MB04BV – A FORTRAN 77 Subroutine to Compute the Eigenvectors Associated to the Purely Imaginary Eigenvalues of Skew-Hamiltonian/Hamiltonian Matrix Pencils [1]

Peihong Jiang[2] Matthias Voigt[3]

September 2013

[1]This document presents research results obtained during the development of the SLICOT (Subroutine Library in Systems and Control Theory) software. The SLICOT Library and the related CACSD tools based on SLICOT were partially developed within the Numerics in Control Network (NICONET) funded by the European Community BRITE-EURAM III RTD Thematic Networks Programme (contract number BRRT-CT97-5040), see http://www.icm.tu-bs.de/NICONET. SLICOT can be used free of charge by academic users, see http://www.slicot.org, for solving analysis and synthesis problems of modern and robust control. This report is available from www.slicot.org/REPORTS/SLWN2013-3.pdf
[2]University of Rochester, New York, USA; Email: pjiang3@u.rochester.edu
[3]Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany; Email: voigtm@mpi-magdeburg.mpg.de

Abstract

We implement a structure-preserving numerical algorithm for extracting the eigenvectors associated to the purely imaginary eigenvalues of skew-Hamiltonian/Hamiltonian matrix pencils. We compare the new algorithm with the QZ algorithm using random examples with different difficulty. The results show that the new algorithm is significantly faster, more robust, and more accurate, especially for hard examples.

Keywords: Eigenvalues, eigenvectors, skew-Hamiltonian/Hamiltonian matrix pencil, reliability, structure-preserving algorithm.

1 Introduction

Skew-Hamiltonian/Hamiltonian matrix pencils have a wide range of applications in systems and control theory, for instance in linear-quadratic optimal control [17], $\mathcal{H}_\infty$ optimal control [19], for computing system norms [2], or analyzing system properties, such as passivity or contractivity [9, 8, 10], or a counterclockwise input/output dynamics [16, 6]. In these applications, different spectral information is needed, such as eigenvalues, eigenvectors, and deflating subspaces. The results presented in this paper are motivated by the following problem. In electrical engineering it is common to model electrical circuits by differential-algebraic equations or descriptor systems. Electrical circuits have many important properties, for instance they cannot internally generate energy which is called *passivity*. This property should also be reflected in the equations that model the circuit. However, due to errors introduced in the modeling process, e.g., by linearization, model approximation, or model order reduction, the resulting model is not always passive. In this case we need to perform a post-processing of the equations in order to restore the passivity property which is important to obtain meaningful results when simulating this model. This process is called passivity enforcement and is typically done by perturbing the system or parts of it [10, 14, 15].

Passivity can be characterized by the purely imaginary eigenvalues of an associated skew-Hamiltonian/Hamiltonian matrix pencil. Passivity is then enforced by perturbing this pencil which results in a movement of the purely imaginary eigenvalues. This process is iteratively repeated until no purely imaginary eigenvalues exist. To compute optimal perturbations not only the eigenvalues have to be computed but also the corresponding eigenvectors, see, e.g., [10, 14, 15]. This report focuses on the actual computation of these eigenvectors.

Standard numerical methods to compute eigenvalues, eigenvectors and deflating subspaces rely on the generalized Schur decomposition [7]. However, this factorization does not respect the structure of the pencil. Therefore, one is interested in a skew-Hamiltonian/Hamiltonian Schur-like form. Unfortunately, when purely imaginary eigenvalues exist, difficulties arise because of the possible non-existence of such a structured Schur form. To avoid this problem, the pencil is embedded into a skew-Hamiltonian/Hamiltonian pencil of double size. A structure-preserving algorithm for computing the purely imaginary eigenvalues in a very accurate and reliable manner is presented in [1]. An algorithm for the computation of the corresponding eigenvectors based on the structured canonical forms given in [1], is stated in [5].

This paper describes the FORTRAN 77 routine `MB04BV` that has been implemented to extract the eigenvectors corresponding to the simple, finite, purely imaginary eigenvalues. We give a brief overview of the theoretical foundations, the algorithm outline, details of the implementation, and present numerical results obtained by `MB04BV`.

2 Theory

In this report we deal with the following matrix structures.

Definition 1. *[1] Let* $\mathcal{J} := \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$, *where* I_n *is the* $n \times n$ *identity matrix.*

(i) A matrix $\mathcal{H} \in \mathbb{R}^{2n\times 2n}$ *is* Hamiltonian *if* $(\mathcal{H}\mathcal{J})^T = \mathcal{H}\mathcal{J}$.

(ii) A matrix $\mathcal{N} \in \mathbb{R}^{2n\times 2n}$ *is* skew-Hamiltonian *if* $(\mathcal{N}\mathcal{J})^T = -\mathcal{N}\mathcal{J}$.

