

Accurate eigenvalue decomposition of arrowhead matrices and applications

N. Jakovčević Stor^{a,1,*}, I. Slapničar^{a,1}, J. Barlow^{b,2}

^a*Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture,
University of Split, Rudjera Boškovića 32, 21000 Split, Croatia*

^b*Department of Computer Science and Engineering, The Pennsylvania State University,
University Park, PA 16802-6822, USA*

Abstract

We present a new algorithm for solving an eigenvalue problem for a real symmetric arrowhead matrix. The algorithm computes all eigenvalues and all components of the corresponding eigenvectors with high relative accuracy in $O(n^2)$ operations. The algorithm is based on a shift-and-invert approach. Double precision is eventually needed to compute only one element of the inverse of the shifted matrix. Each eigenvalue and the corresponding eigenvector can be computed separately, which makes the algorithm adaptable for parallel computing. Our results extend to Hermitian arrowhead matrices, real symmetric diagonal-plus-rank-one matrices and singular value decomposition of real triangular arrowhead matrices.

Keywords: eigenvalue decomposition, arrowhead matrix, high relative accuracy, singular value decomposition

2000 MSC: 65F15

1. Introduction and Preliminaries

In this paper we consider eigenvalue problem for a real symmetric matrix A which is zero except for its main diagonal and one row and column. Since eigenvalues are invariant under similarity transformations, we can symmetrically permute the rows and the columns of the given matrix. Therefore, we assume without loss of generality that the matrix A is a $n \times n$ real symmetric arrowhead matrix of the form

*Corresponding author

Email addresses: nevena@fesb.hr (N. Jakovčević Stor), ivan.slapnicar@fesb.hr (I. Slapničar), barlow@cse.psu.edu (J. Barlow)

¹The research of Ivan Slapničar and Nevena Jakovčević Stor was supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant 023-0372783-1289.

²The research of Jesse L. Barlow was supported by the National Science Foundation under grant CCF-1115704.

$$A = \begin{bmatrix} D & z \\ z^T & \alpha \end{bmatrix}, \quad (1)$$

where

$$D = \text{diag}(d_1, d_2, \dots, d_{n-1})$$

is diagonal matrix of order $n - 1$,

$$z = [\zeta_1 \quad \zeta_2 \quad \cdots \quad \zeta_{n-1}]^T \quad (2)$$

is a vector and α is a scalar.

Such matrices arise in the description of radiationless transitions in isolated molecules [3], oscillators vibrationally coupled with a Fermi liquid [8], quantum optics [15] (see also Example 4). Such matrices also arise in solving symmetric real tridiagonal eigenvalue problems with the divide-and-conquer method [11].

In this paper we present an algorithm which computes all eigenvalues and all components of the corresponding eigenvectors with high relative accuracy in $O(n^2)$ operations.

Without loss of generality we may assume that A is irreducible, that is,

$$\zeta_i \neq 0, \text{ for all } i$$

and

$$d_i \neq d_j, \text{ for all } i \neq j, i, j = 1, \dots, n - 1.$$

If A has a zero in the last column, say $\zeta_i = 0$, then the diagonal element d_i is an eigenvalue whose corresponding eigenvector is the i -th unit vector, and we can reduce the size of the problem by deleting the i -th row and column of the matrix, eventually obtaining a matrix for which all elements ζ_j are nonzero. If $d_i = d_j$, then d_i is eigenvalue of matrix A (this follows from the interlacing property (7)), and we can reduce the size of the problem by annihilating ζ_j with a Givens rotation in the (i, j) -plane and proceeding as in the previous case.

Further, by symmetric row and column pivoting, we can order elements of D such that

$$d_1 > d_2 > \cdots > d_{n-1}. \quad (3)$$

Hence, we will consider only ordered and irreducible arrowhead matrices. Without loss of generality we can also assume that $\zeta_i > 0$ for all i , which can be attained by pre- and post-multiplication of the matrix A with $D = \text{diag}(\text{sign}(\zeta_i))$.

Let

$$A = V\Lambda V^T \quad (4)$$

be the eigenvalue decomposition of A . Here

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$$

is a diagonal matrix whose diagonal elements are the eigenvalues of A , and

$$V = [v_1 \quad \cdots \quad v_n]$$

is an orthonormal matrix whose columns are the corresponding eigenvectors.

The eigenvalues of A are the zeros of the Pick function (see [4, 16])

$$f(\lambda) = \alpha - \lambda - \sum_{i=1}^{n-1} \frac{\zeta_i^2}{d_i - \lambda} = \alpha - \lambda - z^T (D - \lambda I)^{-1} z, \quad (5)$$

and the corresponding eigenvectors are given by

$$v_i = \frac{x_i}{\|x_i\|_2}, \quad x_i = \begin{bmatrix} (D - \lambda_i I)^{-1} z \\ -1 \end{bmatrix}, \quad i = 1, \dots, n. \quad (6)$$

Diagonal elements of the matrix D , d_i , are called poles of the function f .

Notice that (3) and the Cauchy interlacing theorem [10, Theorem 8.1.7] applied to matrices D and A imply the interlacing property

$$\lambda_1 > d_1 > \lambda_2 > d_2 > \dots > d_{n-2} > \lambda_{n-1} > d_{n-1} > \lambda_n. \quad (7)$$

Since A is symmetric, its eigenvalues may be computed by invoking any of a number of standard programs (LAPACK [1]). However, these programs usually begin with an initial reduction of the matrix to tridiagonal form [17], or as proposed in [16], with an alternative which takes advantage of the structure of A by finding the zeros of the Pick function given in (5), for the eigenvalues of A . This results in an algorithm which requires only $O(n^2)$ computations and $O(n)$ storage. Although the idea is conceptually simple and in fact has been used to solve other eigenvalue problems of special structure [2, 5, 6, 7], the computation is not always stable [11]. Namely, if the computed eigenvalues λ_i are not accurate enough, then the computed eigenvectors v_i may not be sufficiently orthogonal (see Example 3). The existing algorithms for arrowhead matrices [11, 16] obtain orthogonal eigenvectors with the following procedure:

- compute the eigenvalues $\tilde{\lambda}_i$ of A by solving (5);
- construct a new matrix

$$\tilde{A} = \begin{bmatrix} D & \tilde{z} \\ \tilde{z}^T & \tilde{\alpha} \end{bmatrix}$$

by solving inverse problem with the prescribed eigenvalues $\tilde{\lambda}$, and diagonal matrix D , that is, compute new \tilde{z} and $\tilde{\alpha}$ as

$$\tilde{\zeta}_i = \sqrt{(d_i - \tilde{\lambda}_n) (\tilde{\lambda}_1 - d_i) \prod_{j=2}^i \frac{(\tilde{\lambda}_j - d_i)}{(d_{j-1} - d_i)} \prod_{j=i+1}^{n-1} \frac{(\tilde{\lambda}_j - d_i)}{(d_j - d_i)}},$$

$$\tilde{\alpha} = \tilde{\lambda}_n + \sum_{j=1}^{n-1} (\tilde{\lambda}_j - d_j).$$

- compute eigenvectors of \tilde{A} by (6).

Since the formulas for $\tilde{\zeta}_i$ involve only multiplications, division and subtractions of exact quantities, each ζ_i is computed with relative error of $O(\varepsilon_M)$, where ε_M denotes the machine precision.³ Therefore, $\tilde{A} = A + \delta A$, where $\|\delta A\|_2 = O(\varepsilon_M)$. Here $\|\cdot\|_2$ denotes the spectral matrix norm. We conclude that the computed eigenvalues $\tilde{\lambda}_i$ satisfy standard perturbation bounds like those from [10, Corollary 8.1.6]. Further, since $\tilde{\lambda}_i$ are the eigenvalues of the matrix \tilde{A} computed to higher relative accuracy, the eigenvectors computed by (6) are orthogonal to machine precision. For details see [11, 16].

Our algorithm uses a different approach. Accuracy of the eigenvectors and their orthogonality follows from high relative accuracy of the computed eigenvalues and there is no need for follow-up orthogonalization. The algorithm is based on shift-and-invert technique. Basically, the eigenvalue λ is computed as the largest or the smallest eigenvalue of the inverse of the matrix shifted to the pole d_i which is nearest to λ , that is,

$$\lambda = \frac{1}{\nu} + d_i, \quad (8)$$

where ν is either smallest or largest eigenvalue of the matrix

$$A_i^{-1} \equiv (A - d_i I)^{-1}.$$

Inverses of arrowhead matrices are structured in the following manner (here \times stands for non-zero element): the inverse of an arrowhead matrix with zero on the shaft is a permuted arrowhead matrix with zero on the shaft,

$$\begin{bmatrix} \times & & & \times \\ & \times & & \times \\ & & 0 & \times \\ & & & \times \\ \times & \times & \times & \times \end{bmatrix}^{-1} = \begin{bmatrix} \times & & & \times \\ & \times & \times & \\ \times & \times & \times & \times \\ & & \times & \times \\ & & \times & 0 \end{bmatrix},$$

and the inverse of the full arrowhead matrix is a diagonal-plus-rank-one (DPR1) matrix,

$$\begin{bmatrix} \times & & & \times \\ & \times & & \times \\ & & \times & \times \\ & & & \times \\ \times & \times & \times & \times \end{bmatrix}^{-1} = \begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \\ & & & & 0 \end{bmatrix} \pm uu^T.$$

³The machine precision ε_M is defined as a smallest positive number such that in the floating-point arithmetic $1 + \varepsilon_M \neq 1$. In Matlab or FORTRAN REAL(8) arithmetic $\varepsilon_M = 2.2204 \cdot 10^{-16}$, thus the floating-point numbers have approximately 16 significant decimal digits. The term “double of the working precision” means that the computations are performed with numbers having approximately 32 significant decimal digits, or with the machine precision equal to ε_M^2 .

