Journal of Mathematical Modeling
Vol. 7, No. 3, 2019, pp. 337-355

Generalized subspace iteration method for solving matrix pair eigenproblem

Abdeslem Hafid Bentbib[†], **Ahmed Kanber**[‡] and **Kamal Lachhab**[†*]

[†] *University of Cadi Ayyad, Marrakesh, Morocco*
[‡] *CRMEF, Marrakesh, Morocco*
Emails: a.bentbib@uca.ac.ma, kanber@uca.ac.ma, kamal.lachhab@ced.uca.ac.ma

Abstract. The main purpose of this work is to give a generalization of the Subspace Iteration Method to compute the largest eigenvalues and their corresponding eigenvectors of the matrix pencil $A-\lambda B$. An effective single shift procedure is given. Several numerical experiments are presented to illustrate the effectiveness of the proposed methods.

Keywords: Matrix pencil, Generalized eigenvalues, Generalized QR-Francis, Generalized Subspace Iteration Method.
AMS Subject Classification: 34A34, 65L05.

1 Introduction

In this paper we discuss the numerical solution of the generalized eigenvalue problem (matrix pencil) $A - \lambda B$, where $A, B \in \mathbb{R}^{n\times n}$. The roots of $\det(A-\lambda B)$ are called generalized eigenvalues associated to matrix pencil $A-\lambda B$. The most popular method for computing generalized eigenvalues is certainly the so-called QZ method; see e.g [5, 8]. First, we introduce a Generalized Power Method (GPM) that computes the largest (in magnitude) generalized eigenvalue of the matrix pencil $A-\lambda B$ [3]. This problem encompasses a wide variety of applications that have been extensively studied in many different research areas [1]. Thereafter, we suppose that B is nonsingular and may be ill-conditioned. The naive approach consists in

*Corresponding author.
Received: 24 July 2019 / Revised: 9 September 2019 / Accepted: 9 September 20119.
DOI: 10.22124/jmm.2019.13944.1297

transforming matrix pencil $A-\lambda B$ to a standard eigenproblem $B^{-1}Av = \lambda v$ is then prohibited. A Generalized Subspace Iteration Method (GSIM) is given to compute a part of largest (in magnitude) generalized eigenvalues. A Generalized Francis-QR based method is also developed to compute the generalized Schur decomposition of the matrix pair (A, B). We show that at each step of the proposed method, the Hessenberg/Triangular structure is preserved. A theoretical study of the Generalized Francis-QR method is also followed.

This paper is organized as follows. In Section 2, we give some definitions and properties and we briefly recall the well-known QZ method. The generalization of the Francis's QR method is presented in Section 3. Section 4 is dedicated to a generalization of the so called power method to compute the largest (in magnitude) generalized eigenvalue, and to introduce a shift-invert strategy. In Section 5, GSIM, which is a block version of GPM is presented. In Section 6 , the numerical test are performed to compare the proposed approaches with MATLAB based function.

2 Definitions and properties

We start with the definition of generalized eigenvalues and eigenvectors, for more details, see [5]. For $A, B \in \mathbb{C}^{n\times n}$, the generalized eigenvalues of $A - \lambda B$ are elements of the set $\lambda(A, B)$ defined by

$$\lambda(A, B) = \{z \in \mathbb{C} : \det(A - zB) = 0\}.$$

If $\lambda \in \lambda(A, B)$ and $0 \neq x \in \mathbb{C}^n$ satisfy

$$Ax = \lambda Bx, \tag{1}$$

then x is an eigenvector of $A-\lambda B$. Note that generalized eigenvalue problem (1) has n eigenvalues if and only if $rank(B) = n$ [5].

Definition 1. A matrix pencil $A - \lambda B$ is regular if there exists $\lambda \in \mathbb{C}^n$ such that $\det(A - \lambda B) \neq 0$. We also say that the pair (A, B) is regular.

The most popular method for solving the generalized eigenvalue problem is the QZ algorithm, which is known in the literature as a numerically backward stable method [5, 8]. Recall that the QZ algorithm was developed by Moler and Stewart in [8] and has undergone some changes in recent years by Ward [9, 10], Kaufman [6], Dackland and Kăgström [2]. We recall the main result by the following theorem from [8].

Theorem 1 (Generalized Real Schur Decomposition)**.** *If A and B are in $\mathbb{R}^{n\times n}$, then there exist orthogonal matrices Q and Z such that Q^TAZ is upper quasi-triangular and Q^TBZ is upper triangular.*

The purpose of the QZ algorithm is to compute the generalized real Schur decomposition of the pair (A, B), i.e., to compute two orthogonal matrices Q and Z such that $S = Q^TAZ$ is quasi-upper triangular with 1×1 and 2×2 blocks on the diagonal, while the matrix $T = Q^TBZ$ is upper triangular. The algorithm proceeds in two stages. In the first one by applying a unitary equivalence transformations from the left and right, A is reduced to upper Hessenberg form and B is reduced to upper triangular form (See Algorithm 1). The second stage is a double shift QR-Francis method implicitly applied to AB^{-1}. This step generates a sequence of orthogonally equivalent matrix pairs $(A_0, B_0) \leftarrow (A, B)$, (A_1, B_1), (A_2, B_2), ... that converge to quasi-triangular/upper triangular equivalent matrix pair (A_∞, B_∞). The Implicit Q Theorem given below is the fundamental tool used at the implicit QR factorization step of the QZ method. For more details see [8].

