Improving the Convergence of the Periodic QZ Algorithm

Vasile Sima[1][a] and Pascal Gahinet[2][b]

[1] *Modelling, Simulation, Optimization Department, National Institute for Research & Development in Informatics,*
Bd. Mareşal Averescu, Nr. 8–10, Bucharest, Romania
[2] *MathWorks, 3 Apple Hill Drive, Natick, MA, U.S.A.*

Keywords: Eigenvalue Problem, Hamiltonian Matrix, Numerical Methods, Optimal Control.

Abstract: The periodic QZ algorithm involved in the structure-preserving skew-Hamiltonian/Hamiltonian algorithm is investigated. These are key algorithms for many applications in diverse theoretical and practical domains such as periodic systems, (robust) optimal control, and characterization of dynamical systems. Although in use for several years, few examples of skew-Hamiltonian/Hamiltonian eigenproblems have been discovered for which the periodic QZ algorithm either did not converge or required too many iterations to reach the solution. This paper investigates this rare bad convergence behavior and proposes some modifications of the periodic QZ and skew-Hamiltonian/Hamiltonian solvers to avoid nonconvergence failures and improve the convergence speed. The results obtained on a generated set of one million skew-Hamiltonian/Hamiltonian eigenproblems of order 80 show no failures and a significant reduction (sometimes of over 240 times) of the number of iterations.

1 INTRODUCTION

A special, structured eigenvalue problem of much theoretical and practical interest is defined by $\lambda S - H$, where S is a *skew-Hamiltonian* matrix, H is a *Hamiltonian* matrix, and $\lambda \in \mathbf{C}$. In the real case, considered in this paper, often used definitions for such matrices are $(SJ)^T = -SJ$ and $(HJ)^T = HJ$, where $J := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$, or

$$S := \begin{bmatrix} A & D \\ E & A^T \end{bmatrix}, \quad H := \begin{bmatrix} C & V \\ W & -C^T \end{bmatrix}, \qquad (1)$$

where $A, D, E, C, V, W \in \mathbf{R}^{n\times n}$, D and E are skew-symmetric ($D = -D^T$, $E = -E^T$), V and W are symmetric ($V = V^T$, $W = W^T$), and I_n is the identity matrix of order n.

The matrix pencil $\lambda S - H$ (or $H - \lambda S$), defined above, is *skew-Hamiltonian/Hamiltonian* (sHH). These pencils have spectra that are symmetric with respect to both the real and imaginary axes. But this symmetry cannot be preserved by general eigensolvers such as those in the LAPACK Library (Anderson et al., 1999) or the `eig` command from MATLAB® (MathWorks, 2019). For some applications, including the computation of $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norms of linear time-invariant multivariable systems, and solution of algebraic matrix Riccati equations, it is very important to guarantee the symmetry of the returned spectra. Ensuring this is possible using structured eigensolvers such as those implemented in the SLICOT Library (Benner et al., 1999) and available in (MathWorks, 2012) and subsequent releases. The related theory is exposed, for example, in (Benner et al., 2002; Benner et al., 2007; Kressner, 2005), and the basic algorithms are described in (Benner et al., 2016) and the references therein. The use of these algorithms for the $\mathcal{L}_\infty$-norms computation and linear-quadratic and $\mathcal{H}_\infty$ optimization is presented in (Benner et al., 2012a; Benner et al., 2012b; Benner et al., 2016). The sHH solver is core to the calculation of the H_∞-norm, based on (Bruinsma and Steinbuch, 1990), and to nonsmooth minimization of the $\mathcal{L}_\infty$-norm, which is central to the fixed-order controller tuning—systune—(Apkarian et al., 2014). The sHH solver is applied in (Xia et al., 2017) for computing the R-index of quadratic sector bounds, which offer a characterization of, for instance, dynamical systems behavior, including passivity, dissipativity, and input/output gain. The SHH solver also comes into play in the new "safe" approach in robust control for finding μ upper bounds for real uncertainty.

The sHH solver uses the *periodic QZ algorithm*, sometimes called *periodic QR algorithm* (Van Loan, 1975; Bojanczyk et al., 1992; Sreedhar and

[a] https://orcid.org/0000-0003-1445-345X
[b] https://orcid.org/0000-0002-9485-5127

Sima, V. and Gahinet, P.
Improving the Convergence of the Periodic QZ Algorithm.
DOI: 10.5220/0007876902610268
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 261-268
ISBN: 978-989-758-380-3

Van Dooren, 1994). This algorithm is also independently implemented in many applications, for example, for periodic linear systems, or for *k*-cyclic matrices and pencils encountered in the investigation of Markov chains and the solution of two-point boundary value problems (Bojanczyk et al., 1992).

Although the sHH solver has been in use for several years and has been exhaustively tested, some examples have been recently discovered for which the underlying periodic QZ solver (pQZ) fails to converge. Such cases are rare events. However, a failure is very undesirable for applications requiring the full eigenspectrum, since, in the best situation, only part of the eigenvalues can be obtained. Therefore, an effort has been undertaken to investigate the reason for failures and find a correction. Nonconvergence may appear because of a too tight tolerance and increasing it may correct the behavior. However, this may result in a lower accuracy of the computed eigenvalues.

