

Algorithm 961: Fortran 77 Subroutines for the Solution of Skew-Hamiltonian/Hamiltonian Eigenproblems

PETER BENNER, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg
 VASILE SIMA, National Institute for Research and Development in Informatics, Bucharest
 MATTHIAS VOIGT, Institut für Mathematik, Technische Universität Berlin, Berlin

Skew-Hamiltonian/Hamiltonian matrix pencils $\lambda S - \mathcal{H}$ appear in many applications, including linear-quadratic optimal control problems, \mathcal{H}_∞ -optimization, certain multi-body systems and many other areas in applied mathematics, physics, and chemistry. In these applications it is necessary to compute certain eigenvalues and/or corresponding deflating subspaces of these matrix pencils. Recently developed methods exploit and preserve the skew-Hamiltonian/Hamiltonian structure and hence increase the reliability, accuracy, and performance of the computations. In this paper we describe the corresponding algorithms which have been implemented in the style of subroutines of the Subroutine Library in Control Theory (SLICOT). Furthermore we address some of their applications. We describe variants for real and complex problems as well as implementation details and perform numerical tests using real-world examples to demonstrate the superiority of the new algorithms compared to standard methods.

Categories and Subject Descriptors: D.3.2 [Programming Languages]: Language Classifications—Fortran 77; G.1.3 [Numerical Analysis]: Numerical Linear Algebra—eigenvalues and eigenvectors (direct and iterative methods)

General Terms: Algorithms, Documentation, Reliability

Additional Key Words and Phrases: Deflating subspaces, eigenvalue reordering, generalized eigenvalues, generalized Schur form, skew-Hamiltonian/Hamiltonian matrix pencil, software, structure-preservation

ACM Reference Format:

Benner, P., Sima, V. and Voigt, M. 2015. Algorithm xxx – Fortran 77 Subroutines for the Solution of Skew-Hamiltonian/Hamiltonian Eigenproblems. *ACM Trans. Math. Softw.* 42, 3, Article 24 (June 2016), 26 pages. DOI = 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

1. INTRODUCTION

In this article we discuss algorithms for the solution of generalized eigenvalue problems with skew-Hamiltonian/Hamiltonian structure. We are interested in the computation of certain eigenvalues and corresponding deflating subspaces. In this work we deal with the following algebraic structures [Benner et al. 2002]. As a notational convention we use calligraphic letters for block matrices and standard letters for ordinary

This work was supported by the *Deutsche Forschungsgemeinschaft* in the project “Numerical algorithms for generalized eigenvalue problems of even structure with application in robust control of descriptor systems” under grant BE-2174/6-1.

Author’s addresses: Peter Benner, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany, e-mail: benner@mpi-magdeburg.mpg.de; Vasile Sima, National Institute for Research & Development in Informatics (ICI), Lab.07, 8-10 Averescu Ave., Bucharest 1, Romania, code 011455, e-mail: vsima@ici.ro; Matthias Voigt, Technische Universität Berlin, Sekretariat MA 4-5, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: mvoigt@math.tu-berlin.de.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2016 ACM 0098-3500/2016/06-ART24 \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

matrices. Moreover, we denote by $\mathbb{R}[\lambda]^{n \times n}$ and $\mathbb{C}[\lambda]^{n \times n}$ the rings of polynomials with coefficients in $\mathbb{R}^{n \times n}$ and $\mathbb{C}^{n \times n}$, respectively.

Definition 1.1. Let $\mathcal{J} := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$, where I_n is the $n \times n$ identity matrix. For brevity of notation, we do not indicate the dimension of the matrix \mathcal{J} and use it for all possible values of n .

- (i) A matrix $\mathcal{H} \in \mathbb{C}^{2n \times 2n}$ is *Hamiltonian* if $(\mathcal{H}\mathcal{J})^H = \mathcal{H}\mathcal{J}$.
- (ii) A matrix $\mathcal{S} \in \mathbb{C}^{2n \times 2n}$ is *skew-Hamiltonian* if $(\mathcal{S}\mathcal{J})^H = -\mathcal{S}\mathcal{J}$.
- (iii) A matrix pencil $\lambda\mathcal{S} - \mathcal{H} \in \mathbb{C}[\lambda]^{2n \times 2n}$ is *skew-Hamiltonian/Hamiltonian* if \mathcal{S} is skew-Hamiltonian and \mathcal{H} is Hamiltonian.
- (iv) A matrix $\mathcal{S} \in \mathbb{C}^{2n \times 2n}$ is *symplectic* if $\mathcal{S}\mathcal{J}\mathcal{S}^H = \mathcal{J}$.
- (v) A matrix $\mathcal{U} \in \mathbb{C}^{2n \times 2n}$ is *unitary symplectic* if $\mathcal{U}\mathcal{J}\mathcal{U}^H = \mathcal{J}$ and $\mathcal{U}\mathcal{U}^H = I_{2n}$.

Note that a similar definition can be given for real matrices. As a convention, all following considerations also hold for real skew-Hamiltonian/Hamiltonian matrix pencils. Then, all matrices \cdot^H must be replaced by \cdot^T , all (skew-)Hermitian matrices become (skew-)symmetric, and unitary matrices become orthogonal. More significant differences to the complex case are explicitly mentioned.

Skew-Hamiltonian/Hamiltonian matrix pencils satisfy certain properties which we will recall briefly. Every skew-Hamiltonian/Hamiltonian matrix pencil can be written as

$$\lambda\mathcal{S} - \mathcal{H} = \lambda \begin{bmatrix} A & D \\ E & A^H \end{bmatrix} - \begin{bmatrix} B & F \\ G & -B^H \end{bmatrix}$$

with skew-Hermitian matrices D , E and Hermitian matrices F , G . If λ_0 is a (generalized) eigenvalue of $\lambda\mathcal{S} - \mathcal{H}$, so is also $-\bar{\lambda}_0$. In other words, eigenvalues which are not purely imaginary, occur in pairs. For real skew-Hamiltonian/Hamiltonian matrix pencils we also have a pairing of complex conjugate eigenvalues, i. e., if λ_0 is an eigenvalue of $\lambda\mathcal{S} - \mathcal{H}$, so are also $\bar{\lambda}_0$, $-\lambda_0$, $-\bar{\lambda}_0$. This leads to eigenvalue pairs $(\lambda_0, -\lambda_0)$ if λ is purely real or purely imaginary, or otherwise to eigenvalue quadruples $(\lambda_0, \bar{\lambda}_0, -\lambda_0, -\bar{\lambda}_0)$.

The structure of skew-Hamiltonian/Hamiltonian matrix pencils is preserved under \mathcal{J} -congruence transformations, that is, $\lambda\tilde{\mathcal{S}} - \tilde{\mathcal{H}} := \mathcal{J}\mathcal{P}^H \mathcal{J}^T(\lambda\mathcal{S} - \mathcal{H})\mathcal{P}$ with nonsingular \mathcal{P} is again skew-Hamiltonian/Hamiltonian. If we choose \mathcal{P} unitary, we additionally preserve the condition of the problem. In this way there is hope that we can choose a unitary \mathcal{J} -congruence transformation to transform $\lambda\mathcal{S} - \mathcal{H}$ into a condensed form which reveals its eigenvalues and deflating subspaces. A suitable candidate for this condensed form is the *structured Schur form*, i. e., we compute a unitary matrix \mathcal{Q} such that

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

with the subpencil $\lambda S_{11} - H_{11}$ in generalized Schur form, where S_{11} is upper triangular, H_{11} is upper triangular (upper quasi-triangular in the real case), S_{12} is skew-Hermitian, and H_{12} is Hermitian. However, such a structured Schur form does *not* necessarily exist. Conditions for the existence are proven in [Mehl 1999; Mehl 2000] for the complex case or in [Voigt 2010] for the real case. This problem can be circumvented by embedding $\lambda\mathcal{S} - \mathcal{H}$ into a skew-Hamiltonian/Hamiltonian matrix pencil of double dimension in an appropriate way, as explained in Section 3.

Skew-Hamiltonian/Hamiltonian pencils are closely related to *even pencils* $\lambda\mathcal{E} - \mathcal{A}$, that is, $\mathcal{E} = -\mathcal{E}^H$ and $\mathcal{A} = \mathcal{A}^H$. It is easy to see that if $\lambda\mathcal{S} - \mathcal{H}$ is skew-Hamiltonian/Hamiltonian, then $\mathcal{J}(\lambda\mathcal{S} - \mathcal{H})$ is even. However, the converse is not true.

If $\lambda\mathcal{E} - \mathcal{A} \in \mathbb{C}[\lambda]^{k \times k}$ is even and of odd dimension, then it cannot be transformed directly to skew-Hamiltonian/Hamiltonian structure. Instead one has to inflate the pencil and introduce one artificial infinite eigenvalue. Define the number $r := k \bmod 2$. Then, the pencil

$$\lambda\mathcal{S} - \mathcal{H} := \mathcal{J} \left(\begin{bmatrix} \lambda\mathcal{E} - \mathcal{A} & 0 \\ 0 & I_r \end{bmatrix} \right)$$

is skew-Hamiltonian/Hamiltonian.

Usually, the eigenvalue problems that are obtained in practice are even. In this case we have to do the above transformation to skew-Hamiltonian/Hamiltonian form. Moreover, sometimes the pencils are singular or have a higher index. Then it is often necessary to remove the singular and higher-index parts before, so that one works with a well-behaved eigenvalue problem [Byers et al. 2007; Brüll and Mehrmann 2007].

Throughout this paper we denote by $\Lambda_-(\mathcal{S}, \mathcal{H})$, $\Lambda_0(\mathcal{S}, \mathcal{H})$, $\Lambda_+(\mathcal{S}, \mathcal{H})$ the set of finite eigenvalues of $\lambda\mathcal{S} - \mathcal{H}$ with negative, zero, and positive real parts, respectively. The set of all eigenvalues is denoted by $\Lambda(\mathcal{S}, \mathcal{H})$. Similarly, we denote by $\text{Defl}_-(\mathcal{S}, \mathcal{H})$, $\text{Defl}_0(\mathcal{S}, \mathcal{H})$, $\text{Defl}_+(\mathcal{S}, \mathcal{H})$, and $\text{Defl}_\infty(\mathcal{S}, \mathcal{H})$ the right deflating subspaces corresponding to $\Lambda_-(\mathcal{S}, \mathcal{H})$, $\Lambda_0(\mathcal{S}, \mathcal{H})$, $\Lambda_+(\mathcal{S}, \mathcal{H})$, and the infinite eigenvalues, respectively.

2. APPLICATIONS

2.1. Linear-Quadratic Optimal Control

First we consider the continuous-time, infinite horizon, linear-quadratic optimal control problem:

Choose a control function $u : [0, \infty) \rightarrow \mathbb{R}^m$ to minimize the cost functional

$$\mathcal{J}(x, u) := \int_0^\infty \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \quad (1)$$

subject to the linear time-invariant *descriptor system*

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad Ex(0) = Ex_0. \quad (2)$$

Here, $u : [0, \infty) \rightarrow \mathbb{R}^m$ is a control input, $x : [0, \infty) \rightarrow \mathbb{R}^n$ is the state, and $\lambda E - A \in \mathbb{R}[\lambda]^{n \times n}$ is regular, $B \in \mathbb{R}^{n \times m}$, $Q = Q^T \in \mathbb{R}^{n \times n}$, $R = R^T \in \mathbb{R}^{m \times m}$, $S \in \mathbb{R}^{n \times m}$. For well-posedness, the matrix R is assumed to be positive semidefinite.

Typically, in addition to minimizing (1), the control $u(\cdot)$ must make $x(\cdot)$ asymptotically stable. Under some regularity conditions, the application of the maximum principle [Pontryagin et al. 1962; Mehrmann 1991] yields as a necessary optimality condition that the control $u(\cdot)$ satisfies the two-point boundary value problem of Euler-Lagrange equations

$$\mathcal{E} \begin{bmatrix} \dot{\mu}(t) \\ \dot{x}(t) \\ \dot{u}(t) \end{bmatrix} = \mathcal{A} \begin{bmatrix} \mu(t) \\ x(t) \\ u(t) \end{bmatrix}, \quad Ex(0) = Ex_0, \quad \lim_{t \rightarrow \infty} E^T \mu(t) = 0$$

with the even matrix pencil

$$\lambda\mathcal{E} - \mathcal{A} = \lambda \begin{bmatrix} 0 & E & 0 \\ -E^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ A^T & Q & S \\ B^T & S^T & R \end{bmatrix}. \quad (3)$$

Under certain stabilizability conditions, the optimal control $u_*(\cdot)$ that stabilizes the descriptor system (2) can be constructed via a stabilizing solution of certain matrix equations. For the case that R is positive definite, this includes several versions of

generalized algebraic Riccati equations [Mehrman 1991; Kawamoto et al. 1999], for the general case with possibly singular R , one has to turn to so-called (descriptor) Lur'e equations [Reis 2011; Reis et al. 2015; Voigt 2015]. In all cases, the desired solution can be determined using the deflating subspaces of $\lambda\mathcal{E} - \mathcal{A}$ to the eigenvalues with negative real parts and to some purely imaginary and infinite eigenvalues.

2.2. \mathcal{H}_∞ -Optimization

Similar structures as in Subsection 2.1 occur in \mathcal{H}_∞ -optimization [Losse et al. 2008]. Consider a descriptor system of the form

$$\mathbf{P} : \begin{cases} E\dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t), \\ z(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t), \\ y(t) = C_2x(t) + D_{21}w(t) + D_{22}u(t), \end{cases} \quad (4)$$

where $\lambda E - A \in \mathbb{R}[\lambda]^{n \times n}$ is regular, $B_i \in \mathbb{R}^{n \times m_i}$, $C_i \in \mathbb{R}^{p_i \times n}$, and $D_{ij} \in \mathbb{R}^{p_i \times m_j}$ for $i, j = 1, 2$. In this system, $x : [0, \infty) \rightarrow \mathbb{R}^n$ is the state, $u : [0, \infty) \rightarrow \mathbb{R}^{m_2}$ is the control input, and $w : [0, \infty) \rightarrow \mathbb{R}^{m_1}$ is an exogenous input that may include noise, linearization errors, and unmodeled dynamics. The function $y : [0, \infty) \rightarrow \mathbb{R}^{p_2}$ contains measured outputs, while $z : [0, \infty) \rightarrow \mathbb{R}^{p_1}$ is a regulated output or an estimation error.