(iii) A real matrix pencil $\lambda\mathcal{N} - \mathcal{H}$ *is* skew-Hamiltonian/Hamiltonian *if* $\mathcal{N}$ *is skew-Hamiltonian and* $\mathcal{H}$ *is Hamiltonian.*

Skew-Hamiltonian/Hamiltonian pencils satisfy the Hamiltonian spectral symmetry, i.e., eigenvalues occur in pairs $\{\lambda, -\lambda\}$ if they are purely real or imaginary, or otherwise in quadruples $\{\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda}\}$,

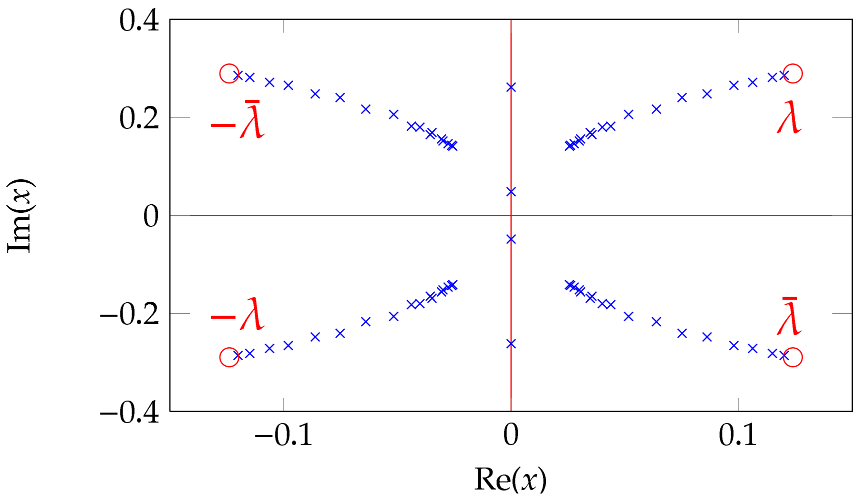

Figure 1: Hamiltonian eigensymmetry

see Figure 1. Numerical algorithms should preserve this structure in order to get meaningful results. To compute the eigenvalues of a skew-Hamiltonian/Hamiltonian matrix pencil, we use the fact that $\mathcal{J}$-congruence transformations of the form

$$\lambda\tilde{\mathcal{N}} - \tilde{\mathcal{H}} := \mathcal{J}\mathcal{Q}^T\mathcal{J}^T(\lambda\mathcal{N} - \mathcal{H})\mathcal{Q}$$

with a nonsingular matrix $\mathcal{Q}$ preserve the skew-Hamiltonian/Hamiltonian structure. Therefore, we hope that we can compute an *orthogonal* matrix $\mathcal{Q}$ such that

$$\mathcal{J}\mathcal{Q}^T\mathcal{J}^T(\lambda\mathcal{N} - \mathcal{H})\mathcal{Q} = \lambda\begin{bmatrix}\mathcal{N}_{11} & \mathcal{N}_{12}\\ 0 & \mathcal{N}_{11}^T\end{bmatrix} - \begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{12}\\ 0 & -\mathcal{H}_{11}^T\end{bmatrix}$$

is in skew-Hamiltonian/Hamiltonian Schur form, i.e., the subpencil $\lambda\mathcal{N}_{11} - \mathcal{H}_{11}$ is in generalized Schur form [12]. Unfortunately, not every skew-Hamiltonian/Hamiltonian pencil has this form, since pairs of simple purely imaginary eigenvalues cannot be represented in this structure. In this case, we embed the matrix pencil into another double-sized matrix pencil to solve the problem as follows. We introduce the orthogonal matrices

$$\mathcal{Y} = \frac{\sqrt{2}}{2}\begin{bmatrix}I_{2n} & I_{2n}\\ -I_{2n} & I_{2n}\end{bmatrix}, \quad \mathcal{P} = \begin{bmatrix}I_n & 0 & 0 & 0\\ 0 & 0 & I_n & 0\\ 0 & I_n & 0 & 0\\ 0 & 0 & 0 & I_n\end{bmatrix}, \quad \mathcal{X} = \mathcal{Y}\mathcal{P}.$$

Then we define the double-sized matrix pencils

$$\lambda\mathcal{B}_{\mathcal{N}} - \mathcal{B}_{\mathcal{H}} := \lambda\begin{bmatrix}\mathcal{N} & 0\\ 0 & \mathcal{N}\end{bmatrix} - \begin{bmatrix}\mathcal{H} & 0\\ 0 & -\mathcal{H}\end{bmatrix},$$

and

$$\lambda\tilde{\mathcal{B}}_{\mathcal{N}} - \tilde{\mathcal{B}}_{\mathcal{H}} := \mathcal{X}^T\left(\lambda\mathcal{B}_{\mathcal{N}} - \mathcal{B}_{\mathcal{H}}\right)\mathcal{X}.$$

The $4n \times 4n$ matrix pencil $\lambda\tilde{\mathcal{B}}_{\mathcal{N}} - \tilde{\mathcal{B}}_{\mathcal{H}}$ is again real skew-Hamiltonian/Hamiltonian with the same eigenvalues (with double algebraic, geometric, and partial multiplicities) as the original pencil. To compute the eigenvalues one uses the generalized symplectic URV decomposition which is summarized in the following theorem (see the real-case versions of the results in [1]).