Theorem 2 (Implicit Q Theorem)**.** *Suppose $Q = [q_1, \ldots, q_n]$ and $V = [v_1, \ldots, v_n]$ are orthogonal matrices with the property that the matrices $Q^TAQ = H$ and $V^TAV = G$ are each upper Hessenberg where $A \in \mathbb{R}^{n\times n}$. If H is unreduced and $q_1 = v_1$, then $q_i = \pm v_i$ and $|h_{i,i-1}| = |g_{i,i-1}|$ for $i = 1, 2, \ldots, n$.*

The following algorithm reduces the matrix pair (A, B) to the Hessenberg/Triangular matrix pair $(\bar{A}, \bar{B})$. This algorithm uses Givens rotations: for scalars a and b we compute $c = \cos\theta$ and $s = \sin\theta$ so that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \; r \in \mathbb{R}.$$

Algorithm 1. *Hessenberg/Triangular Reduction [8]*
Input: *A and B in $\mathbb{R}^{n\times n}$,*
Output*: A an upper Hessenberg matrix and B an upper triangular matrix.*
1. *$B = QR$ (Householder QR), $A = Q^TA$, $B = Q^TB$.*
2. ***for*** $j = 1 : n-2$
  for $i = n : -1 : j+2$
    $[c, s] = givens(A(i-1, j), A(i, j))$;
    $A(i-1 : i, j : n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A(i-1 : i, j : n)$;

$B(i-1:i,j:n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T B(i-1:i,j:n);$
$[c,s] = givens(-B(i,i), B(i,i-1));$
$B(1:i,i-1:i) = B(1:i,i-1:i) \begin{bmatrix} c & s \\ -s & c \end{bmatrix};$
$A(1:n,i-1:i) = A(1:n,i-1:i) \begin{bmatrix} c & s \\ -s & c \end{bmatrix};$

end

end

3 The generalized Francis QR

We will present a generalized version of the Francis QR method (GFQR) to compute eigenvalues of the problem $A - \lambda B$, where A and B are square matrices and B is nonsingular. We first reduce the pair (A, B) to Hessenberg/triangular form by using Algorithm 1. At each step k, we implicitly compute the QR factorization of ${B^{(k-1)}}^{-1}A^{(k-1)}$. Note that neither the inverse of the matrix $B^{(k-1)}$ nor the matrix product are explicitly computed. In fact this method seeks to construct the Schur decomposition of the matrix pair (A, B) (Theorem 1).

3.1 Generalized Francis QR method

Algorithm 2.
Input: $A, B \in \mathbb{R}^{n\times n}$
Output: *Generalized Schur decomposition of matrix pair* (A, B) *(Theorem 1)*
1. *Reduce the pair* (A, B) *to Hessenberg/Triangular form by applying the Algorithm 1.*
2. *Set* $A^{(0)} = A$, $B^{(0)} = B$, $Q = I$, $Z = I$ *and* $k = 0$
3. **while** $A^{(k)}$ *is not quasi-triangular* **do**
(a) $k = k + 1$
(b) Compute $A^{(k-1)} = PR$ *(QR-factorization)*
(c) Compute $P^T B^{(k-1)} = NU$ *(RQ-factorization)*
(d) Set $A^{(k)} = RU^T$, *and* $B^{(k)} = N$
(e) $Q \longleftarrow P^T Q$ *and* $Z \longleftarrow ZU$
End
4. ***return*** Q, Z, $T = A^{(\infty)}$ *and* $S = B^{(\infty)}$

To take advantage of the Hessenberg/Triangular structure of the matrix pair (A, B), we use only Givens rotations in steps (b), (c), (d) and (e) of Algorithm 2. This leads us to Algorithm 3, that preserves the Hessenberg/Triangular structure at each step k. It is mathematically equivalent to Algorithm 2.

