SR and SZ Algorithms for the Symplectic (Butterfly) Eigenproblem

Peter Benner and Heike Faßbender

Zentrum für Technomathematik, Fachbereich 3 – Mathematik and Informatik, Universität Bremen, 28357 Bremen, FRG. Email: benner@math.uni-bremen.de, heike@math.uni-bremen.de

David S. Watkins *

Department of Pure and Applied Mathematics, Washington State University,Pullman, Washington 99164-3113, USA. Email: watkins@wsu.edu. Mailing address: 6835 24th Avenue NE, Seattle, WA 98115-7037, USA.

Abstract

SR and SZ algorithms for the symplectic (generalized) eigenproblem that are based on the reduction of a symplectic matrix to symplectic butterfly form are discussed. A $2n \times 2n$ symplectic butterfly matrix has $8n - 4$ (generically) nonzero entries, which are determined by $4n - 1$ parameters. While the SR algorithm operates directly on the matrix entries, the SZ algorithm works with the $4n - 1$ parameters. The algorithms are made more compact and efficient by using Laurent polynomials, instead of standard polynomials, to drive the iterations.

Dedicated to Professor Ludwig Elsner
on the occasion of his 60th birthday.

1. Introduction

This paper furthers the development of the family of QR-like algorithms for solving eigensystem problems. Professor Ludwig Elsner, to whom this paper is dedicated, has made significant contributions to this development, as demonstrated by both his own publications [13, 17, 30, 31, 32] and those of his students [5, 11, 12, 14, 19, 25] (for example) and grandstudents. Here we focus on symplectic matrices and pencils. Recent developments in this area have been guided by the unitary case, which the symplectic case

*Supported by the National Science Foundation under grant DMS-9403569.

655 Avenue of the Americas, New York, NY 10010 0024-3795/92/$5.00

resembles to some degree. An important landmark has been the work of Bunse-Gerstner and Elsner [13] on the unitary eigenvalue problem.

Symplectic (generalized) eigenvalue problems occur in many applications, e.g., in discrete linear quadratic optimal control, discrete Kalman filtering, the solution of discrete algebraic Riccati equations, discrete stability radii and H_∞-norm computations (see, e.g., [23, 26] and the references therein) and discrete Sturm-Liouville equations (see, e.g., [9]). The solution of the symplectic (generalized) eigenvalue problem has been the topic of numerous publications during the last 30 years. Even so, a numerically sound method, i.e., a strongly backward stable method in the sense of [10], is yet not known. The numerical computation of an invariant (deflating) subspace is usually carried out by an iterative procedure like the QR (QZ) algorithm; see, e.g., [26, 28]. The QR (QZ) algorithm is numerically backward stable but it ignores the symplectic structure. In order to develop fast, efficient, and reliable methods, the symplectic structure of the problem should be preserved and exploited. Then important properties of symplectic matrices (e.g., eigenvalues occurring in reciprocal pairs) will be preserved and not destroyed by rounding errors.

Using the analogy to the continuous-time case, i.e., Hamiltonian eigenvalue problems, Flaschka, Mehrmann, and Zywietz show in [19] how to construct structure-preserving methods for the symplectic eigenproblem based on the SR method [16, 25]. This method is a QR-like method based on the SR decomposition. In an initial step, the $2n \times 2n$ symplectic matrix is reduced to a more condensed form, the symplectic J-Hessenberg form, which in general contains $2n^2 + 3n - 1$ nonzero entries. As in the general framework of GR algorithms [30], the SR iteration preserves the symplectic J-Hessenberg form at each step and is supposed to converge to a form from which eigenvalues and invariant (deflating) subspaces can be read off. A $2n \times 2n$ symplectic J-Hessenberg matrix is determined by $4n-1$ parameters. The SR algorithm can be modified to work only with these parameters instead of the $2n^2+3n-1$ nonzero matrix elements. Thus only $\mathcal{O}(n)$ arithmetic operations per SR step are needed compared to $\mathcal{O}(n^2)$ arithmetic operations when working on the actual J-Hessenberg matrix. The authors note that the algorithm "*...forces the symplectic structure, but it has the disadvantage that it needs* $4n-1$ *terms to be nonzero in each step, which makes it highly numerically unstable. ... Thus, so far, this algorithm is mainly of theoretical value.*" [19, page 186, last paragraph].

Recently, Banse and Bunse-Gerstner [5, 3, 4] presented a new condensed form for symplectic matrices, the *symplectic butterfly form*. The $2n \times 2n$ condensed matrix is symplectic, contains $8n - 4$ nonzero entries, and is, similar to the symplectic J-Hessenberg form of [19], determined by $4n - 1$ parameters. As observed in [3] the SR algorithm preserves the butterfly form in its iterations. It is pointed out that the SR algorithm can be rewritten in a parameterized form that works with $4n-1$ parameters instead of the $(2n)^2$ matrix elements in each iteration. Hence, the symplectic structure, which will be destroyed in the numerical process due to roundoff errors, can easily be restored in each iteration for this condensed form. There is reason to believe that an SR algorithm based on the symplectic butterfly form has better numerical properties than one based on the symplectic J-Hessenberg form; see Sections 3 and 5.

The $4n-1$ parameters that determine a symplectic butterfly matrix B cannot be read off of B directly. Computing the parameters can be interpreted as factoring B into the product of two even simpler matrices M and N: $B = M^{-1}N$. The parameters can then be read off of M and N directly. Up to now two different ways of factoring symplectic butterfly matrices have been proposed in the literature [2, 6]. In Section 2 we will introduce these factorizations and consider their drawbacks and advantages.

In Section 3 we will revisit the SR algorithm for symplectic butterfly matrices. Such an algorithm was already considered in [3, 6]. In those publications, it is proposed to use a polynomial of the form $p(\lambda) = \prod_{i=1}^{k}(\lambda - \mu_i)$ to drive the SR step, just as in the implicit QR (bulge-chasing) algorithm for upper Hessenberg matrices. Here we will show that it is better to use a Laurent polynomial to drive the SR step. This reduces the size of the bulges that are introduced, thereby decreasing the number of computations required per iteration. It also improves the convergence and stability properties of the algorithm by effectively treating each reciprocal pair of eigenvalues as a unit. The method still suffers from loss of the symplectic structure due to roundoff errors, but the loss of symplecticity is normally less severe than in an implementation using a standard polynomial, because less arithmetic is done and the similarity transformations are generally better conditioned. Moreover, using the factors M and N of the symplectic butterfly matrix B, one can easily and cheaply restore the symplectic structure of the iterates whenever necessary.

To derive a method that is purely based on the $4n-1$ parameters that determine B and that thus forces the symplectic structure, one needs to work with the factors M and N. This leads us to develop (in Section 4) an SZ algorithm for the matrix pencil $M - \lambda N$, whose eigenvalues are the same as those of the symplectic matrices $M^{-1}N$, NM^{-1}, MN^{-1}, and $N^{-1}M$.

Numerical examples are presented in Section 5.

2. Symplectic butterfly matrices and pencils

A matrix $M \in \mathbb{R}^{2n \times 2n}$ is called *symplectic* (or *J-orthogonal*) if

$$MJM^T = J \tag{1}$$

(or equivalently, $M^TJM = J$) and a *symplectic matrix pencil* $M - \lambda N$, $M, N \in \mathbb{R}^{2n \times 2n}$ is defined by the property

$$MJM^T = NJN^T, \tag{2}$$

where

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \tag{3}$$

and I_n is the $n \times n$ identity matrix. While symplectic matrices are nonsingular ($M^{-1} = JM^TJ^T$), a symplectic matrix pencil $M - \lambda N$ is not necessarily regular, i.e., there is no guarantee that $\det(M - \lambda N)$ does not vanish identically for all $\lambda \in \mathbb{C}$. M and N may be nonsingular or singular. Hence (2) is in general not equivalent to $M^TJM = N^TJN$.