Theorem 2. *Let $\lambda\mathcal{N} - \mathcal{H}$ be a regular real $2n \times 2n$ skew-Hamiltonian/Hamiltonian pencil. Then there exist real orthogonal $2n \times 2n$ matrices $\mathcal{Q}_1$, $\mathcal{Q}_2$ such that*

$$\begin{aligned} \mathcal{Q}_1^T \mathcal{N} \mathcal{J} \mathcal{Q}_1 \mathcal{J}^T &= \begin{bmatrix} N_1 & N_2 \\ 0 & N_1^T \end{bmatrix}, \\ \mathcal{J} \mathcal{Q}_2^T \mathcal{J}^T \mathcal{N} \mathcal{Q}_2 &= \begin{bmatrix} M_1 & M_2 \\ 0 & M_1^T \end{bmatrix} := \mathcal{M}, \\ \mathcal{Q}_1^T \mathcal{H} \mathcal{Q}_2 &= \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22}^T \end{bmatrix}, \end{aligned} \tag{1}$$

where N_1, M_1, and H_{11} are upper triangular, H_{22} is upper quasi-triangular, N_2 and M_2 are skew-symmetric and the generalized matrix product $N_1^{-1} H_{11} M_1^{-1} H_{22}$ is in real periodic Schur form [7].

Then, by using the matrix decomposition from (1) we can compute an orthogonal matrix $\mathcal{Q}$ such that

$$\begin{aligned} \lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}} &:= \mathcal{J}\mathcal{Q}^T\mathcal{J}^T \left(\lambda\tilde{\mathcal{B}}_{\mathcal{N}} - \tilde{\mathcal{B}}_{\mathcal{H}}\right)\mathcal{Q} \\ &= \lambda \left[\begin{array}{cc|cc} N_1 & 0 & N_2 & 0 \\ 0 & M_1 & 0 & M_2 \\ \hline 0 & 0 & N_1^T & 0 \\ 0 & 0 & 0 & M_1^T \end{array}\right] - \left[\begin{array}{cc|cc} 0 & H_{11} & 0 & H_{12} \\ -H_{22} & 0 & H_{12} & 0 \\ \hline 0 & 0 & 0 & H_{22}^T \\ 0 & 0 & -H_{11}^T & 0 \end{array}\right], \end{aligned} \tag{2}$$

with $\mathcal{Q} = \mathcal{P}^T \left[\begin{smallmatrix} \mathcal{J}\mathcal{Q}_1\mathcal{J}^T & 0 \\ 0 & \mathcal{Q}_2 \end{smallmatrix}\right] \mathcal{P}$. Note, that we never explicitly construct the embedded pencils. It is sufficient to compute the necessary parts of the matrices in (1). The spectrum of $\lambda\mathcal{N} - \mathcal{H}$ is given by

$$\Lambda(\mathcal{N}, \mathcal{H}) = \pm \mathrm{i}\sqrt{\Lambda\left(N_1^{-1} H_{11} M_1^{-1} H_{22}\right)},$$

which can be determined by evaluating the entries on the 1×1 and 2×2 diagonal blocks of the matrices only. In particular, the simple, finite, purely imaginary eigenvalues correspond to the 1×1 diagonal blocks of this matrix product. Provided that the pairwise distance of the simple, finite, purely imaginary eigenvalues is sufficiently large, they can be computed in a robust way without any error in the real part. However, if a purely imaginary eigenvalue has an algebraic multiplicity larger than one or if two purely imaginary eigenvalues are close, they might still be perturbed off the imaginary axis. This depends on the sign-characteristic of the involved eigenvalues, similarly as in [18]. Therefore, for the rest of the paper we assume for simplicity that all finite, purely imaginary eigenvalues are simple.

To compute the eigenvectors corresponding to the finite, purely imaginary eigenvalues we will make use of the structure of $\lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}}$. As in passivity enforcement we only need the positive imaginary eigenvalues, i.e., those with positive imaginary parts, we restrict ourselves to the computation of the eigenvectors corresponding to these eigenvalues.

To derive an algorithm for computing the desired eigenvectors we make use of the following two lemmas. In the following we assume that the matrix pencil $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ H_{22} & 0 \end{smallmatrix}\right]$ is regular which is also equivalent to the regularity of the pencils $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ -H_{22} & 0 \end{smallmatrix}\right]$ and $\lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}}$ in (2).

Lemma 3. *The vector $\left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$ is a right eigenvector of the matrix pencil $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ H_{22} & 0 \end{smallmatrix}\right]$ corresponding to the eigenvalue ω_0 if and only if $\left[\begin{smallmatrix} -\mathrm{i}v_1 \\ v_2 \end{smallmatrix}\right]$ is a right eigenvector of the matrix pencil $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ -H_{22} & 0 \end{smallmatrix}\right]$ corresponding to the eigenvalue $\mathrm{i}\omega_0$.*

Proof. Let $\left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$ be a right eigenvector of $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ H_{22} & 0 \end{smallmatrix}\right]$ corresponding to the eigenvalue ω_0. Then we have

$$\begin{aligned} \omega_0 N_1 v_1 &= H_{11} v_2, \\ \omega_0 M_1 v_2 &= H_{22} v_1. \end{aligned}$$

This is equivalent to

$$
\begin{aligned}
\mathrm{i}\omega_0 N_1(-\mathrm{i}v_1) &= H_{11}v_2,\\
\mathrm{i}\omega_0 M_1 v_2 &= -H_{22}(-\mathrm{i}v_1).
\end{aligned}
$$