The \mathcal{H}_∞ control problem is naturally formulated in the frequency domain. For this we need the space $\mathcal{RH}_\infty^{p \times m}$ which consists of all real-rational $\mathbb{C}^{p \times m}$ -valued functions that are analytic and bounded in the open right half-plane \mathbb{C}^+ . For $G \in \mathcal{RH}_\infty^{p \times m}$, the \mathcal{H}_∞ -norm is defined by

$$\|G\|_{\mathcal{H}_\infty} := \sup_{s \in \mathbb{C}^+} \sigma_{\max}(G(s)) = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(i\omega)),$$

where $\sigma_{\max}(M)$ denotes the largest singular value of the matrix M . In robust control, $\|G\|_{\mathcal{H}_\infty}$ is used as a measure of the worst-case influence of the disturbances w on the output z , where in this case G is the transfer function mapping noise or disturbance inputs to error signals [Zhou and Doyle 1998]. Solving the optimal \mathcal{H}_∞ control problem is the task of designing a dynamic controller

$$\mathbf{K} : \begin{cases} \widehat{E}\dot{\widehat{x}}(t) = \widehat{A}\widehat{x}(t) + \widehat{B}y(t), \\ u(t) = \widehat{C}\widehat{x}(t) + \widehat{D}y(t), \end{cases} \quad (5)$$

with regular $\lambda\widehat{E} - \widehat{A} \in \mathbb{R}[\lambda]^{N \times N}$, $\widehat{B} \in \mathbb{R}^{N \times p_2}$, $\widehat{C} \in \mathbb{R}^{m_2 \times N}$, $\widehat{D} \in \mathbb{R}^{m_2 \times p_2}$ such that the closed-loop system resulting from inserting (5) into (4), that is,

$$\begin{aligned} E\dot{x}(t) &= (A + B_2\widehat{D}Z_1C_2)x(t) + B_2Z_2\widehat{C}\widehat{x}(t) + (B_1 + B_2\widehat{D}Z_1D_{21})w(t), \\ \widehat{E}\dot{\widehat{x}}(t) &= \widehat{B}Z_1C_2x(t) + (\widehat{A} + \widehat{B}Z_1D_{22}\widehat{C})\widehat{x}(t) + \widehat{B}Z_1D_{21}w(t), \\ z(t) &= (C_1 + D_{12}Z_2\widehat{D}C_2)x(t) + D_{12}Z_2\widehat{C}\widehat{x}(t) + (D_{11} + D_{12}\widehat{D}Z_1D_{21})w(t), \end{aligned} \quad (6)$$

with $Z_1 = (I_{p_2} - D_{22}\widehat{D})^{-1}$, and $Z_2 = (I_{m_2} - \widehat{D}D_{22})^{-1}$ has the following properties:

- (i) System (6) is internally stable, that is, the solution $\begin{bmatrix} x(t) \\ \widehat{x}(t) \end{bmatrix}$ of the system with $w \equiv 0$ is asymptotically stable, i. e., $\lim_{t \rightarrow \infty} \begin{bmatrix} x(t) \\ \widehat{x}(t) \end{bmatrix} = 0$.
- (ii) The closed-loop transfer function T_{zw} from w to z satisfies $T_{zw} \in \mathcal{RH}_\infty^{p_1 \times m_1}$ and is minimized in the \mathcal{H}_∞ -norm.

Closely related to the optimal \mathcal{H}_∞ control problem is the modified optimal \mathcal{H}_∞ control problem. For a given descriptor system of the form (4) we search the infimum value γ for which there exists an internally stabilizing dynamic controller of the form (5) such that the corresponding closed-loop system (6) satisfies $T_{zw} \in \mathcal{RH}_\infty^{p_1 \times m_1}$ with $\|T_{zw}\|_{\mathcal{H}_\infty} < \gamma$. For the construction of optimal controllers, one can make use of the following *even* matrix pencils (see [Pope et al. 2009] for a definition and related software)

$$\lambda\mathcal{E}_H - \mathcal{A}_H(\gamma) = \left[\begin{array}{cc|ccc} 0 & -\lambda E^T - A^T & 0 & 0 & -C_1^T \\ \lambda E - A & 0 & -B_1 & -B_2 & 0 \\ \hline 0 & -B_1^T & -\gamma^2 I_{m_1} & 0 & -D_{11}^T \\ 0 & -B_2^T & 0 & 0 & -D_{12}^T \\ -C_1 & 0 & -D_{11} & -D_{12} & -I_{p_1} \end{array} \right], \quad (7)$$

and

$$\lambda\mathcal{E}_J - \mathcal{A}_J(\gamma) = \left[\begin{array}{cc|ccc} 0 & -\lambda E - A & 0 & 0 & -B_1 \\ \lambda E^T - A^T & 0 & -C_1^T & -C_2^T & 0 \\ \hline 0 & -C_1 & -\gamma^2 I_{p_1} & 0 & -D_{11} \\ 0 & -C_2 & 0 & 0 & -D_{12} \\ -B_1^T & 0 & -D_{11}^T & -D_{12}^T & -I_{m_1} \end{array} \right]. \quad (8)$$

Using appropriate deflating subspaces of the matrix pencils (7) and (8) it is possible to state conditions for the existence of an optimal \mathcal{H}_∞ controller. Then we can check if these conditions are fulfilled for a given value of γ . Using a bisection scheme we can iteratively refine γ until a desired accuracy is achieved (see [Losse et al. 2008; Benner et al. 2008b] for details). Finally, when a suboptimal value γ has been found, one can compute the actual controller. The controller formulas are rather cumbersome and therefore, they are omitted. For details, see [Losse 2012].

2.3. \mathcal{H}_∞ -Norm Computation

Finally, we briefly describe a method to compute the \mathcal{H}_∞ -norm of an LTI system using even matrix pencils [Benner et al. 2012a; Benner et al. 2012b]. This norm plays an important role in robust control or model order reduction (see [Antoulas 2005; Mehrmann and Stykel 2005; Zhou and Doyle 1998] and references therein). Consider a descriptor system

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (9)$$

with regular $\lambda E - A \in \mathbb{R}[\lambda]^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$, and state $x : [0, \infty) \rightarrow \mathbb{R}^n$, control input $u : [0, \infty) \rightarrow \mathbb{R}^m$, and output $y : [0, \infty) \rightarrow \mathbb{R}^p$. For such a system the transfer function is given by

$$G(s) := C(sE - A)^{-1}B + D,$$

which directly maps inputs to outputs in the frequency domain [Dai 1989]. Assume that $G \in \mathcal{RH}_\infty^{p \times m}$ and consider the even matrix pencils

$$\lambda\mathcal{E} - \mathcal{A}(\gamma) := \left[\begin{array}{cc|cc} 0 & \lambda E - A & 0 & -B \\ -\lambda E^T - A^T & 0 & -C^T & 0 \\ \hline 0 & -C & -\gamma I_p & D \\ -B^T & 0 & D^T & -\gamma I_m \end{array} \right].$$

It can be shown that if $\lambda E - A$ has no purely imaginary eigenvalue and $\gamma > \inf_{\omega \in \mathbb{R}} \sigma_{\max}(G(i\omega))$, then $\|G\|_{\mathcal{H}_\infty} \geq \gamma$ if and only if $\lambda\mathcal{E} - \mathcal{A}(\gamma)$ has purely imaginary

eigenvalues. In this way we can again use an iterative scheme to update the value of γ until a desired accuracy for the \mathcal{H}_∞ -norm is achieved.

3. THEORY AND ALGORITHM DESCRIPTION

In this section we briefly describe the theory behind the algorithms that we will use. We refer to [Benner et al. 2002; Benner et al. 2007] for a detailed analysis of the algorithms. We consider complex and real problems separately since there are significant differences in the theory. We also distinguish the cases of unfactored and factored skew-Hamiltonian matrices S . This is motivated by the fact that in certain cases, a skew-Hamiltonian matrix S admits a factorization

$$S = \mathcal{J}Z^H \mathcal{J}^T Z. \quad (10)$$

For example, if $S = \begin{bmatrix} E & 0 \\ 0 & E^H \end{bmatrix}$, then $Z = \begin{bmatrix} I & 0 \\ 0 & E^H \end{bmatrix}$. The factorization (10) can be understood as a Cholesky-like decomposition of S with respect to the indefinite inner product $\langle x, y \rangle := x^H \mathcal{J} y$, since $\mathcal{J}Z^H \mathcal{J}^T$ is the adjoint of Z with respect to $\langle \cdot, \cdot \rangle$. We also say that a skew-Hamiltonian matrix S is \mathcal{J} -semidefinite, if it admits a factorization of the form (10). Hence, in our implementation we distinguish the cases that the full matrix S or just its "Cholesky factor" Z is given. In all cases we apply an embedding strategy to the matrix pencil $\lambda S - \mathcal{H}$ to ensure the existence of a structured Schur form.

3.1. The Complex Case

Let $\lambda S - \mathcal{H} \in \mathbb{C}[\lambda]^{2n \times 2n}$ be a given skew-Hamiltonian/Hamiltonian matrix pencil with \mathcal{J} -semidefinite skew-Hamiltonian part $S = \mathcal{J}Z^H \mathcal{J}^T Z$. We split the skew-Hamiltonian matrix $i\mathcal{H} =: \mathcal{N} = \mathcal{N}_1 + i\mathcal{N}_2$, where \mathcal{N}_1 is real skew-Hamiltonian and \mathcal{N}_2 is real Hamiltonian, i. e.,

$$\begin{aligned} \mathcal{N}_1 &= \begin{bmatrix} F_1 & G_1 \\ H_1 & F_1^T \end{bmatrix}, & G_1 &= -G_1^T, & H_1 &= -H_1^T, \\ \mathcal{N}_2 &= \begin{bmatrix} F_2 & G_2 \\ H_2 & -F_2^T \end{bmatrix}, & G_2 &= G_2^T, & H_2 &= H_2^T, \end{aligned}$$

and $F_j, G_j, H_j \in \mathbb{R}^{n \times n}$ for $j = 1, 2$. We define the matrices

$$\mathcal{Y}_c = \frac{\sqrt{2}}{2} \begin{bmatrix} I_{2n} & iI_{2n} \\ I_{2n} & -iI_{2n} \end{bmatrix}, \quad \mathcal{P} = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & I_n \end{bmatrix}, \quad \mathcal{X}_c = \mathcal{Y}_c \mathcal{P}. \quad (11)$$

When denoting by \overline{M} the complex conjugate matrix of M , and performing the embedding $\mathcal{B}_\mathcal{N} := \text{diag}(\mathcal{N}, \overline{\mathcal{N}})$ we obtain that

$$\mathcal{B}_\mathcal{N}^c := \mathcal{X}_c^H \mathcal{B}_\mathcal{N} \mathcal{X}_c = \begin{bmatrix} F_1 & -F_2 & G_1 & -G_2 \\ F_2 & F_1 & G_2 & G_1 \\ H_1 & -H_2 & F_1^T & F_2^T \\ H_2 & H_1 & -F_2^T & F_1^T \end{bmatrix} \quad (12)$$

is a real $4n \times 4n$ skew-Hamiltonian matrix. Similarly, we define

$$\mathcal{B}_Z := \begin{bmatrix} Z & 0 \\ 0 & \overline{Z} \end{bmatrix}, \quad \mathcal{B}_\mathcal{T} := \begin{bmatrix} \mathcal{J}Z^H \mathcal{J}^T & 0 \\ 0 & \mathcal{J}Z^H \mathcal{J}^T \end{bmatrix}, \quad \mathcal{B}_S := \begin{bmatrix} S & 0 \\ 0 & \overline{S} \end{bmatrix} = \mathcal{B}_\mathcal{T} \mathcal{B}_Z.$$

It can be shown that

$$\mathcal{B}_Z^c := \mathcal{X}_c^H \mathcal{B}_Z \mathcal{X}_c, \quad \mathcal{B}_\mathcal{T}^c := \mathcal{X}_c^H \mathcal{B}_\mathcal{T} \mathcal{X}_c, \quad \mathcal{B}_S^c := \mathcal{X}_c^H \mathcal{B}_S \mathcal{X}_c \quad (13)$$

are all real. Hence,

$$\lambda \mathcal{B}_S^c - \mathcal{B}_N^c = \mathcal{X}_c^H (\lambda \mathcal{B}_S - \mathcal{B}_N) \mathcal{X}_c = \mathcal{X}_c^H \left(\begin{bmatrix} \lambda \mathcal{S} - \mathcal{N} & 0 \\ 0 & \lambda \overline{\mathcal{S}} - \overline{\mathcal{N}} \end{bmatrix} \right) \mathcal{X}_c$$

is a real $4n \times 4n$ skew-Hamiltonian/skew-Hamiltonian matrix pencil. To compute the eigenvalues of this matrix pencil we can compute the structured decomposition of the following theorem [Benner et al. 2002].