Algorithm 3.
Input: $A, B \in \mathbb{R}^{n\times n}$
Output: *Generalized Schur decomposition of matrix pair* (A, B) *(Theorem 1)*
1. *Reduce the pair* (A, B) *to Hessenberg/Triangular form by applying the Algorithm 1.*
2. *Set* $A^{(0)} = A$, $B^{(0)} = B$, $Q = I$, $Z = I$ *and* $k = 0$
3. **while** $A^{(k)}$ *is not quasi-triangular* **do**
$k = k + 1$
for $i = 1 : n - 1$
- *Compute the Givens rotation* $G_{i,i+1}$ *plane that annihilate component* $A^{(k-1)}(i, i+1)$,
$A^{(k-1)} \longleftarrow G_{i,i+1}A^{(k-1)}$; $B^{(k-1)} \longleftarrow G_{i,i+1}B^{(k-1)}$;
and $Q \longleftarrow QG^T_{i,i+1}$;
- *Compute the Givens rotation* $P_{i,i+1}$ *plane to restore* $B^{(k-1)}$*'s triangularity;*
$B^{(k-1)} \longleftarrow B^{(k-1)}P^T_{i,i+1}$; $Z \longleftarrow ZP^T_{i,i+1}$;

if $i > 1$
$A^{(k-1)} \longleftarrow A^{(k-1)}P^T_{i-1,i}$;
Endif
Endfor
End
$A^{(k-1)} \longleftarrow A^{(k-1)}P^T_{n-1,n}$;
4. ***return*** Q, Z, $T = A^{(\infty)}$ *and* $S = B^{(\infty)}$.

Let us now describe the method that we use in practice. To show this, we consider $n = 5$. At each step k, we compute Givens rotations to annihilate the element $A^{(k-1)}_{2,1}$,

$$A^{(k-1)} \leftarrow G_{2,1}A^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix};$$

$$B^{(k-1)} \leftarrow G_{2,1}B^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix},$$

the nonzero entry arising in the (2,1) position in $B^{(k-1)}$ can be zeroed by multiplying with an appropriate Givens rotation $P_{1,2}^T$,

$$B^{(k-1)} \leftarrow B^{(k-1)} P_{1,2}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix},$$

we again multiply by Givens rotations to do zeroing $(3, 2)$ position of $A^{(k-1)}$,

$$A^{(k-1)} \leftarrow G_{3,2} A^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix};$$

$$B^{(k-1)} \leftarrow G_{3,2} B^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \boldsymbol{\times} & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix},$$

the Givens rotations $P_{2,3}^T$ is determined to restore $B^{(k-1)}$'s triangularity. We also multiply the matrix $A^{(k-1)}$ by rotation $P_{1,2}^T$

$$B^{(k-1)} \leftarrow B^{(k-1)} P_{2,3}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix};$$

$$A^{(k-1)} \leftarrow A^{(k-1)} P_{1,2}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix};$$

Zeros are similarly introduced into the $(4, 3)$ and $(5, 4)$ positions in $A^{(k-1)}$, and preserving the triangularity of $B^{(k-1)}$

$$A^{(k-1)} \leftarrow G_{4,3} A^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \mathbf{0} & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix};$$

$$B^{(k-1)} \leftarrow G_{4,3}B^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix},$$

$$B^{(k-1)} \leftarrow B^{(k-1)}P_{3,4}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \mathbf{0} & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix};$$

$$A^{(k-1)} \leftarrow A^{(k-1)}P_{2,3}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix};$$

$$A^{(k-1)} \leftarrow G_{5,4}A^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \mathbf{0} & \times \end{pmatrix};$$

$$B^{(k-1)} \leftarrow G_{5,4}B^{(k-1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix},$$

$$B^{(k-1)} \leftarrow B^{(k-1)}P_{4,5}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \mathbf{0} & \times \end{pmatrix};$$

$$A^{(k-1)} \leftarrow A^{(k-1)}P_{3,4}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{pmatrix},$$

$$A^{(k-1)} \leftarrow A^{(k-1)} P_{4,5}^T = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix}.$$

$A^{(k)}$ and $B^{(k)}$ are obtained by the last update of $A^{(k-1)}$ and $B^{(k-1)}$, respectively, Q and Z are computed as $Q = P_{1,2}^T P_{2,3}^T \cdots P_{4,5}^T$; $Z = G_{1,2}^T G_{2,3}^T \cdots G_{4,5}^T$. Note that the obtained matrices $A^{(k)}$ and $B^{(k)}$ are upper Hessenberg and upper triangular, respectively.

Proposition 1. *Algorithm 3 preserves the Hessenberg/Triangular structure at each step k.*

3.2 Convergence theory of the algorithm

The Generalized Francis QR algorithm applied to a matrix pair (A, B) is formally equivalent to applying implicitly the QR algorithm to $B^{-1}A$. Indeed, we have

$$A^{(k-1)} = PR,\ P^T B^{(k-1)} = NU.$$

Then

$$B^{(k-1)^{-1}} A^{(k-1)} = B^{(k-1)^{-1}} (PR),$$

and

$$\begin{aligned} B^{(k-1)^{-1}} A^{(k-1)} &= U^T N^{-1} P^T (PR), \\ &= U^T (N^{-1} R), \end{aligned}$$

which is nothing but the QR factorization of $B^{(k-1)^{-1}} A^{(k-1)}$. By setting $A^{(k)} = RU^T$ and $B^{(k)} = N$, we find the QR-Francis iteration

$$B^{(k)^{-1}} A^{(k)} = (N^{-1} R) U^T.$$

4 Generalized power method

In this section we give a generalization of the so-called power pethod to compute the generalized eigenvalues of the matrix pair (A, B), where A and B are square matrices [11]. This method computes the largest (in magnitude) eigenvalue of generalized eigenproblem. We use both QR and RQ factorizations at the normalization step.

