer, J. Reine Angew. Math. 252 (1972) 195–199.
[13] N. Van Buggenhout, On generating Sobolev orthogonal polynomials, Numer. Math. 155 (2023) 415–443.
[14] M. Zhang, Sensitivity analysis for computing orthogonal polynomials of Sobolev type, in: R.V.M. Zahar (Ed.), Approximation and Computation, vol. 119, Birkhäuser Verlag, Boston, 1994, pp. 563–576.