Pitfalls When Solving Eigenproblems
With Applications in Control Engineering

Vasile Sima[1] and Peter Benner[2]
[1]*National Institute for Research & Development in Informatics, Bd. Mareşal Averescu, Nr. 8–10, Bucharest, Romania*
[2]*Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106, Magdeburg, Germany*

Keywords: Deflating Subspaces, Eigenvalue Reordering, Generalized Eigenvalues, Generalized Schur Form, Numerical Methods, Skew-Hamiltonian/Hamiltonian Matrix Pencil, Software, Structure-preservation.

Abstract: There is a continuous research effort worldwide to improve the reliability, efficiency, and accuracy of numerical computations in various domains. One of the most promising research avenues is to exploit the structural properties of the mathematical problems to be solved. This paper investigates some numerical algorithms for the solution of common and structured eigenproblems, which have many applications in automatic control (e.g., linear-quadratic optimization and H_∞-optimization), but also in various areas of applied mathematics, physics, and computational chemistry. Of much interest is to find the eigenvalues and certain deflating subspaces, mainly those associated to the stable eigenvalues. Several simple examples are used to highlight the pitfalls which may appear in such numerical computations, using state-of-the-art solvers. Balancing the matrices and the use of condition numbers for eigenvalues are shown to be essential options in investigating the behavior of the solvers and problem sensitivity.

1 INTRODUCTION

Many control system analysis and design numerical techniques and procedures require the computation of eigenvalues and bases of certain invariant or deflating subspaces. Often, the corresponding eigenproblems have special structure, which implies structural properties of their spectra. Common structures are Hamiltonian and symplectic matrices or matrix pencils. One relevant computational problem in control theory and its applications is the evaluation of the H_∞- and L_∞-norms of linear dynamic systems, which are used, e.g., to quantify the trade-off between performance and robust stability. State-of-the-art quadratically convergent algorithms for the calculation of these norms use the purely imaginary eigenvalues of a matrix or matrix pencil at each iteration. This matrix (pencil) is Hamiltonian or symplectic, in the continuous- and discrete-time case, respectively. (Actually, the pencils arising in the continuous-time descriptor case can be put in a skew-Hamiltonian/Hamiltonian form (Benner et al., 2012a).) Another fundamental computation in control systems analysis and design is the solution of continuous-time and discrete-time algebraic Riccati equations (CAREs and DAREs). CAREs and DAREs arise in many applications, such as factorization techniques for transfer functions matrices, model reduction procedures based on stochastic bounded- or positive real balancing, stabilization and linear-quadratic regulator problems, Kalman filtering, linear-quadratic Gaussian (H_2-)optimal control problems, computation of (sub)optimal H_∞ controllers, etc. Usually, the stabilizing solution is required, which can be used to stabilize the closed-loop system matrix or matrix pencil. State-of-the-art CARE/DARE solvers (Laub, 1979; Mehrmann, 1991; Sima, 1996; Benner, 1999; MathWorks, 2014) rely on computing stable invariant or deflating subspaces of some Hamiltonian or symplectic matrices or pencils. Finding such subspaces involves eigenvalue reordering.

Recently, structure-exploiting techniques have been investigated for solving (skew-)Hamiltonian and skew-Hamiltonian/Hamiltonian eigenproblems, see, e.g., (Benner et al., 2002; Benner and Kressner, 2006; Benner et al., 2007), and the references therein. These algorithms are more involved than those used in the non-structured case. The initial reduction step is more complicated, since the structure should be exploited. Moreover, the standard QR or QZ algorithm (Golub and Van Loan, 1996) for matrices or pencils is replaced by the periodic QR/QZ algorithm, see (Bo-

Sima V. and Benner P..
Pitfalls When Solving Eigenproblems - With Applications in Control Engineering.
DOI: 10.5220/0005533301710178
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 171-178
ISBN: 978-989-758-122-9

janczyk et al., 1992; Granat et al., 2007) and the references inside. The periodic QR/QZ algorithm operates on formal matrix products $\Pi_{i=1}^{p} A_i^{s_i}$, where $s_i = 1$ or $s_i = -1$, with possibly singular factors in the pencil case. The number of factors p to process (the "period") is two, four or six (Benner et al., 2002; Benner and Kressner, 2006; Benner et al., 2007). Eigenvalue reordering is also performed on these formal matrix products. The main difference to the standard reordering procedures implemented in the LAPACK package (Anderson et al., 1999) is in the algorithm for swapping two adjacent sequences of diagonal blocks, involving the solution of periodic Sylvester-like equations. Details are given, e.g., in (Granat et al., 2007; Sima, 2010) and the references therein.

Advanced structure-exploiting solvers have been incorporated in the SLICOT Library (www.slicot.org), see (Benner et al., 1999; Van Huffel et al., 2004; Benner et al., 2010; Sima et al., 2012; Benner et al., 2013a; Benner et al., 2013b). Fortran and MATLAB software for eigenvalues and invariant or deflating subspaces has been developed, for both real and complex matrices. Versions with a factored or not factored skew-Hamiltonian matrix $\mathcal{S}$ in a pencil are covered. Some performance results for computing the eigenvalues or eigenvalues and stable deflating subspaces for real or complex matrix pencils, with factored or not factored matrix $\mathcal{S}$ are given in (Sima, 2011a; Sima, 2011b; Sima et al., 2012; Benner et al., 2012b; Benner et al., 2013b). The results have shown that these solvers provide reliable and accurate solutions, and are often faster than the state-of-the-art tools.

This paper addresses the numerical difficulties in computations with structured solvers, mainly for the periodic QR/QZ steps involved, e.g., difficulties associated to multiple or close eigenvalues. Small examples, including standard eigenproblems and standard products of 2-by-2 matrices, illustrating the numerical behavior, are analyzed based on experimental evidence. Our examples show the need to use specialized measures, such as (reciprocal) condition numbers, and special techniques, like balancing, even when solving or analyzing small eigenproblems.