Description: We first reduce the pair (A, B) to Hessenberg/triangular form by using Algorithm 1. At each step k of Generalized Power Method (GPM), we implicitly normalize vector $B^{-1}Av^{(k-1)}$ by using only Givens rotations, neither the inverse of the matrix B nor the matrix product are explicitly computed. We set $y^{(k)} = Av^{(k-1)}$ and we compute $(n-1)$ Givens rotations to obtain

$$G_{1,2}G_{2,3}\cdots G_{n-2,n-1}G_{n-1,n}y^{(k)} = \alpha_k e_1,$$

where $G_{i-1,i}$ represents the Givens rotation in the (e_{i-1}, e_i) plane that annihilate the component i of $y^{(k)}$. When multiplying $y^{(k)}$ by the rotation $G_{i-1,i}$ (from $i = n$ by -1 to $i = 2$) to annihilate the i-th component of $y^{(k)}$, the triangularity of B is destroyed by a nonzero $(i, i-1)$ subdiagonal element. We restore the triangularity of B by Givens rotation $Q^T_{i,i-1}$ right-side multiplication. We set $Q^{(k)} = Q^T_{n,n-1}Q^T_{n-1,n-2}\cdots Q^T_{2,1}$ and $v^{(k)} = Q^{(k)}e_1$, where e_1 is the first element of the canonical basis. Let us describe the k-th step when $n = 4$. We first initialize $Q^{(k)}$ with n-by-n identity matrix. The Givens rotation $G_{3,4}$ is introduced to zero $y^{(k)}_4$

$$y^{(k)} \leftarrow G_{3,4}y^{(k)} = \begin{pmatrix} \times \\ \times \\ \times \\ \mathbf{0} \end{pmatrix} \text{ and } B \leftarrow G_{3,4}B = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}.$$

The nonzero entry arising in the $(4, 3)$ position of B can be zeroed by multiplying with an appropriate Givens rotation $Q^T_{4,3}$,

$$B \leftarrow BQ^T_{4,3} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q^T_{4,3}.$$

The third component of $y^{(k)}$ is zeroed by using Givens rotation $G_{2,3}$

$$y^{(k)} \leftarrow G_{2,3}y^{(k)} = \begin{pmatrix} \times \\ \times \\ \mathbf{0} \\ 0 \end{pmatrix} \text{ and } B \leftarrow G_{2,3}B = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & 0 & \times \end{pmatrix}.$$

We multiply again by Givens rotation to restore B 's triangularity,

$$B \leftarrow BQ^T_{3,2} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & 0 & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q^T_{3,2}.$$

Similarly, the Givens rotations $G_{1,2}$ can be used to zero the 2^{nd} position of $y^{(k)}$. And the upper triangularity structure of B can be restored by multiplying it from the right by $Q_{2,1}^T$.

$$y^{(k)} \leftarrow G_{1,2} y^{(k)} = \begin{pmatrix} \times \\ \mathbf{0} \\ 0 \\ 0 \end{pmatrix}, \; B \leftarrow G_{1,2} B = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{pmatrix},$$

$$B \leftarrow B Q_{2,1}^T = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{pmatrix}, \text{ and } Q^{(k)} \leftarrow Q^{(k)} Q_{2,1}^T.$$

Note that two Givens rotations are required for j-th component $y^{(k)}$, $j > 1$, that is zeroed one to do the zeroing and the other to restore B's triangularity. If $G^{(k)} = G_{2,1} G_{3,2} \cdots G_{n-1,n-2} G_{n,n-1}$, then $G^{(k)} B Q^{(k)} = R^{(k)}$. is upper triangular and is obtained by the last update of B. We finally obtain $v^{(k)} = Q^{(k)} e_1$, which is the normalized vector of $B^{-1} A v^{(k-1)}$. We summarize the above method in the following algorithm.

