EFFICIENT COMPUTATIONS WITH STRUCTURED MATRIX PENCILS IN AUTOMATIC CONTROL

Vasile SIMA[1,2]

[1] Coresponding Member of the Academy of Technical Sciences in Romania,
[2]National Institute for Research & Development in Informatics – Bucharest

Abstract. There is a continuous research effort worldwide to improve the reliability, efficiency, and accuracy of numerical calculations in various domains. One of the most promising research avenue is to exploit the structural properties of the mathematical problems to be solved. This paper investigates some improved algorithms for the solution of skew-Hamiltonian/Hamiltonian eigenproblems, which have applications not only in automatic control (linear-quadratic optimization and H_∞-optimization), but also for multi-body systems, and in various areas of applied mathematics, physics, and chemistry. Such eigenproblems involve matrix pencils in which one matrix is skew-Hamiltonian and the other is either Hamiltonian or skew-Hamiltonian. Of much interest is to find the eigenvalues and certain deflating subspaces, mainly those associated to the stable eigenvalues. So-called *J*-congruence transformations are used, which preserve the pencil structure. By employing unitary *J*-congruence transformations, the best accuracy of the solutions is guaranteed, since the problem numerical conditioning is preserved. The algorithms reduce the original problem, or an equivalent problem of double size, to a structured Schur form, which reveals the eigenvalues. The embedding is needed when the structured Schur form does not exist for the original problem. The pencil matrices may have either real or complex entries.
Keywords.Deflating subspaces, eigenvalue reordering, generalized eigenvalues, generalized Schur form, numerical algorithms, skew-Hamiltonian/Hamiltonian matrix pencil, software, structure-preservation.

1. INTRODUCTION

Special, structured$n \times n$ matrix pencils $A - \lambda B$, where B may be singular, are investigated in this paper. The eigenvalues of $A - \lambda B$are the complex numbers λ_i, $i = 1{:}n$ (multiplicities counted), satisfying $\det(A - \lambda B) = 0$. (The MATLAB-style notation for $i = 1, \ldots, n$ is used.)It is assumed that the determinant is not identically 0, i.e., the matrix pencil is regular. The eigenvalues also satisfy the relations $Ax_i = \lambda_i B x_i$, with $x_i \neq 0$, $i = 1{:}n$. The vector $x_i$is a right eigenvector corresponding to λ_i.Similarly, a nonzero vector y_i 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 that obtained for$B = I_n$, the identity matrix of order *n*.

Structured matrices and matrix pencils appear in many domains, including automatic control. Common structures are Hamiltonian and symplectic matrices or matrix pencils. Two important and relevant computational problems, with many applications, are discussed below. One such basic computation is the evaluation of the H_∞-/L_∞-norms, which are used to quantify the trade-off between performance and robust stability. 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 skew-Hamiltonian/Hamiltonian form [7].) Exploiting the special structure in L_∞-norm calculations is discussed, e.g., in [6,7,15] and the references therein. The performance of the developed solvers on a large set of standard or descriptor linear control systems is shown in [6,21].

One of the fundamental computation in system analysis and control systems 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, stabilization and linear-quadratic regulator problems, Kalman filtering, linear-quadratic Gaussian (H_2-)optimal control problems, computation of (sub)optimal H_∞controllers, model reduction techniques based on stochastic, positive or bounded real

LQG balancing, factorization procedures for transfer functions. Usually, the stabilizing solution is required, which can be used to stabilize the closed-loop system matrix or matrix pencil. A very important class of CARE/DARE solvers makes use of stable invariant or deflating subspaces of some structured matrices or pencils. The use of stable invariant subspaces of Hamiltonian or symplectic matrices assumes certain nonsingularity and eigenvalue dichotomy assumptions [12,14]. Finding such subspaces involves eigenvalue reordering.

Recently, structure-exploiting techniques have been investigated for solving skew-Hamiltonian/ Hamiltonian eigenproblems, see, e.g., [2,3], and the references therein. These techniques can be employed for continuous-time systems, e.g., for CARE solvers. For discrete-time systems, it is possible to preprocess the pencils by an extended Cayley transformation, which only involves matrix additions and subtractions [23], to obtain equivalent skew-Hamiltonian/Hamiltonian pencils. Finding subspaces requires to reorder the eigenvalues of a formal matrix product. The logic for such reordering is similar to that for reordering the eigenvalues of a matrix, or of a matrix pencil. Standard procedures are implemented in the LAPACK package [1]. The main difference is in the algorithm for swapping two adjacent sequences of diagonal blocks. Details are given, e.g., in [11,16,17] and the references therein. A direct algorithm for swapping the sequences of diagonal blocks is based on the solution of periodic Sylvester-like equations; such an equation can be rewritten as a single structured linear algebraic system, which is lower block-bidiagonal, plus a nonzero block in the top-right corner. This system can be efficiently solved by a numerically stable overlapping QR factorization algorithm.

Advanced structure-exploiting solvers have been incorporated in the SLICOT Library [4,5,8,9,20-22].Fortran and MATLAB software for eigenvalues and deflating subspaces have been developed, for both real and complex matrices. Versions with a factored or not factored matrix *S*are covered. Auxiliary routines for problems of even order with (quasi-)triangular structure and optimized kernels for problems of order 2, 3, or 4, called by the general solvers, are used.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 *S*are given in [19]. Additional results can be found in [17,18,20]. The results have shown that these solvers provide reliable and accurate results, and are often faster than the state-of-the-art tools.