This paper investigates the pQZ solver failures in the sHH context and describes the currently adopted solution to avoid them. Section 2 presents some details about the algorithms needed for further discussion. Section 3 analyzes a specific failure case. Section 4 proposes the solution for forcing convergence and discusses the obtained results, which illustrate a much faster convergence than for the previous version of the solvers. Section 5 summarizes the conclusions.

2 BASIC THEORY AND UNDERLYING ALGORITHMS

The skew-Hamiltonian/Hamiltonian structure is preserved under *J-congruence transformations*, defined as $\lambda\widetilde{S} - \widetilde{H} := JP^T J^T(\lambda S - H)P$, where P is a nonsingular matrix. The pencils $\lambda\widetilde{S} - \widetilde{H}$ and $\lambda S - H$ are *equivalent*, that is, they have the same spectrum. For numerical reasons, P is chosen to be orthogonal so that the eigenproblem conditioning is also preserved. Such J-congruence transformations with orthogonal P are used to reduce the pencil (1) to a condensed form, which reveals its eigenvalues. A desirable condensed form is the skew-Hamiltonian/Hamiltonian Schur form, also called the *structured Schur form*,

$$\lambda\widetilde{S} - \widetilde{H} = \lambda \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix} - \begin{bmatrix} H_{11} & H_{12} \\ 0 & -H_{11}^T \end{bmatrix}, \quad (2)$$

where $S_{11}, H_{11} \in \mathbf{R}^{n\times n}$, with S_{11} upper triangular and H_{11} in *real Schur form*. Since the structured Schur form does not exist in general, the theory makes use of an embedding of $\lambda S - H$ into an sHH matrix pencil of double size. Briefly speaking, the sHH algorithm for computing the eigenvalues of $\lambda S - H$ proceeds as follows (see (Benner et al., 2013) for more details):

1. Reduce S to *skew-Hamiltonian triangular form*, using an orthogonal matrix Q_1 (built from Householder transformations and Givens rotations),
$$S := Q_1^T SJQ_1J^T = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix},$$
where S_{11} is upper triangular. Update $H := Q_1^T HJQ_1J^T$.
2. Set $T := S$. Reduce H to *Hessenberg-triangular form* using orthogonal matrices Q_1 and Q_2 (built from Givens rotations), which also preserve the structure of S and T,
$$\begin{aligned} S &:= Q_1^T SJQ_1J^T = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{11}^T \end{bmatrix}, \\ T &:= JQ_2^T J^T TQ_2 = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{11}^T \end{bmatrix}, \\ H &:= Q_1^T HQ_2 = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22}^T \end{bmatrix}, \end{aligned}$$
where S_{11}, T_{11}, and H_{11} are upper triangular and H_{22} is upper Hessenberg.
3. Apply the periodic QZ algorithm to the formal matrix product $H_{22}S_{11}^{-1}H_{11}T_{11}^{-1}$, using orthogonal matrices V_i, $i = 1:4$, such that $S_{11} := V_3^T S_{11}V_2$, $H_{11} := V_3^T H_{11}V_4$, $T_{11} := V_1^T T_{11}V_4$ are upper triangular, and $H_{22} := V_1^T H_{22}V_2$ is upper quasi-triangular (i.e., it is block upper triangular, with 1×1 and 2×2 diagonal blocks). MATLAB-style notation is used for index ranges.
4. Set $\Lambda(S,H) = \pm\imath\sqrt{\Lambda(H_{22}S_{11}^{-1}H_{11}T_{11}^{-1})}$, where $\Lambda(\cdot(,\cdot))$ denotes the spectrum of the matrix $\cdot$ (or of the matrix pencil $\cdot,\cdot$).

Taking symmetry into account, only n eigenvalues are returned by the solver, namely those with nonnegative imaginary part and those positive, if real. The remaining eigenvalues have opposite signs. The matrices S_{11} and T_{11} in the formal matrix product $H_{22}S_{11}^{-1}H_{11}T_{11}^{-1}$ may be singular, meaning that there are infinite eigenvalues.[1] However, it is assumed that the original matrix pencil is *regular*, that is, $\det(\lambda S - H) \not\equiv 0$. Note that the transformed formal matrix product is *similar* with the initial matrix product at Step 3, that is, they have the same spectrum. Even if the matrices S_{11} and T_{11} are nonsingular, evaluating the product and calling a standard eigensolver

[1] This is the reason of referring to a “formal” matrix product, since when any of the matrices S_{11} or T_{11} is singular, that product does not exist.

is not a good idea in general because of a risk of numerical cancellations, especially for large number of factors and of ill-conditioned inverses.

This paper investigates in more detail the numerical behavior of the periodic QZ algorithm at Step 3 in the context of solving sHH eigenproblems.