Algorithm 4.
Input : $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times n}$, *a tolerance* tol *and* $iter_{max}$.
Output : *The largest generalized eigenvalue of problem* $Av - \lambda Bv = 0$.
1. *Reduce the pair* (A, B) *to Hessenberg/Triangular form by applying Algorithm 1.*
2. Initialization : $k := 0$; $v^{(0)} = v$ *and* $y = Av$;
3. While $err > tol$ *and* $k \leq iter_{max}$ ***do***
  $k = k + 1$; $G = Q = I$;
  for $i = n : -1 : 2$
   • *Compute the Givens rotations* $G_{i-1,i}$ *that annihilate component* $y(i)$;
    $y \leftarrow G_{i-1,i} y$; $B \leftarrow G_{i-1,i} B$;
   • *Compute the Givens rotations* $Q_{i,i-1}$ *that annihilate component* $B(i, i-1)$;
    $B \leftarrow B Q_{i,i-1}^T$; $Q = Q Q_{i,i-1}^T$;
  Endfor
  $w = \frac{y(1)}{B_{1,1}} Q^T e_1$; $\lambda = {v^{(k-1)}}^T w$;
  $v^{(k)} = Q^T e_1$; $y \leftarrow A v^{(k)}$ *and* $err = norm(v^{(k)} - v^{(k-1)})$;
 EndWhile
4. *The largest generalized eigenvalue is* $\lambda_{max} = \lambda$.

4.1 Shift inverse generalized power method

In this section we present a Shift Inverse Generalized Power Method (SIGPM) that can be used for computing the closest generalized eigenvalue to σ, where σ is called a shift [7]. The idea is to implicitly apply the power method to $(B^{-1}A - \sigma I)^{-1}$. The method has the ability to converge to any desired eigenvalue starting from a fairly good approximation. We first reduce the pair (A, B) to Hessenberg/triangular form by applying Algorithm 1 and we compute the RQ-factorization of the Hessenberg matrix $(A - \sigma B)$ by using $n-1$ Givens rotations. We use the same technique used in GPM given above. At each step k, we compute and normalize implicitly the vector $(B^{-1}A - \sigma I)^{-1}v^{(k-1)} = (A - \sigma B)^{-1}Bv^{(k-1)} = Q^T R^{-1} B v^{(k-1)}$. Here, neither the inverse of the matrix $(A - \sigma B)$ nor the matrix product are explicitly computed. By setting $y^{(k)} = Bv^{(k-1)}$ and using appropriate Givens rotations, we compute the orthogonal matrix $H^{(k)}$ to obtain

$$H^{(k)} y^{(k)} = \alpha_k e_1.$$

$H^{(k)}$ is the product of $(n-1)$ Givens rotations; i.e.,

$$H^{(k)} := H_{1,2} H_{2,3} \cdots H_{n-2,n-1} H_{n-1,n}.$$

Then, we obtain,

$$(B^{-1}A - \sigma I)^{-1} v^{(k-1)} \quad = \quad (A - \sigma B)^{-1} B v^{(k-1)}.$$

Let $(A - \sigma B) = RQ$ be the RQ-factorization of $(A - \sigma B)$, then

$$\begin{aligned}
(B^{-1}A - \sigma I)^{-1} v^{(k-1)} &= Q^T R^{-1} {H^{(k)}}^T H^{(k)} y^{(k)}, \\
&= \alpha_k Q^T (H^{(k)} R)^{-1} e_1, \\
&= \alpha_k Q^T (H^{(k)} R)^{-1} e_1.
\end{aligned}$$

Again, we compute the RQ-factorization, $H^{(k)} R = R^{(k)} G^{(k)}$, using $n-1$ Givens rotations, ${G^{(k)}}^T = G^T_{n,n-1} G^T_{n-1,n-2} \cdots G^T_{2,1}$. Then

$$\begin{aligned}
(B^{-1}A - \sigma I)^{-1} v^{(k-1)} &= \alpha_k (R^{(k)} G^{(k)})^{-1} e_1, \\
&= \alpha_k Q^T {G^{(k)}}^T (R^{(k)})^{-1} e_1, \\
&= \frac{\alpha_k}{r^{(k)}_{1,1}} Q^T {G^{(k)}}^T e_1,
\end{aligned}$$

$v^{(k)} = Q^T {G^{(k)}}^T e_1$, is the normalized vector of $(B^{-1}A - \sigma I)^{-1} v^{(k-1)}$. Note that $H^{(k)}$ and the RQ-factorization of $H^{(k)} R$ are computed simultaneously.