A matrix pencil $\lambda M - N$ is *Hamiltonian* if $NJM^H = -MJN^H$, and *symplectic* if $NJN^H = MJM^H$, where $J = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix}$. Definitions for Hamiltonian and symplectic matrices follow if $= I_{2m}$; for instance, N is *Hamiltonian* if $(NJ)^H = NJ$, and *skew-Hamiltonian* if $(NJ)^H = -NJ$.In the sequel, a pencil$\lambda M - N$ will often be written in the numerically better form$\alpha M - \beta N$, with $\lambda = \alpha/\beta$ (possibly ∞).

Let$\alpha S - \beta H$be a skew-Hamiltonian/Hamiltonian pencil of order $n = 2m$, that is,$(SJ)^H = -SJ$,$(HJ)^H = HJ$. (The extended pencils associated to CAREs become skew-Hamiltonian/ Hamiltonian after an eventual additional extension with one row and column, permutation and scaling.) For some problems, including linear-quadratic optimization, S can be given in a factored form, the *skew-Hamiltonian Cholesky factorization*,defined by $S = JZ^H J^T Z$. Such a matrix *S*is said to be *J-semidefinite*.For instance, for a block-diagonal matrix *S* = diag(*E*, E^H), a factor *Z*can be written as $Z = \text{diag}(I_m , E^H)$.

Some properties of skew-Hamiltonian/Hamiltonian pencils are proven, e.g., in [3]. The *skew-Hamiltonian/Hamiltonian Schur form* of a skew-Hamiltonian/Hamiltonian pencil $\lambda M - N$is

$$JQJ^T \left(\lambda \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{11}^H \end{bmatrix} - \begin{bmatrix} N_{11} & N_{12} \\ 0 & -N_{11}^H \end{bmatrix} \right) Q^H , \qquad (1)$$

where *Q*is unitary, M_{11} and $N_{11}$are upper triangular, M_{12} is *skew-Hermitian*, and $N_{12}$is *Hermitian*, i.e.,$M_{12} = -M_{12}^H$, and$N_{12} = N_{12}^H$.This form,*J*-congruent to$\lambda M - N$,displays the pencil eigenvalues. Some pencils which lack this form can be embedded in double size pencils which always have a form (1).For a matrix, the definition above can be specialized to (skew-)Hamiltonian Schur form. Hamiltonian matrices without purely imaginary eigenvalues, and real skew-Hamiltonian matrices have Hamiltonian Schur forms.

Algorithms for solving skew-Hamiltonian/Hamiltonian eigenproblems are investigated in many papers; see, e.g., [2,3,8,9] and the references therein. Special QR and RQ decompositions are used to

reduce the matrix pencil to the Hamiltonian/Hessenberg form. Then, the periodic QZ algorithm is used to transform the matrices of a formal matrix product to periodic generalized Schur form, which reveals the eigenvalues. The formal matrix product has 4 factors if matrix *S*is given, and 6 factors, if *S*is factored. A specific algorithm is briefly discussed below.Consider a real skew-Hamiltonian/Hamiltonian pencil $\alpha S - \beta H$,

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

where D and E are skew-symmetric ($D = -D^T$, $E = -E^T$), V and W are symmetric. Theoretically, the eigenvalues for the real pencil $\alpha S - \beta H$ 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, −λ (also λ,$\bar{\lambda}$, in the second case), while complex conjugate eigenvalues appear in quadruples, λ, −λ , $\bar{\lambda}$, and −$\bar{\lambda}$. Using two orthogonal transformations, Q_1 and Q_2, the algorithm computes transformed matrices $\tilde{S}$, $\tilde{T}$, and $\tilde{H}$, so that

$$Q_1^T S J Q_1 J^T = \begin{bmatrix} \tilde{A} & \tilde{D} \\ 0 & \tilde{A}^T \end{bmatrix} := \tilde{S}, \quad J^T Q_2^T J S Q_2 = \begin{bmatrix} \tilde{B} & \tilde{F} \\ 0 & \tilde{B}^T \end{bmatrix} := \tilde{T}, \quad Q_1^T H Q_2 = \begin{bmatrix} \tilde{C}_1 & \tilde{V} \\ 0 & \tilde{C}_2^T \end{bmatrix} := \tilde{H},$$

where $\tilde{A}$, $\tilde{B}$,$\tilde{C}_1$are upper triangular,$\tilde{C}_2$ is upper quasi-triangular and $\tilde{D}$, $\tilde{F}$ are skew-symmetric. (A matrix is upper quasi-triangular if it is block upper triangular with 1×1 or 2×2 diagonal blocks.) The reduction procedure is summarized below.