Consider now a formal matrix product P,

$$P=\prod_{i=1}^{k}A_i^{s_i}, \tag{3}$$

where $A_i \in \mathbf{R}^{n\times n}$, and $s_i=\pm 1$, $i=1:k$. The pQZ algorithm does not evaluate the product but just transforms the factors to reveal the eigenstructure. A negative exponent s_i means that the "inverse" of the corresponding factor, A_i, should be considered. If a factor with such an exponent is singular, the algorithm will still provide a solution, revealing one or more infinite eigenvalues. The pQZ algorithm operates with general formal matrix products, where the factors have no structure. Any such product can be reduced to a similar one, where one of the factors is upper Hessenberg and all other factors are upper triangular. Such a reduction is described in (Bojanczyk et al., 1992) for the case with alternating exponents in (3). In the context of this paper, consider that A_h is upper Hessenberg and A_i are upper triangular, $i \neq h$. Without loss of generality it is assumed that $s_h=1$. Otherwise, all exponents can be virtually multiplied by -1.[2] Since

$$\Lambda(P)=\Lambda\Big(\prod_{i=1}^{h-1}A_i^{s_i}\prod_{i=h}^{k}A_i^{s_i}\Big)=\Lambda\Big(\prod_{i=h}^{k}A_i^{s_i}\prod_{i=1}^{h-1}A_i^{s_i}\Big),$$

it can be assumed that $h=1$ or $h=k$. For the sHH problem described above, $h=1$, $k=4$, and $s=\big[1,-1,1,-1\big]$.

The pQZ algorithm is a generalization of the QZ algorithm that was treated in, for example, (Golub and Van Loan, 1996). The essential ingredients are the same: reduction to a Hessenberg-triangular form, deflation, computation of the shifts, and the QZ step. All transformations applied during the computations are defined as follows:

$$\widetilde{A}_i := \begin{cases} Q_i^T A_i Q_{i\oplus 1}, & \text{if } s_i = \ \ 1,\\ Q_{i\oplus 1}^T A_i Q_i, & \text{if } s_i = -1, \end{cases} \tag{4}$$

where Q_i, $i=1:k$, are orthogonal matrices (built by multiplying plane rotations, in the context of this paper), and $i\oplus 1 := \mathrm{mod}(i,k)+1$. Using (4) it follows that $\widetilde{A}_i^{s_i}=Q_i^T A_i^{s_i} Q_{i\oplus 1}$. It is easy to verify that the definitions in (4) preserve similarity between the original and transformed formal product. Indeed,

$$\widetilde{A}_1^{s_1}\widetilde{A}_2^{s_2}\cdots\widetilde{A}_k^{s_k}=Q_1^T A_1^{s_1}A_2^{s_2}\cdots A_k^{s_k}Q_1=Q_1^T P Q_1\,.$$

Two deflation strategies are implemented in the periodic QZ solver. The first strategy is a "careful" (or cautious) one, where the convergence criteria are based on the magnitudes of neighboring elements. This is the recommended option and it is used by the sHH solver when calling the periodic QZ algorithm. The second one is a more "aggressive" strategy, when elements on the subdiagonal or diagonal are set to zero as soon as they become smaller in magnitude than the norm of the corresponding factor times the relative machine precision, ε_M. This option is only recommended if balancing is applied beforehand and convergence problems are observed.

The processing following a deflation detection is performed in specific ways for the Hessenberg matrix and for triangular matrices. More details are given for the Hessenberg matrix case. The notation $a_{pq}^{(i)}$ denotes the (p,q) entry of A_i, also written as $A^{(i)}$ when subscripts are needed. For the cautious case, define $t=\varepsilon_M(|a_{j-1,j-1}^{(1)}|+|a_{j,j}^{(1)}|)$, if $t\neq 0$, and $t=\varepsilon_M\|A_{1:j,1:j}^{(1)}\|_1$, otherwise, where $\|\cdot\|_1$ refers to the 1-norm of a matrix. If $|a_{j,j-1}^{(1)}|\leq t$, then $a_{j,j-1}^{(1)}$ is considered negligible, the Hessenberg matrix is split into two Hessenberg submatrices, $A_{1:j-1,1:j-1}^{(1)}$ and $A_{j:n,j:n}^{(1)}$, and the eigenvalue problem is solved separately for each of them, starting with the trailing part. Of course, if the previous deflation took place at an index l, then the current subproblem to be solved is defined by the range of indices $j:l$. In the "aggressive" deflation case, the tolerance used is $t=\varepsilon_M\|A_1\|_F$, where the subscript F refers to the Frobenius norm. Summarizing, a deflation in the Hessenberg matrix reduces to partitioning of the problem into subproblems. No transformations are necessary.

Similar tests are performed for the triangular matrices of the formal matrix product. When there is a zero diagonal element in a triangular matrix A_i with $s_i=1$, about n suitably chosen Givens rotations applied on each side of each factor will deflate a zero eigenvalue. Similarly, when there is a zero diagonal element in a triangular matrix A_i with $s_i=-1$, an infinite eigenvalue will be deflated. More details are given in (Bojanczyk et al., 1992; Kressner, 2001).