2 ACCURACY OF COMPUTED EIGENVALUES

Several kinds of $n \times n$ matrix pencils $A - \lambda B$ are considered in this paper. The *eigenvalues* of $A - \lambda B$ are the complex numbers λ_i, $i = 1 : n$ (multiplicities counted), satisfying the relations $Ax_i = \lambda B x_i$, with $x_i \neq 0$, $i = 1 : n$. The vector x_i is a *right eigenvector* corresponding to the eigenvalue λ_i; a vector $y_i \neq 0$ satisfying $y_i^H A = \lambda_i y_i^H B$ is a *left eigenvector* corresponding to λ_i, where y_i^H is the conjugate-transpose of y_i. Finding the eigenvalues and eigenvectors of $A - \lambda B$ is the generalized eigenvalue problem. A standard eigenvalue problem is obtained for $B = I_n$, the identity matrix of order n. Simple eigenvalues (i.e., with multiplicity 1) can be computed accurately, if well-conditioned. Multiple eigenvalues are inherently ill-conditioned, and so they cannot generally be found with high accuracy, even for standard small order eigenproblems. Briefly speaking, an eigenvalue λ_i is well-conditioned if small perturbations in the matrix A or matrix pair (A, B) produce small changes in the corresponding computed eigenvalue, $\widehat{\lambda}_i$. Eigenvalue condition numbers are used to assess the conditioning, or, equivalently, the sensitivity to perturbations in the data. Since condition numbers can be infinite, their reciprocals are used in practical calculations.

LAPACK driver routines compute estimates of the reciprocal condition numbers for individual or clusters of eigenvalues and eigenvectors. We will denote these estimates by $\mathrm{rcond}(\lambda_i)$ and $\mathrm{rcond}(x_i)$, respectively. A balancing option allows to perform initial row and column permutations (to isolate the eigenvalues available by inspection in the leading/trailing parts of A and B), scale the matrices to make the 1-norms of rows and corresponding columns as close as possible, or to both permute and scale. Balancing often increases the accuracy of computed eigenvalues. An approximate error bound on the chordal distance between the computed generalized eigenvalue $\widehat{\lambda}_i$ and the corresponding exact eigenvalue λ_i is

$$\chi(\widehat{\lambda}_i, \lambda_i) \leq \varepsilon_M \, \| \, [\, \|A\|, \|B\| \,] \, \| / \mathrm{rcond}(\lambda_i),$$

where ε_M is the relative machine accuracy, and the norms are for the matrices after balancing, if applied. (The 1-norm is used in LAPACK routines for A and B, but 2-norm for the vector of norms in square brackets.) This means that perturbations in the elements of A and B can be amplified by $1/\mathrm{rcond}(\lambda_i)$ in the computed eigenvalue $\widehat{\lambda}_i$. The chordal distance between two points $(\widehat{\alpha}, \widehat{\beta})$ and (α, β) is defined by

$$\chi([\widehat{\alpha}, \widehat{\beta}], [\alpha, \beta]) = \frac{|\widehat{\alpha}\beta - \alpha\widehat{\beta}|}{\sqrt{|\widehat{\alpha}|^2 + |\widehat{\beta}|^2}\sqrt{|\alpha|^2 + |\beta|^2}}.$$

An approximate error bound for the acute angle between the computed left or right eigenvectors $\widehat{y}_i$ or $\widehat{x}_i$, and the true eigenvectors y_i or x_i, respectively, is given by $\varepsilon_M \, \| \, [\, \|A\|, \|B\| \,] \, \| / \mathrm{rcond}(x_i)$. See (Anderson et al., 1999) for further details, where tighter bounds are also given. A very simple example illustrates what can be expected in eigenvalue computations using numerical algorithms.

Example 1. Consider a stable linear system with transfer-function

$$H(s) = \frac{1}{(s+1)^2(s+2)},$$

hence with a double pole at -1, and another pole at -2. A state-space matrix for this system is

$$A = \begin{bmatrix} -4 & -5 & -2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Indeed, this is the companion matrix of the polynomial z^3+4z^2+5z+2, which has exact roots -1, with multiplicity 2, and -2. The eigenvalues of A should coincide with the system poles. However, the eigenvalues corresponding to the double pole cannot generally be computed accurately. Indeed, the MATLAB R2014b (MathWorks, 2014) function `eig`, based on the state-of-the-art LAPACK routines, returns in double precision arithmetic

$$\begin{bmatrix} -2 \\ -0.9999999999999997 \pm 2.83263461777919 \cdot 10^{-8} \imath \end{bmatrix}.$$

when using the command `eig(A)`. The pole at -2 is computed to full accuracy, but the double pole at -1 became a pair of complex conjugate values. The (absolute and relative) errors of the eigenvalues of this pair are about $2.83 \cdot 10^{-8}$, slightly smaller than $2\sqrt{\varepsilon_M}$, i.e., twice the square root of the relative machine accuracy, hence more than half of the accuracy, has been lost. Calling the function `eig` with no optional argument, as above, computes the eigenvalues after a preliminary scaling of matrix A with a diagonal matrix D, i.e., the iterative QR algorithm is applied to the matrix $D^{-1}AD$, with $D = \mathrm{diag}(2,1,1)$ for this example. Optionally, the LAPACK driver `DGEESX` also returns the reciprocal condition numbers for eigenvalues and eigenvectors, which in this case are

$$\begin{aligned} \mathrm{rcond}(\lambda_{1:3}) &= [0.11111, 9.1602 \cdot 10^{-9}, 9.1602 \cdot 10^{-9}], \\ \mathrm{rcond}(x_{1:3}) &= [0.26087, 1.0950 \cdot 10^{-8}, 1.0950 \cdot 10^{-8}], \end{aligned}$$

respectively (rounded to 5 significant digits). The transformed matrix A returned by `DGEESX` is block diagonal with diagonal blocks of order 1 and 2, and the second block is

$$\begin{bmatrix} -0.9999999999999997 & -2.833333333333334 \\ 2.831936074532138 \cdot 10^{-16} & -0.9999999999999997 \end{bmatrix}.$$