Our algorithm is completely parallel, since the computation of one eigenvalue and its eigenvector is completely independent of the computation of other eigenvalues and eigenvectors.

In Section 2 we describe the basic idea of our algorithm named *ah eig* (ArrowHead EIGenvalues). In Section 3 we discuss the accuracy of the algorithm. In Section 4 we present the complete algorithm which uses double of the working precision, if necessary. In Section 5 we illustrate algorithm with few examples and in Section 6 we apply our results to eigenvalue decomposition of Hermitian arrowhead matrix, singular value decomposition of real triangular arrowhead matrix and eigenvalue decomposition of real symmetric diagonal-plus-rank-one matrix. The proofs are given in Appendix A.

2. Basic shift-and-invert algorithm

Let λ be an eigenvalue of A , let v be its eigenvector, and let x be the unnormalized version of v from (6). Let d_i be the pole which is closest to λ . Clearly, from (7) it follows that either $\lambda = \lambda_i$ or $\lambda = \lambda_{i+1}$. Let A_i be the shifted matrix

$$A_i = A - d_i I = \begin{bmatrix} D_1 & 0 & 0 & z_1 \\ 0 & 0 & 0 & \zeta_i \\ 0 & 0 & D_2 & z_2 \\ z_1^T & \zeta_i & z_2^T & a \end{bmatrix}, \quad (9)$$

where

$$\begin{aligned} D_1 &= \text{diag}(d_1 - d_i, \dots, d_{i-1} - d_i), \\ D_2 &= \text{diag}(d_{i+1} - d_i, \dots, d_{n-1} - d_i), \\ z_1 &= [\zeta_1 \quad \zeta_2 \quad \cdots \quad \zeta_{i-1}]^T, \\ z_2 &= [\zeta_{i+1} \quad \zeta_{i+2} \quad \cdots \quad \zeta_{n-1}]^T, \\ a &= \alpha - d_i. \end{aligned}$$

Notice that D_1 (D_2) is positive (negative) definite.

Obviously, if λ is an eigenvalue of A , then

$$\mu = \lambda - d_i$$

is an eigenvalue of A_i , and vice versa, and they both have the same eigenvector. The inverse of A_i is

$$A_i^{-1} = \begin{bmatrix} D_1^{-1} & w_1 & 0 & 0 \\ w_1^T & b & w_2^T & 1/\zeta_i \\ 0 & w_2 & D_2^{-1} & 0 \\ 0 & 1/\zeta_i & 0 & 0 \end{bmatrix}, \quad (10)$$

where

$$\begin{aligned} w_1 &= -D_1^{-1} z_1 \frac{1}{\zeta_i}, \\ w_2 &= -D_2^{-1} z_2 \frac{1}{\zeta_i}, \\ b &= \frac{1}{\zeta_i^2} (-a + z_1^T D_1^{-1} z_1 + z_2^T D_2^{-1} z_2). \end{aligned} \quad (11)$$

Notice that

$$b = \bar{f}(d_i) / \zeta_i^2$$

where

$$\bar{f}(d_i) = \alpha - d_i - \bar{z}^T (\bar{D} - d_i I)^{-1} \bar{z}$$

where \bar{D} is the diagonal matrix D without d_i and \bar{z} is z without ζ_i .

The eigenvector x from (6) is given by

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} (D_1 - \mu I)^{-1} z_1 \\ -\frac{\zeta_i}{\mu} \\ (D_2 - \mu I)^{-1} z_2 \\ -1 \end{bmatrix}. \quad (12)$$

If λ is an eigenvalue of A which is closest to the pole d_i , then μ is the eigenvalue of matrix A_i which is closest to zero and

$$\nu = \frac{1}{\mu} = \pm \|A_i^{-1}\|_2.$$

In this case, if all entries of A_i^{-1} are computed with high relative accuracy, then, according to standard perturbation theory, ν is computed to high relative accuracy (by any reasonable algorithm). In Section 3 we show that all entries of A_i^{-1} are indeed computed to high relative accuracy, except possibly b (see (11)). If b is not computed to high relative accuracy and it influences $\|A_i^{-1}\|_2$, it is sufficient to compute it in double of the working precision (see Section 4).

Further, if μ is not the eigenvalue of A_i which is closest to zero, then $|\nu| < \|A_i^{-1}\|_2$, and the quantity

$$K_\nu = \frac{\|A_i^{-1}\|_2}{|\nu|} \quad (13)$$

tells us how far is ν from the absolutely largest eigenvalue of A_i^{-1} . If $K_\nu \gg 1$, then the standard perturbation theory does not guarantee that the eigenvalue μ will be computed with high relative accuracy. Remedies of this situation are described in Remark 2.

Finally, the exterior eigenvalues λ_1 and λ_n of A are excluded from the above reasoning. The absolutely larger of those is equal to $\|A\|_2$ and is, according to standard perturbation theory, computed with high relative accuracy. The

other exterior eigenvalue can also be computed with high relative accuracy if it is of the same order of magnitude as the larger one. If this is not the case, the absolutely smaller eigenvalue is computed as interior eigenvalues of A .

With this approach the componentwise high relative accuracy of the eigenvectors computed by (12) follows from high relative accuracy of the computed eigenvalues (see Theorem 3). Componentwise high relative accuracy of the computed eigenvectors implies, in turn, their orthogonality.

The described procedure is implemented in algorithm *aeig_basic* (Algorithm 1). The computation of the inverse of the shifted matrix, A_i^{-1} , according to formulas (10) and (11), is implemented in Algorithm 2. Algorithm 3 computes the largest or the smallest zero of the Pick function (5) by bisection. Given eigenvalue λ , Algorithm 4 computes the corresponding eigenvector by (6) or (12), respectively.

3. Accuracy of the algorithm

We now consider numerical properties of Algorithms 1, 2, 3, and 4. We assume the standard model of floating point arithmetic where subtraction is performed with guard digit, such that [9, 18, 10, 19]

$$fl(a \circ b) = (a \circ b)(1 + \varepsilon_\circ), \quad |\varepsilon_\circ| \leq \varepsilon_M, \quad \circ \in \{+, -, *, /\},$$

where ε_M is machine precision. In the statements of the theorems and their proofs we shall use the standard first order approximations, that is, we neglect the terms of order $O(\varepsilon_M^2)$ or higher. Moreover, we assume that neither overflow or underflow occurs during the computation.

We shall use the following notation:

Matrix	Exact eigenvalue	Computed eigenvalue
A	λ	$\tilde{\lambda}$
A_i	μ	$-$
$\tilde{A}_i = fl(A_i)$	$\hat{\mu}$	$\tilde{\mu} = fl(\hat{\mu})$
A_i^{-1}	ν	$-$
$\tilde{(A_i^{-1})} = fl(A_i^{-1})$	$\hat{\nu}$	$\tilde{\nu} = fl(\hat{\nu})$

(14)

Here

$$\tilde{A}_i = fl(A_i) = \begin{bmatrix} D_1(I + E_1) & 0 & 0 & z_1 \\ 0 & 0 & 0 & \zeta_i \\ 0 & 0 & D_2(I + E_2) & z_2 \\ z_1^T & \zeta_i & z_2^T & a(1 + \varepsilon_a) \end{bmatrix},$$

where E_1 and E_2 are diagonal matrices whose elements are bounded by ε_M in absolute values and $|\varepsilon_a| \leq \varepsilon_M$.

Algorithm 1

```
 $[\lambda, v] = \mathbf{aheig\_basic}(D, z, \alpha, k)$   
% Computes the  $k$ -th eigenpair of an irreducible arrowhead matrix  
%  $A = [\text{diag}(D) \ z; z' \ \alpha]$   
 $n = \max(\text{size}(D)) + 1$   
% Set the shift index  $i$  initially to zero  
 $i = 0$   
% Compute exterior eigenvalue ( $k = 1$  or  $k = n$ )  
if  $k == 1$   
     $\lambda = \mathbf{bisect}(D, z, \alpha, 'R')$   
     $\lambda_n = \mathbf{bisect}(D, z, \alpha, 'L')$   
    if  $|\lambda_n/\lambda| < 10$   
        % the eigenvalue is accurate  
         $v = \mathbf{vect}(D, z, \lambda)$   
    else  
        % determine the shift  $\sigma$ , the shift index  $i$ , and whether  $\lambda$  is on the left  
        % or the right side of the nearest pole  
         $\sigma = d_1$   
         $i = 1$   
         $side = 'R'$   
    end  
elseif  $k == n$   
     $\lambda = \mathbf{bisect}(D, z, \alpha, 'L')$   
     $\lambda_1 = \mathbf{bisect}(D, z, \alpha, 'R')$   
    if  $|\lambda_1/\lambda| < 10$   
        % the eigenvalue is accurate  
         $v = \mathbf{vect}(D, z, \lambda)$   
    else  
        % determine the shift, shift index, and side  
         $\sigma = d_{n-1}$   
         $i = n - 1$   
         $side = 'L'$   
    end  
end
```

Algorithm 1 (Continued)

```
else
  % Compute interior eigenvalue ( $k \in \{2, \dots, n-1\}$ )
  % Determine the shift (the nearest pole) and the shift index
   $Dtemp = D - d_k$ 
   $atemp = \alpha - d_k$ 
   $middle = Dtemp_{k-1}/2$ 
   $Fmiddle = atemp - middle - \sum(z^2./(Dtemp - middle))$ 
  if  $Fmiddle < 0$ 
     $\sigma = d_k$ 
     $i = k$ 
     $side = 'R'$ 
  else
     $\sigma = d_{k-1}$ 
     $i = k - 1$ 
     $side = 'L'$ 
  end
end
if  $i \neq 0$ 
  % Compute the inverse of the shifted matrix,  $A_i^{-1}$ 
   $[invD_1, invD_2, w_1, w_2, w_\zeta, b] = \mathbf{invA}(D, z, \alpha, i)$ 
  % Compute the leftmost or the rightmost eigenvalue of  $A_i^{-1}$ 
   $\nu = \mathbf{bisect}([invD_1; 0; invD_2], [w_1; w_\zeta; w_2], b, side)$ 
  % Compute the corresponding eigenvector
   $\mu = 1/\nu$ 
   $v = \mathbf{vect}(D - \sigma, z, \mu)$ 
  % Shift the eigenvalue back
   $\lambda = \mu + \sigma$ 
end
```