THEOREM 3.1. *Let $\lambda \mathcal{S} - \mathcal{N} \in \mathbb{R}[\lambda]^{2n \times 2n}$ be a regular skew-Hamiltonian/skew-Hamiltonian matrix pencil with $\mathcal{S} = \mathcal{J} \mathcal{Z}^T \mathcal{J}^T \mathcal{Z}$. Then there exist a real orthogonal matrix $\mathcal{Q} \in \mathbb{R}^{2n \times 2n}$ and a real orthogonal symplectic matrix $\mathcal{U} \in \mathbb{R}^{2n \times 2n}$ such that*

$$\begin{aligned} \mathcal{U}^T \mathcal{Z} \mathcal{Q} &= \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}, \\ \mathcal{J} \mathcal{Q}^T \mathcal{J}^T \mathcal{N} \mathcal{Q} &= \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^T \end{bmatrix}, \end{aligned} \tag{14}$$

where Z_{11} and Z_{22}^T are upper triangular, N_{11} is upper quasi-triangular and N_{12} is skew-symmetric. Moreover,

$$\begin{aligned} \mathcal{J} \mathcal{Q}^T \mathcal{J}^T (\lambda \mathcal{S} - \mathcal{N}) \mathcal{Q} &= \lambda \begin{bmatrix} Z_{22}^T Z_{11} & Z_{22}^T Z_{12} - Z_{12}^T Z_{22} \\ 0 & Z_{11}^T Z_{22} \end{bmatrix} - \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^T \end{bmatrix} \\ &=: \lambda \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix} - \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^T \end{bmatrix} \end{aligned} \tag{15}$$

is a \mathcal{J} -congruent skew-Hamiltonian/skew-Hamiltonian matrix pencil.

PROOF. See [Benner et al. 2002]. \square

By defining

$$\mathcal{B}_\mathcal{H} = \begin{bmatrix} \mathcal{H} & 0 \\ 0 & -\overline{\mathcal{H}} \end{bmatrix}, \quad \mathcal{B}_\mathcal{H}^c = \mathcal{X}_c^H \mathcal{B}_\mathcal{H} \mathcal{X}_c,$$

and using Theorem 3.1 we can compute factorizations

$$\begin{aligned} \tilde{\mathcal{B}}_\mathcal{Z}^c &:= \mathcal{U}^T \mathcal{B}_\mathcal{Z}^c \mathcal{Q} = \begin{bmatrix} \mathcal{Z}_{11} & \mathcal{Z}_{12} \\ 0 & \mathcal{Z}_{22} \end{bmatrix}, \\ \tilde{\mathcal{B}}_\mathcal{H}^c &:= \mathcal{J} \mathcal{Q}^T \mathcal{J}^T \mathcal{B}_\mathcal{H}^c \mathcal{Q} = \mathcal{J} \mathcal{Q}^T \mathcal{J}^T (-i \mathcal{B}_\mathcal{N}^c) \mathcal{Q} = -i \tilde{\mathcal{B}}_\mathcal{N}^c = \begin{bmatrix} -i \mathcal{N}_{11} & -i \mathcal{N}_{12} \\ 0 & -(-i \mathcal{N}_{11})^H \end{bmatrix}, \end{aligned}$$

where $\lambda \tilde{\mathcal{B}}_\mathcal{S}^c - \tilde{\mathcal{B}}_\mathcal{H}^c = \mathcal{J} \mathcal{Q}^T \mathcal{J}^T (\lambda \mathcal{B}_\mathcal{S}^c - \mathcal{B}_\mathcal{H}^c) \mathcal{Q}$ are \mathcal{J} -congruent complex skew-Hamiltonian/Hamiltonian matrix pencils and $\lambda \tilde{\mathcal{B}}_\mathcal{S}^c - \tilde{\mathcal{B}}_\mathcal{H}^c$ is in a structured quasi-triangular form. Then, the structured Schur form can be obtained by further triangularizing the diagonal 2×2 blocks of $\lambda \tilde{\mathcal{B}}_\mathcal{S}^c - \tilde{\mathcal{B}}_\mathcal{H}^c$ via a \mathcal{J} -congruence transformation. From the symmetry of the eigenvalues it follows that $\Lambda(\mathcal{S}, \mathcal{H}) = \Lambda(\mathcal{Z}_{22}^H \mathcal{Z}_{11}, -i \mathcal{N}_{11})$. Now we can reorder the eigenvalues of $\lambda \tilde{\mathcal{B}}_\mathcal{S}^c - \tilde{\mathcal{B}}_\mathcal{H}^c$ to the top in order to compute the desired deflating subspaces (corresponding to the eigenvalues with negative real parts). The following theorem makes statements about the deflating subspaces [Benner et al. 2002].

THEOREM 3.2. *Let $\lambda \mathcal{S} - \mathcal{H} \in \mathbb{C}[\lambda]^{2n \times 2n}$ be a skew-Hamiltonian/Hamiltonian matrix pencil with \mathcal{J} -semidefinite skew-Hamiltonian matrix $\mathcal{S} = \mathcal{J} \mathcal{Z}^H \mathcal{J}^T \mathcal{Z}$. Consider the extended matrices $\mathcal{B}_\mathcal{Z} = \text{diag}(\mathcal{Z}, \overline{\mathcal{Z}})$, $\mathcal{B}_\mathcal{T} = \text{diag}(\mathcal{J} \mathcal{Z}^H \mathcal{J}^T, \overline{\mathcal{J} \mathcal{Z}^H \mathcal{J}^T})$, $\mathcal{B}_\mathcal{S} = \mathcal{B}_\mathcal{T} \mathcal{B}_\mathcal{Z} =$*

$\text{diag}(\mathcal{S}, \overline{\mathcal{S}})$, $\mathcal{B}_{\mathcal{H}} = \text{diag}(\mathcal{H}, -\overline{\mathcal{H}})$. Let \mathcal{U} , \mathcal{V} , \mathcal{W} be unitary matrices such that

$$\begin{aligned}\mathcal{U}^H \mathcal{B}_{\mathcal{Z}} \mathcal{V} &= \begin{bmatrix} \mathcal{Z}_{11} & \mathcal{Z}_{12} \\ 0 & \mathcal{Z}_{22} \end{bmatrix} =: \mathcal{R}_{\mathcal{Z}}, \\ \mathcal{W}^H \mathcal{B}_{\mathcal{T}} \mathcal{U} &= \begin{bmatrix} \mathcal{T}_{11} & \mathcal{T}_{12} \\ 0 & \mathcal{T}_{22} \end{bmatrix} =: \mathcal{R}_{\mathcal{T}}, \\ \mathcal{W}^H \mathcal{B}_{\mathcal{H}} \mathcal{V} &= \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ 0 & \mathcal{H}_{22} \end{bmatrix} =: \mathcal{R}_{\mathcal{H}},\end{aligned}$$

where $\Lambda_-(\mathcal{B}_{\mathcal{S}}, \mathcal{B}_{\mathcal{H}}) \subset \Lambda(\mathcal{T}_{11} \mathcal{Z}_{11}, \mathcal{H}_{11})$ and $\Lambda(\mathcal{T}_{11} \mathcal{Z}_{11}, \mathcal{H}_{11}) \cap \Lambda_+(\mathcal{B}_{\mathcal{S}}, \mathcal{B}_{\mathcal{H}}) = \emptyset$. Here, \mathcal{Z}_{11} , \mathcal{T}_{11} , $\mathcal{H}_{11} \in \mathbb{C}^{m \times m}$. Suppose $\Lambda_-(\mathcal{S}, \mathcal{H})$ contains p eigenvalues. If $\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \in \mathbb{C}^{4n \times m}$ are the first m columns of \mathcal{V} , $2p \leq m \leq 2n - 2p$, then there are subspaces \mathbb{L}_1 and \mathbb{L}_2 such that

$$\begin{aligned}\text{range } V_1 &= \text{Defl}_-(\mathcal{S}, \mathcal{H}) + \mathbb{L}_1, \quad \mathbb{L}_1 \subseteq \text{Defl}_0(\mathcal{S}, \mathcal{H}) + \text{Defl}_\infty(\mathcal{S}, \mathcal{H}), \\ \text{range } \overline{V_2} &= \text{Defl}_+(\mathcal{S}, \mathcal{H}) + \mathbb{L}_2, \quad \mathbb{L}_2 \subseteq \text{Defl}_0(\mathcal{S}, \mathcal{H}) + \text{Defl}_\infty(\mathcal{S}, \mathcal{H}).\end{aligned}$$

If $\Lambda(\mathcal{T}_{11} \mathcal{Z}_{11}, \mathcal{H}_{11}) = \Lambda_-(\mathcal{B}_{\mathcal{S}}, \mathcal{B}_{\mathcal{H}})$, and $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$, $\begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ are the first m columns of \mathcal{U} , \mathcal{W} , respectively, then there exist unitary matrices Q_U , Q_V , Q_W such that

$$\begin{aligned}U_1 &= [P_U^- \ 0] Q_U, \quad U_2 = [0 \ P_U^+] Q_U, \\ V_1 &= [P_V^- \ 0] Q_V, \quad V_2 = [0 \ P_V^+] Q_V, \\ W_1 &= [P_W^- \ 0] Q_W, \quad W_2 = [0 \ P_W^+] Q_W,\end{aligned}$$

and the columns of P_V^- and $\overline{P_V^+}$ form orthogonal bases of $\text{Defl}_-(\mathcal{S}, \mathcal{H})$ and $\text{Defl}_+(\mathcal{S}, \mathcal{H})$, respectively. Moreover, the matrices P_U^- , P_U^+ , P_W^- , and P_W^+ have orthonormal columns and the following relations are satisfied

$$\begin{aligned}\mathcal{Z} P_V^- &= P_U^- \tilde{\mathcal{Z}}_{11}, \quad \mathcal{J} \mathcal{Z}^H \mathcal{J}^T P_U^- = P_W^- \tilde{\mathcal{T}}_{11}, \quad \mathcal{H} P_V^- = P_W^- \tilde{\mathcal{H}}_{11}, \\ \mathcal{Z} P_V^+ &= \overline{P_U^+} \tilde{\mathcal{Z}}_{22}, \quad \mathcal{J} \mathcal{Z}^H \mathcal{J}^T P_U^+ = \overline{P_W^+} \tilde{\mathcal{T}}_{22}, \quad \mathcal{H} P_V^+ = -\overline{P_W^+} \tilde{\mathcal{H}}_{22}.\end{aligned}$$

Here, $\tilde{\mathcal{Z}}_{kk}$, $\tilde{\mathcal{T}}_{kk}$, and $\tilde{\mathcal{H}}_{kk}$, $k = 1, 2$, satisfy $\Lambda(\tilde{\mathcal{T}}_{11} \tilde{\mathcal{Z}}_{11}, \tilde{\mathcal{H}}_{11}) = \Lambda(\tilde{\mathcal{T}}_{22} \tilde{\mathcal{Z}}_{22}, \tilde{\mathcal{H}}_{22}) = \Lambda_-(\mathcal{S}, \mathcal{H})$.

PROOF. See [Benner et al. 2002]. \square

If the matrix \mathcal{S} is not given in factored form, we can use the following algorithm for the computation of the deflating subspaces [Benner et al. 2002]. The other algorithms are similar, however for brevity we refer to [Benner et al. 2002; Benner et al. 2007].

ALGORITHM 1. Computation of stable deflating subspaces of complex skew-Hamiltonian/Hamiltonian matrix pencils in unfactored form.

Input: A skew-Hamiltonian/Hamiltonian matrix pencil $\lambda \mathcal{S} - \mathcal{H} \in \mathbb{C}[\lambda]^{2n \times 2n}$.

Output: Structured Schur form of the extended skew-Hamiltonian/Hamiltonian matrix pencil $\lambda \mathcal{B}_{\mathcal{S}}^c - \mathcal{B}_{\mathcal{H}}^c$, eigenvalues of $\lambda \mathcal{S} - \mathcal{H}$, orthonormal basis P_V^- of the deflating subspace $\text{Defl}_-(\mathcal{S}, \mathcal{H})$, as in Theorem 3.2.

- 1: Set $\mathcal{N} = i\mathcal{H}$ and determine the matrices $\mathcal{B}_S^c, \mathcal{B}_N^c$ as in (13) and (12), respectively. Perform Algorithm 2 to compute the factorization

$$\begin{aligned}\widehat{\mathcal{B}}_S^c &= \mathcal{J}\mathcal{Q}^T \mathcal{J}^T \mathcal{B}_S^c \mathcal{Q} = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ 0 & \mathcal{S}_{11}^T \end{bmatrix}, \\ \widehat{\mathcal{B}}_N^c &= \mathcal{J}\mathcal{Q}^T \mathcal{J}^T \mathcal{B}_N^c \mathcal{Q} = \begin{bmatrix} \mathcal{N}_{11} & \mathcal{N}_{12} \\ 0 & \mathcal{N}_{11}^T \end{bmatrix},\end{aligned}\tag{16}$$

- where \mathcal{Q} is real orthogonal, \mathcal{S}_{11} is upper triangular and \mathcal{N}_{11} is upper quasi-triangular.
2: Apply the QZ algorithm [Golub and Van Loan 1996] to the 2×2 diagonal blocks of the matrix pencil $\lambda\mathcal{S}_{11} - \mathcal{N}_{11}$ to determine unitary matrices Q_1, Q_2 such that $Q_2^H \mathcal{S}_{11} Q_1, Q_2^H \mathcal{N}_{11} Q_1$ are both upper triangular. Define $\widehat{\mathcal{Q}} := \text{diag}(Q_1, Q_2)$ and set

$$\widetilde{\mathcal{B}}_S^c = \mathcal{J}\widehat{\mathcal{Q}}^H \mathcal{J}^T \widehat{\mathcal{B}}_S^c \widehat{\mathcal{Q}}, \quad \widetilde{\mathcal{B}}_N^c = \mathcal{J}\widehat{\mathcal{Q}}^H \mathcal{J}^T \widehat{\mathcal{B}}_N^c \widehat{\mathcal{Q}}.$$

- 3: Use Algorithm 3 to determine a unitary matrix $\widetilde{\mathcal{Q}}$ such that

$$\begin{aligned}\mathcal{J}\widetilde{\mathcal{Q}}^H \mathcal{J}^T \widetilde{\mathcal{B}}_S^c \widetilde{\mathcal{Q}} &= \begin{bmatrix} \widetilde{\mathcal{S}}_{11} & \widetilde{\mathcal{S}}_{12} \\ 0 & \widetilde{\mathcal{S}}_{11}^H \end{bmatrix}, \\ \mathcal{J}\widetilde{\mathcal{Q}}^H \mathcal{J}^T (-i\widetilde{\mathcal{B}}_N^c) \widetilde{\mathcal{Q}} &= \begin{bmatrix} \widetilde{\mathcal{H}}_{11} & \widetilde{\mathcal{H}}_{12} \\ 0 & -\widetilde{\mathcal{H}}_{11}^H \end{bmatrix},\end{aligned}$$

where $\widetilde{\mathcal{S}}_{11}, \widetilde{\mathcal{H}}_{11}$ are upper triangular such that $\Lambda_-(\widetilde{\mathcal{B}}_S^c, -i\widetilde{\mathcal{B}}_N^c)$ is contained in the spectrum of the $2p \times 2p$ leading principal subpencil of $\lambda\widetilde{\mathcal{S}}_{11} - \widetilde{\mathcal{H}}_{11}$.