The small, but nonzero value of the subdiagonal element is responsible for getting a pair of complex conjugate eigenvalues. Without balancing (e.g., using the command `eig(A,'nobalance')`), the last two eigenvalues are even less accurate,

$$-0.9999999999999986 \pm 3.817602171155931 \cdot 10^{-8} \imath,$$

with errors of about $3.82 \cdot 10^{-8}$, and the nonzero subdiagonal entry of the transformed matrix A is $-4.7705 \cdot 10^{-16}$. The reciprocal condition numbers for eigenvalues and eigenvectors are

$$\begin{aligned} \mathrm{rcond}(\lambda_{1:3}) &= [8.9087 \cdot 10^{-2}, 1.1781 \cdot 10^{-8}, 1.1781 \cdot 10^{-8}], \\ \mathrm{rcond}(x_{1:3}) &= [0.24661, 1.3038 \cdot 10^{-8}, 1.3038 \cdot 10^{-8}]. \end{aligned}$$

Consider now a perturbation in the system poles. Specifically, replacing the double pole by $-1-\varepsilon$ and $-1+\varepsilon$, the characteristic polynomial becomes

$$z^3 + (4+\varepsilon)z^2 + (5+2\varepsilon-\varepsilon^2)z + 2 + \varepsilon - 2\varepsilon^2 - \varepsilon^3,$$

and the eigenvalues of the associated companion matrix have been computed for values of ε set to $\sqrt{\varepsilon_M}$, $10\sqrt{\varepsilon_M}$, and $\varepsilon_M^{1/4}$. Three real eigenvalues have been obtained for all tried parameter values, but just one real eigenvalue for $\varepsilon = \sqrt{\varepsilon_M}$ when balancing was used. The first eigenvalue was always accurately computed, with errors of about $5.33 \cdot 10^{-15}$ or less, with or without balancing. On the other hand, the other two eigenvalues had errors of about $3.87 \cdot 10^{-8}$ or smaller. Specifically, for $\varepsilon = 10\varepsilon_M$ the errors were about $5.45 \cdot 10^{-10}$ without balancing, and for $\varepsilon = \sqrt{\varepsilon_M}$, the errors were $6.75 \cdot 10^{-13}$ and $5.14 \cdot 10^{-12}$, with and without balancing, respectively. The reciprocal condition numbers have been comparable with their values in the unperturbed case for ε set to $\sqrt{\varepsilon_M}$ or $10\sqrt{\varepsilon_M}$, but increased to over $3.7 \cdot 10^{-5}$ for $\lambda_{2:3}$ and $\varepsilon_M^{1/4}$. Note that the error bounds given by the formulas above are quite tight for all the investigated cases. The better accuracy for the computed eigenvalues $\widehat{\lambda}_{2:3}$ corresponding to $-1-\varepsilon$ and $-1+\varepsilon$ when $\varepsilon = \varepsilon_M^{1/4}$ is due to the bigger gap, $2\varepsilon_M^{1/4}$, between the true eigenvalues than in the other cases. When this gap is zero, about half of the accuracy is lost for this example. For a third order system with a triple pole, about two thirds of accuracy (about 12 significant digits in double precision computations) might be lost for each $\widehat{\lambda}_i$. □

3 ACCURACY OF EIGENVALUES FOR MATRIX PRODUCTS

Numerical difficulties appear also when computing the periodic Schur decomposition of a matrix product, $\Pi_{i=1}^{p} A_i$. This may happen even for 2×2 matrices and $p = 2$. Such a decomposition is used, e.g., for computing the eigenvalues and invariant subspaces of Hamiltonian matrices (Benner and Kressner, 2006). When A_i are general matrices, the algorithm first reduces the matrix sequence to the periodic Hessenberg

form, using unitary transformations so that the eigenvalues of the product are preserved. In this form, one matrix, usually the first one, is upper Hessenberg (i.e., all elements below the first subdiagonal are zero), and all other factors are upper triangular. Then, the sequence is further transformed similarly to one in which the Hessenberg matrix is reduced to Schur form, while the other matrices remain upper triangular. For complex matrices, the Schur form is also upper triangular, while in the real case, it is block upper triangular with diagonal blocks of order 1 or 2, corresponding to real or complex conjugate eigenvalues, respectively. The real case only will be considered in the sequel, for convenience. As suggested by the above presentation, the matrix product itself is not evaluated, since forming it may yield large errors, depending on its conditioning. However, the algorithm computes some products of elements on the diagonals (and uses sums with other products involving subdiagonal or superdiagonal elements), after the preliminary reduction to the Hessenberg-triangular form. If some factors are ill-conditioned, it is possible that inaccuracies in the intermediate results be strongly amplified in the computed decomposition. Nevertheless, the returned eigenvalues are usually highly accurate.