The spectrum of a symplectic matrix pencil/matrix is symmetric with respect to the unit circle. Or, in other words, the eigenvalues of symplectic symplectic matrix pencils occur in reciprocal pairs: if $\lambda \neq 0$ is a (generalized finite) eigenvalue, then so is λ^{-1}. Furthermore, if $\lambda = 0$ is an eigenvalue of a symplectic pencil, then so is ∞. Let $y^T \in \mathbb{R}^{2n} \backslash \{0\}$ be a left eigenvector of $M - \lambda N$ to the eigenvalue λ, then $x = JM^Ty$ is a right eigenvector to the eigenvalue λ^{-1}. Hence for symplectic matrices we have: if λ is an eigenvalue of M with right eigenvector x, then λ^{-1} is an eigenvalue of M with left eigenvector $(Jx)^T$. Further, if $\lambda \in \mathbb{C}$ is an eigenvalue of M (or $M - \lambda N$), then so are $\overline{\lambda}, \lambda^{-1}, \overline{\lambda^{-1}}$.

A symplectic matrix

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} \diagdown & \diagdown\!\!\diagdown\!\!\diagdown \\ \diagdown & \diagdown\!\!\diagdown\!\!\diagdown \end{bmatrix}, \qquad \text{where } B_{ij} \in \mathbb{R}^{n \times n},$$

is called a *butterfly* matrix if B_{11} and B_{21} are diagonal, and B_{12} and B_{22} are tridiagonal. Banse and Bunse-Gerstner [3, 5] showed that for every symplectic matrix M, there exist numerous symplectic matrices S such that $B = S^{-1}MS$ is a symplectic butterfly matrix. In [3], an elimination process for computing the butterfly form of a symplectic matrix is presented (see also [6]).

In [3], a *strict butterfly matrix* is introduced in which the upper left diagonal matrix of the butterfly form is nonsingular. This allows the decomposition of B into two simpler symplectic matrices:

$$B = \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{11}^{-1} \end{bmatrix} \begin{bmatrix} I & V \\ 0 & I \end{bmatrix} = \begin{bmatrix} \diagdown & 0 \\ \diagdown & \diagdown \end{bmatrix} \begin{bmatrix} I & \diagdown\!\!\diagdown\!\!\diagdown \\ 0 & I \end{bmatrix}, \tag{4}$$

where $V = B_{11}^{-1}B_{12}$ is tridiagonal and symmetric. Hence $4n-1$ parameters that determine the symplectic matrix can be read off directly. Obviously, n of these parameters have to be nonzero (the diagonal elements of B_{11}). If any of the $n-1$ subdiagonal elements of V is zero, deflation can take place; that is, the problem can be split into at least two problems of smaller dimension, but with the same symplectic butterfly structure.

This decomposition was introduced because of its close resemblance to symplectic matrix pencils that appear naturally in control problems. These pencils are typically of the form

$$M - \lambda N = \begin{bmatrix} F & 0 \\ H & I \end{bmatrix} - \lambda \begin{bmatrix} I & -G \\ 0 & F^T \end{bmatrix}, \qquad F, G = G^T, H = H^T \in \mathbb{R}^{n \times n}.$$

(Note: For $F \neq I$, M and N are not symplectic.) Assuming that M and N are nonsingular (that is, F is nonsingular), we can rewrite the above equation

$$\begin{bmatrix} I & 0 \\ 0 & F^{-T} \end{bmatrix} (M - \lambda N) =: \widetilde{M} - \lambda \widetilde{N} = \begin{bmatrix} F & 0 \\ F^{-T}H & F^{-T} \end{bmatrix} - \lambda \begin{bmatrix} I & -G \\ 0 & I \end{bmatrix}.$$

(Note: $\widetilde{M}$ and $\widetilde{N}$ are symplectic matrices.) Solving this generalized eigenproblem is equivalent to solving the eigenproblem for the symplectic matrix

$$\widetilde{M}^{-1}\widetilde{N} = \begin{bmatrix} F^{-1} & 0 \\ -HF^{-1} & F^T \end{bmatrix} \begin{bmatrix} I & -G \\ 0 & I \end{bmatrix}.$$

In [6], an *unreduced butterfly matrix* is introduced in which the lower right tridiagonal matrix is unreduced, that is, the subdiagonal elements of B_{22} are nonzero. Using the definition of a symplectic matrix, one easily verifies that if B is an unreduced butterfly matrix, then B_{21} is nonsingular. As above, this allows the decomposition of B into two simpler symplectic matrices:

$$B = \begin{bmatrix} B_{21}^{-1} & B_{11} \\ 0 & B_{21} \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & T \end{bmatrix} = \begin{bmatrix} \diagdown & \diagdown \\ 0 & \diagdown \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & \diagdown\!\!\diagdown\!\!\diagdown \end{bmatrix}, \tag{5}$$

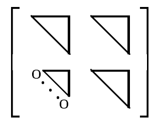
where $T = B_{21}^{-1}B_{22}$ is tridiagonal and symmetric. Hence $4n-1$ parameters that determine the symplectic matrix can be read off directly. Obviously, the diagonal elements of B_{21} have to be nonzero. If any of the $n-1$ subdiagonal elements of T is zero, deflation can take place; that is, the problem can be split into at least two problems of smaller dimension, but with the same symplectic butterfly structure. Hence, $2n-1$ of the parameters determining an unreduced butterfly matrix $B \in \mathbb{R}^{2n\times 2n}$ have to be nonzero.

The introduction of the class of unreduced butterfly matrices and the associated decomposition (5) was motivated by purely theoretical considerations. The unreduced butterfly matrices play a role analogous to that of unreduced Hessenberg matrices in the standard QR theory [3, 6]. We need a couple of definitions to start with. Given a symplectic matrix M, we define *generalized Krylov matrices* $\mathcal{K}(M,v)$ by

$$\mathcal{K}(M,v) \;=\; [\,v, M^{-1}v, \ldots, M^{-n+1}v, Mv, M^2v, \ldots, M^n v\,]. \tag{6}$$

A matrix R is called J*-triangular* if

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \tag{7}$$



where all submatrices $R_{ij} \in \mathbb{R}^{n\times n}$ are upper triangular and R_{21} is strictly upper triangular. By $e_j, j = 1, \ldots, m$ we will denote the jth unit vector in $\mathbb{R}^m$.

LEMMA 1. *Let $B = S^{-1}MS$, where S and M are symplectic, and B is a symplectic butterfly matrix. Let $R = S^{-1}\mathcal{K}(M, Se_1) = \mathcal{K}(B, e_1)$. Then R is J-triangular. Furthermore, R is nonsingular if and only if B is an unreduced butterfly matrix.*

The proof is straightforward. As a first consequence of Lemma 1, we note that the equation $\mathcal{K}(M, Se_1) = SR$ gives a factorization of $\mathcal{K}(M, Se_1)$ into a symplectic matrix times a J-triangular matrix, i.e., an SR *factorization*. If B is unreduced, then R is nonsingular, and the SR factorization is essentially unique [12]. (Precisely, the factorization is unique up to symplectic, J-triangular factors, which can be passed back and forth between S and R. Symplectic, J-triangular matrices have the form

$$\begin{bmatrix} C^{-1} & F \\ 0 & C \end{bmatrix}, \tag{8}$$

where C and F are diagonal matrices. We will call these *trivial* matrices.) Given M, the matrix $\mathcal{K}(M, Se_1)$ is determined by the first column of S.

The essential uniqueness of the factorization $\mathcal{K}(M, Se_1) = SR$ tells us that the transforming matrix S for the similarity transformation $B = S^{-1}MS$ is essentially uniquely determined by its first column. This *implicit-S* theorem can serve as the basis for the construction of an implicit SR algorithm for butterfly matrices, just as the implicit-Q theorem [21, § 7.4.5] provides a basis for the implicit QR algorithm on upper Hessenberg matrices. In both cases uniqueness depends on the unreduced character of the matrix.