In other words, $\left[\begin{smallmatrix} -\mathrm{i}v_1 \\ v_2 \end{smallmatrix}\right]$ is a right eigenvector of the matrix pencil $\lambda\left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ -H_{22} & 0 \end{smallmatrix}\right]$ corresponding to the eigenvalue $\mathrm{i}\omega_0$. The converse statement can be proven in a completely analogous manner. □

Lemma 4. *The vector v is a right eigenvector of the matrix pencil $\lambda\left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ -H_{22} & 0 \end{smallmatrix}\right]$ to the eigenvalue λ_0 if and only if the vector $\left[\begin{smallmatrix} v \\ 0 \end{smallmatrix}\right]$ is a right eigenvector of the skew-Hamiltonian/Hamiltonian matrix pencil $\lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}}$ in (2) corresponding to the eigenvalue λ_0.*

Proof. Trivial. □

As an intermediate step, we compute a matrix X whose columns contain the eigenvectors to the positive real eigenvalues of the pencil $\lambda\left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ H_{22} & 0 \end{smallmatrix}\right]$. This is done by the following basic steps already summarized in [5].

Step 1: Reorder the positive real eigenvalues of the generalized matrix product $P := N_1^{-1}H_{11}M_1^{-1}H_{22}$ to the top, i.e., compute orthogonal matrices $U_i = \begin{bmatrix} U_i^{(1)} & U_i^{(2)} \end{bmatrix}$, $i = 1, \ldots, 4$, such that

$$
\begin{aligned}
U_2^T N_1 U_1 &= \begin{bmatrix} N_1^{(11)} & N_1^{(12)} \\ 0 & N_1^{(22)} \end{bmatrix}, \quad U_2^T H_{11} U_3 = \begin{bmatrix} H_{11}^{(11)} & H_{11}^{(12)} \\ 0 & H_{11}^{(22)} \end{bmatrix},\\
U_4^T M_1 U_3 &= \begin{bmatrix} M_1^{(11)} & M_1^{(12)} \\ 0 & M_1^{(22)} \end{bmatrix}, \quad U_4^T H_{22} U_1 = \begin{bmatrix} H_{22}^{(11)} & H_{22}^{(12)} \\ 0 & H_{22}^{(22)} \end{bmatrix}
\end{aligned}
$$

are still in upper (quasi-)triangular form, but the eigenvalues of the generalized matrix product $P^{(11)} := \left(N_1^{(11)}\right)^{-1} H_{11}^{(11)} \left(M_1^{(11)}\right)^{-1} H_{22}^{(11)}$ are the positive real ones of P [13].

Step 2: Reorder the eigenvalues $\lambda\begin{bmatrix} N_1^{(11)} & 0 \\ 0 & M_1^{(11)} \end{bmatrix} - \begin{bmatrix} 0 & H_{11}^{(11)} \\ H_{22}^{(11)} & 0 \end{bmatrix}$ by computing orthogonal matrices $V_1 = \begin{bmatrix} V_1^{(1)} & V_1^{(2)} \end{bmatrix}$, $V_2 = \begin{bmatrix} V_2^{(1)} & V_2^{(2)} \end{bmatrix}$ such that

$$
V_1^T \left(\lambda \begin{bmatrix} N_1^{(11)} & 0 \\ 0 & M_1^{(11)} \end{bmatrix} - \begin{bmatrix} 0 & H_{11}^{(11)} \\ H_{22}^{(11)} & 0 \end{bmatrix} \right) V_2 = \lambda \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} - \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix},
$$

where $\Lambda(R_{11}, S_{11}) \subset \mathbb{R}^+$ and $\Lambda(R_{22}, S_{22}) \subset \mathbb{R}^-$.

Step 3: Compute the eigenvectors of $\lambda R_{11} - S_{11}$, i.e., compute a matrix W such that $S_{11}W = R_{11}WD$, where D is an appropriate diagonal matrix composed of the eigenvalues of $\lambda R_{11} - S_{11}$.

Step 4: Collect the information contained in the relevant columns of the transformation matrices to obtain

$$
X := \begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} := \begin{bmatrix} U_1^{(1)} & 0 \\ 0 & U_3^{(1)} \end{bmatrix} V_2^{(1)} W.
$$

Now, using Lemma 3 it turns out that

$$
\tilde{X} := \begin{bmatrix} -\mathrm{i}X^{(1)} \\ X^{(2)} \end{bmatrix}
$$

contains the eigenvectors corresponding to the positive imaginary eigenvalues of the pencil $\lambda \left[\begin{smallmatrix} N_1 & 0 \\ 0 & M_1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} 0 & H_{11} \\ -H_{22} & 0 \end{smallmatrix}\right]$. Then, by employing Lemma 4, the columns of the matrix $\left[\begin{smallmatrix} \tilde{X} \\ 0 \end{smallmatrix}\right]$ contain eigenvectors to the positive imaginary eigenvalues of the pencil $\lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}}$. Note that all eigenvalues of this pencil have double algebraic, geometric, and partial multiplicities. So the matrix $\left[\begin{smallmatrix} \tilde{X} \\ 0 \end{smallmatrix}\right]$ contains only *half* of the eigenvectors to each positive imaginary eigenvalue of $\lambda\hat{\mathcal{B}}_{\mathcal{N}} - \hat{\mathcal{B}}_{\mathcal{H}}$. However, this is no problem, since by later turning over to the original pencil $\lambda\mathcal{N} - \mathcal{H}$, we do not need the other half of the eigenvectors.