The periodic Schur algorithm finds orthogonal transformation matrices, Z_i, $i = 1 : p$, such that the transformed matrices,

$$\widetilde{A}_i = Z_i^T A_i Z_{\mathrm{mod}\,(i,p)+1}, \quad i = 1 : p, \qquad (1)$$

have the required structure, i.e., $\widetilde{A}_1$ is in upper real Schur form, and $\widetilde{A}_j$, $j = 2 : p$, are upper triangular. The eigenvalues are then readily obtained. Note that (1) performs a similarity transformation, since $\Pi_{i=1}^{p}\widetilde{A}_i = Z_1^T(\Pi_{i=1}^{p}A_i)Z_1$, hence the eigenvalues are theoretically preserved. The algorithm delivers the eigenvalues and the matrices Z_i and $\widetilde{A}_i$, $i = 1 : p$. Simple and well separated eigenvalues are accurately computed; moreover, the formulas (1) are satisfied with (high) precision for $i > 1$, but sometimes this is not the case for $i = 1$, i.e., the obtained Schur form may not be close to the true real Schur form. Also, the eigenvalues of the matrix product $\Pi_{i=1}^{p}\widetilde{A}_i$ may differ from the computed eigenvalues, as shown by numerical experiments. In some applications it is important to have accurate matrices $\widetilde{A}_i$, $i = 1 : p$. Let $R_i = Z_i^T A_i Z_{\mathrm{mod}\,(i,p)+1} - \widetilde{A}_i$, $i = 1 : p$, be the residuals of the computed transformed matrices. Large norms of some of these residuals indicate an inaccurate periodic Schur decomposition. The accuracy can be increased by an iterative refinement process. Specifically, another algorithm iteration (called sweep below) is applied to the sequence $\widetilde{A}_i + R_i$, $i = 1 : p$, and the process continues similarly till the residual norms are smaller than a given tolerance, or a specified number of iterations is exhausted. Often, one additional sweep ensures acceptable accuracy, and sometimes it is enough to update the matrix $\widetilde{A}_1$ only. If the transformations used in the additional sweep(s) are denoted by Y_i, then Z_iY_i, $i = 1 : p$, usually transform the original problem to one having the right structure and accurate eigenvalues.

Table 1: Absolute difference between the returned (2,1) element of $\widetilde{A}_1$ and its value computed using (1) for various number of factors, p.

p	(2,1)-element error	p	(2,1)-element error
10	$7.78{\cdot}10^{-11}$	26	$3.65{\cdot}10^{-2}$
11	$7.43{\cdot}10^{-10}$	28	0.7068
16	$8.66{\cdot}10^{-7}$	29	0.7743
19	$1.52{\cdot}10^{-4}$	30	0.7163
22	$1.85{\cdot}10^{-3}$	31	0.7205
25	$4.98{\cdot}10^{-2}$	32	0.5688

In a series of tests, the periodic Schur solver implementation in the SLICOT Library has been applied to several products of 2×2 real matrices. All values of p from 2 to 32 have been tried. Since the investigated examples had real eigenvalues only, all factors $\widetilde{A}_i$, $i \geq 1$, are upper triangular. The computed eigenvalues have been compared to those obtained using the MATLAB functions `eig` and its symbolic counterpart, `symbolic/eig`, applied to the product of factors, $\Pi_{i=1}^{p}A_i$, the symbolic product of symbolic factors, $\Pi_{i=1}^{p}\mathcal{A}_i$, the product of transformed factors, $\Pi_{i=1}^{p}\widetilde{A}_i$, and the symbolic product of symbolic transformed factors, $\Pi_{i=1}^{p}\widetilde{\mathcal{A}}_i$. The eigenvalues of all these products agreed well. The difference between the computed periodic matrix and the transformed original periodic matrix, obtained using (1), has also been computed. For several values of p, the (2,1) element of the factor $\widetilde{A}_1$ from (1) differed significantly from its zero value returned by the solver. All other elements for all factors agreed very well to the returned values. Table 1 gives the absolute value of the difference for various p and one series of factor matrices.

The reciprocal condition numbers of the eigenvalues and eigenvectors of the matrix product have also been computed, and their values have usually been quite close to 1, agreeing with the eigenvalues' good observed accuracy. However, the conditioning for the product of the transformed factors has been several orders of magnitude worse. The computed eigenvalues most often correspond accurately to those of $\Pi_{i=1}^{p}A_i$, but not to those of $\Pi_{i=1}^{p}\widetilde{A}_i$. Using one additional sweep, after updating the factor $\widetilde{A}_1$ only, the absolute difference was of the order 10^{-15} or less, for the values p in Table 1. Moreover, the eigenvalues

have usually been more accurate.

The results for several other series of tests with some mildly ill-conditioned factors have been similar, but with different values of p and of the absolute difference. Actually, the numerical difficulties described above can be encountered even for a product of two 2×2 matrices, as illustrated by the example below.

Example 2. Let

$$A_1 = \begin{bmatrix} 1.237 & 2.058 \\ 2.058 & 3.425 \end{bmatrix}, A_2 = \begin{bmatrix} 16.825 & 13.890 \\ 13.890 & 11.467 \end{bmatrix},$$

be exact representations of the two matrices. Note that the condition numbers for A_1, A_2, and A_1A_2 are about $1.5967 \cdot 10^4$, $4.5739 \cdot 10^6$, and $6.4931 \cdot 10^{10}$, respectively. There is an absolute error (in the (2,1) element only) of order 10^{-14} between the product A_1A_2 computed in double precision arithmetic and that computed using symbolic calculations, via the MATLAB command `double(sym(A_1)*sym(A_2))`. The eigenvalues obtained using symbolic calculations and conversion to double precision numbers, are

$$\lambda = [2.031200536560380 \cdot 10^{-9}; 1.172582399979688 \cdot 10^{2}];$$

the SLICOT periodic Schur eigensolver returned

$$\widehat{\lambda} = [2.031200097007968 \cdot 10^{-9}; 1.172582399979688 \cdot 10^{2}],$$

and the transformed matrices $\widetilde{A}_1$ and $\widetilde{A}_2$

$$\begin{bmatrix} -3.096329932867903 \cdot 10^{-4} & 1.552701519915428 \\ 0 & -4.395524985047830 \end{bmatrix}$$
$$\begin{bmatrix} -6.560024032069280 \cdot 10^{-6} & -9.422786660123904 \\ 0 & -26.67673153813540 \end{bmatrix}$$

respectively. Therefore, the eigenvalues of the product of the transformed matrices, computed as products of the corresponding diagonal elements are

$$\widetilde{\lambda} = [2.031199877082891 \cdot 10^{-9}; 1.172582399952876 \cdot 10^{2}].$$