Obviously, not every unreduced butterfly matrix B is a strict butterfly matrix, but B can be turned into a strict one by a similarity transformation by a trivial matrix (8). Numerous choices of C and F will work. Thus, it is practically true that every unreduced butterfly matrix is strict. In [6] it is shown that the converse is false. There are strict butterfly matrices that are not similar to any unreduced butterfly matrix.

Because not every strict butterfly matrix is unreduced, the class of strict butterfly matrices lacks the theoretical basis for an implicit SR algorithm. If one wishes to build an algorithm based on the decomposition (4), one is obliged to restrict oneself to unreduced, strict butterfly matrices. The following considerations show that this is not a serious restriction.

REMARK 2. If both B_{11} and B_{21} are nonsingular, then the matrices $V = B_{11}^{-1}B_{12}$ and $T = B_{21}^{-1}B_{22}$ are related by $V = T - B_{11}^{-1}B_{21}^{-1}$. Thus the off-diagonal entries of V and T are the same. It follows that corresponding off-diagonal entries of B_{12} and B_{22} are either zero or nonzero together. In connection with the decomposition (4), this implies that whenever B is not unreduced, V will also be reducible, and we can split the eigenvalue problem into smaller ones.

This relationship breaks down, however, if B_{21} is singular. Consider, for example, the class of matrices

$$B = \left[\begin{array}{cc|cc} 1 & 0 & a & g \\ 0 & 1 & g & c \\ \hline 1 & 0 & 1+a & g \\ 0 & 0 & 0 & 1 \end{array}\right]$$

with $g \neq 0$. These are strict butterfly matrices for which B_{12} is unreduced but B_{22} is not. Notice that the $(2,2)$ and $(4,4)$ entries are eigenvalues and can be deflated from the problem.

In general, if B_{21} is singular, a deflation (and usually a splitting) is possible. If $(B_{21})_{i,i} = 0$, then $(B_{11})_{i,i}$ must be nonzero, since B is nonsingular. This forces $(B_{22})_{i,i-1} = (B_{22})_{i,i+1} = 0$, because $B_{11}^T B_{22} - B_{21}^T B_{12} = I$. It follows that $(B_{11})_{i,i}$ and $(B_{22})_{i,i}$ are a reciprocal pair of eigenvalues, which can be deflated from the problem. Unless $i = 1$ or $i = n$, the remaining problem can be split into two smaller problems.

In [3] (see also [6]), an elimination process for transforming a symplectic matrix to butterfly form is given. Based on this reduction process, an SR algorithm for computing the eigenvalues of symplectic butterfly matrices can be developed; see Section 3. The method works explicitly with the butterfly matrix B. Roundoff errors will destroy the symplectic structure. However, because the butterfly form is very compact, one can easily and cheaply restore the symplectic structure of the iterates whenever necessary by making use of the decompositions (4) or (5). See the next section for details. As the parameters that determine a symplectic butterfly matrix cannot be read off directly, one should work with the decompositions $B = M^{-1}N$ (4) or (5) in order to develop a method that is purely based on the parameters and thus forces the symplectic structure. This leads us to take a closer look at the symplectic pencil $M - \lambda N$.

Before doing so, we introduce some notation. The diagonal entries of B_{11} will be denoted by $b_1, \ldots, b_n$ and the diagonal entries of B_{21} by $a_1, \ldots, a_n$. The symmetric tridiagonal matrix $T = B_{21}^{-1} B_{22}$ will be denoted by

$$T = \begin{bmatrix} c_1 & d_2 & & \\ d_2 & c_2 & \ddots & \\ & \ddots & \ddots & d_n \\ & & d_n & c_n \end{bmatrix}. \tag{9}$$

The symplectic butterfly matrix B can be decomposed into the product $M^{-1}N$ as in (4) or (5). Instead of considering the symplectic eigenproblem $Bx = \lambda x$, the generalized eigenproblem $(\lambda M - N)x = 0$ or equivalently $(M - \lambda N)x = 0$ can be considered. For the decomposition (4) we obtain

$$M_s - \lambda N_s = \begin{bmatrix} B_{11}^{-1} & 0 \\ -B_{21} & B_{11} \end{bmatrix} - \lambda \begin{bmatrix} I & V \\ 0 & I \end{bmatrix} \tag{10}$$

while for (5) we obtain

$$M_u - \lambda N_u = \begin{bmatrix} B_{21} & -B_{11} \\ 0 & B_{21}^{-1} \end{bmatrix} - \lambda \begin{bmatrix} 0 & -I \\ I & T \end{bmatrix}. \tag{11}$$

It is well-known, see, e.g., [24, 26] that if $M - \lambda N$ is a symplectic matrix pencil, $Q \in \mathbb{R}^{2n \times 2n}$ is nonsingular, and $S \in \mathbb{R}^{2n \times 2n}$ is symplectic, then $Q(M - \lambda N)S$ is a symplectic matrix pencil and the eigenproblems $M - \lambda N$ and $QMS - \lambda QNS$ are equivalent. Obviously the eigenproblems $M_s - \lambda N_s$ and $M_u - \lambda N_u$ are equivalent: $Q(M_s - \lambda N_s) = M_u - \lambda N_u$ where

$$Q = \begin{bmatrix} 0 & -I \\ I & B_{11}^{-1} B_{21}^{-1} \end{bmatrix}.$$

Hence, if x_s is a right eigenvector of $M_s - \lambda N_s$, then $x_u = x_s$ is a right eigenvector of $M_u - \lambda N_u$. If y_u is a left eigenvector of $M_u - \lambda N_u$, then $y_s = Q^T y_u$ is a left eigenvector of $M_s - \lambda N_s$.

Which of these two equivalent eigenproblems should be preferred in terms of accuracy of the computed eigenvalues? As a measure of the sensitivity of a simple eigenvalue of the generalized eigenproblem $A - \lambda B$, one usually considers the reciprocal of

$$\frac{\sqrt{(y^H A x)^2 + (y^H B x)^2}}{||x||_2 \; ||y||_2} \tag{12}$$

as the condition number, where x is the right eigenvector, y the left eigenvector corresponding to the same eigenvalue μ. If the expression (12) is small, one says that the eigenvalue μ is ill conditioned. Let λ be an eigenvalue of B, x_u and y_u the corresponding right and left eigenvectors of $M_u - \lambda N_u$, and $x_s = x_u$ and $y_s = Q^T y_u$ the corresponding right and left eigenvectors of $M_s - \lambda N_s$. Simple algebraic manipulations show

$$\begin{aligned} ||x_s||_2 &= ||x_u||_2, \\ |y_s^H M_s x_s| &= |y_u^H M_u x_u|, \\ |y_s^H N_s x_s| &= |y_u^H N_u x_u|, \end{aligned}$$

while $||y_s||_2 = ||y_u^T Q||_2$. Therefore, the expressions for the eigenvalue condition number differs only in the 2-norm of the respective left eigenvector. Tests in MATLAB[†] indicate that the pencil $M_s - \lambda N_s$ resolves eigenvalues near 1 better then the pencil $M_u - \lambda N_u$, while $M_u - \lambda N_u$ resolves eigenvalues near $\sqrt{-1}$ better. For other eigenvalues both pencils show the same behavior. Hence, from this short analysis there is no indication whether to prefer one of the pencils because of better numerical behavior.

In [3] an elimination process for computing the reduced matrix pencil form (10) of a symplectic matrix pencil (in which both matrices are symplectic) is given. Based on this reduction process, an SZ algorithm for computing the eigenvalues of symplectic matrix pencils of the form (10) can be developed. As the algorithm works on the factors of the butterfly matrix, it works directly on the $4n - 1$ parameters that determine a symplectic butterfly matrix. An elimination process for computing the reduced matrix pencil form (11) of a symplectic matrix pencil (in which both matrices are symplectic) is given in Section 4. Based on this reduction process, an SZ algorithm for computing the eigenvalues of symplectic matrix pencils of the form (11) is developed. It turns out that the SZ algorithm for the pencil (11) requires slightly fewer operations than the SZ algorithm for the pencil (10); see Section 4 for details.