Indeed, when multiplying R by the Givens rotation $H_{i-1,i}$, the triangularity of R can be restored by right multiplication with Givens rotation $G^T_{i,i-1}$. Let us describe the method for the case $n = 4$. First, a Givens rotation $H_{3,4}$ is introduced to zero $y_4^{(k)}$,

$$y^{(k)} \leftarrow H_{3,4}y^{(k)} = \begin{pmatrix} \times \\ \times \\ \times \\ \mathbf{0} \end{pmatrix} \text{ and } R^{(k)} \leftarrow H_{3,4}R = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}.$$

The nonzero entry arising in the $(4,3)$ position of $R^{(k)}$ can be zeroed by right multiplying with an appropriate Givens rotation $G^T_{4,3}$,

$$R^{(k)} \leftarrow R^{(k)}G^T_{4,3} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

We introduce zero in the 3-th position of $y^{(k)}$ by using Givens rotation $H_{2,3}$

$$y^{(k)} \leftarrow H_{2,3}y^{(k)} = \begin{pmatrix} \times \\ \times \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$

and

$$R^{(k)} \leftarrow H_{2,3}R^{(k)} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

We again multiply by Givens rotations to do zeroing $(3,2)$ position of $R^{(k)}$,

$$R^{(k)} \leftarrow R^{(k)}G^T_{3,2} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

Zeros are similarly introduced into the 2-nd position of $y^{(k)}$ and preserving the $R^{(k)}$'s triangularity

$$y^{(k)} \leftarrow H_{1,2}y^{(k)} = \begin{pmatrix} \times \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, R^{(k)} \leftarrow H_{1,2}R^{(k)} = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix},$$

and

$$R^{(k)} \leftarrow R^{(k)}G_{2,1}^T = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

We summarize the above method in following algorithm.

Algorithm 5.

Input : *$A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times n}$, a tolerance tol, $iter_{max}$ and a shift σ.*

Output : *The closest generalized eigenvalue to the shift σ.*

1. *Reduce the pair (A, B) to Hessenberg/triangular form by applying Algorithm 1.*

2. Initialization : *$k := 0$; $v^{(0)} = v$ and $y = Bv$;*

3. *Compute RQ factorization $A - \sigma B = RQ$;*

4. While *$err > tol$ and $k \leq iter_{max}$* ***do***

$k = k + 1$; $G = Q^T$;

for *$i = n : -1 : 2$*

- *Compute the Givens rotation $H_{i-1,i}$ that annihilate component $y(i)$;*
  $y \leftarrow H_{i-1,i}y$; $R \leftarrow H_{i-1,i}R$;
- *Compute the Givens rotation $G_{i,i-1}$ that annihilate component $R(i, i-1)$;*
  $R \leftarrow RQ_{i,i-1}^T$; $G = GG_{i,i-1}^T$;

Endfor

$w = \frac{y(1)}{R_{1,1}}Ge_1$; $v^{(k)} = Ge_1$; $y \leftarrow Bv^{(k)}$ and

$err = norm(v^{(k)} - v^{(k-1)})$;

EndWhile

5. *The closest generalized eigenvalue to the shift σ is given by $\lambda_{max} = \dfrac{1}{\mu} + \sigma$, where $\mu = v^{(k-1)T}w$.*

5 Generalized subspace iteration method

In this section we introduce the block version of GPM called the Generalized Subspace Iteration Method (GSIM) for computing the s-largest generalized eigenvalues of a square pencil (A, B). The idea is based on the technique of Algorithm 4 and using the block-vector instead of single vector [2] . It is also based on the QR factorization and RQ factorization. We first reduce the pair (A, B) to Hessenberg/triangular form by using Algorithm 1. In the following, at each step k, we do implicitly a QR factorisation of the n-by-s matrix $B^{-1}AV^{(k-1)} = Q^{(k)}R^{(k)}$ by only using Givens rotations. We recall that neither the inverse of the matrix B nor the matrix product are explicitly computed. The generalized subspace iteration method applied to a matrix pair (A, B) is nothing but the subspace iteration method implicitly applying to $B^{-1}A$. Let us describe the case $n = 4$ and $s = 2$. At step k, we first initialize $Q^{(k)}$ by n-by-n identity matrix. Set $Y^{(k)} = AV^{(k-1)}$ and apply a Givens rotation $G_{3,4}$ to zeroed the $(4, 1)$ position of $Y^{(k)}$

$$Y^{(k)} \leftarrow G_{3,4}Y^{(k)} = \begin{pmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \mathbf{0} & \times \end{pmatrix}; \quad B \leftarrow G_{3,4}B = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}.$$

The nonzero entry arising in the $(4, 3)$ position of B can be zeroed by right-multiplication with an appropriate Givens rotation $Q_{4,3}^T$,

$$B \leftarrow BQ_{4,3}^T = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q_{4,3}^T.$$

We again applied a Givens rotation $G_{2,3}$ to zero element $(3, 1)$ of $Y^{(k)}$

$$Y^{(k)} \leftarrow G_{2,3}Y^{(k)} = \begin{pmatrix} \times & \times \\ \times & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \times \end{pmatrix}; \quad B \leftarrow G_{2,3}B = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

The nonzero entry arising in the $(3, 2)$ position of B can be zeroed by right-multiplication with an appropriate Givens rotation $Q_{3,2}^T$,

$$B \leftarrow BQ_{3,2}^T = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q_{3,2}^T.$$