Algorithm 2

```
 $[invD_1, invD_2, w_1, w_2, w_\zeta, b] = \mathbf{invA}(D, z, \alpha, i)$ 
% Computes the inverse of an arrowhead matrix  $A = [\text{diag}(D - d_i) \ z; \ z' \ \alpha - d_i]$ 
% according to (10) and (11).
 $n = \max(\text{size}(D)) + 1$ 
 $D = D - d_i$ 
 $a = \alpha - d_i$ 
 $w_1 = -z_{1:i-1} ./ D_{1:i-1} ./ z_i$ 
 $w_2 = -z_{i+1:n-1} ./ D_{i+1:n-1} ./ z_i$ 
 $w_\zeta = 1/z_i$ 
 $invD_1 = 1 ./ D_{1:i-1}$ 
 $invD_2 = 1 ./ D_{i+1:n-1}$ 
 $b = (-a + \text{sum}(z_{1:i-1}.^2 ./ D_{1:i-1}) + \text{sum}(z_{i+1:n-1}.^2 ./ D_{i+1:n-1})) ./ z_i.^2$ 
```

Algorithm 3

```
 $\lambda = \text{bisect}(D, z, \alpha, \text{side})$   
% Computes the leftmost (for  $\text{side}='L'$ ) or the rightmost (for  $\text{side}='R'$ ) eigenvalue  
% of an arrowhead matrix  $A = [\text{diag}(D) \ z; \ z' \ \alpha]$  by bisection.  
 $n = \max(\text{size}(D)) + 1$   
% Determine the starting interval for bisection,  $[\text{left}, \text{right}]$   
if  $\text{side} == 'L'$   
     $\text{left} = \min\{D - |z|, \alpha - \|z\|_1\}$   
     $\text{right} = \min d_i$   
else  
     $\text{right} = \max\{D + |z|, \alpha + \|z\|_1\}$   
     $\text{left} = \max d_i$   
end  
% Bisection  
 $\text{middle} = (\text{left} + \text{right})/2$   
while  $(\text{right} - \text{left})/\text{abs}(\text{middle}) > 2 * \text{eps}$   
     $F_{\text{middle}} = \alpha - \text{middle} - \text{sum}(z.^2 ./ (D - \text{middle}))$   
    if  $F_{\text{middle}} > 0$   
         $\text{left} = \text{middle}$   
    else  
         $\text{right} = \text{middle}$   
    end  
     $\text{middle} = (\text{left} + \text{right})/2$   
end  
% Eigenvalue  
 $\lambda = \text{right}$ 
```

Algorithm 4

```
 $v = \text{vect}(D, z, \lambda)$   
% Computes the eigenvector of an arrowhead matrix  $A = [\text{diag}(D) \ z; \ z' \ \alpha]$   
% which corresponds to the eigenvalue  $\lambda$  by using (6).  
 $v = [z ./ (D - \lambda); -1]$   
 $v = v / \|v\|_2$ 
```

Further we define the quantities κ_λ , κ_μ and κ_b as follows:

$$\tilde{\lambda} = fl(\lambda) = \lambda(1 + \kappa_\lambda \varepsilon_M), \quad (15)$$

$$\tilde{\mu} = fl(\mu) = \mu(1 + \kappa_\mu \varepsilon_M), \quad (16)$$

$$\tilde{b} = fl(b) = b(1 + \kappa_b \varepsilon_M). \quad (17)$$

We also define the quantity

$$K_b = \frac{|a| + |z_1^T D_1^{-1} z_1| + |z_2^T D_2^{-1} z_2|}{|-a + z_1^T D_1^{-1} z_1 + z_2^T D_2^{-1} z_2|}. \quad (18)$$

3.1. Connection between accuracy of λ and μ

Let

$$\lambda = \mu + d_i$$

be an eigenvalue of the matrix A , where μ is the corresponding eigenvalue of the shifted matrix $A_i = A - d_i$ from which λ is computed. Let

$$\tilde{\lambda} = fl(\tilde{\mu} + d_i)$$

be the computed eigenvalue. Theorem 1 gives us dependency of accuracy of $\tilde{\lambda}$ in (15) upon accuracy of $\tilde{\mu}$ in (16).

Theorem 1. *For λ and $\tilde{\lambda}$ from (15) and μ and $\tilde{\mu}$ from (16) we have*

$$|\kappa_\lambda| \leq \frac{|d_i| + |\mu|}{|\lambda|} (|\kappa_\mu| + 1). \quad (19)$$

Proofs of this theorem and subsequent theorems are given in Appendix A.

From Theorem 1 we see that the accuracy of $\tilde{\lambda}$ depends on κ_μ and the size of the quotient

$$\frac{|d_i| + |\mu|}{|\lambda|}. \quad (20)$$

Theorem 2 analyzes the quotient (20) with respect to the position of λ and signs of μ and the neighboring poles.

Theorem 2. *Let the assumptions of Theorem 1 hold.*

(i) *If (see Figure 1 (i))*

$$\text{sign}(d_i) = \text{sign}(\mu),$$

then

$$\frac{|d_i| + |\mu|}{|\lambda|} = 1.$$

(ii) *If λ is between two poles of the same sign and $\text{sign}(d_i) \neq \text{sign}(\mu)$ (see Figure 1 (ii)), then*

$$\frac{|d_i| + |\mu|}{|\lambda|} \leq 3.$$

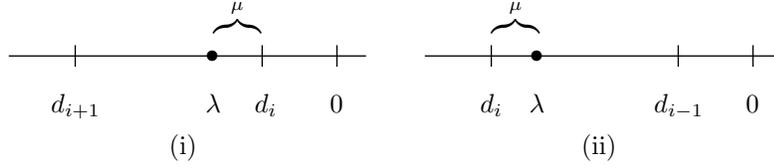


Figure 1: Typical situations from Theorem 2

Theorem 2 does not cover the following cases:

- (a) If $d_1 < 0$, then $\mu > 0$. If, further, $|d_1| \approx |\mu|$, then λ_1 is near zero, and $(|d_1| + |\mu|)/|\lambda_1| \gg 1$ (see Figure 2 (a)).⁴
- (b) If $d_n > 0$, then $\mu < 0$. If, further, $|d_n| \approx |\mu|$, then λ_n is near zero, and again $(|d_n| + |\mu|)/|\lambda_n| \gg 1$.⁵
- (c) If λ is between two poles of the different signs and $\text{sign}(d_i) \neq \text{sign}(\mu)$, then either $d_{i+1} < 0 < d_i$ and $\mu < 0$, or $d_i < 0 < d_{i-1}$ and $\mu > 0$. In both cases, if, additionally, $|d_i| \approx |\mu|$, then λ is near zero, and $(|d_i| + |\mu|)/|\lambda| \gg 1$ (see Figure 2 (c)).

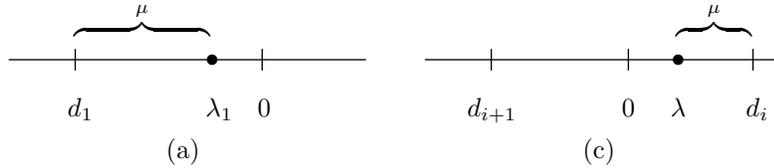


Figure 2: Typical situations for special cases

Since only one of these three cases can occur, Theorems 1 and 2 imply that for all eigenvalues $\lambda \in \sigma(A)$, but eventually one, it holds

$$\frac{|d_i| + |\mu|}{|\lambda|} \leq 3.$$

If one of the above cases does occur, remedies are given in the following remark.

Remark 1. If one of the cases (a), (b) or (c) occurs, then λ is an eigenvalue of A nearest to zero, and we can accurately compute it from the inverse of A .

⁴In this case λ_1 is computed as a difference of two close quantities and cancellation can occur.

⁵In this case λ_n is computed as a difference of two close quantities and cancellation can occur.

Notice that the inverse is of an unreduced arrowhead matrix with non-zero shaft is a diagonal-plus-rank-one (DPR1) matrix of the form

$$A^{-1} = \begin{bmatrix} D^{-1} & \\ & 0 \end{bmatrix} + \rho uu^T,$$

where

$$u = [z^T D^{-1} \quad -1]^T, \quad \rho = \frac{1}{a - z^T D^{-1} z}.$$

Eigenvalues of A^{-1} are zeros of (see [2, 14])

$$\varphi(\lambda) = 1 + \rho \sum_{j=1}^n \frac{u_j^2}{d_j - \lambda}.$$

Since the absolutely largest eigenvalue of A^{-1} is computed accurately according to standard perturbation theory, and $1/|\lambda| = \|A^{-1}\|_2$, λ is also computed with high relative accuracy. In computing matrix A^{-1} , eventually ρ needs to be computed in higher precision. For more details see Remark 2.

We still need to bound the quantity κ_μ from (19). This quantity essentially depends on the accuracy of $fl(b)$. The bound for κ_μ is given in Theorem 6.

