Improving the Convergence of the Periodic QZ Algorithm

Vasile Sima\textsuperscript{1} and Pascal Gahinet\textsuperscript{2}\textsuperscript{b}

\textsuperscript{1}Modelling, Simulation, Optimization Department, National Institute for Research & Development in Informatics,
 Bd. Mareşal Averescu, Nr. 8–10, Bucharest, Romania
\textsuperscript{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 eigenvlue 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 \( \lambda \in \mathbb{C} \). In the real case, considered in this paper, often used definitions for such matrices are \( (SJ)^T = -SJ \) and \( (HJ)^T = HJ \), where

\[
J := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},
\]

\[
S := \begin{bmatrix} A & D \\ E & \lambda I \end{bmatrix}, \quad H := \begin{bmatrix} C & V \\ W & -C^T \end{bmatrix},
\]

where \( A, D, E, C, V, W \in \mathbb{R}^{n\times n} \), \( D \) 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 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 \( L_\infty \)-norms computation and linear-quadratic and \( H_\infty \) 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 \( H_\infty \)-norm, based on (Brunnsma and Steinbuch, 1990), and to nonsmooth minimization of the \( L_\infty \)-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 \( \mu \) upper bounds for real uncertainty.

The sHH solver uses the periodic QR algorithm, sometimes called periodic QR algorithm (Van Loan, 1975; Bojanczyk et al., 1992; Sreedhar and
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 (Bojanetzky 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 \( \lambda S - H \equiv JP^T J^T (\lambda S - H) P \), where \( P \) is a nonsingular matrix. The pencils \( \lambda S - H \) 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,

\[
\begin{bmatrix}
S_{11} & S_{12} \\
0 & S_{11}^T
\end{bmatrix}
- \begin{bmatrix}
H_{11} & H_{12} \\
0 & -H_{11}^T
\end{bmatrix},
\]

where \( S_{11}, H_{11} \in \mathbb{R}^{n \times n} \), 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),

\[
S := Q_1^T S Q_1 J^T = \begin{bmatrix}
S_{11} & S_{12} \\
0 & S_{11}^T
\end{bmatrix},
\]

where \( S_{11} \) is upper triangular. Update \( H := Q_1^T H J Q_1 J^T \).

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 \( T \),

\[
S := Q_1^T S Q_1 J^T = \begin{bmatrix}
S_{11} & S_{12} \\
0 & S_{11}^T
\end{bmatrix},
\]

\[
T := J Q_2^T J^T T Q_2 = \begin{bmatrix}
T_{11} & T_{12} \\
0 & T_{11}^T
\end{bmatrix},
\]

\[
H := Q_1^T H Q_2 = \begin{bmatrix}
H_{11} & H_{12} \\
0 & H_{22}^T
\end{bmatrix},
\]

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 \( S_{11} := V_2^T S_{11} V_2 \), \( H_{11} := V_1^T H_{11} V_1 \), \( T_{11} := V_4^T T_{11} V_4 \) are upper triangular, and \( H_{22} := V_3^T H_{22} V_3 \) is upper quasi-triangular (i.e., it is block upper triangular, with \( 1 \times 1 \) and \( 2 \times 2 \) diagonal blocks). MATLAB-style notation is used for index ranges.

4. Set \( \Lambda(S,H) = \pm \sqrt{\Lambda(H_{22} S_{11}^{-1} H_{11} T_{11}^{-1})} \), 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, \( \det(\lambda S - H) \neq 0 \). 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 \( S_{11} \) or \( T_{11} \) 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 $P$, 

$$P = \prod_{i=1}^{k} A_i^s,$$

where $A_i \in \mathbb{R}^{m_i \times n_i}$, 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 $-1$. Since

$$A(P) = A(h-1) \prod_{i=h}^{k} A_i^s = A(h-1) \prod_{i=h}^{k} A_i^s,$$

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

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:

$$\tilde{A}_i = \begin{cases} Q_i^T A_i Q_i \text{ if } s_i = 1, \\ Q_i^T A_i Q_i \text{ if } s_i = -1, \end{cases}$$

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

$$\tilde{A}_1^s \tilde{A}_2^s \cdots \tilde{A}_k^s = Q_i^T A_i^s Q_i = Q_i^T P Q_i.$$