Now, the corresponding eigenvectors to the positive imaginary eigenvalues of the double-sized matrix pencil $\lambda\mathcal{B}_{\mathcal{N}} - \mathcal{B}_{\mathcal{H}}$ are given by

$$Y := \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \mathcal{X}\mathcal{Q}\begin{bmatrix} \tilde{X} \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -\mathrm{i}\mathcal{Q}_1^{(22)}X^{(1)} + \mathcal{Q}_2^{(11)}X^{(2)} \\ \mathrm{i}\mathcal{Q}_1^{(12)}X^{(1)} + \mathcal{Q}_2^{(21)}X^{(2)} \\ \mathrm{i}\mathcal{Q}_1^{(22)}X^{(1)} + \mathcal{Q}_2^{(11)}X^{(2)} \\ -\mathrm{i}\mathcal{Q}_1^{(12)}X^{(1)} + \mathcal{Q}_2^{(21)}X^{(2)} \end{bmatrix},$$

where $\mathcal{Q}_1 := \left[\begin{smallmatrix} \mathcal{Q}_1^{(11)} & \mathcal{Q}_1^{(12)} \\ \mathcal{Q}_1^{(21)} & \mathcal{Q}_1^{(22)} \end{smallmatrix}\right]$ with $\mathcal{Q}_1^{(ij)} \in \mathbb{R}^{n\times n}$ and $\mathcal{Q}_2 := \left[\begin{smallmatrix} \mathcal{Q}_2^{(11)} & \mathcal{Q}_2^{(12)} \\ \mathcal{Q}_2^{(21)} & \mathcal{Q}_2^{(22)} \end{smallmatrix}\right]$ with $\mathcal{Q}_2^{(ij)} \in \mathbb{R}^{n\times n}$. In the equation above, only Y_1 contains the desired eigenvectors of the matrix pencil $\lambda\mathcal{N} - \mathcal{H}$ and Y has not to be computed explicitly. More specifically, we can express Y_1 as

$$Y_1 := \frac{1}{\sqrt{2}}\begin{bmatrix} -\mathrm{i}\mathcal{Q}_1^{(22)}X^{(1)} + \mathcal{Q}_2^{(11)}X^{(2)} \\ \mathrm{i}\mathcal{Q}_1^{(12)}X^{(1)} + \mathcal{Q}_2^{(21)}X^{(2)} \end{bmatrix}.$$

Within the next sections we will focus on the implementation of the SLICOT routine `MB04BV` that enables us to compute Y_1.

3 Specification

```
      SUBROUTINE MB04BV( N, N1, LDN1, M1, LDM1, H11, LDH11, H22, LDH22,
     $                   Q1, LDQ1, Q2, LDQ2, NEIG, ALPHAI, BETA, EVEC,
     $                   LDEVEC, DWORK, LDWORK, BWORK, INFO )
C
C     .. Scalar Arguments ..
      INTEGER           INFO, LDEVEC, LDH11, LDH22, LDM1, LDN1,
     $                  LDQ1, LDQ2, LDWORK, N, NEIG
C
C     .. Array Arguments ..
      LOGICAL           BWORK( * )
      DOUBLE PRECISION  ALPHAI( * ), BETA( * ), DWORK( * ),
     $                  H11( LDH11, * ), H22( LDH22, * ),
     $                  M1( LDM1, * ), N1( LDN1, * ), Q1( LDQ1, * ),
     $                  Q2( LDQ2, * )
      COMPLEX*16        EVEC( LDEVEC, * )
```

4 Argument List

4.1 Input/Output Parameters

N - (input) INTEGER

The order of the matrix pencil. `N` has to be greater than or equal to 0. `N` also must be even.

`N1` - (input) DOUBLE PRECISION array, dimension (`LDN1`, `N/2`)

On entry, the leading `N/2`-by-`N/2` part of this array must contain the upper triangular matrix N_1 in (1).

`LDN1` - INTEGER

The leading dimension of the array `N1`. The parameter `LDN1` has to be greater or equal than max {1, `N/2`}.

`M1` - (input) DOUBLE PRECISION array, dimension (`LDM1`, `N/2`)

On entry, the leading `N/2`-by-`N/2` part of this array must contain the upper triangular matrix M_1 in (1).

`LDM1` - INTEGER

The leading dimension of the array `M1`. The parameter `LDM1` has to be greater or equal than max {1, `N/2`}.

`H11` - (input) DOUBLE PRECISION array, dimension (`LDH11`, `N/2`)

On entry, the leading `N/2`-by-`N/2` part of this array must contain the upper triangular matrix H_{11} in (1).

`LDH11` - INTEGER

The leading dimension of the array `H11`. The parameter `LDH11` has to be greater or equal than max {1, `N/2`}.