3.2. Accuracy of the eigenvectors

Since the eigenvector is computed by (12), its accuracy depends on the accuracy of $\tilde{\mu}$ as described by the following theorem:

Theorem 3. *Let (16) hold and let*

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = fl\left(\begin{bmatrix} (D_1(I + E_1) - \tilde{\mu}I)^{-1} z_1 \\ -\frac{\zeta_i}{\tilde{\mu}} \\ (D_2(I + E_2) - \tilde{\mu}I)^{-1} z_2 \\ -1 \end{bmatrix} \right) \quad (21)$$

be the computed un-normalized eigenvector corresponding to μ and λ . Then

$$\tilde{x}_j = x_j (1 + \varepsilon_{x_j}), \quad |\varepsilon_{x_j}| \leq 3(|\kappa_\mu| + 3)\varepsilon_M, \quad j = 1, \dots, n.$$

In other words, if κ_μ is small, then all components of the eigenvector are computed to high relative accuracy. Since the accuracy of $\tilde{\lambda}$ and \tilde{x} depends on the accuracy of $\tilde{\mu}$ (on the size of κ_μ) in the next three subsections tells we discuss the accuracy of $\tilde{\mu}$. Since $\tilde{\mu}$ is computed as an inverse of the eigenvalue of the matrix $fl(A_i^{-1})$, we first discuss the accuracy of that matrix.

3.3. Accuracy of the matrix A_i^{-1}

We have the following theorem:

Theorem 4. *For the computed elements of the matrix A_i^{-1} from (10) and (11) for all $(j, k) \neq (i, i)$ we have*

$$\widetilde{(A_i^{-1})}_{jk} = fl(A_i^{-1})_{jk} = (A_i^{-1})_{jk}(1 + \varepsilon_{jk}), \quad |\varepsilon_{jk}| \leq 3\varepsilon_M.$$

For the computed element $b \equiv (A_i^{-1})_{ii}$ from (17) we have

$$|\kappa_b| \leq 3K_b,$$

where K_b is defined by (18).

The above theorem states that all elements of the matrix A_i^{-1} are computed with high relative accuracy except possibly b . Therefore, we have to monitor whether b is computed accurately, and, if not, it needs to be computed in double of the working precision (see Section 4 for details).

3.4. Accuracy of bisection

Let λ_{\max} be the absolutely largest eigenvalue of a symmetric arrowhead matrix A , and let $\widetilde{\lambda}_{\max}$ be the eigenvalue computed by bisection as implemented in Algorithm 3. The error bound from [16, Section 3.1] immediately implies that

$$\frac{|\widetilde{\lambda}_{\max} - \lambda_{\max}|}{|\lambda_{\max}|} = \kappa_{bis}\varepsilon_M, \quad \kappa_{bis} \leq 1.06n(\sqrt{n} + 1). \quad (22)$$

Notice that the similar error bound holds for all eigenvalues which are of the same order of magnitude as $|\lambda_{\max}|$.

3.5. Accuracy of exterior eigenvalues of A_i^{-1}

The desired interior eigenvalue and, in some cases, also absolutely smaller exterior eigenvalue λ of A is in Algorithm 1 computed by (8), where ν is one of the exterior eigenvalues of the matrix A_i^{-1} .

The following theorem covers the case when ν is the absolutely largest eigenvalue of $\|A_i^{-1}\|_2$, and gives two different bounds.

Theorem 5. *Let A_i^{-1} be defined by (10) and let ν be its eigenvalue such that*

$$|\nu| = \|A_i^{-1}\|_2. \quad (23)$$

Let $\widehat{\nu}$ be the exact eigenvalue of the computed matrix $\widetilde{(A_i^{-1})} = fl(A_i^{-1})$. Let

$$\widehat{\nu} = \nu(1 + \kappa_\nu\varepsilon_M). \quad (24)$$

Then

$$|\kappa_\nu| \leq \min \left\{ 3\sqrt{n} \max\{1, K_b\}, 6(\sqrt{n} + \frac{1}{|\zeta_i|} \sum_{\substack{k=1 \\ k \neq i}}^{n-1} |\zeta_k|) \right\}, \quad (25)$$

where K_b is defined by (18).

3.6. Final error bounds

All previous error bounds are summarized as follows.

Theorem 6. *Let $\tilde{\lambda}$ be the computed eigenvalue of an unreduced arrowhead matrix A , let $\tilde{\mu}$ be computed eigenvalue of the matrix $\widetilde{A_i}$ from (9), and let $\tilde{\nu}$ be the corresponding computed eigenvalue of the matrix $(\widetilde{A_i^{-1}})$ from (10). If μ is the eigenvalue of A_i closest to zero (or, equivalently, if (23) holds), then the error in the computed eigenvalue $\tilde{\lambda}$ is given by (15) with*

$$|\kappa_\lambda| \leq 3(|\kappa_\nu| + \kappa_{bis}) + 4, \quad (26)$$

and the error in the computed un-normalized eigenvector \tilde{x} is given by Theorem 3 with

$$|\kappa_\mu| \leq |\kappa_\nu| + \kappa_{bis} + 1, \quad (27)$$

where $|\kappa_\nu|$ is bounded by (25) and κ_{bis} is defined by (22).

Since we are essentially using the shift-and-invert technique, we can guarantee high relative accuracy of the computed eigenvalue and high componentwise relative accuracy of the computed eigenvector if ν is such that $|\nu| = O(\|A_i^{-1}\|_2)$ and it is computed accurately. This is certainly fulfilled if the following conditions are met:

- C1. The quantity K_ν from (13) is moderate, and
- C2. (i) either the quantity K_b from (18) is small, or
(ii) the quantity $\frac{1}{|\zeta_i|} \sum_{\substack{k=1 \\ k \neq i}}^{n-1} |\zeta_k|$ from (25) is of order $O(n)$.

The condition C1 implies that ν will be computed accurately according to the standard perturbation theory. The conditions C2 (i) or C2 (ii) imply that κ_ν from (25) is small, which, together with C1, implies that ν is computed accurately.

If the condition C1 does not hold, that is, if $K_\nu \gg 1$, remedies are given in Remark 2 below. If neither of the conditions C2 (i) and C2 (ii) holds, the remedy is to compute b in double of the working precision as described in Section 4.

Remark 2. If $\lambda = \lambda_1$ or $\lambda = \lambda_n$, it can be computed from the starting matrix A as in Algorithm 1. We have two more possibilities:

- (a) we can compute λ by shifting to another neighboring pole provided that K_ν is in this case small (shifting to the pole d_{i-1} instead of d_i in Figure 3 (a)),
- (b) if shifting to another neighboring pole is not possible ($K_\nu \gg 1$, see Figure 3 (b)), we can invert $A - \sigma I$, where shift σ is chosen near λ , and $\sigma \notin \{\lambda, d_i, d_{i-1}\}$. This results in a DPR1 matrix

$$(A - \sigma I)^{-1} = \begin{bmatrix} (D - \sigma I)^{-1} & \\ & 0 \end{bmatrix} + \rho uu^T,$$

where

$$u = [z^T(D - \sigma I)^{-1} \quad -1]^T, \quad \rho = \frac{1}{a - z^T(D - \sigma I)^{-1}z}.$$

Eigenvalues of this matrix are zeros of

$$\varphi(\lambda) = 1 + \rho \sum_{j=1}^n \frac{u_j^2}{(d_j - \sigma) - \lambda},$$

and the absolutely largest eigenvalue is computed accurately. Eventually, ρ needs to be computed in higher precision.⁶

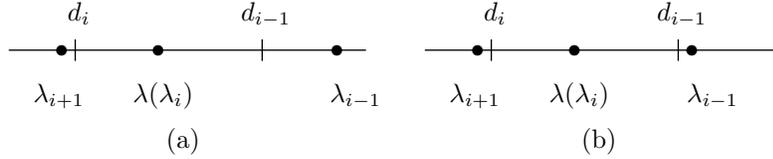


Figure 3: Typical situations from Remark 2

4. Final algorithm

If neither of the conditions $C2(i)$ and $C2(ii)$ hold, in order to guarantee that λ will be computed with high relative accuracy, the element b from the matrix A_i^{-1} needs to be computed in higher precision. The following theorem implies that if $1 \ll K_b \leq O(1/\varepsilon_M)$, it is sufficient to evaluate (11) in double of the working precision.^{7 8}

Theorem 7. *If $-a > 0$ in (11), set*

$$P = -a + z_1^T D_1^{-1} z_1, \quad Q = -z_2^T D_2^{-1} z_2,$$

and if $-a < 0$ in (11) set

$$P = z_1^T D_1^{-1} z_1, \quad Q = a - z_2^T D_2^{-1} z_2.$$

⁶Determining whether ρ needs to be computed in higher precision is done similarly as determining whether element b of A_i^{-1} needs to be computed in higher precision, which is described in Section 4. Further, Theorem 7 implies that it suffices to compute ρ in double of the working precision.

⁷If $K_b \geq O(1/\varepsilon_M)$, that is, if $K_b = 1/\varepsilon_E$ for some $\varepsilon_E < \varepsilon_M$, then, in view of Theorem 7, b needs to be computed with extended precision ε_E .

⁸Usage of higher precision in conjunction with the eigenvalue computation for DPR1 matrices is analysed in [2], but there the higher precision computation is potentially needed in the iterative part. This is less convenient than our approach where the higher precision computation is used only to compute one element.

Notice that in both cases $P, Q \geq 0$ and $b = (P - Q)/\zeta_i^2$. Let $\tilde{P} = fl(P)$ and $\tilde{Q} = fl(Q)$ be evaluated in standard precision, ε_M . Assume that $\tilde{P} \neq \tilde{Q}$ and $K_b \leq O(1/\varepsilon_M)$. If P , Q and b are all evaluated in double of the working precision, ε_M^2 , then (17) holds with $|\kappa_b| \leq O(n)$.

We summarize the above results in one, complete algorithm, *ah eig*. The algorithm first checks the components of the vector z . If they are of the same order of magnitude, the eigenpair (λ, v) is computed by Algorithm 1. If that is not the case, the quantity K_b is computed, and if $K_b \gg 1$, the eigenpair (λ, v) is computed by Algorithm 1 but with evaluation of b in double of the working precision. At the end, the quantity K_ν is computed, and if $K_\nu \gg 1$, one of the remedies from Remark 2 is applied.