Suitable sequences of Givens rotations and Householder reflections are used to annihilate the elements in *S*and*H*, such that the partially transformed matrices $\hat{A}$, $\hat{B}$, and $\hat{C}_1$ are upper triangular and $\hat{C}_2$is upper Hessenberg. The first step to do this reduces S to skew-Hamiltonian triangular form, and updates A and D in S, as well as C, V, and Win H correspondingly, producing A^1, ..., W^1, respectively. Let $S^1$and H^1 be theobtained matrices, where the two diagonal blocks of $H^1$are $C_1 = C^1$ and $C_2 = -C^1$. The initial skew Hamiltonian block upper triangular matrix T^1 is built using $B^1 = A^1$ and $F^1 = D^1$. Then, additional transformations are used to reduce the matrix 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 $\hat{A}$, $\hat{D}$, $\hat{B}$, $\hat{F}$, $\hat{C}_1$, $\hat{V}$, and$\hat{C}_2$be the obtained matrices, and $\hat{S}$, $\hat{T}$, and $\hat{H}$ the block matrices built from them, where $\hat{A}$, $\hat{B}$, and $\hat{C}_1$ are upper triangular, $\hat{D}$ and$\hat{F}$are skew-symmetric, and $\hat{C}_2$ is upper Hessenberg. (The (2,1) $m \times m$ blocks of $\hat{S}$, $\hat{T}$, and $\hat{H}$are zero.) Finally, the periodic QZ algorithm is applied to transform $\hat{C}_2$ to upper quasi-triangular form, while preserving the upper triangular form of $\hat{A}$, $\hat{B}$, and $\hat{C}_1$. Specifically, the formal matrix product$\hat{C}_2\hat{A}^{-1}\hat{C}_1\hat{B}^{-1}$is transformed without using matrix inversions. (Actually, $\hat{A}$ and $\hat{B}$ may be singular.) The eigenvalues of the pencil $\alpha S - \beta H$are the positive and negative square roots of the eigenvalues of the formal matrix product $-\hat{C}_2\hat{A}^{-1}\hat{C}_1\hat{B}^{-1}$. Finite purely imaginary eigenvalues, γι, correspond to the 1×1 diagonal blocks in the transformed product, and they cannot be perturbed off the imaginary axis if they are well separated from other such eigenvalues. An interesting example with $m = 2697$, representing a linearized gyroscopic system, is presented in [9]. The eigenvalues computed by the standard QZ algorithm are spread on both sides of the complex plane axes, the maximum absolute value of the real part being about $1.48 \cdot 10^{-3}$, while the structure-exploiting algorithm placed all eigenvalues on the imaginary axis, proving stability of the system.

2. RELIABILITY AND ACCURACY OF COMPUTED EIGENVALUES

Simple eigenvalues (i.e., with multiplicity 1) can be computed accurately, if well-conditioned, but this is not true for multiple eigenvalues, even for standard small order eigenproblems. Consider the matrix

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

This is the companion matrix of the polynomial $z^3 - z^2 - z + 1$, which has exact roots −1 and 1, the second one with multiplicity 2. Standard eigensolvers, like those included in LAPACK [1], are not guaranteed to return accurate values for eigenvalues with multiplicities. Indeed, the MATLAB R2014b function eig, based on LAPACK routines, returns, in double precision arithmetic

$$\lambda = [-0.9999999999999999, 0.9999999948725736, 1.000000005127427].$$

The (absolute and relative) errors of the eigenvalues corresponding to 1 are slightly smaller than the square root of the relative machine accuracy, ε_M, hence almost half of the accuracy has been lost. Calling the function eig with no optional argument will compute 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 $D^{-1}AD$, but for this example $D = I_3$. Optionally, the LAPACK driver DGEESX also returns the reciprocal condition numbers for eigenvalues and eigenvectors, which in this case are

$$(\text{rcond}(\lambda_i),\ i = 1:3) = [0.94281,\ 8.3731\cdot10^{-9},\ 8.3731\cdot10^{-9}],$$
$$(\text{rcond}(x_i),\ i = 1:3) = [1.2679,\ 1.0255\cdot10^{-8},\ 1.0255\cdot10^{-8}],$$

respectively (rounded to 5 significant digits). This means that perturbations in the elements of A can be amplified by $1/\text{rcond}(\lambda_i)$ in the computed eigenvalue λ_i. More generally, an approximate error bound on the chordal distance between the computed generalized eigenvalue $\hat{\lambda}_i$ of a matrix pencil $\alpha A - \beta B$ and the corresponding exact eigenvalue λ_i is bounded as follows,

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

where the norms are for the matrices after balancing, if applied. Balancing involves row and column permutations (to isolate the eigenvalues available by inspection in the leading and/or trailing part), as well as scaling.The chordal distance between two points $(\hat{\alpha},\hat{\beta})$ and (α, β) is defined by

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

An approximate error bound for the acute angle between the computed left or right eigenvector $\hat{y}_i$ or $\hat{x}_i$, and the true eigenvector, is given by $\varepsilon_M \|[\|A\|, \|B\|]\| / \text{rcond}(x_i)$. See [1] for further details.

Consider now the companion matrix of the polynomial $z^4 - 2z^2 + 1$, which has exact roots 1 and −1, eachwith multiplicity 2. The MATLAB function eig, based on LAPACK routines, returns

$$\lambda = [-0.9999999999999998, -0.9999999999999998, 0.9999999999999994 \pm 1.553410702992418\cdot10^{-8}\iota].$$

The double root in 1 became a pair of complex conjugate values. The (absolute and relative) errors of those eigenvalues aer very close to the square root of ε_M. The diagonal scaling matrix used is D = diag(2, 1, 1, 1). The LAPACK driver DGEESX also returns

$$(\text{rcond}(\lambda_i),\ i = 1:4) = [2.3277\cdot10^{-17},\ 6.9830\cdot10^{-17},\ 2.6055\cdot10^{-8},\ 2.6055\cdot10^{-8}],$$
$$(\text{rcond}(x_i),\ i = 1:4) = [2.6211\cdot10^{-16},\ 2.6211\cdot10^{-16},\ 2.2121\cdot10^{-8},\ 2.2121\cdot10^{-8}],$$

respectively (rounded to 5 significant digits). Without the balancing option (e.g., using the MATLAB command eig(A,'nobalance'), the computed eigenvalues are

$$\lambda = [-1.000000000000001 \pm 1.114121661500053\cdot10^{-8}\iota,\ 1.000000007316396,\ 0.9999999926836054].$$

In this case, all eigenvalues have errors of order $\varepsilon_M^{1/2}$. All reciprocal condition numbers for these eigenvalues and associated eigenvectors are also of order $\varepsilon_M^{1/2}$. Clearly, the computed eigenvalues and their accuracy and reciprocal condition numbers are influenced by balancing.