`H22` - (input) DOUBLE PRECISION array, dimension (`LDH22`, `N/2`)

On entry, the leading `N/2`-by-`N/2` part of this array must contain the upper quasi-triangular matrix H_{22} in (1).

`LDH22` - INTEGER

The leading dimension of the array `H22`. The parameter `LDH22` has to be greater or equal than max {1, `N/2`}.

`Q1` - (input) DOUBLE PRECISION array, dimension (`LDQ1`, `N`)

On entry, the leading `N`-by-`N` part of this array must contain the orthogonal transformation matrix Q_1 in (1).

`LDQ1` - INTEGER

The leading dimension of the array `Q1`. The parameter `LDQ1` has to be greater or equal than max {1, `N`}.

`Q2` - (input) DOUBLE PRECISION array, dimension (`LDQ2`, `N`)

On entry, the leading `N`-by-`N` part of this array must contain the orthogonal transformation matrix Q_2 in (1).

`LDQ2` - INTEGER

The leading dimension of the array `Q2`. The parameter `LDQ2` has to be greater or equal than max $\{1, \mathtt{N}\}$.

NEIG - (output) INTEGER

The number of simple, finite, positive imaginary eigenvalues.

ALPHAI - (output) DOUBLE PRECISION array, dimension (N/2),
BETA - (output) DOUBLE PRECISION array, dimension (N/2)

On exit, the ratio $\mathrm{i} \cdot \mathtt{ALPHAI}(j)/\mathtt{BETA}(j)$, $j = 1, \ldots, \mathtt{NEIG}$, represents the simple, finite, positive imaginary eigenvalues of the matrix pencil.

EVEC - (output) COMPLEX*16 array, dimension (LDEVEC, N/2)

On exit, the leading N-by-NEIG part of this array contains the eigenvectors corresponding to the eigenvalues represented by the arrays ALPHAI and BETA in the same order.

LDEVEC - INTEGER

The leading dimension of the array EVEC. The parameter LDEVEC has to be greater or equal than max $\{1, \mathtt{N}\}$.

4.2 Workspace

DWORK - DOUBLE PRECISION array, dimension (LDWORK)

On exit, if $\mathtt{INFO} = 0$, DWORK(1) returns the optimal LDWORK. On exit, if $\mathtt{INFO} = -20$, DWORK(1) returns the minimum value of LDWORK.

LDWORK - INTEGER

The dimension of the array DWORK. $\mathtt{LDWORK} \geq 6(\mathtt{N}/2)^2 + \max\left\{3\mathtt{N}^2 + 9\mathtt{N} + 16, 272\right\}$. For good performance LDWORK should be generally larger.

If $\mathtt{LDWORK} = -1$, then a workspace query is assumed; the routine only calculates the optimal size of the DWORK array, returns this value as the first entry of the DWORK array, and no error message related to LDWORK is issued by XERBLA.

BWORK - LOGICAL array, dimension(N/2)

4.3 Error Indicator

INFO - INTEGER

$\mathtt{INFO} = 0$: Successful exit;
$\mathtt{INFO} < 0$: If $\mathtt{INFO} = -i$, the i-th argument had an illegal value;
$\mathtt{INFO} = 1$: The eigenvalue reordering in MB03KD failed;
$\mathtt{INFO} = 2$: The triangularization or reordering in DGGES failed;
$\mathtt{INFO} = 3$: The eigenvector computation in DGGEV failed.

5 Numerical Experiments

5.1 Setup of Test Examples

In order to test the performance of MB04BV, we feed it with random examples which have purely imaginary eigenvalues. We randomly generate matrices $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times m}$, and define the transfer function $G(s) = C(sE - A)^{-1}B + D$. Let

$$\lambda\mathcal{N} - \mathcal{H} = \lambda \left[\begin{array}{cc|cc} E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & E^T & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] - \left[\begin{array}{cc|cc} A & B & 0 & 0 \\ C & D & 0 & \gamma I_m \\ \hline 0 & 0 & -A^T & -C^T \\ 0 & -\gamma I_m & -B^T & -D^T \end{array}\right]. \tag{3}$$

Under some assumptions, the matrix pencil $\lambda\mathcal{N} - \mathcal{H}$ is guaranteed to have purely imaginary eigenvalues if $\min_{\omega\in\mathbb{R}} \sigma_{\max}(G(\mathrm{i}\omega)) < \gamma < \|G\|_{\mathcal{L}_\infty}$, where $\sigma_{\max}(\cdot)$ denotes the largest singular value and $\|\cdot\|_{\mathcal{L}_\infty}$ is the $\mathcal{L}_\infty$-norm [19]. Theoretically, when the distance between γ and $\|G\|_{\mathcal{L}_\infty}$ decreases, the difficulty of the example will increase, in the sense that the eigenvalues will be increasingly sensitive to perturbations, as the numerical results will later demonstrate. This is due to the fact, that there are two purely imaginary eigenvalues which almost form a non-trivial Jordan block in the Weierstraß canonical form [11]. Then, the transformation matrices will be ill-conditioned, which leads to a higher sensitivity of these eigenvalues.