Algorithm 5

```

 $[\lambda, v] = \mathbf{ah eig}(D, z, \alpha, k)$ 
% Computes the  $k$ -th eigenpair of an irreducible arrowhead matrix
%  $A = [\text{diag}(D) \ z; z' \ \alpha]$ 
if the quantity  $\left( \sum_{\substack{k=1 \\ k \neq i}}^{n-1} |\zeta_k| \right) / |\zeta_i|$  from (25) is of  $O(n)$ 
    % standard precision is enough
     $[\lambda, v] = \mathbf{ah eig.basic}(D, z, \alpha, k)$ 
else
    compute the quantity  $K_b$  from (18)
    if  $K_b \gg 1$ 
        % double precision is necessary
         $[\lambda, v] = \mathbf{ah eig.basic}(D, z, \alpha, k)$  with evaluation of  $b$  in double precision
    else
        % standard precision is enough
         $[\lambda, v] = \mathbf{ah eig.basic}(D, z, \alpha, k)$ 
    end
end
compute the quantity  $K_\nu$  from (13)
if  $K_\nu \gg 1$ 
    apply one of the remedies from Remark 2
end

```

4.1. On implementing double precision

Implementation of the double of the working precision depends upon whether the input is considered to be binary or decimal.

Double standard precision in Matlab, which assumes that input is binary, is obtained by using a combination of commands `vpa`, `digits` and `double` [13], where

- `digits(d)` specifies the number of significant decimal digits d used to do variable precision arithmetic `vpa`,

- `vpa(x)` uses variable-precision arithmetic to compute x to d decimal digits of accuracy,
- `double(x)` converts x to standard precision.

The assignment `a1=vpa(a,32)` pads the binary representation of a with zeros, which means that the decimal interpretation of the variable `a1` may have non-zero entries after 16-th significant decimal digit. The same effect is obtained in Intel FORTRAN compiler `ifort` [12] by the following program segment

```
real(8) a
real(16) a1
...
a1=a
```

However, the user can assume that the true input is given as a decimal number, which is, for example, assumed by extended precision computation in Mathematica [20]. In this case, the options in Matlab are to either use symbolic computation, or to cast the input to a string, and then convert it to extended precision:

```
a1=vpa(num2str(a,16),32)
```

In this case, the the decimal interpretation of the variable `a1` has all zero entries after 16-th significant decimal digit, but the binary representation of the variable `a` is, in general, padded with non-zero entries. The same effect is obtained in `ifort` writing to and reading from a string variable as in the following program segment:

```
real(8) a
real(16) a1
character(25) string
...
write(string,*) a
read(string,*) a1
```

If the input consists of numbers for which decimal and binary representation are equal (for example, integers, as in Example 3 below), then the two above approaches give the same results.

5. Numerical Examples

We illustrate our algorithm with four numerically demanding examples. Examples 1 and 2 illustrate Algorithm 1, Example 3 illustrates the use of double precision arithmetic, and Example 4 illustrates an application of higher dimension.

Example 1. In this example both quantities K_ν from (13) and K_b from (18) are for all eigenvalues approximately equal to 1, so we guarantee that all eigenvalues and all components of their corresponding eigenvectors are computed with high relative accuracy by Algorithm 5 (*ah eig*) using only standard machine precision.

Let

$$A = \begin{bmatrix} 2 \cdot 10^{-3} & 0 & 0 & 0 & 0 & 10^7 \\ 0 & 10^{-7} & 0 & 0 & 0 & 10^7 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -10^{-7} & 0 & 10^7 \\ 0 & 0 & 0 & 0 & -2 \cdot 10^{-3} & 10^7 \\ 10^7 & 10^7 & 1 & 10^7 & 10^7 & 10^{20} \end{bmatrix}.$$

The eigenvalues computed by Matlab [13] routine `eig`, Algorithm 5 and Mathematica [20] with 100 digits precision, are, respectively:

$\lambda^{(eig)}$	$\lambda^{(ah eig)}$	$\lambda^{(Math)}$
$1.0000000000000000 \cdot 10^{20}$	$1.0000000000000000 \cdot 10^{20}$	$1.0000000000000000 \cdot 10^{20}$
$1.999001249000113 \cdot 10^{-3}$	$1.999001249000113 \cdot 10^{-3}$	$1.999001249000113 \cdot 10^{-3}$
$4.987562099695390 \cdot 10^{-9}$	$4.987562099722817 \cdot 10^{-9}$	$4.987562099722817 \cdot 10^{-9}$
$-1.000644853973479 \cdot 10^{-20}$	$-9.99999999980001 \cdot 10^{-20}$	$-9.99999999980001 \cdot 10^{-20}$
$-2.004985562101759 \cdot 10^{-6}$	$-2.004985562101717 \cdot 10^{-6}$	$-2.004985562101717 \cdot 10^{-6}$
$-2.001001251000111 \cdot 10^{-3}$	$-2.001001251000111 \cdot 10^{-3}$	$-2.001001251000111 \cdot 10^{-3}$

We see that even the tiniest eigenvalues λ_3 and λ_4 , computed by Algorithm 5, are exact to the machine precision, which is not true for the eigenvalues computed by `eig`. Because of the accuracy of the computed eigenvalues, the eigenvectors computed by Algorithm 5 are componentwise accurate up to machine precision, and therefore, orthogonal up to machine precision. For example:

$v_4^{(eig)}$	$v_4^{(ah eig)}$	$v_4^{(Math)}$
$4.999993626151683 \cdot 10^{-11}$	$-4.99999999985000 \cdot 10^{-11}$	$-4.99999999985000 \cdot 10^{-11}$
$9.99999962328609 \cdot 10^{-7}$	$-9.99999999969000 \cdot 10^{-7}$	$-9.99999999969000 \cdot 10^{-7}$
$9.99999999990000 \cdot 10^{-1}$	$-9.99999999899999 \cdot 10^{-1}$	$-9.99999999899999 \cdot 10^{-1}$
$-9.99999964673912 \cdot 10^{-7}$	$9.99999999970999 \cdot 10^{-7}$	$9.99999999970999 \cdot 10^{-7}$
$-5.000006338012225 \cdot 10^{-11}$	$4.99999999985000 \cdot 10^{-11}$	$4.99999999985000 \cdot 10^{-11}$
$-9.99999963825105 \cdot 10^{-21}$	$9.99999999970000 \cdot 10^{-21}$	$9.99999999969999 \cdot 10^{-21}$

Example 2. In this example, despite very close diagonal elements, we again guarantee that all eigenvalues and all components of their corresponding eigenvectors are computed with high relative accuracy, without deflation. Let

$$A = \begin{bmatrix} 1 + 4\varepsilon_M & 0 & 0 & 0 & 1 \\ 0 & 1 + 3\varepsilon_M & 0 & 0 & 2 \\ 0 & 0 & 1 + 2\varepsilon_M & 0 & 3 \\ 0 & 0 & 0 & 1 + \varepsilon_M & 4 \\ 1 & 2 & 3 & 4 & 0 \end{bmatrix}$$

where $\varepsilon_M = 2 \cdot 2^{-53} = 2.2204 \cdot 10^{-16}$. For this matrix the quantities K_ν and K_b are again of order one for all eigenvalues, so Algorithm 5 uses only

standard working precision. The eigenvalues computed by Matlab, Algorithm 5 and Mathematica with 100 digits precision, are, respectively:

$\lambda^{(eig)}$	$\lambda^{(aheig)}$	$\lambda^{(Math)}$
6.000000000000000	6.000000000000001	6.000000000000000
1.000000000000001	1.000000000000001	1.000000000000001
1.000000000000001	1.000000000000001	1.000000000000001
1.000000000000000	1.000000000000000	1.000000000000000
-5.000000000000000	-4.999999999999999	-5.000000000000000

Although the eigenvalue computed by Matlab appear to be accurate, they are not. Namely,

$$\lambda_2^{(aheig)} - \lambda_3^{(aheig)} = 2.220446049250313 \cdot 10^{-16},$$

while $\lambda_2^{(eig)} - \lambda_3^{(eig)} = 0$, so the eigenvalues computed by Matlab do not satisfy the interlacing property. Notice that despite of very close eigenvalues, Algorithm 5 works without deflation. Due to the accuracy of the computed eigenvalues, the eigenvectors computed by Algorithm 5 are componentwise accurate up to the machine precision, and are therefore orthogonal. For example:⁹

$v_3^{(eig)}$	$v_3^{(aheig)}$	$v_3^{(Math)}$
$-4.99999999985000 \cdot 10^{-1}$	$-7.758019882294899 \cdot 10^{-2}$	$-7.758019882294900 \cdot 10^{-2}$
$-9.99999999969000 \cdot 10^{-2}$	$-9.085893014435115 \cdot 10^{-1}$	$-9.085893014435116 \cdot 10^{-1}$
$-9.99999999989999 \cdot 10^{-2}$	$3.53462724159426 \cdot 10^{-1}$	$3.53462724159426 \cdot 10^{-1}$
$9.99999999970999 \cdot 10^{-1}$	$2.085926573079233 \cdot 10^{-1}$	$2.085926573079233 \cdot 10^{-1}$
0	$2.077382436267882 \cdot 10^{-17}$	$2.077382436267882 \cdot 10^{-17}$

Example 3. In this example we can guarantee all eigenvalues and eigenvectors, componentwise will be computed with high relative accuracy only if we use double of the working precision when computing b from (11) in matrices A_2^{-1} , A_3^{-1} , A_4^{-1} and A_5^{-1} . Let

$$A = \begin{bmatrix} 10^{10} & 0 & 0 & 0 & 0 & 10^{10} \\ 0 & 4 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 10^{10} & 1 & 1 & 1 & 1 & 10^{10} \end{bmatrix}.$$

⁹Since, as described in Section 4.1, Mathematica uses decimal representation of the input, in order to obtain accurate eigenvectors we need to define ε_M in Mathematica with the output of Matlab's command `vpa(eps)`, $\varepsilon_M = 2.2204460492503130808472633361816 \cdot 10^{-16}$.