Givens rotation $G_{1,2}$ to zero element $(2, 1)$ of $Y^{(k)}$

$$Y^{(k)} \leftarrow G_{1,2}Y^{(k)} = \begin{pmatrix} \times & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \times \end{pmatrix}; \quad B \leftarrow G_{1,2}B = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

The nonzero entry arising in the $(2,1)$ position of B can be zeroed by right-multiplication with an appropriate Givens rotation $Q_{2,1}^T$,

$$B \leftarrow BQ_{2,1}^T = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q_{2,1}^T.$$

Now dealing with the second column of the matrix $Y^{(k)}$. We first give rotation $G_{3,4}$ to zero element $(4,2)$ of $Y^{(k)}$

$$Y^{(k)} \leftarrow G_{3,4}Y^{(k)} = \begin{pmatrix} \times & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \mathbf{0} \end{pmatrix}; \quad B \leftarrow G_{3,4}B = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}.$$

The nonzero entry arising in the $(4,3)$ position of B can be zeroed by right-multiplication with an appropriate Givens rotation $Q_{4,3}^T$,

$$B \leftarrow BQ_{4,3}^T = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q_{4,3}^T.$$

Givens rotation $G_{2,3}$ is given to zero element $(3,2)$ of $Y^{(k)}$

$$Y^{(k)} \leftarrow G_{2,3}Y^{(k)} = \begin{pmatrix} \times & \times \\ \mathbf{0} & \times \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}; \quad B \leftarrow G_{2,3}B = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix}.$$

The nonzero entry arising in the $(3,2)$ position of B can be zeroed by right-multiplication with an appropriate Givens rotation $Q_{3,2}^T$,

$$B \leftarrow BQ_{3,2}^T = \begin{pmatrix} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ 0 & \mathbf{0} & \times & \times \\ 0 & 0 & \mathbf{0} & \times \end{pmatrix} \text{ and update } Q^{(k)} \leftarrow Q^{(k)}Q_{3,2}^T.$$

We finally obtain $V^{(k)} = Q^{(k)}I_{n\times s}$ the normalized block vector of $B^{-1}AV^{(k-1)}$, where, $I_{n\times s}$ are the s-th first columns of n-by-n identity matrix. The above process is repeated until obtaining convergence and it is summarize in the following algorithm.

Algorithm 6.

Input : *$A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times n}$, a tolerance tol, $iter_{max}$ and a number s.*

Output: *The s largest generalized eigenvalues of the pair (A, B).*

1. *Reduce the pair (A, B) to Hessenberg/Triangular form by applying Algorithm 1.*

2. ***Initialization :*** *$k := 0$; $V^{(0)} \in \mathbf{R}^{n\times s}$; $Z = AV^{(0)}$.*

3. ***While*** *$err > tol$ and $k \leq iter_{max}$* ***do***

$k = k + 1$; $G = Q = I$.

For *$j = 1 : s$*

$y = Z(:, j)$;

For *$i = n : -1 : j + 1$*

- *Compute the Givens rotation $G_{i-1,i}$ that annihilate component y_i;*
  $y \leftarrow G_{i-1,i}y$;$Z \leftarrow G_{i-1,i}Z$;
  $B \leftarrow G_{i-1,i}B$;
- *Compute the Givens rotation $Q_{i,i-1}$ that annihilate component $B(i, i-1)$;*
  $B \leftarrow BQ^T_{i,i-1}$; $Q = QQ^T_{i,i-1}$;

End*i*

End*j*

$V^{(k)} \leftarrow Q(:, 1 : s)$; $Z = AV^{(k)}$;

$err = norm(V^{(k)} - V^{(k-1)})$;

EndWhile

5. *The s largest generalized eigenvalues are $\dfrac{Z(i,i)}{B(i,i)}$, $i = 1, ..., s$.*

6 Numerical examples

In this section we compare the numerical results obtained by GPM algorithm, GSIM algorithm and eig function of MATLAB in terms of the relative error. All of the reported experiments were performed on a 32-bit 2.4 GHz Intel Core Duo Processor and 2 GB RAM on 2013a MATLAB version.

Example 1. We compared and tested the numerical results obtained by Algorithm 4 with MATLAB eig function for different sizes. For the first test we take a 500 by 500 real matrix pencil (A, B), where the condition number of $B^{-1}A$ is $8.8274.10^{24}$. Table 1 gives the largest eigenvalue of the pair (A, B) computed by Algorithm 4 and the one computed by MATLAB eig function, and Table 2 presents the relative errors of both methods. For the second test we take a 1000 by 1000 real matrix pencil (A, B), where the condition number of $B^{-1}A$ is $1.3462.10^{33}$. The exact and the computed largest eigenvalues of the pair (A, B) are given in Table 3. The corresponding relative errors are given in Table 4.

Table 1: Eigenvalue calculated by each method ($n = 500$).