[†]MATLAB is a trademark of The MathWorks, Inc.

3. *SR* algorithm

Eigenvalues and eigenvectors of symplectic butterfly matrices can be computed efficiently by the SR algorithm [12], which is a QR-like algorithm in which the QR decomposition is replaced by the SR decomposition. Almost every matrix $A \in \mathbb{R}^{2n\times 2n}$ can be decomposed into a product $A = SR$ where S is symplectic and R is J-triangular (7). The SR algorithm is an iterative algorithm that performs an SR decomposition at each iteration. If B is the current iterate, then a *spectral transformation function* q is chosen (such that $q(B) \in \mathbb{R}^{2n\times 2n}$) and the SR decomposition of $q(B)$ is formed, if possible:

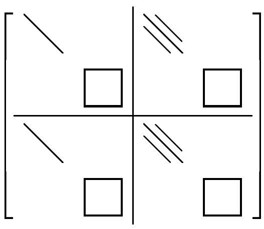
$$q(B) = SR.$$

Then the symplectic factor S is used to perform a similarity transformation on B to yield the next iterate, which we will call $\widehat{B}$:

$$\widehat{B} = S^{-1}BS. \tag{13}$$

If $\operatorname{rank}(q(B)) = 2n$ and B is a symplectic butterfly matrix, then so is $\widehat{B}$ in (13) [3, 5]. If rank $(q(B)) = 2n - \nu =: 2k$ and B is an unreduced symplectic butterfly matrix, then $\widehat{B}$ in (13) is of the form (see [6] for a proof)

$\widehat{B} =$



$$= \underbrace{}_{k}\underbrace{}_{n-k}\underbrace{}_{k}\underbrace{}_{n-k}\left[\begin{array}{cc|cc} \widehat{B}_{11} & & \widehat{B}_{13} & \\ & \widehat{B}_{22} & & \widehat{B}_{24} \\ \hline \widehat{B}_{31} & & \widehat{B}_{33} & \\ & \widehat{B}_{42} & & \widehat{B}_{44} \end{array}\right]\begin{array}{l} \}k \\ \}n-k \\ \}k \\ \}n-k \end{array} \tag{14}$$

where

- $\begin{bmatrix} \widehat{B}_{11} & \widehat{B}_{13} \\ \widehat{B}_{31} & \widehat{B}_{33} \end{bmatrix}$ is a symplectic butterfly matrix and
- the eigenvalues of $\begin{bmatrix} \widehat{B}_{22} & \widehat{B}_{24} \\ \widehat{B}_{42} & \widehat{B}_{44} \end{bmatrix}$ are just the ν shifts that are eigenvalues of B.

An algorithm for computing S and R explicitly is presented in [14]. As with explicit QR steps, the expense of explicit SR steps comes from the fact that $q(B)$ has to be computed explicitly. A preferred alternative is the implicit SR step, an analogue to the Francis QR step [20, 21, 22]. The first implicit transformation S_1 is selected so that the first columns of the implicit and the explicit S are equivalent. That is, a symplectic matrix S_1 is determined such that

$$S_1^{-1} q(B) e_1 = \alpha e_1, \qquad \alpha \in \mathbb{R}.$$

Applying this first transformation to the butterfly matrix yields a symplectic matrix $S_1^{-1} B S_1$ with almost butterfly form having a small bulge. The remaining implicit transformations perform a bulge-chasing sweep down the subdiagonals to restore the butterfly form. That is, a symplectic matrix S_2 is determined such that $S_2^{-1} S_1^{-1} B S_1 S_2$ is of butterfly form again. As the implicit SR step is analogous to the implicit QR step, this technique will not be discussed here. The algorithm for reducing a symplectic matrix to butterfly form as given in [3, 6] can be used as a building block for the implicit SR step. An efficient implementation of the SR step for symplectic butterfly matrices involves $\mathcal{O}(n)$ arithmetic operations. Hence a gain in efficiency is obtained compared to the SR algorithm on J-Hessenberg matrices where each SR step involves $\mathcal{O}(n^2)$ arithmetic operations.

A natural way to choose the spectral transformation function q is to choose a polynomial $p_2(\lambda) = (\lambda - \mu)(\lambda - \mu^{-1})$ for $\mu \in \mathbb{R}$ (or $\mu \in \mathbb{C}, |\mu| = 1$) or $p_4(\lambda) = (\lambda - \mu)(\lambda - \mu^{-1})(\lambda - \overline{\mu})(\lambda - \overline{\mu^{-1}})$ for $\mu \in \mathbb{C}$, as these choices make use of the symmetries of the spectrum of symplectic matrices. A better choice is a Laurent polynomial $q_2(\lambda) = \lambda^{-1} p_2(\lambda)$ or $q_4(\lambda) = \lambda^{-2} p_4(\lambda)$. Each of these is a function of $\lambda + \lambda^{-1}$. For example,

$$\begin{aligned} q_4(\lambda) &= (\lambda + \lambda^{-1})^2 - (\mu + \mu^{-1} + \overline{\mu} + \overline{\mu^{-1}})(\lambda + \lambda^{-1}) \\ &\quad + (\mu + \mu^{-1})(\overline{\mu} + \overline{\mu^{-1}}) - 2. \end{aligned}$$

At first it would appear not to matter whether p_4 or q_4 is used to drive the SR step; the outcome should be essentially the same: An SR iteration driven by p_4 has the form $\widehat{B} = S^{-1} B S$, where S comes from an SR decomposition: $p_4(B) = SR$. On the other hand, $q_4(B) = B^{-2} p_4(B) = (B^{-2} S) R$, which is an SR decomposition of $q_4(B)$. Thus an SR iteration driven by q_4 gives $(B^{-2}S)^{-1} B (B^{-2} S) = S^{-1} B S = \widehat{B}$, the same as for p_4. This equation ignores the fact that the SR decomposition is not uniquely defined. S is at best unique up to right multiplication by a trivial matrix (8). Consequently $\widehat{B}$ is only unique up to a trivial similarity transformation. The $\widehat{B}$ that is obtained in practice will depend upon whether p_4 or q_4 is used to drive the step. In principle any undesirable discrepancy that arises can be corrected by application of a similarity transformation by a trivial matrix. Note, however, that a trivial matrix can be arbitrarily ill conditioned. Thus one transformation could be much better conditioned than the other.

The convergence theory of GR algorithms [30] suggests that Laurent polynomials will be more satisfactory than ordinary polynomials from this standpoint. If symplectic structure is to be preserved throughout the computation, eigenvalues must be deflated in pairs: when λ is removed, λ^{-1} must also be removed. Thus we want eigenvalues λ and λ^{-1} to converge at the same rate. The convergence of GR algorithms is driven by convergence of iterations on a nested sequence of subspaces of dimensions 1, 2, $\ldots$, $2n-1$, [29], [30]. If iterations are driven by a function f, the rate of convergence of the subspaces of dimension k is determined by the ratio $|f(\lambda_{k+1})|/|f(\lambda_k)|$, where the eigenvalues of $f(B)$ are numbered in the order

$$|f(\lambda_1)| \geq |f(\lambda_2)| \geq \cdots \geq |f(\lambda_{2n})|. \tag{15}$$

If f is a Laurent polynomial q of the type we have proposed to use, then $q(\lambda) = q(\lambda^{-1})$ for every λ, so each eigenvalue appears side-by-side with its inverse in the ordering (15). The odd ratios $|f(\lambda_{2j})|/|f(\lambda_{2j-1})|$ are all equal to one; only the even-dimensional subspaces converge. Reciprocal pairs of eigenvalues converge at the same rate and are deflated at the same time.