The quantities K_ν and K_b are:¹⁰

K_ν	K_b
$9.999999090793056 \cdot 10^{-1}$	$3.243243243540540 \cdot 10^9$
$9.999996083428923 \cdot 10^{-1}$	$3.636363636818182 \cdot 10^9$
$1.000000117045544 \cdot 10^0$	$4.444444445000000 \cdot 10^9$
$9.999998561319470 \cdot 10^{-1}$	$5.217390439488477 \cdot 10^9$
$7.941165469988994 \cdot 10^0$	$5.217390439488477 \cdot 10^9$

It is clear, from the condition numbers, that the element b in each of the matrices $A_2^{-1}, A_3^{-1}, A_4^{-1}$ and A_5^{-1} needs to be computed in double of the working precision. For example,

$$A_2 = A - d_2 I = \begin{bmatrix} 10^{10} - 4 & 0 & 0 & 0 & 0 & 10^{10} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & 0 & 1 \\ 0 & 0 & 0 & 0 & -3 & 1 \\ 10^{10} & 1 & 1 & 1 & 1 & 10^{10} - 4 \end{bmatrix}.$$

The element $b = [A_2^{-1}]_{22}$ computed by Algorithm 2 gives $b = 6.16666666667$, Matlab's routine `inv` yields $b = 6.166665889418350$, while computing b in double of the working precision gives the correct value $b = 6.166666668266667$.

Eigenvalues computed by Algorithm 1 (*ah eig_basic*, using only standard working precision), Algorithm 5 (*ah eig*, using double of the working precision to compute respective b 's) and Mathematica with 100 digits precision, respectively, are:

$\lambda_{ah eig_basic}$	$\lambda_{ah eig}$	λ_{Math}
$2.000000000000000 \cdot 10^{10}$	$2.000000000000000 \cdot 10^{10}$	$2.000000000000000 \cdot 10^{10}$
$4.150396802313551 \cdot 10^0$	$4.150396802279712 \cdot 10^0$	$4.150396802279713 \cdot 10^0$
$3.161498641452035 \cdot 10^0$	$3.161498641430967 \cdot 10^0$	$3.161498641430967 \cdot 10^0$
$2.188045596352105 \cdot 10^0$	$2.188045596339914 \cdot 10^0$	$2.188045596339914 \cdot 10^0$
$1.216093560005649 \cdot 10^0$	$1.216093584948579 \cdot 10^0$	$1.216093584948579 \cdot 10^0$
$-7.160348702977373 \cdot 10^{-1}$	$-7.160346250991725 \cdot 10^{-1}$	$-7.160346250991725 \cdot 10^{-1}$

The eigenvectors computed by Algorithm 5 are componentwise accurate to machine precision and therefore orthogonal.

Example 4. This example comes from the research related to decay of excited states of quantum dots in in real photon crystals [15]. In this case

- α is quantum dot transition frequency,
- d_i is a frequency of the i -th optical mode, and

¹⁰Algorithm 5 does not compute K_ν and K_b for the first eigenvalue, since it is an absolutely largest one.

- ζ_i is an interaction constant of the quantum dot with the i -th optical mode.

The size of the matrix is changeable but, in realistic cases, it is between 10^3 and 10^4 . We ran a test example for $n = 2501$ where, typically,

$$\begin{aligned} d_i &\in [5.8769928900036225 \cdot 10^{14}, 1.3709849013800450 \cdot 10^{15}], \\ \zeta_i &\in [3.2087698694339995 \cdot 10^6, 4.9584253488898976 \cdot 10^6], \\ \alpha &= 9.7949881500060375 \cdot 10^{14}. \end{aligned}$$

For this matrix the condition number $K_\nu \sim 1$ for all eigenvalues, and the components of the vector z are all of the same order of magnitude. Therefore, conditions *C1* and *C2* (*ii*) from Section 3 are fulfilled, so all eigenvalues and all components of all eigenvectors are computed with high relative accuracy by Algorithm 5 using only standard working precision. On the other hand about half of the eigenvalues computed by the Matlab routine `eig` do not satisfy the interlacing property.

6. Applications

In this section we extend our results to eigenvalue decompositions of Hermitian arrowhead matrices, singular value decompositions of real triangular arrowhead matrices and eigenvalue decompositions of real symmetric diagonal-plus-rank-one matrices.

6.1. Hermitian arrowhead matrices

Let

$$C = \begin{bmatrix} D & z \\ z^* & \alpha \end{bmatrix},$$

where

$$D = \text{diag}(d_1, d_2, \dots, d_{n-1}),$$

is a real diagonal matrix of order $n - 1$,

$$z = [\zeta_1 \quad \zeta_2 \quad \cdots \quad \zeta_{n-1}]^*,$$

is a complex valued vector and α is a real scalar. Here z^* denotes the conjugate transpose of z . As in Section 1, we assume that C is irreducible. The eigenvalue decomposition of C is given by

$$C = U\Lambda U^*$$

where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues, and $U = [u_1 \quad u_2 \quad \cdots \quad u_n]$ is a unitary matrix of the corresponding eigenvectors.

To apply Algorithm 5 to Hermitian arrowhead matrix we first transform C to real symmetric arrowhead matrix A by diagonal unitary similarity:

$$A = \Phi^* C \Phi = \begin{bmatrix} D & |z| \\ |z|^T & \alpha \end{bmatrix}, \quad (28)$$

where

$$\Phi = \text{diag} \left(\frac{\zeta_1}{|\zeta_1|}, \frac{\zeta_2}{|\zeta_2|}, \dots, \frac{\zeta_{n-1}}{|\zeta_{n-1}|}, 1 \right)$$

We now compute the k -th eigenpair (λ, v) of A by Algorithm 5, and set $u = \Phi v$. Since we guarantee high relative accuracy of the eigenvalue decomposition of A computed by Algorithm 5, we also guarantee high relative accuracy of the eigenvalue decomposition of C . Notice that, if double precision is needed to compute b in Algorithm 5, the modules $|\zeta_i|$ in (28) need to be computed in double of the working precision, as well.

Remark 3. Similarly, for irreducible non-symmetric arrowhead matrix

$$G = \begin{bmatrix} D & \check{z} \\ \check{z}^T & \alpha \end{bmatrix},$$

where $\text{sign}(\check{\zeta}_i) = \text{sign}(\zeta_i)$, $i = 1, \dots, n-1$, we define the diagonal matrix

$$\Psi = \text{diag} \left(\text{sign}(\check{\zeta}_1) \sqrt{\frac{\check{\zeta}_1}{\zeta_1}}, \dots, \text{sign}(\check{\zeta}_{n-1}) \sqrt{\frac{\check{\zeta}_{n-1}}{\zeta_{n-1}}}, 1 \right).$$

The matrix

$$A = \Psi^{-1} G \Psi = \begin{bmatrix} D & z \\ z^T & \alpha \end{bmatrix},$$

where $\zeta_i = \sqrt{\check{\zeta}_1 \zeta_i}$ is an irreducible symmetric arrowhead matrix.

We now compute the k -th eigenpair (λ, v) of A by Algorithm 5. The eigenpair of G is then $(\lambda, \Psi v)$. set $u = \Phi v$. Since we guarantee high relative accuracy of the eigenvalue decomposition of A , we also guarantee high relative accuracy of the eigenvalue decomposition of G . Notice that, if double precision is needed to compute b in Algorithm 5, the elements ζ_i need to be computed in double of the working precision, as well.

6.2. Singular value decomposition of a triangular arrowhead matrix

Let

$$B = \begin{bmatrix} D & z \\ 0 & \alpha \end{bmatrix},$$

be an irreducible upper triangular arrowhead matrix, that is, $d_i \neq d_j$ for $i \neq j$ and $\zeta_i \neq 0$ for all i . The matrix

$$A = B^T B = \begin{bmatrix} D^2 & Dz \\ z^T D & \alpha + z^T z \end{bmatrix},$$

is an irreducible symmetric arrowhead matrix.

When applying Algorithm 5 to the matrix A , we must ensure that all components of A_i^{-1} in (10) are computed to high relative accuracy. This is obviously true for elements of the vectors w_i and w_2 . Diagonal elements, except b , are computed with high relative accuracy as differences of squares of original quantities,

$$[A_i^{-1}]_{jj} = \frac{1}{(d_j - d_i)(d_j + d_i)}, \quad j \neq i.$$

The element $b = [A_i^{-1}]_{ii}$ from (11) is computed as¹¹

$$b = \frac{1}{d_i^2 \zeta_i^2} \left(-\alpha - z^T z + d_i + \sum_{j \neq i} \frac{d_j^2 \zeta_j^2}{(d_j - d_i)(d_j + d_i)} \right).$$

If double precision is needed in Algorithm 5, all entries of A need to be computed in double precision.

Let $B = U\Sigma V^T$ be the singular value decomposition of B , where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ are the singular values, the columns of V are the corresponding right singular vectors and the columns of U are the corresponding left singular vectors. We first compute the k -th eigenpair (λ, v) of A by Algorithm 5. Then $\sigma = \sqrt{\lambda}$ is the corresponding singular value of B and v is the corresponding right singular vector. The value σ and all components of v are computed to almost full accuracy. From the relation $U^T B = \Sigma V^T$ for the k -th row we have

$$\begin{bmatrix} u_{1:n-1}^T & u_n \end{bmatrix} \begin{bmatrix} D & z \\ 0 & \alpha \end{bmatrix} = \sigma \begin{bmatrix} v_{1:n-1}^T & v_n \end{bmatrix},$$

which implies

$$u_{1:n-1} = \sigma v_{1:n-1} D^{-1}.$$

From the relation $BV = U\Sigma$ for the k -th column we have

$$\begin{bmatrix} D & z \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} v_{1:n-1} \\ v_n \end{bmatrix} = \sigma \begin{bmatrix} u_{1:n-1} \\ u_n \end{bmatrix},$$

which implies

$$u_n = \frac{\alpha v_n}{\sigma}.$$

Components of u are computed by multiplication and division of quantities which are accurate to almost full machine precision, so they are accurate to almost full machine precision, as well.