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, $\epsilon_M$. 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 $A_i^{(pq)}$ when subscripts are needed. For the cautious case, define $t = \epsilon_M \| A_{j-1,j}^{(1)} + |a_{j,j}^{(1)}| \| \text{ if } t \neq 0$, and $t = \epsilon_M \| A_{j-1,j}^{(1)} \|$ otherwise, where $\| \cdot \|$ 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, $A_{j-1,j}^{(1)}$ and $A_{j,n,j}$, 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 $t = \epsilon_M \| A_{j} \|_F$, 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 un-reduced, 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.
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 \( \lambda_{j+1}, \ldots, \lambda_n \) 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 x 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 \( \lambda_1 \) and \( \lambda_2 \) of the 2 x 2 matrix (adapted from (Kressner, 2001))

\[
F := \begin{bmatrix}
    (a_{11}^{(1)}) & (a_{11}^{(1)}) & \cdots & (a_{11}^{(1)}) \\
    (a_{m1}^{(1)}) & (a_{m1}^{(1)}) & \cdots & (a_{m1}^{(1)}) \\
    \vdots & \vdots & \ddots & \vdots \\
    (a_{ll}^{(1)}) & (a_{ll}^{(1)}) & \cdots & (a_{ll}^{(1)}) \\
\end{bmatrix} \prod_{i=2}^{k} \begin{bmatrix}
    (a_{mm}^{(i)}) & (a_{ml}^{(i)}) & \cdots & (a_{mj}^{(i)}) \\
    0 & (a_{ml}^{(i)}) & \cdots & (a_{mj}^{(i)}) \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & (a_{mj}^{(i)}) \\
\end{bmatrix}^{s_i},
\]

(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

\[
P_\lambda := (P - \lambda_1 I_q)(P - \lambda_2 I_q)
\]

(6)

where \( P \) 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

\[
\begin{bmatrix}
    G_1 & 0 \\
    0 & I_{q-2}
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 \\
    0 & G_2 & 0 \\
    0 & 0 & I_{q-3}
\end{bmatrix}
\]

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 \( \lambda_1 \) and \( \lambda_2 \). The QR factorization, \( P - \lambda_1 I_q = Q_1 R_1 \), where \( Q_1 \) is orthogonal and \( R_1 \) upper triangular, and the spectrum preserving operation

\[
P_\alpha := R_1 Q_1 + \lambda_1 I_q = Q_1^T P \alpha
\]

define the first QR step. \( Q_\alpha \) is chosen so that \( P_\alpha \) is upper Hessenberg. Similarly, \( P_\alpha - \lambda_2 I_q = Q_2 R_2 \), and

\[
P_\beta := R_2 Q_2 + \lambda_2 I_q = Q_2^T P \beta = Q_2^T Q_1^T P \alpha Q_\alpha
\]

define the second QR step. Setting \( Q = Q_\alpha Q_\beta, R = R_\alpha R_\beta \), premultiplying \( Q_\beta R_\beta = P_\alpha - \lambda_2 I_q \) by \( Q_\alpha \), and postmultiplying it by \( R_\alpha = Q_\alpha^T (P - \lambda_1 I_q) \), it follows that

\[
QR = (P - \lambda_2 I_q)(P - \lambda_1 I_q) = P_\lambda
\]
is a QR factorization of \( P_\lambda \). Moreover, \(QP_\beta = PQ_\alpha \) and \( P_\beta \) is, without loss of generality, an unreduced Hessenberg matrix. But any real unreduced Hessenberg matrix \( H := \overline{Q}^T \overline{P} \), 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 \( P \) one determines an upper Hessenberg matrix \( H \) so that \( \overline{QH} = \overline{P} \), where \( \overline{Q} \) is orthogonal and its first column coincides with that of \( Q \), then \( \overline{Q} = Q \) and \( H = P_\beta \). Now, \( P_\beta \) can be triangularized by a product of \( q - 1 \) Householder transformations, \( U_1 = U_1^T, i = 1 : q - 1 \), 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, \( U_1 P_\alpha U_1 \) has a "bump" of extra possibly nonzero entries in the locations (3,1), (4,1), and (4,2). If \( U_1 P_\alpha U_1 \) is reduced to an upper Hessenberg matrix \( H \), then \( H = P_\beta \).

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

\[
P_\lambda = \begin{bmatrix}
    \Delta_1 & I_q \\
    & 0
\end{bmatrix} \prod_{i=2}^{k} \begin{bmatrix}
    \Delta_1 & 0 \\
    0 & \alpha_{mm} I_q
\end{bmatrix}^{s_i}
\]

(8)

where \( \Delta_i \) 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

\[
P^2 = (\lambda_1 + \lambda_2) P + \lambda_3 \lambda_2 I_q
\]

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,

$$ A = \text{diag}(S_1) A \text{diag}(S_2); \quad \tilde{D} = \text{diag}(S_1) D \text{diag}(S_1); \quad \tilde{E} = \text{diag}(S_2) E \text{diag}(S_2); \quad \tilde{C} = \text{diag}(S_1) C \text{diag}(S_2); \quad \tilde{V} = \text{diag}(S_1) V \text{diag}(S_1); \quad W = \text{diag}(S_2) W \text{diag}(S_2).$$

With this scaling, the sHH problems for $A, D, E, C, V, W$ and $\tilde{A}, \tilde{D}, \tilde{E}, \tilde{C}, \tilde{V}, W$ 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 $1 \times 1$ or $2 \times 2$ 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 $C_{l,l-1}$ 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 $\text{rand}$).