In contrast, if f is a regular polynomial $p(\lambda) = \lambda^k q(\lambda)$, then for any eigenvalue λ satisfying $|\lambda| > 1$, we will have $p(\lambda) = \lambda^{2k} p(\lambda^{-1})$, whence $|p(\lambda)| > |p(\lambda^{-1})|$. Thus the underlying subspace iterations will tend to force λ and λ^{-1} to converge at different rates. Suppose, for example, B has a single real eigenvalue λ_1 such that $p(\lambda_1)$ dominates all the other eigenvalues. Then the odd ratio $|p(\lambda_2)|/|p(\lambda_1)|$ is less than one, and the sequence of one-dimensional subspaces will converge. This tends to force a_1, the first entry of B_{21}, toward zero. If, after some iterations, a_1 becomes effectively zero, then b_1, the first entry of B_1, will have converged to λ_1. As we already noted in Remark 2, the symplectic structure then forces the $(1,1)$ entry of B_{22} to be λ_1^{-1} and allows a deflation. According to the GR convergence theory, the eigenvalue that should emerge in the $(1,1)$ position of B_{22} is λ_2, where $p(\lambda_2)$ is the second largest eigenvalue of $p(B)$. If, as may happen, $\lambda_2 \neq \lambda_1^{-1}$, we have a conflict between the symplectic structure and the convergence theory. This apparent contradiction is resolved as follows. The convergence of the matrix iterates depends not only on the underlying subspace iterations, but also on the condition numbers of the transforming matrices S [30]. Convergence of the subspace iterations may fail to result in convergence of the matrix iterations if *and only if* the transforming matrices are ill conditioned. The tension between the symplectic structure and the convergence of the subspace iterations is inevitably resolved in favor of the symplectic structure through the production of ill-conditioned transforming matrices. This is clearly something we wish to avoid. Example 5 in Section 5 demonstrates that situations like this do arise and can have undesirable consequences.

Apart from these considerations, the Laurent polynomial is superior because it allows a more economical implementation than the standard polynomial does. For symplectic butterfly matrices B,

$$p_4(B)e_1 = \sum_{j=1}^{4}(\beta_j e_j + \gamma_j e_{n+j})$$

has (generically) eight nonzero entries, whereas

$$q_4(B)e_1 = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3$$

has only three nonzero entries. For the implicit SR step driven by q_4, a symplectic matrix $\widetilde{S}$ has to be determined such that $\widetilde{S}q_4(B)e_1 = \lambda e_1$. Applying $\widetilde{S}$ to B yields

$$\left[\begin{array}{ccccccc|cccccccc}
x & + & + & & & & & x & x & + & + & & & & \\
+ & x & + & & & & & x & x & x & + & & & & \\
+ & + & x & & & & & + & x & x & x & & & & \\
& & & x & & & & + & + & x & x & x & & & \\
& & & & x & & & & & & x & x & x & & \\
& & & & & x & & & & & & x & x & x & \\
& & & & & & \ddots & & & & & & \ddots & \ddots & \ddots \\
\hline
x & + & + & & & & & x & x & + & + & & & & \\
+ & x & + & & & & & x & x & x & + & & & & \\
+ & + & x & & & & & + & x & x & x & & & & \\
& & & x & & & & + & + & x & x & x & & & \\
& & & & x & & & & & & x & x & x & & \\
& & & & & x & & & & & & x & x & x & \\
& & & & & & \ddots & & & & & & \ddots & \ddots & \ddots
\end{array}\right]$$

where the undesired elements are denoted by $+$. If we use p_4 instead of q_4, the bulge is one row and one column larger in each quadrant of the matrix. Thus we prefer q_4 over p_4 (and, similarly, q_2 over p_2) also from the point of view of computational cost.

If the chosen shifts are good approximate eigenvalues, we expect deflation at the end of the SR step as indicated in (14). We propose a shift strategy similar to that used in the standard QR algorithm. For example, for a quadruple shift, we choose the 4 eigenvalues of the 4×4 symplectic submatrix

$$G = \left[\begin{array}{cc|cc}
B_{n-1,n-1} & & B_{n-1,2n-1} & B_{n-1,2n} \\
& B_{n,n} & B_{n,2n-1} & B_{n,2n} \\
\hline
B_{2n-1,n-1} & & B_{2n-1,2n-1} & B_{2n-1,2n} \\
& B_{2n,n} & B_{2n,2n-1} & B_{2n,2n}
\end{array}\right]$$

denoting the entries of B by B_{ij}. There is no need to compute the eigenvalues of G. Comparing $p_4(B)$ with the characteristic polynomial of G we obtain

$$p_4(B) = B^4 - \beta(B^3 + B) + \gamma B^2 + I$$

where

$$\begin{aligned}\beta &= \mathrm{trace}(G)\\ \gamma &= (B_{n-1,n-1}+B_{2n-1,2n-1})(B_{n,n}+B_{2n,2n})+2-B_{2n-1,2n}B_{2n,2n-1}\\ &= (b_{n-1}+a_{n-1}c_{n-1})(b_n+a_nc_n)+2-a_{n-1}a_nd_n^2.\end{aligned}$$

Hence $q_4(B)=p_4(B)B^{-2}=(B+B^{-1})^2-\beta(B+B^{-1})+(\gamma-2)I$ and the first column of $q_4(B)$ is given by

$$\begin{aligned}q_4(B)e_1 &= [(b_1+a_1c_1)^2+a_1a_2d_2^2-\beta(b_1+a_1c_1)+\gamma-2]e_1 \\ &\quad +[a_1d_2(b_2+a_2c_2+b_1+a_1c_1-\beta)]e_2+a_1a_2d_2d_3e_3.\end{aligned} \tag{16}$$

This is exactly the generalized Rayleigh-quotient strategy for choosing shifts proposed by Watkins and Elsner in [30]. Hence, the convergence theorems Theorem 6.2, 6.3, and 6.5 from [30] can be applied here. In particular, the butterfly SR algorithm is typically cubically convergent. Let B_0 be a symplectic butterfly matrix with distinct eigenvalues. Let (B_i) be the sequence generated by the SR algorithm starting from B_0, using the generalized Rayleigh-quotient shift strategy with polynomials of degree m. Then, from Theorem 6.5 in [30], it follows that under certain additional assumptions, if each of the iterates

$$\begin{aligned}PB_iP^T &= \begin{bmatrix} X_{11}^{(i)} & X_{12}^{(i)} \\ X_{21}^{(i)} & X_{22}^{(i)} \end{bmatrix},\\ \text{where } P &= [e_1,e_3,\ldots,e_{2n-1},e_2,e_4,\ldots,e_{2n}]\in\mathbb{R}^{2n\times 2n},\end{aligned}$$

satisfies $||X_{12}^{(i)}||=||X_{21}^{(i)}||$ for some fixed norm $||\cdot||$, then the iterates converge cubically if they converge. In order to see that our iterates always satisfy this constraint, we first note that any unreduced symplectic butterfly matrix is similar to an unreduced butterfly matrix with $b_i=1$ and $|a_i|=1$ for $i=1,\ldots,n$ and $\mathrm{sign}(a_i)=\mathrm{sign}(d_i)$ for $i=2,\ldots n$ (this follows as the reduction to butterfly form is not unique, it is only unique up to scaling by a trivial matrix (8)). Clearly, we can modify the butterfly SR algorithm such that each iterate satisfies these constraints. Consider for each iterate B_i

$$PB_iP^T=\begin{bmatrix} X_{11}^{(i)} & X_{12}^{(i)} \\ X_{21}^{(i)} & X_{22}^{(i)} \end{bmatrix}, \qquad X_{22}^{(i)}\in\mathbb{R}^{2l\times 2l},$$

where $2l$ is the degree of the shift polynomials $p_i(\lambda)=\lambda^l q_i(\lambda)$. As for

$k = n - l + 1$

$$X_{21}^{(i)} = \begin{bmatrix} 0 & \cdots & 0 & b_k d_k \\ 0 & \cdots & 0 & a_k d_k \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}, \qquad X_{12}^{(i)} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & b_{k-1} d_k & 0 & \cdots & 0 \\ 0 & a_{k-1} d_k & 0 & \cdots & 0 \end{bmatrix},$$

and $b_{k-1} = b_k = 1, |a_k| = 1$, we have

$$||X_{12}^{(i)}||_F = \sqrt{2d_k^2} = ||X_{21}^{(i)}||_F.$$

Hence from Theorem 6.5 in [30] we obtain that the convergence rate of the butterfly SR algorithm is typically cubic.