The element-wise absolute and relative errors between $\widehat{\lambda}$ and λ are about $4.40 \cdot 10^{-16}$ and $4.26 \cdot 10^{-14}$, and $2.16 \cdot 10^{-7}$ and $3.64 \cdot 10^{-16}$, respectively. The last value is also the approximate global relative error, i.e., $\|\widehat{\lambda} - \lambda\| / \|\lambda\|$, using Euclidean norm. On the other hand, the element-wise absolute and relative errors between $\widetilde{\lambda}$ and λ are about $6.59 \cdot 10^{-16}$ and $2.68 \cdot 10^{-9}$, and $3.25 \cdot 10^{-7}$ and $2.29 \cdot 10^{-11}$; the last value is the approximate global relative error of $\widetilde{\lambda}$ compared to λ.

Clearly, although obtained using orthogonal transformations only, the transformed matrices $\widetilde{A}_i$, $i = 1,2$, are inaccurate, and therefore $\widetilde{\lambda}_i$ are less accurate than $\widehat{\lambda}_i$. The inaccuracy of $\widetilde{A}_i$ is due to the ill-conditioning of the matrix product A_1A_2, and the way the periodic Schur algorithm works. The residual matrices of $\widetilde{A}_i$, $i = 1,2$, have the following approximate values

$$R_1 = \begin{bmatrix} 0 & 4.44 \cdot 10^{-16} \\ 2.85 \cdot 10^{-10} & 0 \end{bmatrix},$$
$$R_2 = \begin{bmatrix} 5.26 \cdot 10^{-16} & 1.78 \cdot 10^{-15} \\ 1.78 \cdot 10^{-15} & 0 \end{bmatrix}.$$

The largest residual appears in the (2,1) element of R_1. While this element should be zero, and numerically its magnitude should be of the order of ε_M, in the best case, the ill-conditioning of the matrix product has determined its increase by six orders of magnitude.

After updating the computed matrix $\widetilde{A}_1$ only and using an additional sweep on the matrix sequence, which is now in periodic Hessenberg-triangular form, the new residuals have all elements equal to 0, except for the (2,1) element of the new residual R_2, which has a value about $4.25 \cdot 10^{-16}$. Moreover, the global residuals, given by the difference between the matrices returned by the additional sweep and those obtained using (1) with Z_i replaced by Z_iY_i, have all values with magnitude comparable to ε_M. The largest residual element is now the (1,2) element in the second residual matrix, whose value is about $3.55 \cdot 10^{-15}$. Also, the eigenvalues of the matrix sequence returned after the additional sweep coincide to those of $\widetilde{A}_1^1\widetilde{A}_2^1$ and are

$$\widehat{\lambda}^1 = [2.031200536459228 \cdot 10^{-9}; 1.172582399979688 \cdot 10^{2}].$$

The second eigenvalue, $\widehat{\lambda}_2^1$, practically coincides to λ_2 and $\widehat{\lambda}_2$, and the first 10 digits of $\widehat{\lambda}_1^1$ coincide to those of λ_1, hence it has three additional correct digits compared to $\widehat{\lambda}_1$. The element-wise absolute and relative errors between $\widehat{\lambda}^1$ and λ are about $1.01 \cdot 10^{-19}$ and $4.42 \cdot 10^{-14}$, and $4.98 \cdot 10^{-11}$ and $1.21 \cdot 10^{-16}$, respectively, hence there is an improvement of 4 and 5 orders of magnitude compared to $\widetilde{\lambda}$. If $\widetilde{A}_2$ is also updated as $\widetilde{A}_1$, there is no noticeable improvement. □

Many other 2×2 examples with a similar or even worse behaviour have been found. In one case, the (2,1) entry of R_1 had a value of order 10^{-6}, but one additional sweep reduced its magnitude to about 10^{-22}. Larger order examples have also been tried, and behaved better than the 2×2 case. For instance, for an example with four 10×10 factors, the relative error between $\widehat{\lambda}$ and λ (obtained with symbolic computations) has been $6.46 \cdot 10^{-16}$, and the Frobenius norm of the stacked residuals was $5.56 \cdot 10^{-12}$.

The experiments have also shown that the eigenvalues conditioning does not depend on the conditioning of the matrix product.

4 ACCURACY OF EIGENVALUES FOR SKEW-HAMILTONIAN/ HAMILTONIAN PROBLEMS

Often, badly scaled matrices or matrix pencils have eigenvalues with magnitudes covering a large interval. A structured matrix pencil example will be discussed below. Consider a real $2m \times 2m$ skew-Hamiltonian/ Hamiltonian pencil $\alpha\mathcal{S} - \beta\mathcal{H}$, with

$$\mathcal{S} = \begin{bmatrix} A & D \\ E & A^T \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} C & V \\ W & -C^T \end{bmatrix}, \qquad (2)$$

where D and E are skew-symmetric ($D = -D^T$, $E = -E^T$), V and W are symmetric, and define $\mathcal{J}$ as

$$\mathcal{J} = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix}.$$

The eigenvalues for the real pencil $\alpha\mathcal{S} - \beta\mathcal{H}$ in (2) are symmetric with respect to both real and imaginary axes of the complex plane. Real or purely imaginary eigenvalues appear in pairs λ, $-\lambda$ (or λ, $\bar{\lambda}$, in the second case), while complex conjugate eigenvalues appear in quadruples, λ, $-\lambda$, $\bar{\lambda}$, and $-\bar{\lambda}$. Using two orthogonal transformations, Q_1 and Q_2, the structure-exploiting algorithm computes the transformed matrices $\widetilde{\mathcal{S}}$, $\widetilde{\mathcal{T}}$, and $\widetilde{\mathcal{H}}$, so that