The periodic QZ step works with subproblems where the Hessenberg submatrix is *unreduced*, that is, without any zero on the first subdiagonal, and the triangular submatrices are nonsingular. Starting with a suitably chosen initial transformation discussed below, new transformations are found and are propagated to all factors via (4) so that the transformed subproblem preserves its Hessenberg-triangular form. Each such "sweep" is equivalent to one step of the standard QR algorithm applied to the formal matrix

[2] Note that in this case the eigenvalues of the original product will be the reciprocals of the eigenvalues computed with opposite exponents.

product. According to the theory of the standard eigenproblem, after a number of QR steps, deflation(s) will occur and the problem decomposes into smaller subproblems. Normally, the pQZ algorithm finishes finding all eigenvalues. However, in rare cases the algorithm may not converge.

The initial transformation is chosen with the aim of increasing the convergence speed. This is performed using *shifts*. Assume that the eigenvalues $\lambda_{l+1}, \ldots, \lambda_n$ have been determined and an unreduced nonsingular subproblem has been found, defined by the range of indices $j:l$, where initially $l = n$. The standard technique for matrices (or matrix pencils) is to use as shifts the eigenvalues of the block(s) defined by $l-1:l$, if $l > 1$, or by l, otherwise. Since the bottom 2×2 part may have complex conjugate eigenvalues, using two shifts simultaneously is necessary to keep the arithmetic real. Actually, the two shifts implicitly used are the eigenvalues λ_1 and λ_2 of the 2×2 matrix (adapted from (Kressner, 2001))

$$F := \begin{bmatrix} a_{mm}^{(1)} & a_{ml}^{(1)} \\ a_{lm}^{(1)} & a_{ll}^{(1)} \end{bmatrix} \prod_{i=2}^{k} \begin{bmatrix} a_{mm}^{(i)} & a_{ml}^{(i)} \\ 0 & a_{ll}^{(i)} \end{bmatrix}^{s_i}, \qquad (5)$$

where $m = l - 1$.

The algorithm annihilates the second and third entries of the first column, P_1, of the double shift Wilkinson polynomial

$$\begin{aligned} P_\lambda &:= \left(\underline{P} - \lambda_1 I_q\right)\left(\underline{P} - \lambda_2 I_q\right) \\ &= \underline{P}^2 - (\lambda_1 + \lambda_2)\underline{P} + \lambda_1\lambda_2 I_q, \end{aligned} \qquad (6)$$

where $\underline{P}$ denotes the submatrix containing the rows and columns $j:l$ of P, and $q = l - j + 1$. Specifically, two Givens rotations, G_1 and G_2, are computed such that

$$\begin{bmatrix} G_1 & 0 \\ 0 & I_{q-2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & G_2 & 0 \\ 0 & 0 & I_{q-3} \end{bmatrix} P_1 =: U_0 P_1 \qquad (7)$$

is transformed to a multiple of e_1, the first column of the identity matrix I_q. Note that the entries $4:q$ of P_1 are zero by construction. Although this is a standard procedure for eigensolvers, a brief explanation is useful. The algorithm actually performs two QR steps with shifts λ_1 and λ_2. The QR factorization, $\underline{P} - \lambda_1 I_q = Q_a R_a$, where Q_a is orthogonal and R_a upper triangular, and the spectrum preserving operation

$$P_a := R_a Q_a + \lambda_1 I_q = Q_a^T \underline{P} Q_a$$

define the first QR step. Q_a is chosen so that P_a is upper Hessenberg. Similarly, $P_a - \lambda_2 I_q = Q_b R_b$, and

$$P_b := R_b Q_b + \lambda_2 I_q = Q_b^T P_a Q_b = Q_b^T Q_a^T \underline{P} Q_a Q_b$$

define the second QR step. Setting $Q = Q_a Q_b$, $R = R_b R_a$, premultiplying $Q_b R_b = P_a - \lambda_2 I_q$ by Q_a, and postmultiplying it by $R_a = Q_a^T(\underline{P} - \lambda_1 I_q)$, it follows that

$$QR = \left(\underline{P} - \lambda_2 I_q\right)\left(\underline{P} - \lambda_1 I_q\right) = P_\lambda$$

is a QR factorization of P_λ. Moreover, $QP_b = \underline{P}Q$ and P_b is, without loss of generality, an unreduced Hessenberg matrix. But any real unreduced Hessenberg matrix $H := Q^T \underline{P} Q$, with Q an orthogonal matrix, has the property that H and Q are uniquely determined by the first column of Q, see, for instance, (Golub and Van Loan, 1996). Therefore, if from $\underline{P}$ one determines an upper Hessenberg matrix H so that $\widetilde{Q}H = \underline{P}\widetilde{Q}$, where $\widetilde{Q}$ is orthogonal and its first column coincides with that of Q, then $\widetilde{Q} = Q$ and $H = P_b$. Now, P_λ can be triangularized by a product of $q-1$ Householder transformations, $U_i = U_i^T$, $i = 1 : q-1$, and the first column of $Q = U_1 U_2 \cdots U_{q-1}$ coincides with that of U_1 (and with that of U_0 in (7)), which has at most the first three entries nonzero. Hence, $U_1 \underline{P} U_1$ has a "bump" of extra possibly nonzero entries in the locations (3,1), (4,1), and (4,2). If $U_1 \underline{P} U_1$ is reduced to an upper Hessenberg matrix H, then $H = P_b$.