REMARK 3. a) The SR algorithm based on the reduction to unreduced J-Hessenberg form as proposed by Flaschka, Mehrmann and Zywietz in [19] does not typically converge cubically; there is no guarantee that the iterates always satisfy the constraint discussed above.

b) One hypothesis of the convergence theorems given in [30] is (modified slightly to fit the situation given here): *Let $A_0 \in \mathbb{R}^{2n \times 2n}$, and let q be a Laurent polynomial. Let $\lambda_1, \ldots \lambda_{2n}$ denote the eigenvalues of A_0, ordered so that $|q(\lambda_1)| \geq |q(\lambda_2)| \geq \ldots \geq |q(\lambda_{2n})|$. Suppose $2k$ is a positive integer less than $2n$ such that $|q(\lambda_{2k})| > |q(\lambda_{2k+1})|$, let $\rho = |q(\lambda_{2k+1})|/|q(\lambda_{2k})|$, and let (q_i) be a sequence of Laurent polynomials such that $q_i \to q$ and $q_i(\lambda_j) \neq 0$ for $j = 1, \ldots, 2k$ and all i. Let $\mathcal{U}$ be the invariant subspace of PA_0P^T associated with $\lambda_{2k+1}, \ldots, \lambda_{2n}$, and suppose $span\{e_1, \ldots, e_{2k}\} \cap \mathcal{U} = \{0\}$.* It is pointed out in [30] that the condition $span\{e_1, \ldots, e_{2k}\} \cap \mathcal{U} = \{0\}$ is automatically satisfied for unreduced Hessenberg and unreduced J-Hessenberg matrices. This condition also holds for any unreduced symplectic butterfly matrix (see [18] for details).

By applying a sequence of double or quadruple shift SR steps to a symplectic butterfly matrix B it is possible to reduce the tridiagonal blocks in B to quasi-diagonal form with 1×1 and 2×2 blocks on the diagonal. The eigenproblem decouples into a number of simple symplectic 2×2 or 4×4 eigenproblems. In doing so, it is necessary to monitor the off-diagonal elements in the tridiagonal blocks of B in order to bring about decoupling whenever possible. Decoupling occurs if $d_j = 0$ for some j as

$$B = \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

$$
= \left[\begin{array}{cccc|cccc}
b_1 & & & & b_1c_1 - a_1^{-1} & b_1d_2 & & \\
 & \ddots & & & b_2d_2 & \ddots & \ddots & \\
 & & \ddots & & & \ddots & \ddots & b_{n-1}d_n \\
 & & & b_n & & & b_nd_n & b_nc_n - a_n^{-1} \\
\hline
a_1 & & & & a_1c_1 & a_1d_2 & & \\
 & \ddots & & & a_2d_2 & \ddots & \ddots & \\
 & & \ddots & & & \ddots & \ddots & a_{n-1}d_n \\
 & & & a_n & & & a_nd_n & a_nc_n
\end{array}\right]. \quad (17)
$$

Or, equivalently, if $(B_{12})_{j,j-1}, (B_{12})_{j-1,j}, (B_{22})_{j,j-1}$, and $(B_{22})_{j-1,j}$ are simultaneously zero.

When dealing with upper Hessenberg matrices, as in the QR setting, decoupling occurs whenever a subdiagonal element becomes zero. In practice, decoupling is said to occur whenever a subdiagonal element in the Hessenberg matrix H is suitably small. For example, in LAPACK [1] if

$$|h_{p+1,p}| \leq c\mathbf{u}(|h_{p,p}| + |h_{p+1,p+1}|)$$

for some small constant c and the unit roundoff $\mathbf{u}$, then $h_{p+1,p}$ is declared to be zero. This is justified since rounding errors of order $\mathbf{u}||H||$ are already present throughout the matrix.

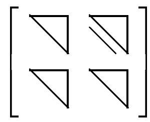
Taking the same approach here, we check whether

$$
\begin{array}{rcl}
\max\{|(B_{12})_{i,i-1}|, |(B_{12})_{i,i+1}|\} & \leq & \epsilon(|(B_{12})_{i-1,i-1}| + |(B_{12})_{ii}|) \\
\max\{|(B_{22})_{i,i-1}|, |(B_{22})_{i,i+1}|\} & \leq & \epsilon(|(B_{22})_{i-1,i-1}| + |(B_{22})_{ii}|)
\end{array}
$$

are simultaneously satisfied, in this case we will have deflation. Here ϵ is some small constant, e.g., $\epsilon = c\mathbf{u}$.

We proceed with the process of applying double or quadruple SR steps to a symplectic butterfly matrix B until the problem has completely split into subproblems of dimension 2 or 4. In a final step we then have to solve these small subproblems in order to compute a real Schur-like form from which eigenvalues and invariant subspaces can be read off. That is, in the 2×2 and 4×4 subproblems we will zero the $(2,1)$ block (if possible) using a symplectic transformation. In case the 4×4 subproblem has real eigenvalues we will further reduce the $(1,1)$ and $(2,2)$ blocks. Moreover, we can sort the eigenvalues such that the eigenvalues inside the unit circle will appear in the $(1,1)$ block. For a detailed discussion see [18].

As mentioned in the introduction, Flaschka, Mehrmann, and Zywietz presented a structure-preserving method for the symplectic eigenproblem based on the SR method in [19]. That method first reduces the symplectic matrix M to symplectic J-Hessenberg form, that is to a matrix of the form



where the $(1,1)-, (2,1)-$ and $(2,2)-$blocks are upper triangular and the $(1,2)-$block is upper Hessenberg. The SR iteration preserves this form at each step and is supposed to converge to a form from which the eigenvalues can be read off. An efficient implementation of the SR step for symplectic J-Hessenberg matrices requires $\mathcal{O}(n^2)$ arithmetic operations; hence no gain in efficiency is obtained compared to the standard Hessenberg QR algorithm. Further, the authors report the loss of the symplectic structure due to roundoff errors after only a few SR steps. As a symplectic J-Hessenberg matrix looks like a general J-Hessenberg matrix, it is not easy to check and to guarantee that the structure is kept invariant in the presence of roundoff errors. Two examples, one involving a 6×6, the other a 12×12 symplectic matrix, are given demonstrating the loss of the symplectic structure.

The symplectic butterfly SR algorithm discussed here also destroys the symplectic structure of the butterfly matrix due to roundoff errors. However, the very compact butterfly form allows one to restore the symplectic structure of the iterates easily and cheaply whenever necessary. This can be done using either one of the two decompositions (4), (5) of a symplectic butterfly matrix discussed in Section 2. Whichever decomposition is used, one assumes that the two diagonal blocks of the butterfly matrix are exact. That is, one assumes that the parameters $a_1, \ldots, a_n, b_1, \ldots, b_n$, which can be read off of the butterfly matrix directly, are correct. Then one uses them to compute the other $2n-1$ parameters. Using, e.g., the decomposition (5) one obtains different formulae for the other parameters

$$
\begin{aligned}
c_k &= B_{k+n,k+n}/a_k \\
&= (B_{k,k+n+1} + a_k^{-1})/b_k \\
d_k &= B_{k,k+n-1}/b_k \\
&= B_{k-1,k+n}/b_{k-1} \\
&= B_{k+n,k+n-1}/a_k \\
&= B_{k+n-1,k+n}/a_{k-1}
\end{aligned}
$$

Adding the terms on the right hand sides and averaging, corrected values for the parameters c_k and d_k are obtained (in actual computations one should use only those terms for which the numerical computations are save, e.g. in case b_k is zero or very small, the equations with this term are not used). Using the so obtained parameters, one computes new entries for the $(1,2)-$

and $(2,2)-$block of the butterfly matrix. Using this procedure to force the symplectic structure whenever necessary, the SR algorithm based on the butterfly form has no problems in solving the two abovementioned examples given by Flaschka, Mehrmann, and Zywietz in [19]; cubic convergence can be observed, see Section 5.