Often, badly scaled matrices or matrix pencils have eigenvalues with magnitudes covering a large interval. A singular matrix pencil case will be discussed below. Omitting infinite eigenvalues, if

the ratio between the largest magnitude and the smallest one exceeds, let's say, 10^{10}, the smallest eigenvalues may make the solver behave similarly to the case of multiple small eigenvalues. 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. Specifically, before those changes, the two smallest eigenvalues were about $2.52 \cdot 10^{-5}$ι and $8.66 \cdot 10^{-4}$ι, and after the changes, they became $2.30 \cdot 10^{-3}$ιand $1.83 \cdot 10^{-5}$. The final effect was obtaining a wrong H_∞-norm in the second case. The author performed a detailed investigation of the associated numerical issue. The test case with changes in scaling will be referred to as Test 1 below, while the test case without changes will be referred to as Test 2. Test 1 example has 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. Test 2 example has the same structure, but different A and C.

Initially, an error was suspected in the SLICOT Library [5,8,9,20,21] routines used, i.e., in 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. The alternative was to suppose that the problem is sensitive to small changes in the data. Actually, the two smallest purely imaginary eigenvalues for the Test 2 data example are quite close, or the algorithm implemented in MB04BDis not guaranteed to work properly for problems with multiple eigenvalues on the imaginary axis. Changes in the data caused one purely imaginary eigenvalue to slightly move off the imaginary axis, and since it was already close to 0, it became real.

The following remarks about the MB04BD behavior support the above statement.

1. The transformed matrices, computed from the Test 1 data, 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 returned eigenvalues have been computed using matrices derived from the returned ones. Orthogonal transformations are used in the calculations.

2. The required structure and the needed relationships with the initial skew-Hamiltonian/Hamiltonian pencil are satisfied by the transformed matrices $\hat{A}$, $\hat{D}$, $\hat{B}$, $\hat{F}$, $\hat{C}_1$, $\hat{V}$, and $\hat{C}_2$, obtained in MB04BDjust before calling the periodic QZ algorithm.

3. The finite and "positive" eigenvalues, 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 separatedthe finite and infinite spectra has been produced at the end of the second iteration of the periodic QZ algorithm, hence few more operations were applied on copies of $\hat{C}_1$, $\hat{A}$, $\hat{C}_2$and $\hat{B}$,and the orthogonal matrices used, Z_1, ..., Z_4, have a very simple structure (slightly modifiedidentity matrices). Also, the transformed matrices, denoted with check accent, satisfy theneeded transformation rules, i.e.,$\check{C}_2 = Z_1^T \hat{C}_2 Z_1$,$\check{A} = Z_3^T \hat{A} Z_2$,$\check{C}_1 = Z_3^T \hat{C}_1 Z_4$,$\check{B} = Z_1^T \hat{B} Z_4$,so that the resulted formal matrix product is $\check{C}_2 \check{A}^{-1} \check{C}_1 \check{B}^{-1} = Z_1^T \hat{C}_2 \hat{A}^{-1} \hat{C}_1 \hat{B}^{-1} Z_1$. The maximum relative error is about $2.25 \cdot 10^{-15}$. Moreover, the 7 × 7 trailing submatrices of $\hat{A}$ and$\hat{B}$, used as coefficient matrices in the two linear systems corresponding to the finite spectrum, solved for obtaining the matrix product using symbolic calculations, have quite small condition numbers, about $9.36 \cdot 10^3$ and 3.39, respectively. Note that using the standard MATLAB function eig on a matrix directly derived from $\check{C}_2$, $\check{A}$, $\check{C}_1$, and $\check{B}$ returned two large,complex conjugate eigenvalues instead of the two smallest eigenvalues.

4. Repeating the calculations on the Test 2 data example, the same conclusion has been obtained: the eigenvalues agreed in position and within a relative error of about $\varepsilon_M^{1/2}$to those returned by MB04BD.

In addition to the analysis above, the reciprocal condition numbers for eigenvalues and eigenvectors for both Test 1 and Test 2 data problems, with or without balancing, have been computed. Omitting the eigenvalues larger than 10^5, including the infinite ones, the LAPACK driver DGGEVX with balancing (permutations and scaling)returned the "small" eigenvalues and associated approximate condition information shown in Table 1for the Test 1 data. Note that pairing can be guessed, and the eigenvalues are close to their required position. There are at most 12 identical

significant digits in the paired eigenvalues. Also, the conditioning is acceptable, except for the complex conjugate eigenvalues. The one-norms of the balanced skew-Hamiltonian and Hamiltonian matrices were about 2.3842 and 4287.7.