$$\begin{aligned} Q_1^T \mathcal{S} \mathcal{J} Q_1 \mathcal{J}^T &= \begin{bmatrix} \widetilde{A} & \widetilde{D} \\ 0 & \widetilde{A}^T \end{bmatrix} := \widetilde{\mathcal{S}}, \\ \mathcal{J}^T Q_2^T \mathcal{J} \mathcal{S} Q_2 &= \begin{bmatrix} \widetilde{B} & \widetilde{F} \\ 0 & \widetilde{B}^T \end{bmatrix} := \widetilde{\mathcal{T}}, \\ Q_1^T \mathcal{H} Q_2 &= \begin{bmatrix} \widetilde{C}_1 & \widetilde{V} \\ 0 & \widetilde{C}_2^T \end{bmatrix} := \widetilde{\mathcal{H}}, \qquad (3) \end{aligned}$$

where $\widetilde{A}$, $\widetilde{B}$, $\widetilde{C}_1$ are upper triangular, $\widetilde{C}_2$ is upper quasi-triangular (i.e., block upper triangular with 1×1 or 2×2 diagonal blocks) and $\widetilde{D}$, $\widetilde{F}$ are skew-symmetric. Specifically, the first step reduces $\mathcal{S}$ to skew-Hamiltonian triangular form, using Givens rotations and Householder reflections, and updates A and D in $\mathcal{S}$, as well as C, V, and W in $\mathcal{H}$ correspondingly, producing A^1, ..., W^1, respectively. Let $\mathcal{S}^1$ and $\mathcal{H}^1$ be the obtained matrices, where the two diagonal blocks of $\mathcal{H}^1$ are $C_1 = C^1$ and $C_2 = -C^1$. A skew-Hamiltonian block upper triangular matrix $\mathcal{T}^1$ is built using $B^1 = A^1$ and $F^1 = D^1$. Then, additional transformations are used to reduce the matrix $\mathcal{H}^1$ to a block upper triangular form, while updating A^1, D^1, B^1 and F^1 to preserve their form or structure. Let $\widehat{A}$, $\widehat{D}$, $\widehat{B}$, $\widehat{F}$, $\widehat{C}_1$, $\widehat{V}$, and $\widehat{C}_2$ be the obtained matrices, and $\widehat{\mathcal{S}}$, $\widehat{\mathcal{T}}$, and $\widehat{\mathcal{H}}$ the block matrices built from them, where $\widehat{A}$, $\widehat{B}$, and $\widehat{C}_1$ are upper triangular, $\widehat{D}$ and $\widehat{F}$ are skew-symmetric, and $\widehat{C}_2$ is upper Hessenberg. Finally, the periodic QZ algorithm is applied to transform $\widehat{C}_2$ to upper quasi-triangular form, while preserving the upper triangular form of $\widehat{A}$, $\widehat{B}$, and $\widehat{C}_1$. Specifically, the formal matrix product $\widehat{C}_2\widehat{A}^{-1}\widehat{C}_1\widehat{B}^{-1}$ is transformed without using matrix products and inverses. (Actually, $\widehat{A}$ and $\widehat{B}$ may be singular.) The eigenvalues of the pencil $\alpha\mathcal{S} - \beta\mathcal{H}$ are the positive and negative square roots of the eigenvalues of the formal matrix product $-\widehat{C}_2\widehat{A}^{-1}\widehat{C}_1\widehat{B}^{-1}$.

Example 3. In the computation of the H_∞-norm of a linear control system, an 18×18 skew-Hamiltonian/Hamiltonian pencil was found for which changes in scaling led to one small imaginary eigenvalue becoming real. The case with changes in scaling will be referred to as Test 1 below, while the case without changes will be referred to as Test 2. Specifically, the two smallest eigenvalues were about $2.52 \cdot 10^{-5}\,\imath$ and $8.66 \cdot 10^{-4}\,\imath$ for Test 2 data, but $2.30 \cdot 10^{-3}\,\imath$ and $1.83 \cdot 10^{-5}$ for Test 1 data. The final effect was obtaining a wrong H_∞-norm in the Test 1 case. A detailed investigation of the associated numerical issue has been performed.

Our specific example has $m = 9$ and a simple structure. Specifically, A is diagonal with $a_{ii} = a_{11}$, $i = 2:7$, $a_{jj} = 0$, $j = 8:9$, C is almost upper triangular, with $c_{:,9} = 0$, but $c_{i,i-1}$, $i = 3:7$, $c_{8,1:2}$ and $c_{9,2:7}$ are nonzero; moreover, $D = E = 0$ and V and W are diagonal, with $v_{ii} = w_{ii} = 0$, $i = 1:7$, $v_{jj} = -1$ and $w_{jj} = 1$, $j = 8:9$; all other elements are 0. There are four infinite eigenvalues. The only differences between Test 1 and Test 2 data are in the nonzero elements of A and C.

The first idea was to suspect an error in the SLICOT Library (Benner et al., 1999; Van Huffel et al., 2004; Benner et al., 2013a; Benner et al., 2013b) routines used, i.e., in subroutine `MB04BD` or in one of the routines it calls, e.g., a wrong decision test. But a step by step analysis of the intermediate results proved their correctness. Indeed, the transformed matrices computed from the Test 1 data, including $\widehat{A}$, $\widehat{D}$, $\widehat{B}$, $\widehat{F}$, $\widehat{C}_1$, $\widehat{V}$, and $\widehat{C}_2$ (obtained in `MB04BD` just before calling the periodic QZ algorithm), satisfy the required structure and the needed relationships with the initial skew-Hamiltonian/Hamiltonian pencil. The maximum relative error is $2.42 \cdot 10^{-16}$. The finite and "positive" eigenvalues of the reduced problem, computed using symbolic calculations, immediately after the infinite part has been deflated, agree in position, and within a relative error of about $\varepsilon_M^{1/2}$, to those returned by `MB04BD`.