5.2 Environment and Configuration

The tests have been performed on a 2.6.32-23-generic-pae Ubuntu machine with Intel® Core™2 Duo CPU with 3.00GHz and 4GB RAM. The algorithms have been implemented and tested in MATLAB 7.14.0.739 (R2012a).

A FORTRAN MEX-file as an interface between MATLAB and MB04BV has also been written for testing purposes and improving the user-friendliness. Using MATLAB, one can easily generate random examples, compute the $\mathcal{L}_\infty$-norm of the descriptor system, determine the eigenvalues by the structure-preserving solver MB04BD (see [3, 4] for implementation details) and the corresponding MEX-file skewHamileig, and profile the performance. The MATLAB interface for MB04BV is given by

```
[ alpha, beta, evecs ] = sHH_evecs( N1, M1, H11, H22, Q1, Q2 ).
```

Here, N1, M1, H11, H22, Q1, and Q2 are the matrices N_1, M_1, H_{11}, H_{22}, Q_1, and Q_2 in (1), respectively. The purely imaginary eigenvalues are given by $\mathrm{i}\frac{\alpha_i}{\beta_i}$, and the corresponding eigenvectors are stored in the columns of evecs in the same order as the eigenvalues.

In this section we also compare our structure-preserving approach with the standard one for general eigenvalue problems, namely the QZ algorithm (with eigenvalue reordering) [12]. In order to have a fair comparison of the performance of both methods, we also implemented a FORTRAN 77 subroutine which combines the QZ algorithm with reordering the purely imaginary eigenvalues to the top by using the LAPACK subroutines DGGES and DGGEV. For testing, a corresponding MEX-file has been written.

5.3 Numerical Results

5.3.1 Performance Comparison

In this paragraph we will discuss the behavior of the new approach compared to the QZ algorithm. We do this by constructing random pencils of the form (3) with $n = 100$ and $m = 5$. Table 1 shows the performance results of both algorithms when computing both desired eigenvalues and eigenvectors for

Table 1: Performance comparison

γ	new algorithm runtime	new algorithm avg. rel. residual	QZ algorithm runtime	QZ algorithm avg. rel. residual	QZ algorithm failure rate
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-2}\right)$	99.48	1.1936e-13	154.62	1.0388e-13	0.0%
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-4}\right)$	99.53	1.5555e-13	153.96	3.2024e-13	0.1%
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-6}\right)$	99.45	1.3882e-13	153.74	3.2727e-12	0.8%
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-8}\right)$	99.92	1.1820e-13	153.69	1.9054e-11	4.6%
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-10}\right)$	99.42	1.3450e-13	151.76	6.6909e-11	28.9%
$\|G\|_{\mathcal{L}_\infty}\left(1-10^{-12}\right)$	99.51	1.3827e-13	147.13	6.5136e-11	78.7%

different values of γ, from the "easier" examples to the "harder" ones. Each row contains the results for a thousand test runs. The accuracy of the results is measured by computing the average of the relative residuals given by $\|(\lambda_i\mathcal{N}-\mathcal{H})\,v_i\|_2 \,/\, \|v_i\|_2$. The runtime is given in seconds, using the `tic` and `toc` commands in MATLAB. Furthermore, for the QZ algorithm we have an additional column that indicates the percentage of examples that could not be solved. This is due to the fact that eigenvalues might be perturbed off the imaginary axis and will not be considered as purely imaginary when the distance to the imaginary axis exceeds a certain threshold. For our tests this value is set to 1e-10. First of all, Table 1 shows that the QZ algorithm needs about 50% more time to execute than the new algorithm. However, the most important aspect of the new algorithm is the improved reliability. We can see that the failure rate of the QZ algorithm is dramatically increasing when the examples become more ill-conditioned. By failure we mean that the algorithm extracts a different number of eigenvectors, compared to the actual number of purely imaginary eigenvalues. Moreover, even in the cases where the QZ algorithm successfully extracts the eigenvectors, the average relative residual becomes significantly larger when the condition gets worse. On the other hand, the performance of the new algorithm is far more reliable and accurate. Its runtime and accuracy remain at the same level from the "easy" examples to the "harder" ones.

5.3.2 On the Failure of the QZ Algorithm

We now briefly describe the nature of QZ algorithm's failure by examining a small example. Consider the randomly generated 6×6 skew-Hamiltonian/Hamiltonian matrix pencil

$$\lambda\mathcal{N}-\mathcal{H} = \lambda\begin{bmatrix} 0.7060 & 0.2769 & 0 & 0 & 0 & 0\\ 0.0318 & 0.0462 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0.7060 & 0.0318 & 0\\ 0 & 0 & 0 & 0.2769 & 0.0462 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$-\begin{bmatrix} 0.7431 & 0.6555 & 0.0971 & 0 & 0 & 0\\ 0.3922 & 0.1712 & 0.8235 & 0 & 0 & 0\\ 0.6948 & 0.3171 & 0.9502 & 0 & 0 & 0.9502\\ 0 & 0 & 0 & -0.7431 & -0.3922 & -0.6948\\ 0 & 0 & 0 & -0.6555 & -0.1712 & -0.3171\\ 0 & 0 & -0.9502 & -0.0971 & -0.8235 & -0.9502 \end{bmatrix},$$

with $\gamma = \|G\|_{\mathcal{L}_\infty}\left(1-10^{-6}\right)$. The spectrum is given by

$$\Lambda = \{927.5\mathrm{i}, -927.5\mathrm{i}, 1.161, -1.161, \infty, \infty\}.$$

In the spectrum, only the purely imaginary eigenvalue 927.5i is interesting to us. The new algorithm successfully extracts one eigenvector with relative residual 1.8594e-15.