- 4: Set $V = [I_{2n} \ 0] \mathcal{X}_c \mathcal{Q} \widetilde{\mathcal{Q}} \begin{bmatrix} I_{2p} \\ 0 \end{bmatrix}$ and compute P_V^- , an orthogonal basis of range V , using any numerically stable orthogonalization scheme.

Now we present the algorithm for the computation of the structured matrix factorization (16).

ALGORITHM 2. Computation of a structured matrix factorization for real skew-Hamiltonian/skew-Hamiltonian matrix pencils in unfactored form.

Input: A skew-Hamiltonian/skew-Hamiltonian matrix pencil $\lambda\mathcal{S} - \mathcal{N} \in \mathbb{R}[\lambda]^{2n \times 2n}$.

Output: A real orthogonal matrix \mathcal{Q} and the structured factorization (15).

- 1: Set $\mathcal{Q} = I_{2n}$. Reduce \mathcal{S} to skew-Hamiltonian triangular form, i. e., determine an orthogonal matrix \mathcal{Q}_1 such that

$$\mathcal{S} := \mathcal{J}\mathcal{Q}_1^T \mathcal{J}^T \mathcal{S}\mathcal{Q}_1 = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ 0 & \mathcal{S}_{11}^T \end{bmatrix}$$

with an upper triangular matrix \mathcal{S}_{11} . Update $\mathcal{N} := \mathcal{J}\mathcal{Q}_1^T \mathcal{J}^T \mathcal{N}\mathcal{Q}_1$, $\mathcal{Q} := \mathcal{Q}\mathcal{Q}_1$. This step is performed by applying a sequence of Householder reflections and Givens rotations in a specific order, see [Benner et al. 2002] for details.

- 2: Reduce \mathcal{N} to skew-Hamiltonian Hessenberg form. Determine an orthogonal matrix \mathcal{Q}_1 such that

$$\begin{aligned}\mathcal{S} &:= \mathcal{J}\mathcal{Q}_1^T \mathcal{J}^T \mathcal{S}\mathcal{Q}_1 = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ 0 & \mathcal{S}_{11}^T \end{bmatrix}, \\ \mathcal{N} &:= \mathcal{J}\mathcal{Q}_1^T \mathcal{J}^T \mathcal{N}\mathcal{Q}_1 = \begin{bmatrix} \mathcal{N}_{11} & \mathcal{N}_{12} \\ 0 & \mathcal{N}_{11}^T \end{bmatrix},\end{aligned}$$

where \mathcal{S}_{11} is upper triangular and \mathcal{N}_{11} is upper Hessenberg. Update $\mathcal{Q} := \mathcal{Q}\mathcal{Q}_1$. This step is performed by applying an appropriate sequence of Givens rotations to annihilate the ele-

ments in \mathcal{N} in a specific order without destroying the structure of \mathcal{S} , for details see [Benner et al. 2002].

- 3: Apply the QZ algorithm to the matrix pencil $\lambda S_{11} - N_{11}$ to determine orthogonal matrices Q_1 and Q_2 such that $Q_2^T S_{11} Q_1$ is upper triangular and $Q_2^T N_{11} Q_1$ is upper quasi-triangular. Set $Q_1 := \text{diag}(Q_1, Q_2)$ and update $S := \mathcal{J} Q_1^T \mathcal{J}^T S Q_1$, $\mathcal{N} := \mathcal{J} Q_1^T \mathcal{J}^T \mathcal{N} Q_1$, $Q := Q Q_1$.

Now, the eigenvalues of $\lambda S - \mathcal{N}$ are determined by the diagonal 1×1 and 2×2 blocks of S_{11} and N_{11} . Next, we consider an eigenvalue reordering routine in order to move the eigenvalues with negative real part of a skew-Hamiltonian/Hamiltonian pencil to the leading principal subpencil.

ALGORITHM 3. Eigenvalue reordering for complex skew-Hamiltonian/Hamiltonian matrix pencils in unfactored form.

Input: A regular skew-Hamiltonian/Hamiltonian matrix pencil $\lambda S - \mathcal{H} \in \mathbb{C}[\lambda]^{2n \times 2n}$ of the form

$$S = \begin{bmatrix} S & W \\ 0 & S^H \end{bmatrix}, \mathcal{H} = \begin{bmatrix} H & D \\ 0 & -H^H \end{bmatrix}, \text{ with upper triangular } S, H.$$

Output: A unitary matrix Q and the transformed matrices $\mathcal{J} Q^H \mathcal{J}^T S Q$, $\mathcal{J} Q^H \mathcal{J}^T \mathcal{H} Q$ which have still the same triangular form as S and \mathcal{H} , respectively, but the eigenvalues in $\Lambda_-(S, \mathcal{H})$ are reordered such that they occur in the leading principal subpencil of $\mathcal{J} Q^H \mathcal{J}^T (\lambda S - \mathcal{H}) Q$.

- 1: Set $Q = I_{2n}$. Reorder the eigenvalues in the subpencil $\lambda S - H$.
 - a) Determine unitary matrices Q_1, Q_2 such that $S := Q_2^H S Q_1$, $H := Q_2^H H Q_1$ are still upper triangular but the m_- eigenvalues with negative real part are reordered to the top of $\lambda S - H$. Set $Q_1 := \text{diag}(Q_1, Q_2)$ and update $Q := Q Q_1$.
 - b) Determine unitary matrices Q_1, Q_2 such that $S := Q_2^H S Q_1$, $H := Q_2^H H Q_1$ are still upper triangular but the m_+ eigenvalues with positive real part are reordered to the bottom of $\lambda S - H$. Set $Q_1 := \text{diag}(Q_1, Q_2)$ and update $Q := Q Q_1$.
- 2: Reorder the remaining $n - m_+ + 1$ eigenvalues with negative real parts which are now in the bottom right subpencil of $\lambda S - \mathcal{H}$. Determine a unitary matrix Q_1 such that the eigenvalues of the top left subpencil of $\lambda S - \mathcal{H}$ with positive real parts and those of the bottom right subpencil of $\lambda S - \mathcal{H}$ with negative real parts are interchanged. Update $Q := Q Q_1$.

3.2. The Real Case

We also briefly recall the theory for the real case which has some significant differences compared to the complex case. For a detailed description we refer to [Benner et al. 2007]. Let $\lambda S - \mathcal{H} \in \mathbb{R}[\lambda]^{2n \times 2n}$ be a skew-Hamiltonian/Hamiltonian matrix pencil with

\mathcal{J} -semidefinite skew-Hamiltonian part $S = \mathcal{J} Z^T \mathcal{J}^T Z$ where $Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$, $\mathcal{H} =$

$\begin{bmatrix} F & G \\ H & -F^T \end{bmatrix}$. We introduce the orthogonal matrices

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

with \mathcal{P} as in (11). Now we define the double-sized matrices

$$\begin{aligned}\mathcal{B}_Z &:= \begin{bmatrix} Z & 0 \\ 0 & Z \end{bmatrix}, \\ \mathcal{B}_T &:= \begin{bmatrix} \mathcal{J}Z^T\mathcal{J}^T & 0 \\ 0 & \mathcal{J}Z^T\mathcal{J}^T \end{bmatrix} = \begin{bmatrix} \mathcal{J} & 0 \\ 0 & \mathcal{J} \end{bmatrix} \mathcal{B}_Z^T \begin{bmatrix} \mathcal{J} & 0 \\ 0 & \mathcal{J} \end{bmatrix}^T, \\ \mathcal{B}_S &:= \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} = \mathcal{B}_T \mathcal{B}_Z, \\ \mathcal{B}_H &:= \begin{bmatrix} \mathcal{H} & 0 \\ 0 & -\mathcal{H} \end{bmatrix}.\end{aligned}$$

Furthermore, we define

$$\begin{aligned}\mathcal{B}_Z^r &:= \mathcal{X}_r^T \mathcal{B}_Z \mathcal{X}_r = \begin{bmatrix} Z_{11} & 0 & Z_{12} & 0 \\ 0 & Z_{11} & 0 & Z_{12} \\ Z_{21} & 0 & Z_{22} & 0 \\ 0 & Z_{21} & 0 & Z_{22} \end{bmatrix}, \\ \mathcal{B}_T^r &:= \mathcal{X}_r^T \mathcal{B}_T \mathcal{X}_r = \mathcal{J} (\mathcal{B}_Z^r)^T \mathcal{J}^T, \\ \mathcal{B}_S^r &:= \mathcal{X}_r^T \mathcal{B}_S \mathcal{X}_r = \mathcal{J} (\mathcal{B}_Z^r)^T \mathcal{J}^T \mathcal{B}_Z^r, \\ \mathcal{B}_H^r &:= \mathcal{X}_r^T \mathcal{B}_H \mathcal{X}_r = \begin{bmatrix} 0 & F & 0 & G \\ F & 0 & G & 0 \\ 0 & H & 0 & -F^T \\ H & 0 & -F^T & 0 \end{bmatrix}.\end{aligned}$$

It can be easily observed, that the $4n \times 4n$ matrix pencil $\lambda \mathcal{B}_S^r - \mathcal{B}_H^r$ is again real skew-Hamiltonian/Hamiltonian. For the computation of the eigenvalues of $\lambda S - \mathcal{H}$ we apply the following structured matrix factorization which is also often referred to as *generalized symplectic URV decomposition* [Benner et al. 2007].

THEOREM 3.3. *Let $\lambda S - \mathcal{H} \in \mathbb{R}[\lambda]^{2n \times 2n}$ be a skew-Hamiltonian/Hamiltonian matrix pencil with $S = \mathcal{J}Z^T\mathcal{J}^T Z$. Then there exist orthogonal matrices Q_1, Q_2 and orthogonal symplectic matrices U_1, U_2 such that*

$$\begin{aligned}Q_1^T (\mathcal{J}Z^T\mathcal{J}^T) U_1 &= \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \\ U_2^T Z Q_2 &= \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}, \\ Q_1^T \mathcal{H} Q_2 &= \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix},\end{aligned}\tag{17}$$

with the formal matrix product $T_{11}^{-1} H_{11} Z_{11}^{-1} Z_{22}^{-T} H_{22}^T T_{22}^{-T}$ in real periodic Schur form [Bojanczyk et al. 1992; Hench and Laub 1994], where $T_{11}, Z_{11}, H_{11}, T_{22}^T, Z_{22}^T$ are upper triangular and H_{22}^T is upper quasi-triangular.

PROOF. The proof is constructive, see [Benner et al. 2007]. \square

By using Theorem 3.3 (with the same notation) we get the following factorization of the embedded matrix pencil $\lambda \mathcal{B}_S^r - \mathcal{B}_H^r$ with a factored matrix \mathcal{B}_S^r . We can compute an

orthogonal matrix \tilde{Q} and an orthogonal symplectic matrix \tilde{U} such that

$$\begin{aligned} \tilde{U}^T \mathcal{B}_Z^r \tilde{Q} &= \left[\begin{array}{cc|cc} T_{22}^T & 0 & -T_{12}^T & 0 \\ 0 & Z_{11} & 0 & Z_{12} \\ \hline 0 & 0 & T_{11}^T & 0 \\ 0 & 0 & 0 & Z_{22} \end{array} \right] =: \begin{bmatrix} \tilde{Z}_{11} & \tilde{Z}_{12} \\ 0 & \tilde{Z}_{22} \end{bmatrix}, \\ \mathcal{J} \tilde{Q}^T \mathcal{J}^T \mathcal{B}_H^r \tilde{Q} &= \left[\begin{array}{cc|cc} 0 & H_{11} & 0 & H_{12} \\ -H_{22}^T & 0 & H_{12} & 0 \\ \hline 0 & 0 & 0 & H_{22} \\ 0 & 0 & -H_{11}^T & 0 \end{array} \right] =: \begin{bmatrix} \tilde{\mathcal{H}}_{11} & \tilde{\mathcal{H}}_{12} \\ 0 & -\tilde{\mathcal{H}}_{11}^T \end{bmatrix}, \end{aligned} \quad (18)$$

where $\tilde{Q} = \mathcal{P}^T \begin{bmatrix} \mathcal{J} \mathcal{Q}_1 \mathcal{J}^T & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix} \mathcal{P}$, $\tilde{U} = \mathcal{P}^T \begin{bmatrix} \mathcal{U}_1 & 0 \\ 0 & \mathcal{U}_2 \end{bmatrix} \mathcal{P}$. From the condensed form (18) we can immediately get the eigenvalues of $\lambda \mathcal{S} - \mathcal{H}$ as

$$\Lambda(\mathcal{S}, \mathcal{H}) = \Lambda \left(\tilde{Z}_{22}^T \tilde{Z}_{11}, \tilde{\mathcal{H}}_{11} \right) = \pm i \sqrt{\Lambda \left(T_{11}^{-1} H_{11} Z_{11}^{-1} Z_{22}^{-T} H_{22}^T T_{22}^{-T} \right)}. \quad (19)$$

Note that all matrices of the product are upper triangular, except H_{22}^T which is upper quasi-triangular. Hence, the eigenvalue information can be extracted directly from the diagonal 1×1 or 2×2 blocks of the main diagonals. Note that the finite, simple, purely imaginary eigenvalues of the initial matrix pencil correspond to the positive eigenvalues of the generalized matrix product. Hence, these eigenvalues can be computed without any error in their real parts. This leads to a high robustness in algorithms which require these eigenvalues, e. g., in the \mathcal{H}_∞ -norm computation [Benner et al. 2012a]. However, if two purely imaginary eigenvalues are very close they might still be slightly perturbed from imaginary axis. This essentially depends on the Kronecker structure of a close-by skew-Hamiltonian/Hamiltonian matrix pencil with double purely imaginary eigenvalues. This problem is similar to the Hamiltonian matrix case, see [Mehrman and Xu 2008].