Exact	GPM	MATLAB eig
$7.0313e-01$	$7.0318e-01$	$7.0328e-01$

Table 2: Relative errors occurred when computing the eigenvalues ($n = 500$).

GPM	MATLAB eig
$8.4711e-05$	$2.2436e-04$

Table 3: Eigenvalue calculated by each method ($n = 1000$).

Exact	GPM	MATLAB eig
6.1875	6.1875	6.1873

Table 4: Relative errors occurred when computing the eigenvalues ($n = 1000$).

GPM	MATLAB eig
$6.3348e-06$	$3.0734e-05$

Example 2. In this example we give two tests that compare the numerical results obtained by GSIM (Algorithm 6) and MATLAB eig function for different sizes. Figure 1 presents the relative errors obtained for $n = 200$ and $s = 5$. In Figure 2, we plot the relative errors for $n = 1000$ and $s = 4$.

7 Conclusion

We have presented a generalization of the subspace iteration method (GSIM) to compute the s-largest (in magnitude) generalized eigenvalues of the matrix pencil $A - \lambda B$. The special case when $s = 1$ is presented (GPM).

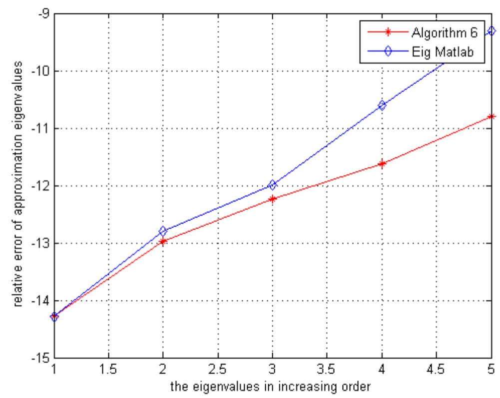

Figure 1: Relative errors of MATLAB eig function and GSIM.

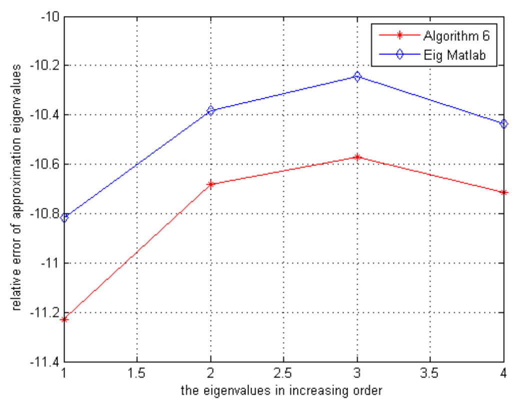

Figure 2: Relative error of MATLAB eig function and GSIM.

A generalization of the well-known Francis-QR method to compute the generalized Schur decomposition of the matrix pair (A, B) is also given. The method is presented in such a way that it preserves the Hessenberg/Triangular structure at each step. In all the proposed methods, neither the inverse of the matrix B nor matrix product are effectively computed.

Acknowledgements

The authors are grateful to the anonymous referee for his/her comments which substantially improved the quality of this paper.

References

[1] A.H. Bentbib, A. Kanber and K. Lachhab, *Subspace iteration method for generalized singular values*, Ann. Uni. Craiova Math. Comp. Sci. Ser. **46**(1) (2019) 78–89.

[2] K. Dackland and B. Kågström. *Blocked algorithms and software for reduction of a regular matrix pair to generalized Schur form*, ACM Trans. Math. Software **25**(4) (1999) 425–454.

[3] D. Kressner, E. Mengi, I. Naki and N. Truhar, *Generalized eigenvalue problems with specified eigenvalues*, IMA J. Numer. Anal. **34** (2) (2014) 480–501.

[4] H. Fassbender, and D. Kressner, *Structured eigenvalue problems*, GAMM-Mitt **29** (2006) 297–318.

[5] G.H. Golub and C.F. Van Loan, *Matrix computation*, 4th Edition, 2013.

[6] L. Kaufman, *Some thoughts on the QZ algorithm for solving the generalized eigenvalue problem*, ACM Trans. Math. Software **3**(1) (1977) 65–75.

[7] K. Meerbergen, A. Spence and D. Roose, *Shift-invert and Cayley transforms for detection of rightmost eigenvalues of nonsymmetric matrices*, BIT Numer. Math. **34** (3) (1994) 409–423.

[8] C.B. Moler and G.W. Stewart, *An Algorithm for Generalized Matrix Eigenvalue Problems* , SIAM J. Num Anal, 10 (1973).

[9] R.C. Ward, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comput. **2** (2) (1981) 141–152.

[10] R.C. Ward, *The combination shift QZ algorithm*, SIAM J. Numer. Anal. **12**(6) (1975) 835–853.

[11] F. Xue and H.C. Elman, *Fast inexact subspace iteration for generalized eigenvalue problems with spectral transformation*, Linear Algebra Appl. **435** (2011) 601–622.