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.

<table>
<thead>
<tr>
<th>runs</th>
<th>maxit</th>
<th>failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>21095</td>
<td>30n</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>60n</td>
<td>1</td>
</tr>
<tr>
<td>100107</td>
<td>60n</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>120n</td>
<td>1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>err</th>
<th>rerr</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>$3.20 \cdot 10^{-11}$</td>
<td>$5.18 \cdot 10^{-11}$</td>
</tr>
<tr>
<td>min</td>
<td>$8.94 \cdot 10^{-14}$</td>
<td>$1.45 \cdot 10^{-15}$</td>
</tr>
<tr>
<td>mean</td>
<td>$7.36 \cdot 10^{-13}$</td>
<td>$1.19 \cdot 10^{-14}$</td>
</tr>
<tr>
<td>norm</td>
<td>$1.09 \cdot 10^{-9}$</td>
<td>$1.77 \cdot 10^{-14}$</td>
</tr>
</tbody>
</table>

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 $10^6$ 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>
<thead>
<tr>
<th>Iterations</th>
<th>Number of examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 3000</td>
<td>1</td>
</tr>
<tr>
<td>(2000, 3000)</td>
<td>14</td>
</tr>
<tr>
<td>(1000, 2000)</td>
<td>96</td>
</tr>
<tr>
<td>(100, 1000)</td>
<td>8,679</td>
</tr>
<tr>
<td>≤ 100</td>
<td>31,214</td>
</tr>
</tbody>
</table>

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 \( 2 \times 2 \) submatrix of \( P_\lambda \) 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 \( 2 \times 2 \) 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

\[
\left( \prod_{i=2}^{k} A_i^n \right) A_1, \tag{10}
\]

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 \( 2 \times 2 \) 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 \( P_l \) is

\[
P_l = \begin{bmatrix} -I_l & I_l & 0 \\ \sum_{i=2}^{k} A_i \prod_{j=2}^{i-1} A_j & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^{s_1} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & a_{nn}I_l & a_{nl}I_l \\ 0 & 0 & 0 \end{bmatrix}^{s_1} = \begin{bmatrix} A_1 & 0 \\ 0 & a_{ii}I_l \end{bmatrix} \prod_{j=2}^{k} A_j.
\]

This embedding is used to find the matrix \( U_0 \) so that when premultiplying \( P_l \) 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>
<thead>
<tr>
<th>Iterations</th>
<th>Number of examples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 4000</td>
<td>8</td>
<td>0.00079</td>
</tr>
<tr>
<td>(3000, 4000)</td>
<td>45</td>
<td>0.0045</td>
</tr>
<tr>
<td>(2000, 3000)</td>
<td>373</td>
<td>0.037</td>
</tr>
<tr>
<td>(1000, 2000)</td>
<td>2,492</td>
<td>0.25</td>
</tr>
<tr>
<td>(100, 1000)</td>
<td>217,554</td>
<td>21.54</td>
</tr>
<tr>
<td>≤ 100</td>
<td>789,528</td>
<td>78.17</td>
</tr>
</tbody>
</table>

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 investiga-
tion 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 $4 \times 4$ nonconverging subproblem was isolated and analyzed separately. The product of its factors has eigenvalues $75.74 \pm 0.07689$ and $116.84 \pm 0.06248$, but the eigenvalues of the trailing $2 \times 2$ (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, $120 \times 4$. 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 $6.18 \cdot 10^{-5}, 5.26 \cdot 10^{-19}$ and 0, respectively. The eigenvalues used as shifts for the last three iterations were $116.84 \pm 0.06248$.

The main convergence difficulty above seems to be due to the existence of two consecutive $2 \times 2$ blocks with small imaginary parts. The explicitly chosen shifts are taken close to the real parts of these last three iterations, at each other iteration, the two real eigenvalues used as shifts for the last three iterations were $6.18 \pm 10^{-5}, 5.26 \pm 10^{-19}$ and 0, respectively. The eigenvalues used as shifts for the last three iterations were $116.84 \pm 0.06248$.

The test case 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 30$n = 1200$. 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.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Number of examples</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&gt; 200$</td>
<td>2</td>
<td>0.0002</td>
</tr>
<tr>
<td>(150, 200)</td>
<td>1.233</td>
<td>0.123</td>
</tr>
<tr>
<td>(100, 150)</td>
<td>219.236</td>
<td>21.92</td>
</tr>
<tr>
<td>(75, 100)</td>
<td>774.329</td>
<td>77.43</td>
</tr>
<tr>
<td>(50, 75)</td>
<td>5.201</td>
<td>0.52</td>
</tr>
</tbody>
</table>

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 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 $10^9$ 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