To compute the deflating subspaces we are interested in, it is necessary to compute the structured Schur form of the embedded matrix pencils $\lambda \mathcal{B}_S^r - \mathcal{B}_H^r$. This can be done by computing a finite number of similarity transformations to the subpencil $\lambda \tilde{Z}_{22}^T \tilde{Z}_{11} - \tilde{\mathcal{H}}_{11}$ to put $\tilde{\mathcal{H}}_{11}$ into upper quasi-triangular form. That is, we compute orthogonal matrices $\mathcal{Q}_3, \mathcal{Q}_4, \mathcal{U}_3$ such that

$$\mathcal{H}_{11} = \mathcal{Q}_3^T \tilde{\mathcal{H}}_{11} \mathcal{Q}_4, \quad \mathcal{Z}_{11} = \mathcal{U}_3^T \tilde{Z}_{11} \mathcal{Q}_4, \quad \mathcal{Z}_{22} = \mathcal{U}_3^T \tilde{Z}_{22} \mathcal{Q}_3,$$

where $\mathcal{Z}_{11}, \mathcal{Z}_{22}^T$ are upper triangular and \mathcal{H}_{11} is upper quasi-triangular. By setting $\mathcal{Q} = \tilde{Q} \begin{bmatrix} \mathcal{Q}_4 & 0 \\ 0 & \mathcal{Q}_3 \end{bmatrix}$, $\mathcal{U} = \tilde{U} \begin{bmatrix} \mathcal{U}_3 & 0 \\ 0 & \mathcal{U}_3 \end{bmatrix}$, $\mathcal{Z}_{12} = \mathcal{U}_3^T \tilde{Z}_{12} \mathcal{Q}_3$, and $\mathcal{H}_{12} = \mathcal{Q}_3^T \tilde{\mathcal{H}}_{12} \mathcal{Q}_3$ we obtain the

structured Schur form of $\lambda \mathcal{B}_S^r - \mathcal{B}_H^r$ as $\lambda \tilde{\mathcal{B}}_S^r - \tilde{\mathcal{B}}_H^r$ with $\tilde{\mathcal{B}}_S^r = \mathcal{J} \left(\tilde{\mathcal{B}}_Z^r \right)^T \mathcal{J}^T \tilde{\mathcal{B}}_Z^r$ and

$$\begin{aligned} \tilde{\mathcal{B}}_Z^r &:= \mathcal{U}^T \mathcal{B}_Z^r \mathcal{Q} = \begin{bmatrix} \mathcal{Z}_{11} & \mathcal{Z}_{12} \\ 0 & \mathcal{Z}_{22} \end{bmatrix}, \\ \tilde{\mathcal{B}}_H^r &:= \mathcal{J} \mathcal{Q}^T \mathcal{J}^T \mathcal{B}_H^r \mathcal{Q} = \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ 0 & -\mathcal{H}_{11}^T \end{bmatrix}. \end{aligned}$$

Now we can reorder the eigenvalues of $\lambda \tilde{\mathcal{B}}_S^r - \tilde{\mathcal{B}}_H^r$ to the top in order to compute the desired deflating subspaces which is similar to the complex case. Then, for the deflating subspaces we find a similar result as Theorem 3.2 which we do not state here for brevity.

If the matrix S is not given in factored form, we need the following slightly modified version of Theorem 3.3 from [Benner et al. 2007].

THEOREM 3.4. *Let $\lambda S - \mathcal{H} \in \mathbb{R}[\lambda]^{2n \times 2n}$ be a skew-Hamiltonian/Hamiltonian matrix pencil. Then there exist orthogonal matrices Q_1, Q_2 such that*

$$\begin{aligned} Q_1^T S J Q_1 J^T &= \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix} && \text{(skew-Hamiltonian),} \\ J Q_2^T J^T S Q_2 &= \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{11}^T \end{bmatrix} =: \mathcal{T} && \text{(skew-Hamiltonian),} \\ Q_1^T \mathcal{H} Q_2 &= \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \end{aligned} \quad (20)$$

with the formal matrix product $S_{11}^{-1} H_{11} T_{11}^{-1} H_{22}^T$ in real periodic Schur form, where S_{11}, T_{11}, H_{11} are upper triangular and H_{22}^T is upper quasi-triangular.

PROOF. The proof is done by construction, see [Benner et al. 2007]. \square

With the notation of Theorem 3.4 we can compute an orthogonal matrix \tilde{Q} such that

$$\begin{aligned} J \tilde{Q}^T J^T \mathcal{B}_S^r \tilde{Q} &= \left[\begin{array}{cc|cc} S_{11} & 0 & S_{12} & 0 \\ 0 & T_{11} & 0 & T_{12} \\ \hline 0 & 0 & S_{11}^T & 0 \\ 0 & 0 & 0 & T_{11}^T \end{array} \right] =: \begin{bmatrix} \tilde{S}_{11} & \tilde{S}_{12} \\ 0 & \tilde{S}_{11}^T \end{bmatrix}, \\ J \tilde{Q}^T J^T \mathcal{B}_H^r \tilde{Q} &= \left[\begin{array}{cc|cc} 0 & H_{11} & 0 & H_{12} \\ -H_{22}^T & 0 & H_{12} & 0 \\ \hline 0 & 0 & 0 & H_{22} \\ 0 & 0 & -H_{11}^T & 0 \end{array} \right] =: \begin{bmatrix} \tilde{\mathcal{H}}_{11} & \tilde{\mathcal{H}}_{12} \\ 0 & -\tilde{\mathcal{H}}_{11}^T \end{bmatrix}, \end{aligned} \quad (21)$$

with $\tilde{Q} = \mathcal{P}^T \begin{bmatrix} J Q_1 J^T & 0 \\ 0 & Q_2 \end{bmatrix} \mathcal{P}$. The spectrum of $\lambda S - \mathcal{H}$ is given by

$$\Lambda(S, \mathcal{H}) = \pm i \sqrt{\Lambda(S_{11}^{-1} H_{11} T_{11}^{-1} H_{22}^T)}$$

which can be determined by evaluating the entries on the 1×1 and 2×2 diagonal blocks of the matrices only. To put the matrix pencil formed of the matrices in (21) into structured Schur form we have to triangularize $\lambda \tilde{S}_{11} - \tilde{\mathcal{H}}_{11}$, i. e., we determine orthogonal matrices Q_3 and Q_4 such that

$$S_{11} = Q_4^T \tilde{S}_{11} Q_3, \quad \mathcal{H}_{11} = Q_4^T \tilde{\mathcal{H}}_{11} Q_3$$

are upper triangular and upper quasi-triangular, respectively. By setting $Q = \tilde{Q} \begin{bmatrix} Q_3 & 0 \\ 0 & Q_4 \end{bmatrix}$, $S_{12} = Q_4^T \tilde{S}_{12} Q_4$, and $\mathcal{H}_{12} = Q_4^T \tilde{\mathcal{H}}_{12} Q_4$, we obtain the structured Schur form as

$$\begin{aligned} \tilde{B}_S^r &:= J Q^T J^T \mathcal{B}_S Q = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix}, \\ \tilde{B}_H^r &:= J Q^T J^T \mathcal{B}_H Q = \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ 0 & -\mathcal{H}_{11}^T \end{bmatrix}. \end{aligned}$$

By properly reordering the eigenvalues we can compute the desired deflating subspaces as explained above. A complete algorithm description can be found in [Benner et al. 2007]. We omit the details for brevity.

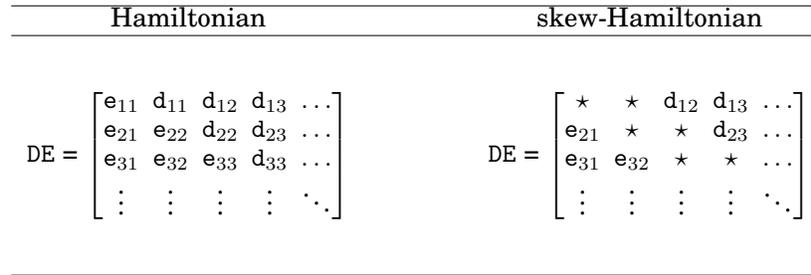


Fig. 1. Storage layout for the (skew-)symmetric submatrices D and E .

4. IMPLEMENTATION DETAILS

In this section we focus on implementation details of the algorithms outlined above.

4.1. General Remarks

Our subroutines are written by employing the rigorous implementation and documentation standards of the Subroutine Library in Control Theory (SLICOT¹), see [Eindhoven University of Technology, The Netherlands 1996b; Eindhoven University of Technology, The Netherlands 1996a]. The parameters of each SLICOT routine can be classified as follows:

- mode parameters,
- input/output parameters,
- tolerances,
- workspace,
- error/warning indicator.

Mode parameters specify the provided functionality with regard to output results and the methods used for the computations. Input/output parameters are usually the dimension of the involved matrices and the matrices themselves with their leading dimensions. The error indicator `INFO` tells the user if an illegal value was used as input (`INFO` returns negative values) or whether an error occurred during program execution (`INFO` returns positive values). A warning indicator `IWARN` informs the user about possibly unreliable or inaccurate results or additional information about the results.

4.2. Storage Layout

Since Hamiltonian and skew-Hamiltonian matrices have certain block structures we use a packed storage layout proposed in [Benner et al. 2000] to avoid saving redundant data. More specifically, if a real $2n \times 2n$ Hamiltonian matrix $\mathcal{H} = \begin{bmatrix} A & D \\ E & -A^T \end{bmatrix}$ is given,

we save the submatrix A in a conventional $n \times n$ array A , the symmetric submatrices D and E are stored in an $n \times (n+1)$ array DE such that the upper triangular part of D is stored in $DE(1:n, 2:n+1)$ and the lower triangular part of E is stored in $DE(1:n, 1:n)$. The skew-symmetric parts of a skew-Hamiltonian matrix are similarly stored with the notable difference that the parts containing the diagonal and the first superdiagonal of the array DE are not referenced. See also Figure 1 for an illustration. Similarly, as every

orthogonal or unitary symplectic $2n \times 2n$ matrix has the block structure $\mathcal{U} = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix}$,

¹<http://slicot.org/>

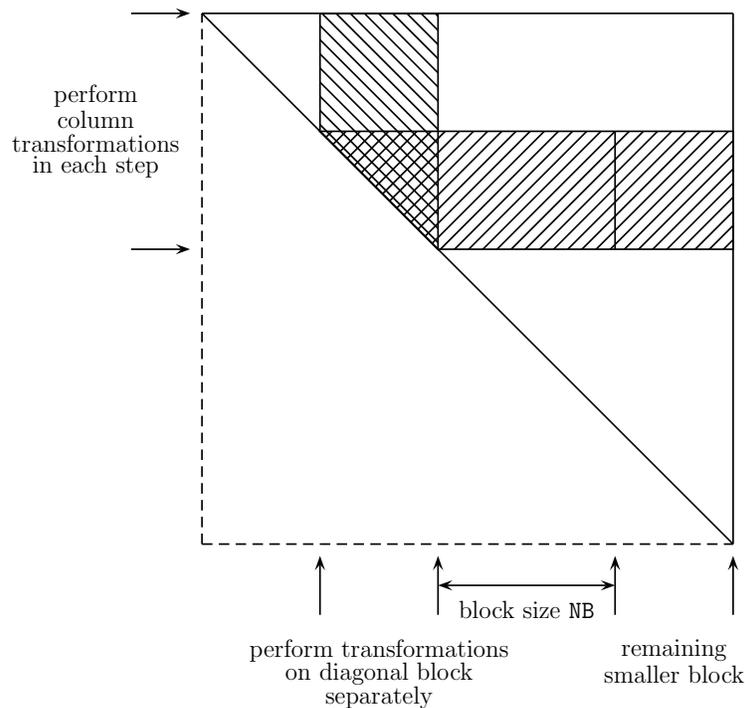


Fig. 2. Panel blocking technique for an upper triangular matrix.

we only store the matrix U_1 in an $n \times n$ array U1 and the matrix U_2 in an $n \times n$ array U2.

A similar storage format is also applied to complex skew-Hamiltonian or Hamiltonian matrices. In contrast to the real case, for skew-Hamiltonian matrices, also the parts containing the diagonal and the first superdiagonal of the array DE are referenced.

4.3. Panel Blocking for Larger Problems

The problems considered here are usually based on applying sequences of Givens rotations. When updating the involved matrices we successively have to transform the corresponding rows and columns in each step. However, for larger matrices this kind of transformations can become very inefficient due to Fortran's memory and cache management. Fortran uses a column-major memory layout, i. e., elements of a column are internally stored one after the other. On the other hand, the distance in the internal memory between two successive elements in a row is exactly the leading dimension of that array. Therefore, rows can only be put into the cache memory by caching also the remaining parts of the columns that contain elements of the rows under consideration. For larger arrays, this easily leads to chunk sizes that do not fit into cache memory anymore. Therefore, we store the information of a certain number of Givens rotations and apply the row transformations only on panels of block size NB in order to better exploit the data locality.