For numerical reasons, $\underline{P}$, F, $\Lambda(F)$, and P_λ are not explicitly computed but a suitable embedding is used. The previous version of the solver used the embedding proposed in (Kressner, 2001),

$$\begin{aligned} P_\lambda = \begin{bmatrix} \underline{A}_1 & I_q \end{bmatrix} \prod_{i=2}^{k} \begin{bmatrix} \underline{A}_i & 0 \\ 0 & a_{mm}^{(i)} I_q \end{bmatrix}^{s_i} \\ \cdot \begin{bmatrix} -I_q & 0 \\ a_{mm}^{(1)} I_q & -a_{lm}^{(1)} I_q \end{bmatrix} \begin{bmatrix} -\underline{A}_1 & a_{lm}^{(1)} I_q & a_{ll}^{(1)} I_q \\ 0 & a_{mm}^{(1)} I_q & a_{ml}^{(1)} I_q \end{bmatrix} \\ \cdot \prod_{i=2}^{k} \begin{bmatrix} \underline{A}_i & 0 & 0 \\ 0 & a_{mm}^{(i)} I_q & a_{ml}^{(i)} I_q \\ 0 & 0 & a_{ll}^{(i)} I_q \end{bmatrix}^{s_i} \begin{bmatrix} I_q \\ 0 \\ I_q \end{bmatrix}, \end{aligned} \qquad (8)$$

where $\underline{A}_i$ is the submatrix defined by the rows and columns $j:l$ of A_i. By exploiting the structure in (8), the rotations G_1 and G_2 can be efficiently computed. Evaluating the embedding (8) indeed gives

$$\underline{P}^2 - (\lambda_1 + \lambda_2)\underline{P} + \lambda_1\lambda_2 I_q\,.$$

3 CONVERGENCE FAILURE EXPERIMENT

Although myriads of sHH eigenproblems have been successfully solved, in recent work a problem was identified where the pQZ solver, hence also the sHH solver, failed to converge. Because this event proved difficult to reproduce across platforms, a small binary scaling was applied to make the solver fail more consistently and obtain failure rate statistics for algorithm comparison purposes and further investigations.

This is briefly described below. The start was with a structured skew-symmetric/symmetric pencil of order $2n = 80$ defined by the skew-symmetric matrix N and symmetric matrix M, given as

$$N := \begin{bmatrix} -D & A \\ -A^T & E \end{bmatrix}, \quad M := \begin{bmatrix} -V & C \\ C^T & W \end{bmatrix}. \quad (9)$$

The (skew-)symmetry of N and M implies that D and E are skew-symmetric and V and W are symmetric matrices. By applying a block-column permutation and sign changes, the pencil $\lambda S - H$, with $S := -NJ$ and $H := -MJ$, is sHH.

Starting from this example, a large number of tests have been performed. Specifically, randomized small scaling factors have been used, chosen as 2^r, with r randomly taking values in the set $\{-2, -1, 0, 1, 2\}$. Two vectors, S_1, S_2, of length 40, with such scaling factors have been used for each new example, generated as,

$$\begin{aligned}\widetilde{A} &= \mathrm{diag}(S_1)A\,\mathrm{diag}(S_2);\ \widetilde{D} = \mathrm{diag}(S_1)D\mathrm{diag}(S_1);\\ \widetilde{E} &= \mathrm{diag}(S_2)E\,\mathrm{diag}(S_2);\ \widetilde{C} = \mathrm{diag}(S_1)C\mathrm{diag}(S_2);\\ \widetilde{V} &= \mathrm{diag}(S_1)V\,\mathrm{diag}(S_1);\ \widetilde{W} = \mathrm{diag}(S_2)W\,\mathrm{diag}(S_2).\end{aligned}$$

With this scaling, the sHH problems for A, D, E, C, V, W and $\widetilde{A}$, $\widetilde{D}$, $\widetilde{E}$, $\widetilde{C}$, $\widetilde{V}$, $\widetilde{W}$ have the same eigenvalues. Moreover, since all problems generated in this manner only differ by a small scaling from the original problem, the convergence behavior of the periodic QZ algorithm should be quite similar for all of them. However, in the first tests with the previous sHH version of the solver, there were 42 cases of nonconvergence failures in 21095 trials. For such a failure, the periodic QZ algorithm (called by the sHH solver) cannot separate a 1×1 or 2×2 submatrix at the bottom of the current Hessenberg matrix, defined by the last row and column index l. This means that the l-th eigenvalue could not be found since the periodic QZ iteration did not converge. The processed submatrix $C_{1:l,1:l}$ is still in the upper Hessenberg form (not necessarily unreduced) and not in full Schur form (at least, its elements $c_{l-1,l-2}$ and $c_{l,l-1}$ are not zero).

One approach for solving the convergence problem is to increase the accepted maximum total number of iterations. The initial version of the pQZ solver has this value set to $30n$. Such a number is also used in the LAPACK eigensolvers. However, a larger value could be used, for example, $60n$, taking into account that a $2n$-order eigenproblem is actually solved. Some statistics for the failure rates for different values of the number of iterations are shown in Table 1, for exactly the same problems (generated using the same initial seed for the MATLAB function `rand`).