Table 1. **The small eigenvalues and associated approximate condition numbers for eigenvalues, rcond($\hat{\lambda}_i$), and for eigenvectors, rcond(x_i), computed using DGGEVX with balancingfor Test 1 data.**

$\hat{\lambda}_i$	rcond($\hat{\lambda}_i$)	rcond(x_i)
$7.982789076557393\cdot10^{2}$	$4.0562\cdot10^{-3}$	$3.1494\cdot10^{-5}$
$-7.982789076562744\cdot10^{2}$	$4.0562\cdot10^{-3}$	$5.5267\cdot10^{-5}$
$-2.258073819582842\cdot10^{1}$	$2.2679\cdot10^{-2}$	$2.7466\cdot10^{-4}$
$2.258073819581395\cdot10^{1}$	$2.2679\cdot10^{-2}$	$1.5415\cdot10^{-4}$
-5.180514839367283	$5.5576\cdot10^{-3}$	$9.8965\cdot10^{-5}$
5.180514839369767	$5.5576\cdot10^{-3}$	$1.0228\cdot10^{-4}$
$2.631383527274714\cdot10^{-9} \pm 8.649722516061533\cdot10^{-4}\,\imath$	$2.2559\cdot10^{-8}$	$1.6116\cdot10^{-7}$
$-2.631570224527866\cdot10^{-9} \pm 2.522221870167959\cdot10^{-5}\,\imath$	$1.0379\cdot10^{-7}$	$1.5965\cdot10^{-5}$

With scaling only, the results are similar, but the pairs of complex conjugate eigenvalues have just three significant digits in agreement, instead of four. Without balancing, there is just one pair of complex conjugate eigenvalues (instead of two), $6.3458\cdot10^{-6} \pm 1.6546\cdot10^{-3}\imath$, and the smallest eigenvalues, which should be paired, are $9.0169\cdot10^{-6}$ and $-2.1810\cdot10^{-5}$. All these four eigenvalues have the smallest reciprocal condition numbers (of order 10^{-15}), hence they are very sensitive. The other "small" eigenvalues have values less than $4.5\cdot10^{-9}$ for rcond($\hat{\lambda}_i$). Pairing is not as needed. There are at most eight identical significant digits in the paired eigenvalues. The one-norms of the matrices *S* and *H* were about $2.3842\cdot10^{-7}$ and 7.3736, respectively. Using the LAPACK subroutine DGGEVX with permutations only (no scaling), all "small" eigenvalues resulted real; they coincide with the result delivered by the MATLAB function eig. 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:

$$\hat{\lambda} = [\ 8.860295592973415\cdot10^{6}\,\imath,\ 1.009148542240609\cdot10^{5}\,\imath,\ 7.982789076561136\cdot10^{2},$$
$$2.258073819582885\cdot10^{1},\ 5.180514839373116,\ 2.521340745136690\cdot10^{-5}\,\imath,\ 8.652245367690063\cdot10^{-4}\,\imath\ ]\ .$$

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 bysymmetry.

Table 2. **The small eigenvalues and associated approximate condition numbers for eigenvalues, rcond($\hat{\lambda}_i$), and for eigenvectors, rcond(x_i), computed using DGGEVX with balancing for Test 2 data.**

$\hat{\lambda}_i$	rcond($\hat{\lambda}_i$)	rcond(x_i)
$7.982789076561603\cdot10^{2}$	$2.0812\cdot10^{-2}$	$7.9167\cdot10^{-5}$
$-7.982789076552374\cdot10^{2}$	$2.0812\cdot10^{-2}$	$1.7633\cdot10^{-4}$
$-2.258073819582906\cdot10^{1}$	$2.8346\cdot10^{-1}$	$1.9272\cdot10^{-3}$
$2.258073819582878\cdot10^{1}$	$2.8346\cdot10^{-1}$	$1.5797\cdot10^{-3}$
-5.180514839373380	$6.2260\cdot10^{-2}$	$2.8105\cdot10^{-3}$
5.180514839372754	$6.2260\cdot10^{-2}$	$2.7842\cdot10^{-3}$
$5.734066638153054\cdot10^{-10} \pm 8.654639002700035\cdot10^{-4}\,\imath$	$6.1486\cdot10^{-7}$	$1.3933\cdot10^{-6}$
$-5.730728370968683\cdot10^{-10} \pm 2.520095998934891\cdot10^{-5}\,\imath$	$1.3609\cdot10^{-6}$	$4.3596\cdot10^{-5}$

All calculations above have been repeated for the Test 2 data, and similar results have been obtained (see Table 2). This problem is less sensitive. Indeed, even using the command eig(*H*, *S*), for the corresponding arguments defined by *H* and *S*, one pair of eigenvalues moved from the real axis (but not onto the imaginary axis, since the real and imaginary parts had values of order 10^{-4}). (On the other

hand, all small eigenvalues for the Test 1 data were real when computed by eig.) Better conditioning than for the Test 1 data has been obtained for all balancing options. For instance, full balancing returned the values in Table 2.Therefore, the reciprocal condition numbers for the small eigenvalues are about one order of magnitude larger than for the Test 1 data problem, hence the condition numbers are smaller. The conclusion of these tests is that it is necessary to include in the computational solver an option for balancing the skew-Hamiltonian/Hamiltonian eigenproblem data, preserving the structure.

3. RELIABILITY AND ACCURACY OF COMPUTED EIGENVALUES OF MATRIX PRODUCTS

Numerical difficulties appear also for periodic Schur decomposition [10] of a standard (not formal)matrix product. This may happen even for 2 × 2 matrices. Usually, the computed eigenvalues arequite accurate, but the transformation of the factor which has been initially reduced to an upperHessenberg form (all other factors are reduced to upper triangular form) may not be close to the true realSchur form.In a series of tests, several products of 2 × 2 real matrices of maximum 32 factors havebeen used. Let p be the number of factors A_i, $i = 1:p$, used in a test. Allvalues of p from 2 to 32 have been tried. The periodic Schur algorithm delivered the eigenvalues,the transformation matrices, Z_i, $i = 1:p$, and the transformed matrices,

$$\tilde{A}_i = Z_i^T A_i Z_{\mathrm{mod}(i,p)+1}, \quad i = 1:p. \tag{3}$$

The periodic Schur solver implementation in the SLICOT Library returns $\tilde{A}_1$ in real Schur form, inparticular, upper triangular if the eigenvalues are both real, and all other factors $\tilde{A}_i$, $i > 1$, in uppertriangular form. (The analyzed examples had real eigenvalues only.) The computed eigenvalueshave been compared to those obtained using MATLAB function eigapplied 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, $\Pi_{i=1}^{p} \tilde{A}_i$, and the symbolic product of symbolic transformed factors, $\Pi_{i=1}^{p} \tilde{\mathcal{A}}_i$. The eigenvalues of all these products agreed well. The difference between the computed periodic matrix and thetransformed original periodic matrix, obtained using (3), has also been computed. For severalvalues of p, the (2,1) element of the factor $\tilde{A}_1$ from (3) differed significantly from its value returnedby the solver. All other elements of all factors agreed very well to the returned values. Table 3gives the absolute value of the difference for various $p$and one series of factor matrices.