An example for such a panel update is depicted in Figure 2. It illustrates the blocking technique for an update of a triangular matrix. Updates on columns are always directly applied after the generation of the Givens rotation, whereas rows are split into certain

Table I. Overview of the implemented Fortran routines

Fortran routine	Algorithm description
ZGHFDF	Computing the eigenvalues and stable deflating subspaces of a complex skew-Hamiltonian/Hamiltonian pencil (factored version)
DGHFST	Computing the eigenvalues of a real skew-Hamiltonian/skew-Hamiltonian pencil (factored version)
ZGHFXC	Moving eigenvalues with negative real parts of a complex skew-Hamiltonian/Hamiltonian pencil in structured Schur form to the leading subpencil (factored version)
ZGHUFD	Computing the eigenvalues and stable deflating subspaces of a complex skew-Hamiltonian/Hamiltonian pencil (unfactored version)
DGHUST	Computing the eigenvalues of a real skew-Hamiltonian/skew-Hamiltonian pencil (unfactored version)
ZGHUXC	Moving eigenvalues with negative real parts of a complex skew-Hamiltonian/Hamiltonian pencil in structured Schur form to the leading subpencil (unfactored version)
DGHFDF	Computing the eigenvalues and stable deflating subspaces of a real skew-Hamiltonian/Hamiltonian pencil (factored version)
DGHURV	Computing the eigenvalues of a real skew-Hamiltonian/Hamiltonian pencil via generalized symplectic URV decomposition (factored version)
DGHFYR	Reducing a special real block (anti-)diagonal skew-Hamiltonian/Hamiltonian pencil in factored form to generalized Schur form (factored version)
DGHFXC	Moving eigenvalues with negative real parts of a real skew-Hamiltonian/Hamiltonian pencil in structured Schur form to the leading subpencil (factored version)
DGHUFD	Computing the eigenvalues and stable deflating subspaces of a real skew-Hamiltonian/Hamiltonian pencil (unfactored version)
DGHUTR	Computing the eigenvalues of a real skew-Hamiltonian/Hamiltonian pencil (unfactored version)
DGHUYR	Reducing a special real block (anti-)diagonal skew-Hamiltonian/Hamiltonian pencil in factored form to generalized Schur form (unfactored version)
DGHUXC	Moving eigenvalues with negative real parts of a real skew-Hamiltonian/Hamiltonian pencil in structured Schur form to the leading subpencil (unfactored version)

subpanels of maximum block size NB . Note that updates on the diagonal block are done separately as then the remaining parts of the rows have equal size and can therefore be easily decomposed into subblocks. We note that each part of the code has to be blocked in a different way. This is due to different matrix structures or dependencies of the updates and generation of the next Givens rotations. Therefore, sometimes parts of rows have to be updated in each step. We have developed blocked versions for some of our codes and we will compare them with the unblocked versions in Section 5.

4.4. Overview of the Implemented Fortran Subroutines

Table I contains a brief overview of the main algorithms that have been implemented along with their corresponding Fortran subroutine names. Full details of the individual interfaces and the structure of the call graphs of the driver routines may be found in the user manual that accompanies the software [Benner et al. 2015].

5. NUMERICAL RESULTS

In this section we present some numerical results of our implementations. The tests have been performed on a 2.6.32-31-generic Ubuntu machine with Intel®Core™2 Quad CPU Q9550 with 2.83GHz in each of the four cores and 8GB RAM. All codes have been compiled using `gfortran` with the optimization level `-O2` (safe optimizations). For better handling the codes, MEX gateway functions have been written for calling the routines from MATLAB 7.14.0.739 (R2012a). For this purpose we also use

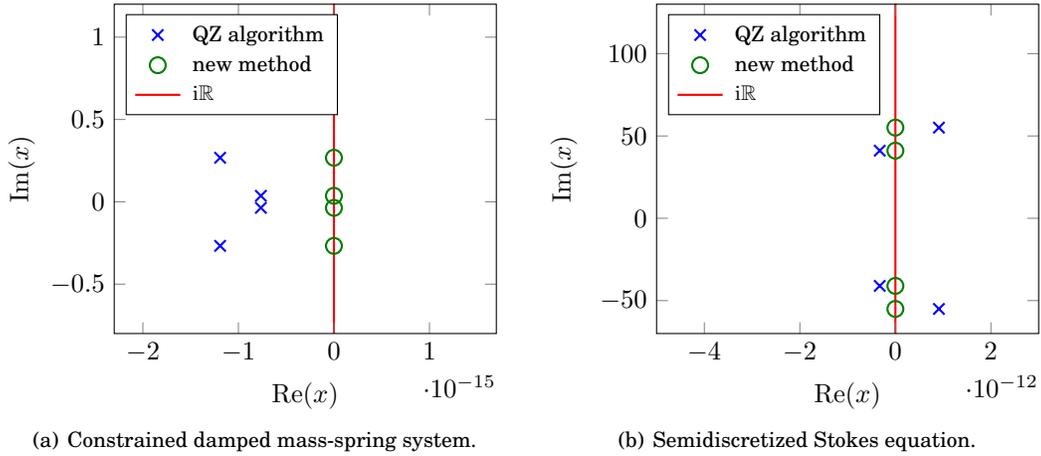


Fig. 3. Computed purely imaginary eigenvalues of two skew-Hamiltonian/Hamiltonian example matrix pencils.

MATLAB’s built-in MKL BLAS and LAPACK libraries which are optimized for using multi-core architectures.

5.1. Structure-Preserving Computations

The most important feature of our algorithms is structure-preservation. This means that only reductions that keep the skew-Hamiltonian/Hamiltonian structure are performed. Therefore, only skew-Hamiltonian/Hamiltonian perturbations of the eigenvalues are possible. In particular, simple, finite, purely imaginary eigenvalues stay on the imaginary axis as long as their pairwise distance is sufficiently large. In such a situation the perturbation off the imaginary axis would not lead to the formation of a quadruple of eigenvalues which is necessary by the Hamiltonian spectral symmetry. In Figure 3, some of the computed eigenvalues by the QZ algorithm [Golub and Van Loan 1996] and our new method are depicted. For the tests we used extended skew-Hamiltonian/Hamiltonian pencils for the \mathcal{L}_∞ -norm computation of descriptor systems [Voigt 2010]. The pencils are related to models for constrained mass-spring systems or semidiscretized Stokes equations (see [Mehrmann and Stykel 2005] and references therein). The figure shows that the eigenvalues computed by the standard QZ algorithm are perturbed off the imaginary axis whereas the new method preserves the eigenvalue symmetry. In particular, the new approach allows a reliable determination of the stable eigenvalues. If we furthermore want to compute the stable deflating subspaces we have to know the stable eigenvalues in advance. For the first example (Figure 3(a)), the QZ algorithm computes more stable than unstable eigenvalues which is impossible by theory. Therefore, also the stable deflating subspace computed by this method will have a too large dimension. This undesired behavior is avoided by our method.

A second example that illustrates the superiority of our method arises in the context of gyroscopic systems of the form

$$M\ddot{x}(t) + G\dot{x}(t) + Kx(t) = 0 \tag{22}$$

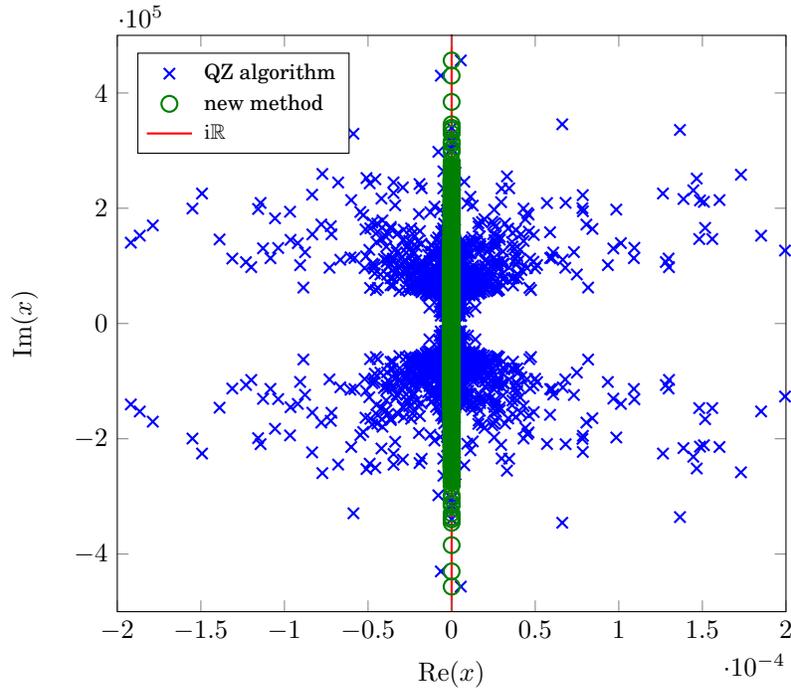


Fig. 4. Computed eigenvalues from a skew-Hamiltonian/Hamiltonian matrix pencil resulting from a linearized gyroscopic system.

with $M = M^T > 0$, $G = -G^T$, and $K = K^T$. To analyze stability of such a system we have to consider the quadratic eigenvalue problem

$$(M\lambda^2 + G\lambda + K)y = 0. \quad (23)$$

It can be shown that a necessary condition for (22) to be stable is that all eigenvalues of (23) are purely imaginary [Lancaster 2013]. A linearization of (23) to second companion form [Tisseur and Meerbergen 2001] leads to an eigenvalue problem for the skew-Hamiltonian/Hamiltonian matrix pencil

$$\lambda \begin{bmatrix} M & G \\ 0 & M \end{bmatrix} - \begin{bmatrix} 0 & -K \\ M & 0 \end{bmatrix}.$$

The example we use here is the “Rolling Tires” system from [Benner et al. 2008a] with a system dimension of $n = 2697$. The computed eigenvalues for both the QZ algorithm and our method are depicted in Figure 4. For our algorithm, all eigenvalues are determined to be on the imaginary axis which means that the necessary stability criterion for the gyroscopic system is fulfilled. However, for the QZ algorithm this is not the case. Since the QZ algorithm does not respect the skew-Hamiltonian/Hamiltonian structure, all eigenvalues are perturbed off the imaginary axis. Some of them are also very far away from the imaginary axis (the maximum absolute value of the real parts is $1.4836e-03$). So in contrast to the structure-preserving approach, one could think that the necessary stability criterion is not fulfilled.

5.2. Solving Algebraic Riccati Equations

In this subsection we use our algorithms for computing the solution of algebraic Riccati equations and compare with the results of the MATLAB function `care`. We consider continuous-time algebraic Riccati equations (AREs) of the form

$$0 = Q + A^T X + X A - X G X, \tag{24}$$

where $A, G, Q, X \in \mathbb{R}^{n \times n}$. In many problems the matrices $Q = Q^T$ and $G = G^T$ are given in factored form $Q = C^T \tilde{Q} C$, $G = B R^{-1} B^T$ with $C \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{n \times m}$, $\tilde{Q} = \tilde{Q}^T \in \mathbb{R}^{p \times p}$, and $R = R^T \in \mathbb{R}^{m \times m}$. If $\tilde{Q} \geq 0$, $R > 0$, (A, B) is stabilizable, and (A, C) is detectable, then (24) has a unique, symmetric positive semidefinite, stabilizing solution X_* .

A popular method for determining X_* is to compute the stable invariant subspace spanned by $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ of the Hamiltonian matrix

$$\mathcal{H} = \begin{bmatrix} A & -G \\ -Q & -A^T \end{bmatrix} = \begin{bmatrix} A & -B R^{-1} B^T \\ -C^T \tilde{Q} C & -A^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n}.$$

If U_1 is invertible, then $X_* = U_2 U_1^{-1}$ (see [Abels and Benner 1999] and references therein). Here, we use a slightly more general approach, namely we compute the right stable deflating subspace of the skew-Hamiltonian/Hamiltonian matrix pencil

$$\lambda \mathcal{S} - \mathcal{H} = \lambda \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix} - \begin{bmatrix} A & -G \\ -Q & -A^T \end{bmatrix} \in \mathbb{R}[\lambda]^{2n \times 2n}$$

which is equal to the stable invariant subspace of \mathcal{H} .

For benchmarking we use the examples collected in [Abels and Benner 1999] which are often difficult to solve due to ill-conditioning of the ARE or the existence of eigenvalues of \mathcal{H} close to the imaginary axis. In Table II the relative residuals for each individual problem are presented. We compare the skew-Hamiltonian/Hamiltonian pencil approach with orthogonalization via pivoted QR factorization (QRP), singular value decomposition (SVD) and the MATLAB solver (`care`). To ensure comparability we use the same scaling technique for the ARE for both our codes and `care` (by calling `arescale` in MATLAB). Except for one example (which also `care` could not solve), our codes could compute X_* in all tests. The relative residuals are most often of the same order of magnitude. For five problems, our codes obtained better results for at least one orthogonalization option (for tests # 5, 13, 14, 31, 32 the relative residual is at least one order of magnitude smaller than the one of `care`). On the other hand, `care` performed better for 6 examples (# 8, 12, 15, 17, 24, 34). In particular, for example 17, the difference is about 10 orders of magnitude, for the other examples (except for # 8) the difference is about one order of magnitude. Note that example # 17 is difficult to solve in the sense that the pencil $\lambda \mathcal{S} - \mathcal{H}$ has eigenvalues on the imaginary axis, i. e., even though the ARE is solvable, it is on the boundary to unsolvability. To construct the stabilizing solution of the ARE via deflating subspaces one has to include a certain subspace of $\text{Defl}_0(\mathcal{S}, \mathcal{H})$ which is however neither correctly extracted by a pivoted QR decomposition nor an SVD. On the other hand, by determining the desired deflating subspace by a non-pivoted QR decomposition, we obtain a relative residual of 1.2232e-15. This indicates that for this orthogonalization scheme, the correct subspace was extracted. This behavior does however not turn over to many of the other examples.