In contrast, using the "aggressive" strategy, convergence was achieved for all nonconverging problems. This observation suggested that after a failure with the cautious strategy, in principle, a second call with the "aggressive" strategy might work. Since the number of nonconverging examples is very small compared to the number of tests, the additional computational effort due to the second call of the solver is negligible.

Table 1: Failure rates statistics for several values of the total number of iterations, maxit, allowed for pQZ algorithm.

runs	maxit	failures
21095	$30n$	42
	$60n$	1
100107	$60n$	17
	$120n$	1

Note that increasing the allowed total number of iterations does not practically affect the computational effort. However, there is no guarantee that all problems could be solved, no matter how many iterations are allowed. Based on these remarks, several changes in the implementation details of the sHH and pQZ solvers have been evaluated. The simplest and effective working solution has been to allow for $120n$ iterations and make a second call of the pQZ solver with the "aggressive" option when the first call (with the "careful" option) returned with an error indicating nonconvergence. With this modification, there were only four cases, out of 10^6 scaled problems, when the first call to the pQZ solver did not converge. For these cases, convergence occurred with the second call. Table 2 shows some error statistics for all these 10^6 runs with respect to the original problem. The notations used are as follows: err is the error norm (the Euclidean norm of the vector of differences in the eigenvalues of the original and a scaled problem); rerr is the relative error norm; max, min, and mean denote the maximum, minimum, and the mean of all these errors, respectively; norm denotes the Euclidean norm of the vector of all error norms.

Table 2: Global error statistics for 10^6 runs of the sHH solver with respect to the original problem.

	err	rerr
max	$3.20 \cdot 10^{-11}$	$5.18 \cdot 10^{-13}$
min	$8.94 \cdot 10^{-14}$	$1.45 \cdot 10^{-15}$
mean	$7.36 \cdot 10^{-13}$	$1.19 \cdot 10^{-14}$
norm	$1.09 \cdot 10^{-9}$	$1.77 \cdot 10^{-11}$

The values in Table 2 are very good results. Moreover, with the previous version of the solver there were about 20-30 fatal failures (meaning nonconvergence) for each batch of 10^4 problems, that is, 2000-3000 failures for a 10^6 problem set.

Table 3 shows the number of examples, from a set of 40004 scaled problems generated as described above, needing a number of iterations of the periodic QZ algorithm in various ranges. The only iteration count that exceeded 3000 was 3090.

Table 3: Histogram data for the number of iterations of the periodic QZ algorithm for 40004 runs.

Iterations	Number of examples
> 3000	1
$(2000, 3000]$	14
$(1000, 2000]$	96
$(100, 1000]$	8,679
≤ 100	31,214

Therefore, less than 100 iterations are needed for almost all cases. The mean number is about 108. But there are about 100 problems that needed more than $30n = 1200$ iterations. Even $60n$ is not sufficiently large for about a dozen problems.

4 IMPROVING PERIODIC QZ ALGORITHM

During the tests, it was discovered that the implicit Wilkinson double shift polynomial used by the periodic QZ solver was not the desired one. Specifically, the first rows of the matrix F in (5) and of the trailing 2×2 submatrix of P_λ in (6) differ, since the contribution of the $l-2$ rows of the factors is not taken into account, where l is the last row of the currently deflated subproblem. The influence of the $l-2$ rows could be avoided if the Hessenberg matrix would be the last factor of the product. But in the implementation of the sHH solver, the Hessenberg matrix is assumed to be the first one. Since F is incorrect, the implicitly used shifts are inaccurate, at least in the first iterations of the pQZ algorithm for the same subproblem. However, the shifts become increasingly accurate if and when the iterative process converges for the current subproblem. The occasionally observed convergence difficulties were supposed to be explainable by the use of possibly poor approximations of the true eigenvalues of the trailing 2×2 submatrix of the product.

Since the eigenvalues of the formal product P in (3) with $s_1 = 1$ are the same as the eigenvalues of

$$\Big(\prod_{i=2}^{k} A_i^{s_i}\Big) A_1 \,, \qquad (10)$$

it was then necessary to find an appropriate embedding for the product having the Hessenberg matrix A_1 as the last factor and to adapt the solver for this new setting. Then, the eigenvalues of the trailing 2×2 subproblem of the new double shift polynomial will be indeed correct.

For the formal matrix product in (10) it can be proven that an embedding of the corresponding P_λ is

$$P_\lambda = \begin{bmatrix} -I_q & I_q & 0 \end{bmatrix} \prod_{i=2}^{k} \begin{bmatrix} \underline{A}_i & 0 & 0 \\ 0 & a_{mm}^{(i)} I_q & a_{ml}^{(i)} I_q \\ 0 & 0 & a_{ll}^{(i)} I_q \end{bmatrix}^{s_i} \cdot \begin{bmatrix} \underline{A}_1 & 0 \\ a_{mm}^{(1)} I_q & a_{ml}^{(1)} I_q \\ a_{lm}^{(1)} I_q & a_{ll}^{(1)} I_q \end{bmatrix} \begin{bmatrix} -I_q & a_{ll}^{(1)} I_q \\ 0 & -a_{lm}^{(1)} I_q \end{bmatrix} \cdot \prod_{j=2}^{k} \begin{bmatrix} \underline{A}_i & 0 \\ 0 & a_{ll}^{(i)} I_q \end{bmatrix}^{s_i} \begin{bmatrix} \underline{A}_1 \\ I_q \end{bmatrix} . \qquad (11)$$

This embedding is used to find the matrix U_0 so that when premultiplying P_λ by U_0, its first column is reduced to a multiple of e_1.