Table 3. **Absolute difference between the returned (2,1) element of $\tilde{A}_1$ and its valuecomputed using (3)**

p	(2,1)-element error	p	(2,1)-element error	p	(2,1)-element error	p	(2,1)-element error
10	$7.78 \cdot 10^{-11}$	19	$1.52 \cdot 10^{-4}$	26	$3.65 \cdot 10^{-2}$	30	0.7163
11	$7.45 \cdot 10^{-10}$	22	$1.85 \cdot 10^{-3}$	28	0.7068	31	0.7205
16	$8.66 \cdot 10^{-7}$	25	$4.98 \cdot 10^{-2}$	29	0.7743	32	0.5688

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. Using balancing improved the conditioning by one or two orders of magnitude. However, the conditioning for the product of the transformed factors has been several (e.g., 3) orders of magnitude worse.

The eigenvalues of the matrix product most often correspond accurately to those of $\Pi_{i=1}^{p} A_i$, but not to those of $\Pi_{i=1}^{p} \tilde{A}_i$. But the results can be easily improved, by computing the residual of the transformed first factor, $Z_1^T A_1 Z_2 - \tilde{A}_1$,adding it to the computed transformed first factor – essentially replacing the returned transformed first factor (which has the (2,1) element set to zero if two real eigenvalues are found) by $Z_1^T A_1 Z_2$ – and applying the periodic Schur algorithm again. Indeed, if the (2,1) element so computed is large in comparison to relative machine accuracy, ε_M, for problemswith real eigenvalues, then a new sweep of the periodic Schur algorithm usually reduces the new (2,1) element to a value close to ε_M, and returns even possibly more accurate eigenvalues, closer to those computed symbolically. If the transformations used in the additional sweep are denoted by Y_i, then

$Z_i Y_i$, $i = 1:p$, transform the original problem to one having the right structure and accurate eigenvalues. Specifically, for the values p in Table 3, the absolute difference after the additional sweep was of the order 10^{-15} or less. The results for other series of tests have been similar, but with different values of $p$and of theabsolute difference.

Actually, the numerical difficulties described above can be encountered even for a product of two2×2matrices, as illustrated by the example below. Let

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

be exact representations of the two matrices. Note that the condition numbers for A_1, A_2, and$A_1 A_2$ 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 $A_1 A_2$ computed in double precisionarithmetic and that computed using symbolic calculations, via the MATLAB commanddouble(sym(A_1)*sym(A_2)).The eigenvalues obtained using symbolic calculations, after conversion to double precision numbers, are

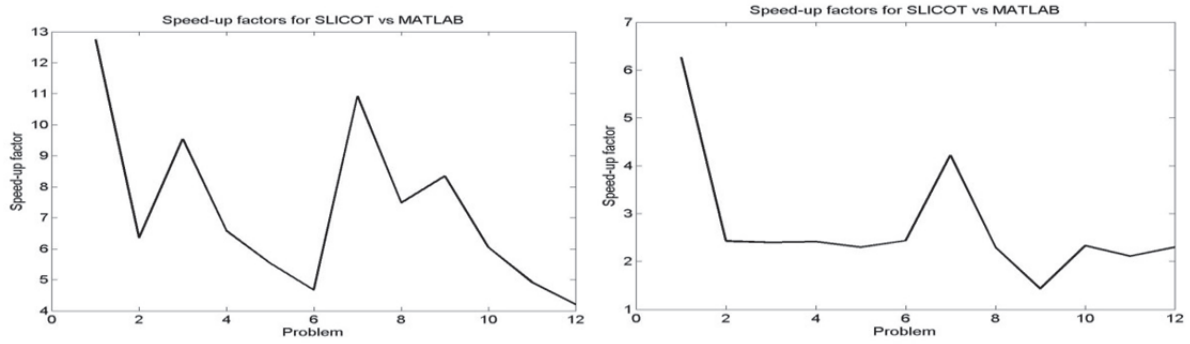
$$\lambda = [2.031200536560380 \cdot 10^{-9};\ 1.172582399979688 \cdot 10^{2}],$$