A similar overall picture is achieved when having a look at the relative errors compared to the analytic solution if it is known. We omit this analysis since it does not give

Table II. Relative residuals of the solution of algebraic Riccati equations: Comparison of the new algorithm with orthogonalization via pivoted QR factorization (QRP), singular value decomposition (SVD), and the MATLAB solver care.

test #	ex. #	n	m	p	parameters	QRP	SVD	care
1	1.1	2	1	2		3.0044e-15	2.5749e-15	3.0062e-15
2	1.2	2	1	2		7.3931e-16	3.4594e-15	6.5338e-16
3	1.3	4	2	4		2.4751e-15	2.4167e-15	3.9430e-15
4	1.4	8	2	8		2.5514e-15	1.5739e-15	1.1924e-15
5	1.5	9	3	9		8.7957e-15	2.3342e-13	9.3663e-14
6	1.6	30	3	5		8.8269e-12	4.4861e-12	1.4481e-12
7	2.1	2	1	1	$\varepsilon = 1$	9.0528e-16	9.5989e-16	7.5037e-16
8	2.1	2	1	1	$\varepsilon = 10^{-6}$	1.7361e-10	3.2218e-10	0
9	2.2	2	2	1	$\varepsilon = 1$	5.5948e-16	3.7261e-16	1.1068e-15
10	2.2	2	2	1	$\varepsilon = 10^{-8}$	1.5895e-09	7.7370e-10	2.3218e-09
11	2.3	2	1	2	$\varepsilon = 1$	7.3951e-16	1.4259e-15	1.1378e-15
12	2.3	2	1	2	$\varepsilon = 10^6$	2.0448e-10	3.7537e-11	6.5854e-13
13	2.3	2	1	2	$\varepsilon = 10^{-6}$	1.6745e-21	4.6784e-18	6.8373e-20
14	2.4	2	2	2	$\varepsilon = 1$	0	1.2684e-14	1.1531e-15
15	2.4	2	2	2	$\varepsilon = 10^{-7}$	2.9441e-15	1.1608e-14	1.6454e-16
16	2.5	2	1	2	$\varepsilon = 1$	1.4121e-15	1.3570e-15	1.9343e-15
17	2.5	2	1	2	$\varepsilon = 0$	3.6694e-05	1.2326e-06	1.2232e-15
18	2.6	3	3	3	$\varepsilon = 1$	5.8902e-15	3.8570e-15	5.7262e-15
19	2.6	3	3	3	$\varepsilon = 10^6$	4.7596e+02	4.4341e+02	6.3670e+02
20	2.7	4	1	2	$\varepsilon = 1$	2.4085e-16	1.6736e-16	1.4054e-15
21	2.7	4	1	2	$\varepsilon = 10^{-6}$	1.9697e-08	3.2989e-11	1.3429e-11
22	2.8	4	1	1	$\varepsilon = 1$	7.4186e-16	4.0395e-15	5.6954e-15
23	2.8	4	1	1	$\varepsilon = 10^{-6}$	3.8032e-15	1.0134e-15	4.6214e-15
24	2.9	55	2	10	#1	1.0737e-11	5.7755e-12	2.4757e-13
25	3.1	9	5	4		3.8305e-15	2.6481e-15	3.2909e-15
26	3.1	39	20	19		3.4076e-15	4.6692e-15	8.0452e-15
27	3.2	8	8	8		2.9567e-15	2.2579e-15	3.7270e-15
28	3.2	64	64	64		9.8352e-15	8.8604e-15	1.2277e-14
29	4.1	21	1	1	$q = r = 1.0$	1.0359e-06	4.4380e-07	6.8088e-07
30	4.1	21	1	1	$q = r = 100.0$	2.1010e-05	2.1627e-05	6.3995e-05
31	4.2	20	1	1	$a = 0.05, b = c = 0.1,$ $[\beta_1, \beta_2] = [0.1, 0.5],$ $[\gamma_1, \gamma_2] = [0.1, 0.5]$	1.4274e-17	1.1291e-13	1.8773e-13
32	4.2	100	1	1	$a = 0.01, b = c = 1.0,$ $[\beta_1, \beta_2] = [0.2, 0.3],$ $[\gamma_1, \gamma_2] = [0.2, 0.3]$	1.3742e-15	1.2528e-12	3.5524e-12
33	4.3	60	2	60	$\ell = 30, \mu = 4.0,$ $\delta = 4.0, \kappa = 1.0$	7.8279e-15	7.9629e-15	2.6545e-14
34	4.4	421	211	211		5.1450e-03	1.0845e-05	7.9411e-07

significantly more information. In conclusion, both approaches give results of similar quality, even though our codes are not specifically designed for solving AREs.

5.3. Comparison of Runtimes

In this subsection we discuss the runtimes of our codes and compare them with standard implementations included in LAPACK. The results are listed in Tables III and IV, respectively. In Figure 5, the runtime ratios of the new codes compared to MATLAB's LAPACK implementations are depicted to summarize these results. In general, pure eigenvalue computations are much faster than the computation of both eigenvalues and deflating subspaces. The reason is that for the subspace computation the transformation matrices for the embedded pencils (*of double size*) are accumulated in the final step. However, during our tests we often observe that LAPACK routines, even though they are faster, are not able to solve some (random) examples. Especially, for larger problems INFO = N+2 is returned which indicates that the desired reordering of

Table III. Comparison of runtimes for the real case (measured in secs.).

Problem size	eigenvalues only		eigenvalues and deflating subspaces	
	DGGEV	DGHUTR	DGGES	DGHUDF
2	3.2480e-06	2.2000e-06	7.7860e-06	2.1138e-05
4	1.1510e-05	1.3725e-05	4.3221e-05	1.1633e-04
8	3.7886e-05	7.3677e-05	1.4393e-04	3.6971e-04
16	1.2640e-04	1.9300e-04	3.6360e-04	1.3859e-03
32	7.1310e-04	7.3620e-04	1.7058e-03	5.0400e-03
64	3.0412e-03	3.0708e-03	8.3355e-03	2.5425e-02
128	1.8980e-02	1.6620e-02	4.3790e-02	1.1256e-01
256	1.4190e-01	1.0272e-01	2.8654e-01	5.8121e-01
512	1.4790e+00	8.9793e-01	2.5960e+00	3.9449e+00
1024	2.2127e+01	1.2964e+01	4.8888e+01	4.5998e+01
2048	4.2508e+02	2.6144e+02	5.6186e+02	6.3338e+02
4096	2.9650e+03	2.8367e+03	4.2058e+03	5.5788e+03

Table IV. Comparison of runtimes for the complex case (measured in secs.).

Problem size	eigenvalues only		eigenvalues and deflating subspaces	
	ZGGEV	DGHUST	ZGGES	ZGHUDF
2	7.7400e-06	4.7300e-06	2.7047e-05	4.1847e-05
4	2.3252e-05	2.2831e-05	5.2271e-05	9.3827e-05
8	8.2346e-05	7.5673e-05	1.2291e-04	2.2438e-04
16	3.1020e-04	3.1190e-04	4.6900e-04	7.7210e-04
32	1.4953e-03	1.4844e-03	2.5219e-03	3.4171e-03
64	8.7930e-03	9.0812e-03	1.4392e-02	1.9041e-02
128	5.8440e-02	5.6700e-02	9.2550e-02	1.1988e-01
256	4.5301e-01	4.5600e-01	6.2856e-01	9.6518e-01
512	3.4875e+00	7.6826e+00	4.6978e+00	1.4286e+01
1024	3.8185e+01	1.4554e+02	5.6904e+01	2.6081e+02
2048	4.9624e+02	1.2935e+03	8.2872e+02	2.1489e+03
4096	4.8410e+03	1.0849e+04	7.7507e+03	1.7189e+04

the eigenvalues could not be successfully performed. On the other hand, our solvers could always return meaningful results. Note that LAPACK routines can much better exploit blocked codes of Level 3 BLAS which is not the case for our codes since they are algorithmically based on Givens rotations. Even though the panel blocking technique we present here gives some improvements for larger examples, there is still the question whether one can find better ways of blocking our codes.

There are also significant differences in the behavior of the real and complex codes. The real codes have relatively constant runtime ratios for small and medium-size problems up to orders of about 128. Then, they increase up to order 2048 and then decrease again. However, for the complex codes, the runtime ratios are constant for problems up to order 256 and get significantly slower for larger problems. The main reason for this qualitatively different behavior is that the complex codes need significantly more memory and also more transfers between the different levels of the memory hierarchy. Fortunately, for larger problems, we have developed blocked codes which are able to avoid this slowdown, see also Subsection 5.5.

5.4. Factored Versus Unfactored Matrix Pencils

In this subsection we compare the results of the previous subsection with the factored versions of the algorithms with respect to accuracy, memory requirements, and speed.

5.4.1. Accuracy. We begin with an analysis of the obtained accuracy. We performed tests on random skew-Hamiltonian/Hamiltonian pencils of order 40. For the factored

Table V. Comparison of the errors of the eigenvalues.

	real case		complex case	
	unfactored	factored	unfactored	factored
$10^{-17} < \kappa_{\max} < 10^{-16}$	0	0	29	44
$10^{-18} < \kappa_{\max} < 10^{-17}$	825	805	932	926
$10^{-19} < \kappa_{\max} < 10^{-18}$	155	162	39	30
$10^{-20} < \kappa_{\max} < 10^{-19}$	6	17	0	0
$\kappa_{\max} < 10^{-20}$	14	16	0	0

algorithms we choose $Z = \begin{bmatrix} A & 0 \\ 0 & I_{20} \end{bmatrix}$ with a random matrix A . Then we can easily form

$$S = \mathcal{J} Z^H \mathcal{J}^T Z = \begin{bmatrix} A & 0 \\ 0 & A^H \end{bmatrix}$$

without any rounding error. This allows a fair comparison between the codes for factored and unfactored problems. First we analyze the accuracy of the computed eigenvalues. Therefore, we perform 1000 random tests and compute the maximum of the reciprocal condition numbers κ_{\max} of the matrices $\lambda_j S - \mathcal{H}$, $j = 1, \dots, 20$ for each problem. The reciprocal condition numbers define the closeness to singularity and should be zero in theory. Thus, small values κ_{\max} indicate a high accuracy. We divide the computed results into different classes and list the number of elements in each class in Table V. Furthermore, we observe that in the real case, the unfactored codes were more accurate for 500 examples. For the complex case this was the case for 516 examples. We can conclude that the computed eigenvalues are similarly accurate for both types of codes.

We also have a look at the accuracy of the deflating subspaces. Again, we perform 1000 random tests. With the computed stable deflating subspace $\text{im } Q$ we measure the error by determining the angle α between the subspaces $\text{im } SQ$ and $\text{im } HQ$. In our tests, S and H are nonsingular and thus by the definition of a deflating subspace, $\text{im } SQ = \text{im } HQ$ is satisfied in theory, i. e., $\alpha = 0$. We divide the results into classes which are listed in Table VI. Now, the unfactored version is more accurate for 615 examples in the

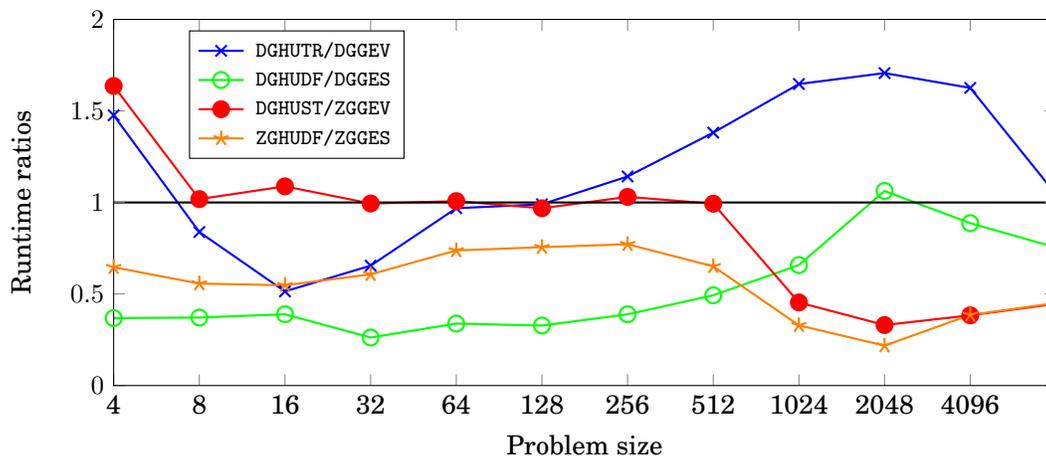


Fig. 5. Runtime ratios of the new routines compared to LAPACK software.

Table VI. Comparison of the errors of the deflating subspaces.

	real case		complex case	
	unfactored	factored	unfactored	factored
$10^{-11} < \alpha < 10^{-10}$	1	0	0	0
$10^{-12} < \alpha < 10^{-11}$	9	11	0	3
$10^{-13} < \alpha < 10^{-12}$	82	96	38	62
$10^{-14} < \alpha < 10^{-13}$	900	888	962	935
$10^{-15} < \alpha < 10^{-14}$	8	5	0	0

Table VII. Comparison of runtimes for factored and unfactored versions (measured in secs.).