6.3. Diagonal-plus-rank-one matrices

Let

$$M = D + uu^T,$$

¹¹In view of Theorem 7, if double precision computation is necessary, the positive and negative parts of this formula should be computed separately, and then added.

where

$$D = \text{diag}(d_1, \dots, d_n), \quad d_1 > d_2 > \dots > d_n,$$

$$u = [u_1 \quad \dots \quad u_n]^T, \quad u_i \neq 0, \quad i = 1, \dots, n,$$

be a $n \times n$ irreducible ordered real symmetric diagonal-plus-rank-one (DPR1) matrix. Let

$$\begin{aligned} \bar{D} &= \text{diag}(d_1, \dots, d_{n-1}), \\ \Delta &= (\bar{D} - d_n)^{1/2}, \\ \bar{u} &= [u_1 \quad \dots \quad u_{n-1}]^T, \\ L &= \begin{bmatrix} u_n \Delta^{-1} & 0 \\ -\bar{u}^T \Delta^{-1} & 1 \end{bmatrix}. \end{aligned}$$

Then

$$A = L^{-1}ML = \begin{bmatrix} \bar{D} & z \\ z^T & \alpha \end{bmatrix},$$

where

$$z = \bar{u}\Delta, \quad \alpha = d_n + u^T u,$$

is an irreducible real symmetric arrowhead matrix.

When applying Algorithm 5 to the matrix A , we must ensure that all components of A_i^{-1} in (10) are computed to high relative accuracy. This is obviously true for elements of the vectors w_i and w_2 . Diagonal elements, except b , are computed with high relative accuracy as differences of original quantities, and the element $b = [A_i^{-1}]_{ii}$ from (11) is computed as

$$b = \frac{1}{\zeta_i} \left(-d_n - u^T u + d_i + \sum_{j \neq i} \frac{\zeta_j}{d_j - d_i} \right).$$

If double precision is needed in Algorithm 5, all entries of A need to be computed in double precision.

Let $M = Q\Lambda Q^T$ and $A = V\Lambda V^T$ be the eigenvalue decompositions of M and A , respectively. Since M is by assumption irreducible, its eigenvalues satisfy interlacing property

$$\lambda_1 > d_1 > \lambda_2 > d_2 > \dots > \lambda_n > d_n. \quad (29)$$

We first compute the k -th eigenpair (λ, v) of A by Algorithm 5. The value λ and all components of v are computed to almost full accuracy. The relation $V^T AV = V^T L^{-1}MLV = \Lambda$ implies that the columns of the matrix $X = LV$ are the unnormalized eigenvectors of the matrix M . Further, since, by (29), all eigenvalues are simple, we conclude that $X = Q\Sigma$, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ is a positive definite matrix. Notice that $Q\Sigma V^T = L$ is, in fact, singular value decomposition of L .

Equating k -th columns of the equation $X = LV$ gives

$$x = \begin{bmatrix} \bar{x} \\ x_n \end{bmatrix} = Lv = \begin{bmatrix} u_n \Delta^{-1} & 0 \\ -\bar{u}^T \Delta^{-1} & 1 \end{bmatrix} \begin{bmatrix} \bar{v} \\ v_n \end{bmatrix},$$

where x and v are partitioned according to L . This immediately implies that

$$\bar{x} = u_n \Delta^{-1} \bar{v}.$$

Notice that, since all components of \bar{v} are computed to almost full, accuracy, the same holds for the components of \bar{x} , and it remains to compute x_n accurately.

Let

$$q = \begin{bmatrix} \bar{q} \\ q_n \end{bmatrix}$$

be the k -th column of Q and let $\sigma = \Sigma_{kk}$. Equating k -th rows of the equation

$$X^{-1} = \Sigma^{-1} Q^T = V^T L^{-1}$$

gives for the n -th element

$$q_n \frac{1}{\sigma} = x_n \frac{1}{\sigma^2} = v_n.$$

Thus,

$$x_n = \sigma^2 v_n$$

and, in order to compute x_n , it is necessary to compute σ^2 . From $X = U\Sigma = LV$ it follows that $V^T L^T L V = \Sigma^2$, or, equivalently, $LV = L^{-T} V \Sigma^2$. Equating k -th columns of this equation gives

$$\Delta^{-1} \bar{v} u_n = \left[\Delta \bar{v} \frac{1}{u_n} + \Delta^{-1} \bar{u} v_n \right] \sigma^2.$$

This gives $n - 1$ equations for σ^2 , and we can choose the numerically most accurate one.

Therefore, x_n will be computed to almost full machine precision, as are the entries of \bar{x} , and it remains to normalize x and obtain $q = x/\sigma$.

Remark 4. Notice that DPR1 matrices of the form $D - uu^T$ cannot be reduced to symmetric arrowhead matrix by the procedure described in this section. By using ideas from this paper, it is possible to derive highly accurate algorithm for DPR1 matrices without prior transformation to arrowhead form. This algorithm, which is a topic of our forthcoming paper, covers more general DPR1 matrices of the form

$$D + \rho uu^T, \quad \rho \in \mathbb{R}.$$

Appendix A. Proofs

Proof of Theorem 1.

Let $\tilde{\mu}$ and $\tilde{\lambda}$ be defined by (14). Then

$$\tilde{\lambda} \equiv fl(d_i + \tilde{\mu}) = (d_i + \tilde{\mu})(1 + \varepsilon_1).$$

By simplifying the equality

$$(d_i + \mu(1 + \kappa_\mu \varepsilon_M))(1 + \varepsilon_1) = \lambda(1 + \kappa_\lambda \varepsilon_M)$$

and using $\lambda = \mu + d_i$, we have

$$d_i \varepsilon_1 + \mu(\kappa_\mu \varepsilon_M + \varepsilon_1) = \lambda \kappa_\lambda \varepsilon_M.$$

Taking absolute value gives

$$|\kappa_\lambda| \leq \frac{|d_i| + |\mu|}{|\lambda|} (|\kappa_\mu| + 1). \quad \square$$

Proof of Theorem 2.

(i) The assumption $\text{sign}(d_i) = \text{sign}(\mu)$ immediately implies

$$\frac{|d_i| + |\mu|}{|\lambda|} = \frac{|d_i + \mu|}{|d_i + \mu|} = 1.$$

(ii) The assumptions imply that either

$$0 < d_{i+1} < \lambda < d_i, \quad \mu < 0,$$

or

$$d_i < \lambda < d_{i-1} < 0, \quad \mu > 0.$$

In the first case λ is closest to the pole d_i and

$$\begin{aligned} \frac{|d_i| + |\mu|}{|\lambda|} &\leq \frac{|d_i| + \frac{1}{2}|d_i - d_{i+1}|}{\frac{1}{2}|d_i + d_{i+1}|} \leq \frac{d_i + \frac{1}{2}d_i - \frac{1}{2}d_{i+1}}{\frac{1}{2}d_i + \frac{1}{2}d_{i+1}} \\ &\leq \frac{\frac{3}{2}d_i - \frac{1}{2}d_{i+1}}{\frac{1}{2}d_i + \frac{1}{2}d_{i+1}} \leq \frac{3d_i}{d_i} = 3. \end{aligned}$$

Here we used the inequalities $|\mu| \leq \frac{1}{2}|d_i - d_{i+1}|$ and $|\lambda| \geq \frac{1}{2}|d_i + d_{i+1}|$ for the first inequality, $d_i - d_{i+1} > 0$ and $d_i + d_{i+1} > 0$ for the second inequality and $d_{i+1} > 0$ for the fourth inequality, respectively.

The proof for the second case is analogous. \square

Proof of Theorem 3.

Let x and \tilde{x} be defined by (12) and (21), respectively. The theorem obviously holds for $x_n = \tilde{x}_n = -1$. For \tilde{x}_i we have

$$\tilde{x}_i = fl\left(-\frac{\zeta_i}{\tilde{\mu}}\right) = -\frac{\zeta_i}{\mu(1 + \kappa_\mu \varepsilon_M)}(1 + \varepsilon_1) = x_i(1 + \varepsilon_{x_i}).$$

By using (16) and (21), the first order approximation gives

$$|\varepsilon_{x_i}| \leq (|\kappa_\mu| + 1) \varepsilon_M.$$

For $j \notin \{i, n\}$, by solving the equality

$$\tilde{x}_j = \frac{\zeta_j}{((d_j - d_i)(1 + \varepsilon_1) - \mu(1 + \kappa_\mu \varepsilon_M))(1 + \varepsilon_2)}(1 + \varepsilon_3) = \frac{\zeta_j}{d_j - \lambda}(1 + \varepsilon_x)$$

for ε_x , using (16) and $\lambda = \mu + d_i$, and ignoring higher order terms, we have

$$\varepsilon_x = \frac{(d_j - d_i)(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) - \mu(\kappa_\mu \varepsilon_M + \varepsilon_2 + \varepsilon_3)}{d_j - \lambda}.$$

Therefore,

$$|\varepsilon_x| \leq \frac{|d_j - d_i| + |\mu|}{|d_j - \lambda|} (|\kappa_\mu| + 3) \varepsilon_M. \quad (\text{A.1})$$

To complete the proof we need to analyze two cases. If

$$\text{sign}(d_j - d_i) = -\text{sign } \mu,$$

then

$$\frac{|d_j - d_i| + |\mu|}{|d_j - \lambda|} = \frac{|d_j - d_i - \mu|}{|d_j - \lambda|} = \frac{|d_j - \lambda|}{|d_j - \lambda|} = 1.$$

If

$$\text{sign}(d_j - d_i) = \text{sign } \mu,$$

then, since d_i is pole closest to λ , we have $|\mu| \leq 0.5|d_j - d_i|$ and

$$\frac{|d_j - d_i| + |\mu|}{|d_j - \lambda|} \leq \frac{|d_j - d_i| + |\mu|}{|d_j - d_i| - |\mu|} \leq \frac{\frac{3}{2}|d_j - d_i|}{\frac{1}{2}|d_j - d_i|} = 3.$$

Finally, the theorem follows by inserting this into (A.1). \square

Proof of Theorem 4.