However, the QZ algorithm fails to extract the eigenvector. Since it does not respect the structure of the pencil, all purely imaginary eigenvalues will be perturbed off the imaginary axis. When we are selecting the purely imaginary eigenvalues, we cannot expect that the real part of an eigenvalue is exactly zero, as it is for the structure-preserving algorithm. Alternatively, what we do is to ask if the real part is smaller than some tolerance. This tolerance is empirically set to 1e-10. This is a rather tight bound, but it illustrates the behavior of the QZ algorithm quite well. However the real part of this eigenvalue after perturbation is 1.4079e-10. Because the real part is slightly larger than the tolerance, this eigenvalue is not selected and thus no eigenvector is extracted.

6 Conclusions

In this report we have presented a FORTRAN 77 implementation of a new algorithm for computing the eigenvectors of a skew-Hamiltonian/Hamiltonian matrix pencil associated to the purely imaginary eigenvalues. The performed numerical tests clearly indicate that compared to the QZ algorithm the new method

1. is more robust, especially for ill-conditioned examples;
2. is comparably accurate for well-conditioned examples, and significantly more accurate for ill-conditioned examples;
3. needs only about 2/3 of time to execute.

Acknowledgment

The main part of this work was done while the first author visited the second author as an intern in summer 2012. The financial support from the RISE program of the *Deutscher Akademischer Austauschdienst* is gratefully acknowledged.

References

[1] P. Benner, R. Byers, V. Mehrmann, and H. Xu. Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Matrix Anal. Appl.*, 24(1):165–190, 2002.

[2] P. Benner, V. Sima, and M. Voigt. $\mathcal{L}_\infty$-norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans. Automat. Control*, 57(1):233–238, January 2012.

[3] P. Benner, V. Sima, and M. Voigt. FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems – Part I: Algorithms and applications. SLICOT Working Note 2013-1, August 2013.

[4] P. Benner, V. Sima, and M. Voigt. FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems – Part II: Implementation and numerical results. SLICOT Working Note 2013-2, August 2013.

[5] P. Benner and M. Voigt. On the computation of particular eigenvectors of Hamiltonian matrix pencils. *Proc. Appl. Math. Mech.*, 11(1):753–754, 2011.

[6] P. Benner and M. Voigt. Spectral characterization and enforcement of negative imaginariness for descriptor systems. *Linear Algebra Appl.*, 439(4):1104–1129, August 2013. 17th Conference of the International Linear Algebra Society, Braunschweig, Germany, August 2011.

[7] A. I. Bojanczyk, G. H. Golub, and P. Van Dooren. The periodic Schur decomposition. Algorithms and applications. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations III*, volume 1770 of *Proc. SPIE*, pages 31–42, 1992.

[8] T. Brüll. Checking dissipativity of linear behaviour systems in kernel representation. *Math. Control Signals Systems*, 23:159–175, 2011.

[9] T. Brüll. *Dissipativity of linear quadratic systems.* PhD thesis, Institut für Mathematik, TU Berlin, March 2011.

[10] T. Brüll and C. Schröder. Dissipativity enforcement via perturbation of para-Hermitian pencils. *IEEE Trans. Circuits Syst. I. Regul. Pap.*, 60(1), January 2013.

[11] L. Dai. *Singular Control Systems*, volume 118 of *Lecture Notes in Control and Inform. Sci.* Springer-Verlag, Berlin/Heidelberg, 1989.

[12] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[13] R. Granat, B. Kågström, and D. Kressner. Computing periodic deflating subspaces associated with a specified set of eigenvalues. *BIT*, 43(1):1–18, 2003.

[14] S. Grivet-Talocia. Passivity enforcement via perturbation of Hamiltonian matrices. *IEEE Trans. Circuits Syst. I. Regul. Pap.*, 51(9):1755–1769, September 2004.

[15] S. Grivet-Talocia and A. Ubolli. On the generation of large passive macromodels for complex interconnect structures. *IEEE Trans. Adv. Packaging*, 29(1):39–54, February 2006.

[16] M. Mabrok, A. G. Kallapur, I. R. Petersen, and A. Lanzon. Spectral conditions for the negative imaginary property of transfer function matrices. In *Proceedings of the 18th IFAC World Congress*, pages 1302–1306, Milan, Italy, September 2011.

[17] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, volume 163 of *Lecture Notes in Control and Inform. Sci.* Springer-Verlag, Heidelberg, July 1991.

[18] V. Mehrmann and H. Xu. Perturbation of purely imaginary eigenvalues of Hamiltonian matrices under structured perturbations. *Electron. J. Linear Algebra*, 17:234 – 257, 2008.

[19] K. Zhou and J. D. Doyle. *Essentials of Robust Control.* Prentice Hall, Upper Saddle River, NJ, 1st edition, 1998.