Problem size	real case		complex case	
	unfactored	factored	unfactored	factored
2	2.1149e-05	3.7359e-05	4.1219e-05	7.4963e-05
4	5.3765e-05	9.4412e-05	8.9305e-05	1.8112e-04
8	3.6373e-04	5.1282e-04	2.3065e-04	4.2514e-04
16	1.4868e-03	1.8846e-03	7.5680e-04	1.1702e-03
32	5.9223e-03	7.8657e-03	3.2732e-03	5.6365e-03
64	2.3258e-02	3.1986e-02	1.8261e-02	2.8526e-02
128	1.0901e-01	1.4402e-01	1.1473e-01	1.9216e-01
256	5.7424e-01	7.9756e-01	9.2289e-01	1.6150e+00
512	3.8463e+00	6.1073e+00	1.4380e+01	3.3246e+01
1024	4.6299e+01	1.0119e+02	2.5326e+02	4.1394e+02
2048	6.0400e+02	9.5667e+02	2.0491e+03	3.5848e+03
4096	5.4444e+03	7.9957e+03	1.6688e+04	2.8164e+04

real and for 592 examples in the complex case, respectively. Therefore, we can conclude that the subspace computation is slightly more accurate in the unfactored case.

5.4.2. Speed and Memory Requirements. We briefly compare the timing results of the factored and unfactored codes which are listed in Table VII. A run of the factored versions needs approximately 1.5 – 2 times as long as one of the unfactored versions. This is simply due to fact that also more matrix entries (usually $\approx 50\%$ more) have to be updated within the factored codes. Also this higher amount of matrices has to be stored which leads to an approximately 50% higher memory usage.

5.4.3. Conclusion. In conclusion, we can say that one should always use the unfactored version of the code whenever the matrix \mathcal{S} is explicitly given or can be formed without any rounding errors. This is due to the lower accuracy, larger runtimes and higher memory usage of the factored versions. However, one might think of situations where only the factor \mathcal{Z} is known and it is not possible to appropriately form \mathcal{S} due to numerical errors. Then we still recommend to use the factored versions even in the presence of the disadvantages mentioned above.

5.5. Blocked Versus Unblocked Code

As already mentioned, the routines get relatively slow if the problem gets too large. This is due to the unoptimized cache usage. Therefore, we have implemented the unfactored algorithms using the panel blocking technique from Subsection 4.3. For illustration we generated a random example of order 2048 and compared the runtimes of the unblocked code with those of the blocked code for different block sizes NB. The results can be found in Table VIII. The smallest runtimes are marked in boldface font. The time savings can be significant. For computing the eigenvalues of a complex pencil, the reduction in runtime can be up to 50% compared to the unblocked code. Note that there is only a slight speedup for the subspace computation in the real case since the time-consuming routine DGHUYR cannot be blocked. Mostly, the best timings are attained for $NB = 8$, however similar results are observed for all $NB = 4, \dots, 128$, so the

Table VIII. Comparison of runtimes for blocked and unblocked code (measured in secs.).

block size NB	eigenvalues only		eigenvalues and deflating subspaces	
	real case	complex case	real case	complex case
unblocked	224.23	1332.72	582.65	2173.03
1	200.91	922.11	546.95	1644.15
2	147.87	738.48	517.41	1458.51
4	133.90	662.78	483.16	1365.91
8	132.31	657.39	466.02	1345.67
16	146.86	680.06	470.36	1383.82
32	139.47	680.46	469.58	1363.19
64	138.72	672.18	469.72	1376.00
128	139.44	700.46	459.48	1400.46
256	139.62	1024.32	468.70	1775.13
512	151.18	1116.98	482.61	1844.09
1024	216.94	1165.43	548.82	1955.30
2048	210.05	1244.88	555.28	2025.61

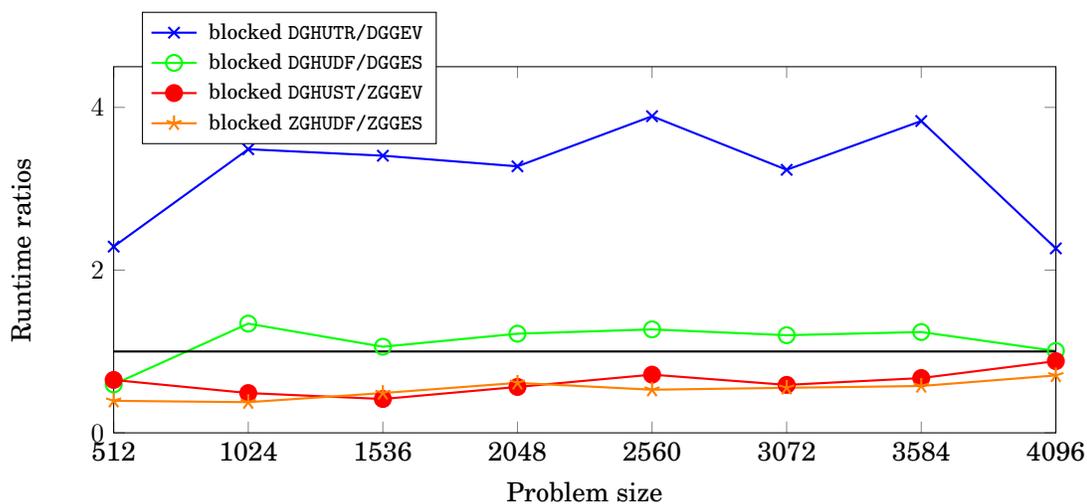


Fig. 6. Runtime ratios of the blocked codes compared to LAPACK software for larger problem sizes.

choice of NB is flexible. An important point is that the problem must be sufficiently large in order to benefit from the panel blocking, otherwise one would even lose performance, especially for small block sizes. Finally, we also compare the performance of our blocked codes for NB = 8 with LAPACK for problem sizes of 512 to 4096. We refer to Figure 6 for the specific LAPACK routines we have used for our comparison and the corresponding runtime ratios. For real problems we can achieve good speedups compared to LAPACK. When computing only the eigenvalues we can achieve a speedup factor of about 3.5. When we also compute the deflating subspaces we still get a factor of 1.2, so we are still faster than LAPACK. However, this is not the case anymore when we consider complex problems. In this case we achieve a slowdown by a factor between 0.4 to 0.9, but the blocked codes are still faster than the unblocked ones.

6. SUMMARY

We have presented algorithms which can be used to compute the eigenvalues and deflating subspaces of skew-Hamiltonian/Hamiltonian matrix pencils in a structure-preserving way which may lead to higher accuracy, reliability, and computational performance. Applications which are based on matrix pencils of this structure have been

introduced to show the importance of our considerations. Moreover, we have presented details of the implementation of the algorithms in Fortran 77. Numerical examples have demonstrated the increased reliability since critical purely imaginary eigenvalues are not perturbed off the imaginary axis (as long as their pairwise distance is large enough). This has also allowed the safe computation of the associated stable deflating subspaces of the pencil since the perturbation of eigenvalues from the left into the right half-plane (or vice versa) is avoided. On the other hand, the runtimes are often higher compared to LAPACK routines. However, a panel blocking technique has significantly improved performance for larger problems.

REFERENCES

- J. Abels and P. Benner. 1999. *CAREX - A collection of benchmark examples for continuous-time algebraic Riccati equations (version 2.0)*. SLICOT Working Note 1999-14. Niconet e. V.
- A. C. Antoulas. 2005. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, PA.
- P. Benner, R. Byers, and E. Barth. 2000. Algorithm 800: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: the square-reduced method. *ACM Trans. Math. Softw.* 26, 1 (2000), 49–77. DOI:<http://dx.doi.org/10.1145/347837.347852>
- P. Benner, R. Byers, P. Losse, V. Mehrmann, and H. Xu. 2007. Numerical solution of real skew-Hamiltonian/Hamiltonian eigenproblems. (Nov. 2007). Unpublished report.
- P. Benner, R. Byers, V. Mehrmann, and H. Xu. 2002. Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Matrix Anal. Appl.* 24, 1 (2002), 165–190.
- P. Benner, H. Fassbender, and M. Stoll. 2008a. Solving large-scale quadratic eigenvalue problems with Hamiltonian eigenstructure using a structure-preserving Krylov subspace method. *Electron. Trans. Numer. Anal.* 29 (2008), 212–229.
- P. Benner, P. Losse, V. Mehrmann, L. Poppe, and T. Reis. 2008b. γ -Iteration for descriptor systems using structured matrix pencils. In *Proc. International Symposium on Mathematical Theory of Networks and Systems*. Blacksburg VA, USA.
- P. Benner, V. Sima, and M. Voigt. 2012a. L_∞ -norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans. Automat. Control* 57, 1 (2012), 233–238. DOI:<http://dx.doi.org/10.1109/TAC.2011.2161833>
- P. Benner, V. Sima, and M. Voigt. 2012b. Robust and efficient algorithms for L_∞ -norm computations for descriptor systems. In *Proc. 7th IFAC Symposium on Robust Control Design*. Aalborg, Denmark, 195–200. DOI:<http://dx.doi.org/10.3182/20120620-3-DK-2025.00114>
- P. Benner, V. Sima, and M. Voigt. 2015. *SHHEIG Users' Guide*.
- A. Bojanczyk, G. H. Golub, and P. Van Dooren. 1992. The periodic Schur decomposition. Algorithms and applications. In *Advanced Signal Processing Algorithms, Architectures, and Implementations III (Proc. SPIE)*, F. T. Luk (Ed.), Vol. 1770. 31–42.
- T. Brüll and V. Mehrmann. 2007. *STCSSP: A FORTRAN 77 routine to compute a structured staircase form for a (skew-)symmetric / (skew-)symmetric matrix pencil*. Preprint 31-2007. Institut für Mathematik, TU Berlin.
- R. Byers, V. Mehrmann, and H. Xu. 2007. A structured staircase algorithm for skew-symmetric/symmetric pencils. *Electron. Trans. Numer. Anal.* 26 (2007), 1–33.
- L. Dai. 1989. *Singular Control Systems*. Lecture Notes in Control and Inform. Sci., Vol. 118. Springer-Verlag, Heidelberg.
- Eindhoven University of Technology, The Netherlands 1996a. *SLICOT Contributors' Kit 2.1*. Eindhoven University of Technology, The Netherlands. WGS Report 96-2.
- Eindhoven University of Technology, The Netherlands 1996b. *SLICOT Implementation and Documentation Standards*. Eindhoven University of Technology, The Netherlands. WGS Report 96-1.
- G. H. Golub and C. F. Van Loan. 1996. *Matrix Computations* (third ed.). The John Hopkins University Press, Baltimore/London.
- J. J. Hench and A. J. Laub. 1994. Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Automat. Control* 39, 6 (1994), 1197–1210.
- A. Kawamoto, K. Takaba, and T. Katayama. 1999. On the generalized algebraic Riccati equation for continuous-time descriptor systems. *Linear Algebra Appl.* 296 (1999), 1–14.
- P. Lancaster. 2013. Stability of linear gyroscopic systems: A review. *Linear Algebra Appl.* 439, 3 (2013), 686–706. Special Issue in Honor of Harm Bart.

- P. Losse. 2012. *The \mathcal{H}_∞ Optimal Control Problem for Descriptor Systems*. Dissertation. Fakultät für Mathematik, Technische Universität Chemnitz.
- P. Losse, V. Mehrmann, L. Poppe, and T. Reis. 2008. The modified optimal \mathcal{H}_∞ control problem for descriptor systems. *SIAM J. Control Optim.* 47, 6 (2008), 2795–2811.
- C. Mehl. 1999. *Compatible Lie and Jordan algebras and applications to structured matrices and pencils*. Dissertation. Chemnitz University of Technology, Faculty of Mathematics, Germany, Berlin.
- C. Mehl. 2000. Condensed forms for skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Matrix Anal. Appl.* 21, 2 (2000), 454–476. DOI:<http://dx.doi.org/10.1137/S0895479898336513>
- V. Mehrmann. 1991. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Lecture Notes in Control and Inform. Sci., Vol. 163. Springer-Verlag, Heidelberg.
- V. Mehrmann and T. Stykel. 2005. Balanced truncation model reduction for large-scale systems in descriptor form. In *Dimension Reduction of Large-Scale Systems*, P. Benner, V. Mehrmann, and D. Sorensen (Eds.). Lect. Notes Comput. Sci. Eng., Vol. 45. Springer-Verlag, Berlin, Heidelberg, New York, Chapter 3, 89–116.
- V. Mehrmann and H. Xu. 2008. Perturbation of purely imaginary eigenvalues of Hamiltonian matrices under structured perturbations. *Electron. J. Linear Algebra* 17 (2008), 234 – 257.
- L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishenko. 1962. *The Mathematical Theory of Optimal Processes*. Interscience, New York.
- L. Poppe, C. Schröder, and I. Thies. 2009. *PEPACK: A software package for computing the numerical solution of palindromic and even eigenvalue problems using the pencil Laub trick*. Preprint 22-2009. Institut für Mathematik, Technische Universität Berlin.
- T. Reis. 2011. Lur’e equations and even matrix pencils. *Linear Algebra Appl.* 434, 1 (2011), 152–173.
- T. Reis, O. Rendel, and M. Voigt. 2015. The Kalman-Yakubovich-Popov inequality for differential-algebraic systems. *Linear Algebra Appl.* 485 (2015), 153–193.
- F. Tisseur and K. Meerbergen. 2001. The quadratic eigenvalue problem. *SIAM Rev.* 43, 2 (2001), 235–286.
- M. Voigt. 2010. *\mathcal{L}_∞ -Norm Computation for Descriptor Systems*. Diploma Thesis. Chemnitz University of Technology, Faculty of Mathematics, Germany. <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-201001050>
- M. Voigt. 2015. *On Linear-Quadratic Optimal Control and Robustness of Differential-Algebraic Systems*. Dissertation. Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg. Accepted.
- K. Zhou and J. D. Doyle. 1998. *Essentials of Robust Control* (1st ed.). Prentice Hall, Upper Saddle River, NJ.