The deflation which separated the finite and infinite spectra has been produced at the end of the sec-

ond iteration of the periodic QZ algorithm, hence few more operations were applied on copies of $\widehat{C}_1$, $\widehat{A}$, $\widehat{C}_2$, and $\widehat{B}$; moreover, the orthogonal matrices used, Z_1, ..., Z_4, have a very simple structure (slightly modified identity matrices). Also, the transformed matrices, denoted with check accent, satisfy the needed transformation rules, i.e., $\check{C}_2 = Z_1^T \widehat{C}_2 Z_2$, $\check{A} = Z_3^T \widehat{A} Z_2$, $\check{C}_1 = Z_3^T \widehat{C}_1 Z_4$, $\check{B} = Z_1^T \widehat{B} Z_4$, so that the resulting formal matrix product is $\check{C}_2\check{A}^{-1}\check{C}_1\check{B}^{-1} = Z_1^T \widehat{C}_2 \widehat{A}^{-1} \widehat{C}_1 \widehat{B}^{-1} Z_1$. The maximum relative error is about $2.25 \cdot 10^{-15}$. Moreover, the 7×7 trailing submatrices of $\widehat{A}$ and $\widehat{B}$, used as coefficient matrices in the two linear systems corresponding to the finite spectrum, solved for obtaining the true matrix product using symbolic calculations, have quite small condition numbers, about $9.36 \cdot 10^3$ and 3.39, respectively.

In addition to the analysis above, the reciprocal condition numbers for eigenvalues and eigenvectors for both Test 1 and Test 2 eigenproblems, with or without balancing, have been computed. It was found that the small eigenvalues for the Test 1 data are more ill-conditioned than those for the Test 2 data. Omitting the eigenvalues larger than 10^5, including the infinite ones, the LAPACK driver `DGGEVX` without balancing returned the "small" eigenvalues and associated approximate condition information shown in Table 2 for the Test 1 data. Note that there is just one pair of complex conjugate eigenvalues instead of two. Pairing is not as needed, and the eigenvalues which should be paired do not agree well. The last eigenvalue should be paired to the third one above it (i.e., to $9.0169 \cdots 73 \cdot 10^{-6}$). There are at most eight identical significant digits in the paired eigenvalues. The complex pair and the real pair of eigenvalues mentioned above have the smallest reciprocal condition numbers (of order 10^{-15}), hence they are very sensitive. The one-norms of the matrices $\mathcal{S}$ and $\mathcal{H}$ were about $2.3842 \cdot 10^{-7}$ and 7.3736, respectively.

Using instead `DGGEVX` with balancing (permutations and scaling), returned good results, as shown in Table 3. Note that the pairing is much better than in Table 2, and the eigenvalues are closer to their required position. The accuracy increased to at most 12 identical significant digits in the paired eigenvalues. Also, the conditioning has been significantly improved for all eigenvalues. The one-norms of the balanced skew-Hamiltonian and Hamiltonian matrices were about 2.3842 and 4287.7. With scaling only, the results are similar, so we omit them.

Using the scaling factors returned by `DGGEVX` to scale the input matrices for a MEX-file based on SLICOT subroutine `MB04BD`, the following finite eigenvalues (including the "large" ones) have been obtained:

Table 2: The small eigenvalues and associated approximate condition numbers for eigenvalues, rcond(λ_i), and for eigenvectors, rcond(x_i), computed using `DGGEVX` without balancing for the Test 1 data.

λ_i	rcond(λ_i)	rcond(x_i)
$7.982789077728958 \cdot 10^2$	$4.5 \cdot 10^{-9}$	$1.4 \cdot 10^{-11}$
$-7.982789039705131 \cdot 10^2$	$4.5 \cdot 10^{-9}$	$4.6 \cdot 10^{-11}$
$-2.258073833805364 \cdot 10^1$	$5.1 \cdot 10^{-12}$	$5.9 \cdot 10^{-11}$
$2.258073628829316 \cdot 10^1$	$5.1 \cdot 10^{-12}$	$2.8 \cdot 10^{-11}$
$6.345831704138784 \cdot 10^{-6}$		
$\pm 1.654595668192905 \cdot 10^{-3}\imath$	$3.4 \cdot 10^{-15}$	$8.4 \cdot 10^{-14}$
$9.016920132336973 \cdot 10^{-6}$	$2.0 \cdot 10^{-15}$	$1.8 \cdot 10^{-18}$
-5.180515230574346	$1.0 \cdot 10^{-13}$	$1.7 \cdot 10^{-11}$
5.180514112635106	$1.0 \cdot 10^{-13}$	$1.5 \cdot 10^{-11}$
$-2.180898342048099 \cdot 10^{-5}$	$3.0 \cdot 10^{-15}$	$1.8 \cdot 10^{-18}$

Table 3: The small eigenvalues and associated approximate condition numbers for eigenvalues, rcond(λ_i), and for eigenvectors, rcond(x_i), computed using `DGGEVX` with balancing for the Test 1 data.