However, the tests with the scaled problems mentioned before have shown a behavior similar to that for the previous version of the solver. Specifically, for four problems out of 1,010,000, the modified solver did not converge in $120n = 4800$ iterations. The results are shown in Table 4.

Table 4: Histogram data for the number of iterations of the modified periodic QZ algorithm for 1,010,000 runs.

Iterations	Number of examples	Percentage
> 4000	8	0.00079
$(3000, 4000]$	45	0.0045
$(2000, 3000]$	373	0.037
$(1000, 2000]$	2,492	0.25
$(100, 1000]$	217,554	21.54
≤ 100	789,528	78.17

Therefore, 99.71% of the problems required less than 1000 iterations. However, 2,245 problems required more than $30n = 1200$ iterations (the usual maximum number of iterations in a QR-like algorithm), of which 151 problems required more than $60n$ iterations. The mean number of iterations is 106, the median is 85, and the standard deviation is 104.55. The error statistics are identical or slightly better than in Table 2. All four problems showing nonconvergence actually converged after a second call to the periodic QZ solver with the "aggressive" strategy.

The irregularity of a nonnegligible number of problems requiring much more iterations than the large majority, for a sequence of problems (with identical eigenvalues) differing only by small, powers of 2, scaling factors, suggested that further investigation is needed. The nonconverging example encountered first in the series has been analyzed in detail. The behavior of the solver for this problem can be summarized as follows: out of $n = 40$ eigenvalues to compute, the last 22 have been found after 61 iterations and the algorithm has further deflated a block of size 4 in the rows and columns 15:18, which has two pairs of complex conjugate eigenvalues. Unfortunately, this subproblem could not be split in the remaining $4800 - 61$ iterations.

The 4×4 nonconverging subproblem was isolated and analyzed separately. The product of its factors has eigenvalues $75.74 \pm 0.07689\imath$ and $116.84 \pm 0.06248\imath$, but the eigenvalues of the trailing 2×2 (product) submatrix are real, about 75.75 and 116.82. So, the pQZ solver implicitly assumed these real eigenvalues as initial shifts. The behavior in the remaining iterations, but the last three ones, was similar. Omitting these last three iterations, at each other iteration, the two real eigenvalues belonged to two clusters centered at 75.74 and 116.84, with standard deviations about $7.74 \cdot 10^{-2}$ and $3.98 \cdot 10^{-2}$, respectively. The problem was solved after 1149 iterations, which is about three times larger than the default value, that is, 120×4. However, the minimum absolute value of the $(3,2)$ element during iterations, except for the last three, was $9.22 \cdot 10^{-5}$. The last three absolute values were $6.18 \cdot 10^{-8}$, $5.26 \cdot 10^{-19}$ and 0, respectively. The eigenvalues used as shifts for the last three iterations were $116.84 \pm 0.06248\imath$.

The main convergence difficulty above seems to be due to the existence of two consecutive 2×2 blocks with small imaginary parts. The implicitly chosen shifts are taken close to the real parts of the complex eigenvalues from both blocks. Consequently, it is impossible to force convergence in this case using such shifts.

This remark suggested how to modify the periodic QZ solver to force convergence when needed. Specifically, if an eigenvalue (pair) does not converge after 60 iterations and the order of the current deflated (hence nonsingular) subproblem is at most six, then a new routine is called, which computes the eigenvalues of that subproblem using the LAPACK (Anderson et al., 1999) routine DLAHQR and finds two rotations making the first column of the real Wilkinson double shift polynomial parallel to the first unit vector. The shifts are chosen as the two eigenvalues with largest moduli. If there are complex conjugate eigenvalues, the real eigenvalues, if any, are not considered. The idea is to force convergence for the 2×2 blocks because such blocks may produce convergence difficulties. Clearly, this strategy evaluates the product of the factors, but only for small order subproblems for which there are convergence difficulties. If the shifts found in this manner are still inaccurate because of nearly singular submatrices and because of multiplying the factors, then convergence difficulties may in principle persist, though this never occurred in the case study of this paper. For most examples, this strategy is not invoked at all. A “window” of 60 iterations for each new eigenvalue (pair) has been chosen in order to allow the use of exceptional transformations, as in other eigensolvers. After invoking an exceptional transformation, a switch is set for calling the new routine finding the eigenvalues to be used as shifts. However, this routine is invoked only if convergence does not occur after 30 additional iterations for the same small order (≤ 6) subproblem. In such a case, the routine can be called for the next, at most 10, consecutive iterations.