and the SLICOT periodic Schur eigensolver returns$\hat{\lambda}$= $[2.031200097007968 \cdot 10^{-9};\ 1.172582399979688 \cdot 10^{2}]$. The eigenvalues of the product of the transformed matrices, obtained as products of the corresponding diagonal elements, $\tilde{A}_1(i,i)\tilde{A}_2(i,i)$, $i = 1:2$, are$\tilde{\lambda}$ = $[2.031199877082891 \cdot 10^{-9};\ 1.172582399952876 \cdot 10^{2}]$.The element-wise absolute errors between $\hat{\lambda}$and λ are about $4.40 \cdot 10^{-16}$ and $4.26 \cdot 10^{-14}$, and the element-wise relative errors are about $2.16 \cdot 10^{-7}$ and $3.64 \cdot 10^{-16}$. On the other hand, the element-wise absolute errors between $\tilde{\lambda}$ and $\lambda$are about $6.59 \cdot 10^{-16}$ and $2.68 \cdot 10^{-9}$, and the element-wise relative errors are about $3.25 \cdot 10^{-7}$ and $2.29 \cdot 10^{-11}$. Clearly, although obtained using orthogonal transformations only, the transformed matrices $\tilde{A}_i$,$i = 1:2$, are inaccurate, and therefore $\tilde{\lambda}_i$ are less accurate than $\hat{\lambda}_i$. The inaccuracy of $\tilde{A}_i$ is due tothe ill-conditioning of the matrix product $A_1 A_2$, and the way the periodic Schur algorithm works.Although this algorithm does not compute the matrix product itself, it computes some products ofelements on the diagonals (and use sums with other products involving subdiagonal or superdiagonal elements), after the preliminary reduction to the Hessenberg-triangular form. To check the results, the differences$R_i$between the matrices $\tilde{A}_i$, $i = 1:2$, returned by the periodic Schur eigensolver, and those obtained using (3) are computed. The largest residual, of order 10^{-10}, appears in the (2,1) element of R_1. While numerically its magnitude should be of the order of ε_M, in the best case, the ill-conditioningof the matrix product has determined its increase by six orders of magnitude. By applying one step of iterative correction, the new residuals have all elements equal to0, except for the (2,1) element of the new residual R_2, which has a value about $4.25 \cdot 10^{-16}$. The eigenvalues of the matrix product after the additional sweep are$\hat{\lambda}^1$ = $[2.031200536459228 \cdot 10^{-9};\ 1.172582399979688 \cdot 10^{2}]$. These values coincide to the eigenvalues $\tilde{\lambda}^1$of the product $\tilde{A}_1^1 \tilde{A}_2^1$of the matrices returned after the additional sweep. Now, the absolute errors between $\hat{\lambda}^1$ and λ are about $1.01 \cdot 10^{-19}$ and $4.42 \cdot 10^{-14}$, and the element-wise relative errors are about $4.98 \cdot 10^{-11}$ and $1.21 \cdot 10^{-16}$, hence there is an improvement of 4 and 5orders of magnitude compared to $\tilde{\lambda}$.If $\tilde{A}_2$ is also updated as $\tilde{A}_1$, there is no noticeable improvement.

4. EFFICIENCY OF STRUCTURE-EXPLOITING SKEW-HAMILTONIAN/ HAMILTONIAN SOLVERS

The efficiency of structure-exploiting skew-Hamiltonian/Hamiltonian solvershas been investigated in several papers [6-9, 17, 19]. Therefore, few results for some systems from the COMPl$_e$ib collection [13] are presented here. Figure 1 (a) plotsthe speed-up factors obtained by the structured L_∞-norm solver in comparison with MATLAB function normfor the 12 examples CM1− CM6 and CM1_IS − CM6_IS. These are standard continuous-time systems, with $E = I_n$. The examples in each group havesystem order $n$equal to 20, 60, 120, 240, 480, and 960, respectively. The speed-up factor is over 4.2. Figure 1 (b) shows similarly the speed-up factor for the corresponding

modified, descriptor examples, with $E(n,n) = 0$. The speed-up factor is over 2.1, except for CM4_IS, for which it is about 1.43.



(a) (b)

Fig. 1. The ratios between the CPU times needed by the MATLAB function norm and SLICOT structure-exploiting solver for COMPl$_e$ib CM1− CM6 and CM1_IS − CM6_IS examples: (a) standard systems ($E = I_n$); (b) descriptor systems($E(n,n) = 0$).

Acknowledgments

Acknowledgments are addressed to P. Benner (Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg), M. Voigt (Technical University Berlin), and D. Kressner (École Polytechnique Fédérale de Lausanne).

REFERENCES

[1] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. LAPACK Users' Guide: Third Edition. Software · Environments · Tools. SIAM, Philadelphia, 1999.

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

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

[4] Benner, P., Kressner, D., Sima, V., and Varga, A. Die SLICOT-Toolboxen für Matlab (The SLICOT toolboxes for Matlab). *at-Automatisierungstechnik*, 58(1), 15–25, 2010.

[5] Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., Varga, A. *SLICOT — A subroutine library in systems and control theory*. In: Aplied and Computational Control, Signals, and Circuits, Datta, B.N. (Ed.), Vol. 1, chapter 10, p. 499–539, Birkhäuser, ISBN 0-8176-3954-2, 3-7643-3954-3, Boston, United States, 1999.

[6] Benner, P., Sima, V., and Voigt, M. Robust and efficient algorithms for L_∞-norm computations for descriptor systems. *Preprints of the 7th IFAC Symposium on Robust Control Design* (ROCOND'12). The International Federation of Automatic Control, Aalborg, Denmark, June 20–22, 2012, pp.189–194.

[7] Benner, P., Sima, V., and Voigt, M. L_∞-norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans. Automat. Contr.*, AC–57(1), 233–238, 2012.

[8] Benner, P., Sima, V., and Voigt, M. FORTRAN 77 Subroutines for the Solution of Skew-Hamiltonian/Hamiltonian Eigenproblems− Part I: Algorithms and Applications, SLICOT Working Note 2013-1, Aug. 2013. www.slicot.org/REPORTS/SLWN2013-1.pdf

[9] Benner, P., Sima, V., and Voigt, M. FORTRAN 77 Subroutines for the Solution of Skew-Hamiltonian/Hamiltonian Eigenproblems− Part II: Implementation and Numerical Results, SLICOT Working Note 2013-2, Aug. 2013. www.slicot.org/REPORTS/SLWN2013-2.pdf

[10] Bojanczyk, A.W., Golub, G., and Van Dooren, P. The periodic Schur decomposition: Algorithms and applications. In F.T. Luk (ed.), *Proc. of the SPIE Conference*, volume 1770, 31-42, 1992.