λ_i	rcond(λ_i)	rcond(x_i)
$7.982789076557393 \cdot 10^2$	$4.1 \cdot 10^{-3}$	$3.1 \cdot 10^{-5}$
$-7.982789076562744 \cdot 10^2$	$4.1 \cdot 10^{-3}$	$5.5 \cdot 10^{-5}$
$-2.258073819582842 \cdot 10^1$	$2.3 \cdot 10^{-2}$	$2.7 \cdot 10^{-4}$
$2.258073819581395 \cdot 10^1$	$2.3 \cdot 10^{-2}$	$1.5 \cdot 10^{-4}$
-5.180514839367283	$5.6 \cdot 10^{-3}$	$9.9 \cdot 10^{-5}$
5.180514839369767	$5.6 \cdot 10^{-3}$	$1.0 \cdot 10^{-4}$
$2.631383527274714 \cdot 10^{-9}$		
$\pm 8.649722516061533 \cdot 10^{-4}\imath$	$2.3 \cdot 10^{-8}$	$1.6 \cdot 10^{-7}$
$-2.631570224527866 \cdot 10^{-9}$		
$\pm 2.522221870167959 \cdot 10^{-5}\imath$	$1.0 \cdot 10^{-7}$	$1.6 \cdot 10^{-5}$

$$\lambda = \begin{bmatrix} 8.860295592973415 \cdot 10^6\imath \\ 1.009148542240609 \cdot 10^5\imath \\ 7.982789076561136 \cdot 10^2 \\ 2.258073819582885 \cdot 10^1 \\ 5.180514839373116 \\ 2.521340745136690 \cdot 10^{-5}\imath \\ 8.652245367690063 \cdot 10^{-4}\imath \end{bmatrix}.$$

This is the expected result for this data, but it was not returned as such by `MB04BD` with no scaling. Note that only the eigenvalues with positive real parts and, for purely imaginary eigenvalues, only those with positive imaginary parts are listed. The other half of the spectrum is obtained by symmetry.

All calculations above have been repeated for the Test 2 data, and similar results have been obtained. This problem is less sensitive. Better conditioning than for the Test 1 data has been obtained for all balancing options. Indeed, the reciprocal condition numbers for the small eigenvalues are about one order of magnitude larger than for the Test 1 data eigenproblem, hence the condition numbers are smaller. □

The analysis in Example 3 suggests that it would be useful to include in the computational solver an op-

tion for balancing the skew-Hamiltonian/Hamiltonian eigenproblem data, preserving the structure.

5 CONCLUSIONS

Numerical algorithms for the solution of common and structured eigenproblems, encountered in many control theory applications and in other domains, have been investigated. Eigenvalue computations for standard as well as formal matrix products have been considered, and their accuracy and conditioning has been discussed. Simple examples highlight the pitfalls which may appear in such numerical computations, using state-of-the-art solvers. An iterative refinement process for the periodic Schur decomposition (not yet offered by software packages) is proposed, and the improvement of the results is illustrated. Balancing the matrices or matrix pencils and the use of condition number estimates for eigenvalues are shown to be essential options in investigating the behavior of the solvers and problem sensitivity.

REFERENCES

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. SIAM, Philadelphia, third edition.

Benner, P. (1999). Computational methods for linear-quadratic optimization. *Supplemento ai Rendiconti del Circolo Matematico di Palermo, II*, (58):21–56.

Benner, P., Byers, R., Losse, P., Mehrmann, V., and Xu, H. (2007). Numerical solution of real skew-Hamiltonian/Hamiltonian eigenproblems. Technical report, Technische Universität Chemnitz, Chemnitz.

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

Benner, P. and Kressner, D. (2006). Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices II. *ACM Trans. Math. Softw.*, 32(2):352–373.

Benner, P., Kressner, D., Sima, V., and Varga, A. (2010). Die SLICOT-Toolboxen für Matlab. *at—Automatisierungstechnik*, 58(1):15–25.

Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., and Varga, A. (1999). SLICOT — A subroutine library in systems and control theory. In *Applied and Computational Control, Signals, and Circuits*, 1, ch. 10, pp. 499–539. Birkhäuser, Boston.

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

Benner, P., Sima, V., and Voigt, M. (2012b). Robust and efficient algorithms for $\mathcal{L}_\infty$-norm computations for descriptor systems. In *7th IFAC Symposium on Robust Control Design*, pp. 189–194.

Benner, P., Sima, V., and Voigt, M. (2013a). FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems. Part I: Algorithms and applications. `www.slicot.org`.

Benner, P., Sima, V., and Voigt, M. (2013b). FORTRAN 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems. Part II: Implementation and numerical results. `www.slicot.org`.

Bojanczyk, A. W., Golub, G., and Van Dooren, P. (1992). The periodic Schur decomposition: algorithms and applications. In *SPIE Conference Advanced Signal Processing Algorithms, Architectures, and Implementations III*, 1770, pp. 31–42.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, third edition.

Granat, R., Kågström, B., and Kressner, D. (2007). Computing periodic deflating subspaces associated with a specified set of eigenvalues. *BIT Numerical Mathematics*, 47(4):763–791.

Laub, A. J. (1979). A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Contr.*, AC–24(6):913–921.

MathWorks (2014). *MATLAB®: The Language of Technical Computing. R2014b*. The MathWorks, Inc.

Mehrmann, V. (1991). *The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution*. Springer-Verlag, Berlin.

Sima, V. (1996). *Algorithms for Linear-Quadratic Optimization*. Marcel Dekker, Inc., New York.

Sima, V. (2010). Structure-preserving computation of stable deflating subspaces. In *10th IFAC Workshop "Adaptation and Learning in Control and Signal Processing"*.

Sima, V. (2011a). Computational experience with structure-preserving Hamiltonian solvers in optimal control. In *8th International Conference on Informatics in Control, Automation and Robotics*, vol. 1, pp. 91–96. SCITEPRESS.

Sima, V. (2011b). Computational experience with structure-preserving Hamiltonian solvers in complex spaces. In *5th International Scientific Conference on Physics and Control*.

Sima, V., Benner, P., and Kressner, D. (2012). New SLICOT routines based on structured eigensolvers. In *2012 IEEE International Conference on Control Applications*, pp. 640–645. Omnipress.

Van Huffel, S., Sima, V., Varga, A., Hammarling, S., and Delebecque, F. (2004). High-performance numerical software for control. *IEEE Control Syst. Mag.*, 24(1):60–76.