The strategy described above proved to be very effective for the 10^6 set of scaled problems. All problems converged and the maximum number of iterations was 204. The second call of the periodic QZ solver, with the “aggressive” deflation strategy, was never needed. The error statistics are actually the same as before (see Table 2) but the convergence is much faster. Specifically, 99.88% of the problems require less than 150 iterations, which is much smaller than $30n = 1200$. The mean number of iterations is 90.65, the median value is 85, and the standard deviation is 13.85, again much smaller than for the previous versions of the periodic QZ algorithm. The summary of the convergence results is given in Table 5. It is possible to avoid using exceptional transformations for small order deflated subproblems, calling the new routine instead, and further increasing the speed by using a smaller window size. Experiments with this approach have resulted in further reduction of the total number of iterations. The value 4 for the maximum order of the subproblem has also been used instead of 6 and the behavior has been the same for our case study. The value 6 has been chosen to possibly enlarge the domain in which this strategy is effective.

Table 5: Histogram data for the number of iterations of the latest modification of the periodic QZ algorithm for 1,000,001 runs.

Iterations	Number of examples	Percentage
> 200	2	0.0002
$(150, 200]$	1,233	0.123
$(100, 150]$	219,236	21.92
$(75, 100]$	774,329	77.43
$(50, 75]$	5,201	0.52

The new routine for finding the shifts is called only when there are convergence difficulties (after 60 iterations in the current version). Otherwise, and also for the next eigenvalues, the implicit scheme is used, as long as it works fast enough. The main difficulty with the original solver was that the implicitly used eigenvalues have been real approximations of eigenvalues from two different blocks with complex eigenvalues. If the real parts of a complex conjugate pair with small imaginary parts would have been used, the implicit scheme would be likely to succeed but this was not the case and too many iterations were required in the situation described above. Actually, all 10^6 problems have been solved by allowing around 5500 iterations.

5 CONCLUSIONS

The periodic QZ algorithm involved in the structure-preserving skew-Hamiltonian/Hamiltonian algorithm has been investigated. The main algorithmic issues have been presented and the convergence behavior has been analyzed for a series of equivalent skew-Hamiltonian/Hamiltonian eigenproblems of order 80, which differ by small, powers of 2, scaling factors. In a few cases, the previous version of the solver did not converge. For other cases the number of iterations required for convergence varied in a very large range (from less than 100 till over 5000). Some modifications of the periodic QZ and skew-Hamiltonian/Hamiltonian solvers have been proposed for which there are no failures and the number of iterations did not exceed 204 for the same large set of examples. These solvers are needed in many domains, including periodic systems and robust optimal control.

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: Third Edition*. Software · Environments · Tools. SIAM, Philadelphia.

Apkarian, P., Gahinet, P. M., and Buhr, C. (2014). Multi-model, multi-objective tuning of fixed-structure controllers. In *Proceedings 13th European Control Conference (ECC)*, 24–27 June 2014, Strasbourg, France, pages 856–861. IEEE.

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., Mehrmann, V., Sima, V., Van Huffel, S., and Varga, A. (1999). SLICOT — A subroutine library in systems and control theory. In Datta, B. N., editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 10, pages 499–539. Birkhäuser, Boston, MA.

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 Bendtsen, J. D., editor, *7th IFAC Symposium on Robust Control Design (ROCOND'12)*, pages 189–194. IFAC.

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

Benner, P., Sima, V., and Voigt, M. (2016). Algorithm 961: Fortran 77 subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems. *ACM Transactions on Mathematical Software (TOMS)*, 42(3):1–26.

Bojanczyk, A. W., Golub, G., and Van Dooren, P. (1992). The periodic Schur decomposition: Algorithms and applications. In Luk, F. T., editor, *Proc. of the SPIE Conference Advanced Signal Processing Algorithms, Architectures, and Implementations III*, volume 1770, pages 31–42.

Bruinsma, N. A. and Steinbuch, M. (1990). A fast algorithm to compute the H_∞-norm of a transfer function. *Systems Control Lett.*, 14(4):287–293.

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

Kressner, D. (2001). An efficient and reliable implementation of the periodic QZ algorithm. In *IFAC Proceedings Volumes (IFAC Workshop on Periodic Control Systems)*, volume 34, pages 183–188.

Kressner, D. (2005). *Numerical Methods for General and Structured Eigenvalue Problems*, volume 46 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin.

MathWorks® (2019). MATLAB® , Release R2019a, March 2019.

MathWorks® (2012). Control System Toolbox™ , Release R2012a, March 2012.

Sreedhar, J. and Van Dooren, P. (1994). Periodic Schur form and some matrix equations. In *Proceedings of the Symposium on the Mathematical Theory of Networks and Systems (MTNS'93)*, Regensburg, Germany, 2–6 August 1993, volume 1, pages 339–362. John Wiley & Sons.

Van Loan, C. F. (1975). A general matrix eigenvalue algorithm. *SIAM J. Numer. Anal.*, 12:819–834.

Xia, M., Gahinet, P.M., Abroug, N., and Buhr, C. (2017). Sector bounds in control design and analysis. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, Australia, pages 1169–1174.