[11] Granat, R., Kågström, B., and Kressner, D. Computing periodic deflating subspaces associated with a specified set of eigenvalues. *BIT Numerical Mathematics*, 47, 763-791, 2007.

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

[13] Leibfritz, F., and Lipinski, W. *COMPleib* 1.0 – User manual and quick reference. Department of Mathematics, University of Trier, D–54286 Trier, Germany, Tech. Rep., 2004.

[14] Sima, V. Algorithms for Linear-Quadratic Optimization, volume 200 of Pure and Applied Mathematics: A Series of Monographs and Textbooks. Marcel Dekker, Inc., New York, 1996.
[15] Sima, V. Efficient algorithm for L_∞-norm calculations. *Preprints of 5th IFAC Symposium on Robust Control Design (ROCOND'06)*, Toulouse, France, July 5–7, 2006.
[16] Sima, V. Structure-preserving computation of stable deflating subspaces. *Proceedings of the 10th IFAC Workshop "Adaptation and Learning in Control and Signal Processing" (ALCOSP 2010)*, Antalya,Turkey, 26–28 August 2010. IFAC-PapersOnLine, Vol. 10, Part 1, http://www.ifac-papersonline.net/Detailed/46793.html, 6 pages.
[17] Sima, V. Computational experience with structure-preserving Hamiltonian solvers in optimal control. *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics(ICINCO 2011)*, Noordwijkerhout, The Netherlands, 28–31 July, 2011, Vol. 1, pp. 91–96.SciTePress, Portugal.
[18] Sima, V. Computational experience with structure-preserving Hamiltonian solvers in complex spaces.*Proceedings of the PHYSCON 2011 — 5th International Scientific Conference on Physics and Control*,Leòn, Spain, September, 5–8, 2011, 6 pages.
[19] Sima, V. Performance of skew-Hamiltonian/Hamiltonian solvers for automatic control computations. *Proceedings of VI-th International Conference: Days of the Academy of Technical Sciences in Romania*, Timişoara, Romania, 22-23 September 2011. Part I, pp. 130–137.
[20] Sima, V. Recent developments in SLICOT Library and related tools. M. Voicu (Ed.), *Proceedings of the 15th International Conference on System Theory, Control and Computing, Joint Conference SINTES 15, SACCS 11, SIMSIS 15 (ICSTCC 2011)*, Sinaia, Romania, October 14–16, 2011,pp. 564–569.
[21] Sima, V., Benner, P., and Kressner, D. New SLICOT routines based on structured eigensolvers. *2012 IEEE International Conference on Control Applications (CCA)*, Part of 2012 IEEE Multi-conference on Systems and Control, October 3-5, 2012, Dubrovnik, Kroatia, pp. 640–645, Omnipress.
[22] Van Huffel, S., Sima, V., Varga, A., Hammarling, S., and Delebecque, F. High performance numerical software for control. *IEEE Control Systems Magazine*, 24(1), 60–76, 2004.
[23] Xu, H. On equivalence of pencils from discrete-time and continuous-time control. *Lin. Alg. Appl.*, 414, 97–124, 2006.

CALCULE EFICIENTE CU FASCICOLE DE MATRICE STRUCTURATE ÎN AUTOMATICĂ

Vasile SIMA[1,2]

[1]Membru corespondent al Academiei de Ştiinţe Tehnice din România
[2]Institutul Naţional de Cercetare-Dezvoltare în Informatică

Rezumat: În întreaga lume există un continuu efort de cercetare pentru a îmbunătăţi fiabilitatea, eficienţa şi precizia calculelor numerice din diferite domenii. Una dintre cele mai promiţătoare direcţii de cercetare este exploatarea proprietăţilor structurale ale problemelor matematice de rezolvat. Această lucrare investighează algoritmi îmbunătăţiţi pentru soluţionarea problemelor de valori proprii anti-Hamiltoniene/Hamiltoniene, care au aplicaţii nu numai în automatică (optimizare liniar-pătratică şi optimizare H_∞), ci, de asemenea, pentru sisteme de corpuri mecanice interconectate, cât şi în variate domenii ale matematicilor aplicate, fizicii şi chimiei. Astfel de probleme de valori proprii implică fascicole de matrice în care una dintre matrice este anti-Hamiltoniană iar cealaltă este fie Hamiltoniană fie anti-Hamiltoniană. De mare interes este găsirea valorilor proprii şi a anumitor subspaţii de deflaţie, în principal a celor asociate valorilor proprii stabile. Sunt utilizate aşa-numitele transformări *J*-congruente, care păstrează structura fascicolului. Prin folosirea unor transformări *J*-congruente unitare, este garantată obţinerea celei mai bune precizii a soluţiilor, deoarece se păstrează conditionarea numerică a problemei. Algoritmii reduc problema originală, sau o problemă echivalentă de dimensiune dublă, la o formă Schur structurată, care expune direct valorile proprii. Extensia dimensională este necesară în cazul în care forma Schur structurată nu există pentru problema originală. Matricele fascicolului pot avea elemente fie reale, fie complexe. Se foloseşte o tehnică bloc specială pentru mărirea performanţei în cazul problemelor de mari dimensiuni.