For the non-zero computed elements of the matrix A_i^{-1} from (10) and (11) we have:

$$fl([A_i^{-1}]_{jj}) = \frac{1}{(d_j - d_i)(1 + \varepsilon_1)}(1 + \varepsilon_2), \quad j \notin \{i, n\},$$

$$fl([A_i^{-1}]_{ji}) = fl([A_i^{-1}]_{ij}) = \frac{-\zeta_j}{(d_j - d_i)(1 + \varepsilon_3)\zeta_i(1 + \varepsilon_4)}(1 + \varepsilon_5), \quad j \notin \{1, n\},$$

$$fl([A_i^{-1}]_{ni}) = fl([A_i^{-1}]_{in}) = \frac{1}{\zeta_i}(1 + \varepsilon_6),$$

where $|\varepsilon_k| \leq \varepsilon_M$ for all indices k . The first statement of the theorem now follows by using standard first order approximations.

Similar analysis of the formula (11) yields

$$fl([A_i^{-1}]_{ii}) = \tilde{b} = b + \delta b,$$

where

$$|\delta b| \leq \frac{3}{\zeta_i^2} (|-a| + |z_1^T D_1^{-1} z_1| + |z_2^T D_2^{-1} z_2|) \varepsilon_M. \quad (\text{A.2})$$

This, in turn, implies (17) with

$$|\kappa_b| \leq \frac{|\delta b|}{|b|} \frac{1}{|\varepsilon_M|} = 3 \frac{|a| + |z_1^T D_1^{-1} z_1| + |z_2^T D_2^{-1} z_2|}{|-a + z_1^T D_1^{-1} z_1 + z_2^T D_2^{-1} z_2|} = 3K_b,$$

where K_b is defined by (18). \square

Proof of Theorem 5.

Let

$$\widetilde{(A_i^{-1})} = A_i^{-1} + \delta A_i^{-1},$$

or, equivalently,

$$\hat{\nu} - \nu = \delta A_i^{-1}.$$

From this, (24), and Theorem 4, we have

$$|\nu \kappa_\nu \varepsilon_M| \leq \|\delta A_i^{-1}\|_2 \leq 3 \|A_i^{-1}\|_2 \max\{1, K_b\} \varepsilon_M.$$

Since $\|A_i^{-1}\|_2 \leq \sqrt{n} \|A_i^{-1}\|_2$, we have

$$|\kappa_\nu| \leq 3\sqrt{n} \max\{1, K_b\}. \quad (\text{A.3})$$

For the second part of the proof, we may assume that the element $a = \alpha - d_i$ is not perturbed. More precisely, we represent the perturbation in a as perturbations in D_1 and D_2 by pre- and post-multiplication of the matrix $\tilde{A}_i = fl(A_i)$ with the diagonal matrix

$$\begin{bmatrix} (1 + \varepsilon_a)^{1/2} I_{n-1} & 0 \\ 0 & (1 + \varepsilon_a)^{-1/2} \end{bmatrix}.$$

This introduces additional errors in D_1 and D_2 bounded by ε_M . Therefore, we can write

$$|\nu \kappa_\nu \varepsilon_M| \leq \|\delta A_i^{-1}\|_2 \leq 6 \|A_i^{-1}\|_2 + |\delta b|. \quad (\text{A.4})$$

By modifying (A.2) to accommodate lack of perturbation in a , we have

$$|\delta b| \leq \frac{6}{\zeta_i^2} (|z_1^T D_1^{-1} z_1| + |z_2^T D_2^{-1} z_2|) \varepsilon_M.$$

By inserting this bound into (A.4) and rearranging, we have

$$|\kappa_\nu| \leq 6 \left(\sqrt{n} + \frac{1}{|\nu|} \frac{|z_1^T D_1^{-1} z_1| + |z_2^T D_2^{-1} z_2|}{\zeta_i^2} \right). \quad (\text{A.5})$$

Since

$$\begin{aligned} \|A_i^{-1}\|_2 &= |\nu| = \max_{\|x\|_2=1} \|A_i^{-1}x\|_2 \geq \|A_i^{-1}e_k\|_2 \\ &= \sqrt{\frac{1}{(d_k - d_i)^2} + \frac{\zeta_k^2}{\zeta_i^2 (d_k - d_i)^2}} \geq \frac{|\zeta_k|}{|\zeta_i| |d_k - d_i|}, \end{aligned}$$

by simply dividing each term

$$\frac{\zeta_k^2}{\zeta_i^2 |d_k - d_i|}$$

in (A.5) with the corresponding quotient

$$\frac{|\zeta_k|}{|\zeta_i| |d_k - d_i|},$$

we obtain

$$|\kappa_\nu| \leq 6 \left(\sqrt{n} + \frac{1}{\zeta_i} \sum_{\substack{k=1 \\ k \neq i}}^{n-1} |\zeta_k| \right). \quad (\text{A.6})$$

The bound (25) now follows from (A.3) and (A.6). \square

Proof of Theorem 6.

We first prove the bound (27). Since $\tilde{\nu} = fl(\hat{\nu})$ is computed by bisection, from (22) we have

$$\tilde{\nu} = \hat{\nu}(1 + \kappa_{bis}\varepsilon_M).$$

This and (24) imply

$$\tilde{\nu} = \nu(1 + \kappa_\nu\varepsilon_M)(1 + \kappa_{bis}\varepsilon_M).$$

Since $\hat{\mu} = fl(1/\hat{\nu})$, the bound (27) follows by ignoring higher order terms. The bound (26) now follows by inserting (27) into Theorems 1 and 2. \square

Proof of Theorem 7.

Let the assumptions of the theorem hold. Let b be computed in double of the working precision, ε_M^2 , and then stored in the standard precision. The standard floating-point error analysis with neglecting higher order terms gives

$$\begin{aligned} \frac{P(1 + \kappa_P\varepsilon_M^2) - Q(1 + \kappa_Q\varepsilon_M^2)}{\zeta_i^2} (1 + \kappa_1\varepsilon_M^2) &= \frac{P - Q}{\zeta_i^2} (1 + \kappa_b\varepsilon_M) \\ &\equiv b(1 + \kappa_b\varepsilon_M), \end{aligned}$$

where $|\kappa_P|, |\kappa_Q| \leq (n+1)$ and $|\kappa_1| \leq 3$. Solving the above equality for κ_b , neglecting higher order terms, and taking absolute values gives

$$|\kappa_b| \leq \frac{|P| + |Q|}{|P - Q|} (n+4) \varepsilon_M \equiv K_b (n+4) \varepsilon_M.$$

Since, by assumption, $K_b \leq O(1/\varepsilon_M)$, this implies

$$|\kappa_b| \leq O(n),$$

as desired. \square

References

- [1] E. Anderson et al., *LAPACK Users' Guide*, SIAM 3rd ed., Philadelphia, (1999).
- [2] J. L. Barlow, Error analysis of update methods for the symmetric eigenvalue problem, *SIAM J. Matrix Anal. Appl.*, 14 (1993) 598-618.
- [3] M. Bixon and J. Jortner, Intramolecular radiationless transitions, *J. Chem. Physics*, 48 (1968) 715-726.
- [4] C. F. Borges, W. B. Gragg, A parallel Divide - and - Conquer Method for the Generalized Real Symmetric Definite Tridiagonal Eigenproblem, in *Numerical Linear Algebra and Scientific Computing*, L. Reichel, A. Ruttan and R. S. Varga, eds., de Gruyter, Berlin (1993) 11-29.
- [5] J. R. Bunch and C. P. Nielsen, Rank-one modification of the symmetric eigenproblem, *Numer. Math.*, 31 (1978) 31-48.
- [6] J. J. M. Cuppen, A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numer. Math.*, 36 (1981) 177-195.
- [7] J. Dongarra and D. Sorensen, A fully parallel algorithm for the symmetric eigenvalue problem, *SIAM J. Sci. Statist. Comput.*, 8 (1987) 139-154.
- [8] J. W. Gadzuk, Localized vibrational modes in Fermi liquids, general Theory, *Phys. Rev. B*, 24 (1981) 1651-1663.
- [9] D. Goldberg, What Every Computer Scientist Should Know About Floating-Point Arithmetic, *ACM Computing Surveys*, 23:1 (1991) 5-48.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, 3rd ed. (1996).
- [11] M. Gu and S. C. Eisenstat, A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, *SIAM J. Matrix Anal. Appl.*, 16 (1995) 79-92.

- [12] *Intel Fortran Compiler*, <http://software.intel.com/en-us/fortran-compilers>
- [13] *MATLAB. The MathWorks*, Inc., Natick, Massachusetts, USA, <http://www.mathworks.com>.
- [14] A. Melman, Numerical solution of a secular equation, *Numer. Math.*, 69 (1995) 483-493.
- [15] D. Mogilevtsev, A. Maloshtan, S. Kilin, L. E. Oliveira and S. B. Cavalcanti, Spontaneous emission and qubit transfer in spin-1/2 chains, *J. Phys. B: At. Mol. Opt. Phys.*, 43 095506 (2010).
- [16] D. P. O'Leary and G.W. Stewart, Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices, *J. Comput. Phys.* 90, 2 (1990) 497-505.
- [17] S. Oliveira, A new parallel chasing algorithm for transforming arrowhead matrices to tridiagonal form, *Math. Comp.*, 221 (1998) 221-235.
- [18] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, (1980).
- [19] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, (1965).
- [20] *Wolfram Mathematica*, Documentation Center, <http://reference.wolfram.com/mathematica/guide/Mathematica.html>