4. SZ algorithm

In this section we develop an algorithm for computing the eigenvalues of a symplectic butterfly matrix B that is purely based on the parameterization of the butterfly matrices in the iteration process and thus preserves the symplectic structure automatically. No additional adjustments like the ones described at the end of the last section will be necessary. The algorithm will work with just the $4n-1$ parameters that determine B.

In order to derive such a method, one should work with the factorization $B = M^{-1}N$ (4) or (5), as the parameters of B can be read off of M and N directly. The eigenvalue problem $M^{-1}Nx = \lambda x$ is equivalent to $(\lambda M - N)x = 0$ and $(M - \lambda N)x = 0$ because of the symmetry of the spectrum. In the latter equations the $4n-1$ parameters are given directly. An SZ algorithm will be developed to solve these generalized eigenproblems. The SZ algorithm is the analogue of the SR algorithm for the generalized eigenproblem, just as the QZ algorithm is the analogue of the QR algorithm for the generalized eigenproblem. Both are instances of the GZ algorithm [32].

Each iteration step begins with M and N such that the corresponding butterfly matrix $B = M^{-1}N$ is unreduced. Choose a spectral transformation function q and compute a symplectic matrix Z_1 such that

$$Z_1^{-1} q(M^{-1}N)e_1 = \alpha e_1$$

for some scalar α. Then transform the pencil to

$$\widetilde{M} - \lambda \widetilde{N} = (M - \lambda N)Z_1.$$

This introduces a bulge into the matrices $\widetilde{M}$ and $\widetilde{N}$. Now transform the pencil to

$$\widehat{M} - \lambda \widehat{N} = S^{-1}(\widetilde{M} - \lambda \widetilde{N})\widetilde{Z},$$

where $\widehat{M}$ and $\widehat{N}$ are in form (10) or (11), depending on the form of M and N, S and $\widetilde{Z}$ are symplectic, and $\widetilde{Z}e_1 = e_1$. This concludes the iteration.

Letting $Z = Z_1\widetilde{Z}$, we have

$$\widehat{M} - \lambda \widehat{N} = S^{-1}(M - \lambda N)Z.$$

The symplectic matrices $\widehat{M}^{-1}\widehat{N}$ and $\widehat{N}\widehat{M}^{-1}$ are similar to $M^{-1}N$ and NM^{-1}, respectively. Indeed

$$\widehat{M}^{-1}\widehat{N} = Z^{-1}(M^{-1}N)Z \quad \text{and} \quad \widehat{N}\widehat{M}^{-1} = S^{-1}NM^{-1}S.$$

The following theorem shows that these similarity transformations amount to iterations of the SR algorithm on $M^{-1}N$ and NM^{-1}.

THEOREM 4. *There exist J-triangular matrices R and U such that*

$$q(NM^{-1}) = SR \quad \textit{and} \quad q(M^{-1}N) = ZU.$$

Proof. The transforming matrix Z was constructed so that $Ze_1 = Z_1\widetilde{Z}e_1 = Z_1e_1 = \alpha^{-1}q(B)e_1$, where $B = M^{-1}N$. Now

$$q(B)\mathcal{K}(B, e_1) = \mathcal{K}(B, q(B)e_1) = \alpha\mathcal{K}(B, Ze_1) = \alpha Z\mathcal{K}(\widehat{B}, e_1),$$

where $\widehat{B} = Z^{-1}BZ = \widehat{M}^{-1}\widehat{N}$. By Lemma 1, $\mathcal{K}(B, e_1)$ and $\mathcal{K}(\widehat{B}, e_1)$ are J-triangular, and $\mathcal{K}(B, e_1)$ is nonsingular. Hence $q(M^{-1}N) = ZU$, where

$$U = \alpha\mathcal{K}(\widehat{B}, e_1)\mathcal{K}(B, e_1)^{-1}$$

is J-triangular.

The proof that $q(NM^{-1})$ equals SR depends on which of the decompositions (10) or (11) is being used. If (11) is being used, M and $\widehat{M}$ are J-triangular matrices, so $\widehat{M}^{-1}e_1 = \beta e_1$ and $Me_1 = \gamma e_1$ for some β and γ. Since $\widehat{M} = S^{-1}MZ$, we have $Se_1 = MZ\widehat{M}^{-1}e_1 = \beta MZe_1 = \beta\alpha^{-1}Mq(M^{-1}N)e_1 = \beta\alpha^{-1}q(NM^{-1})Me_1 = \beta\alpha^{-1}\gamma q(NM^{-1})e_1$. Thus $q(NM^{-1})e_1 = \delta Se_1$ for some nonzero δ. Since the matrices $C = NM^{-1}$ and $\widehat{C} = \widehat{N}\widehat{M}^{-1}$ are butterfly matrices, and C is unreduced, we can now repeat the argument of the previous paragraph with B replaced by C to get $q(NM^{-1}) = SR$, where $R = \delta\mathcal{K}(\widehat{C}, e_1)\mathcal{K}(C, e_1)^{-1}$ is J-triangular.

If the decomposition (10) is being used, M is not J-upper triangular. However, since $\widehat{N}^{-1}e_1 = e_1$ and $Ne_1 = e_1$, we can use the equation $\widehat{N} = S^{-1}NZ$ in the form $S = NZ\widehat{N}^{-1}$ to prove that $q(NM^{-1})e_1 = \delta Se_1$ for some δ, as above. In this case C and $\widehat{C}$ are not butterfly matrices, but their inverses are. Thus one can show, as above, that $q(NM^{-1}) = SR$, where $R = \delta\mathcal{K}(\widehat{C}^{-1}, e_1)\mathcal{K}(C^{-1}, e_1)^{-1}$. □

We now consider the details of implementing an SZ iteration for the symplectic pencil (11). The spectral transformation function q should be chosen as discussed in Section 3. Computation of $q(M^{-1}N)e_1$ does not require explicit inversion of M. As $M^{-1}N$ is of butterfly form, we can use directly the formula (16) in order to determine $q(M^{-1}N)e_1$. Applying Z_1^{-1}

to $M - \lambda N$ introduces a bulge. The main part of the iteration is a bulge chasing process that restores MZ_1^{-1} and NZ_1^{-1} to their original forms.

We refrain from discussing the bulge chasing process in detail as the implicit SZ step is analogous to the implicit QZ step. Instead we will present an algorithm for reducing a symplectic matrix pencil $M - \lambda N$ where M and N are both symplectic to a reduced pencil of the form (11). Such an algorithm can be used as a building block of the SZ step. The algorithm uses the following elementary symplectic transformations:

- symplectic Givens transformation

$$G(k,c,s) = \left[\begin{array}{ccc|ccc} I_{k-1} & & & & & \\ & c & & & s & \\ & & I_{n-k} & & & \\ \hline & & & I_{k-1} & & \\ & -s & & & c & \\ & & & & & I_{n-k} \end{array}\right],$$

- symplectic Householder transformation

$$H(k,v) = \left[\begin{array}{cc|cc} I_{k-1} & & & \\ & P & & \\ \hline & & I_{k-1} & \\ & & & P \end{array}\right], \text{ where } P = I_{n-k+1} - 2\frac{vv^T}{v^Tv},$$

- symplectic Gauss transformation

$$L(k,c,d) = \left[\begin{array}{cccc|cccc} I_{k-2} & & & & & & & \\ & c & & & & & d & \\ & & c & & & d & & \\ & & & I_{n-k} & & & & \\ \hline & & & & I_{k-2} & & & \\ & & & & & c^{-1} & & \\ & & & & & & c^{-1} & \\ & & & & & & & I_{n-k} \end{array}\right].$$

- symplectic Gauss transformation of type II,

$$\widetilde{L}(k,c,d) = \left[\begin{array}{ccc|ccc} I_{k-1} & & & & & \\ & c & & & d & \\ & & I_{n-k} & & & \\ \hline & & & I_{k-1} & & \\ & & & & c^{-1} & \\ & & & & & I_{n-k} \end{array}\right].$$

The symplectic Givens and Householder transformations are orthogonal, while the symplectic Gauss transformations are nonorthogonal. Algorithms to compute the entries of the abovementioned transformations can be found, e.g., in [27] and [15]. The Gaussian transformations can be computed such that among all possible transformations satisfying the same purpose, the one with the minimal condition number is chosen.

Zeros in the rows of M and N will be introduced by applying one of the above mentioned transformations from the right, while zeros in the columns will be introduced by applying the transformations from the left. The basic idea of the algorithm can be summarized as follows:

- bring the first column of N into the desired form
- now iterate for $j = 1$ to n
  - bring the jth row of M into the desired form
  - bring the jth column of M into the desired form
  - bring the $(n+j)$th column of N into the desired form
  - bring the jth row of N into the desired form

The remaining rows and columns in M and N that are not explicitly touched during the process will be in the desired form due to the symplectic structure. For an 8×8 symplectic matrix pencil, the elimination process can be summarized as in the following scheme.

$$
\left[\begin{array}{cccc|cccc}
\star & 8, G^{post} & 7, G^{post} & 6, G^{post} & \star & 10, L^{post} & 9, H^{post} & 9, H^{post} \\
13, G^{pre} & \star & 26, G^{post} & 25, G^{post} & \hat{0}_{13} & \star & 28, L^{post} & 27, H^{post} \\
12, G^{pre} & 30, G^{pre} & \star & 40, G^{post} & \hat{0}_{12} & \hat{0}_{30} & \star & 41, L^{post} \\
11, G^{pre} & 29, G^{pre} & 42, G^{pre} & \star & \hat{0}_{11} & \hat{0}_{29} & \hat{0}_{42} & \star \\
\hline
16, \widetilde{L}^{pre} & \hat{0}_{15} & \hat{0}_{15} & \hat{0}_{15} & \star & \hat{0}_{15} & \hat{0}_{15} & \hat{0}_{15} \\
15, L^{pre} & 33, \widetilde{L}^{pre} & \hat{0}_{32} & \hat{0}_{32} & \hat{0}_{15} & \star & \hat{0}_{32} & \hat{0}_{32} \\
14, H^{pre} & 32, L^{pre} & 44, \widetilde{L}^{pre} & \hat{0}_{43} & \hat{0}_{14} & \hat{0}_{32} & \star & \hat{0}_{43} \\
14, H^{pre} & 31, H^{pre} & 43, L^{pre} & 47, \widetilde{L}^{pre} & \hat{0}_{14} & \hat{0}_{31} & \hat{0}_{43} & \star
\end{array}\right]
$$
$$
-\lambda \left[\begin{array}{cccc|cccc}
4, G^{pre} & \hat{0}_{5} & \hat{0}_{5} & \hat{0}_{5} & \star & \hat{0}_{5} & \hat{0}_{5} & \hat{0}_{5} \\
3, G^{pre} & 23, G^{post} & 22, G^{post} & 21, G^{post} & 19, G^{pre} & \star & 24, H^{post} & 24, H^{post} \\
2, G^{pre} & \hat{0}_{24} & 38, G^{post} & 37, G^{post} & 18, G^{pre} & 35, G^{pre} & \star & 39, H^{post} \\
1, G^{pre} & \hat{0}_{24} & \hat{0}_{39} & 46, G^{post} & 17, G^{pre} & 34, G^{pre} & 45, G^{pre} & \star \\
\hline
\star & \hat{0}_{23} & \hat{0}_{22} & \hat{0}_{21} & \star & \star & \hat{0}_{24} & \hat{0}_{24} \\
5, H^{pre} & \star & \hat{0}_{38} & \hat{0}_{37} & \star & \star & \star & \hat{0}_{39} \\
5, H^{pre} & \hat{0}_{24} & \star & \hat{0}_{46} & 20, H^{pre} & \star & \star & \star \\
5, H^{pre} & \hat{0}_{24} & \hat{0}_{39} & \star & 20, H^{pre} & 36, H^{pre} & \star & \star
\end{array}\right]
$$

The capital letters indicate the type of elimination matrix used to eliminate the entry (G used for a symplectic Givens, H for a symplectic Householder, and $L, \widetilde{L}$ for a symplectic Gauss transformation). The upper index indicates whether the elimination is done by pre- or postmultiplication. The numbers indicate the order in which the entries are annihilated. A zero that is not created by explicit elimination but because of the symplectic structure, is denoted by $\hat{0}$. Its index indicates which transformation causes this zero. E.g., if after five steps the first column of N is denoted by $[0\ 0\ 0\ 0\ n_{51}\ 0\ 0\ 0]^T, n_{51} \neq 0$, and the first row by $[0 \star \star \star\ n_{15} \star \star \star]$, then as N is symplectic throughout the whole reduction process, from $N^T J N = J$ we have $e_1^T(N^T J N) = e_1^T J = e_{n+1}$, or in other words

$$-n_{51}[0\ n_{12}\ n_{13}\ n_{14}\ n_{15}\ n_{16}\ n_{17}\ n_{18}] = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0].$$

Hence, the entries of the first row of N have to be zero, only the $(n+1,1)$ entry is nonzero.

In the following an algorithm for reducing a symplectic matrix pencil $M - \lambda N$, where M and N are both symplectic to a reduced pencil of the form (11) is given. In order to keep the presentation as simple as possible, no pivoting is introduced here, but should be used in an actual implementation. This algorithm can be used to derive a bulge chasing process, e.g., for a quadruple or double shift SZ step.

Algorithm 1 (Transformation to butterfly pencil):

Given a symplectic matrix pencil $M - \lambda N$, where $M, N \in \mathbb{R}^{2n \times 2n}$ are both symplectic matrices, the following algorithm computes symplectic matrices S and Z such that $S(M - \lambda N)Z$ is a symplectic pencil of the form (11). M is overwritten by SMZ and N by SNZ.

$Z = I_{2n}$; $S = I_{2n}$;
for $k = n : -1 : 1$
  compute G such that $(GN)_{k,1} = 0$.
  $N = GN$; $M = GM$; $S = GS$;
end
compute H such that $(HN)_{n+2:2n,1} = 0$.
$N = HN$; $M = HM$; $S = HS$;
for $j = 1 : n$
  if $j > 1$
    for $k = n : -1 : j$
      compute G such that $(NG)_{j,k} = 0$.
      $N = NG$; $M = MG$; $Z = ZG$;
    end
  end
  if $j < n$
    if $j > 1$
      compute H such that $(NH)_{j,j+n+1:2n} = 0$.
      $N = NH$; $M = MH$; $Z = ZH$;

```
        end
        for k = n : -1 : j + 1
            compute G such that (MG)_{j,k} = 0.
            N = NG;   M = MG;   Z = ZG;
        end
    end
    if j < n - 1
        compute H such that (MH)_{j,j+2+n:2n} = 0.
        N = NH;   M = MH;   Z = ZH;
    end
    if j < n
        compute L such that (ML)_{j,j+n+1} = 0.
        Transformation might not exist!!
        N = NL;   M = ML;   Z = ZL;
        for k = n : -1 : j + 1
            compute G such that (GM)_{k,j} = 0.
            N = GN;   M = GM;   S = GS;
        end
    end
    if j < n - 1
        compute H such that (HM)_{j+2+n:2n,j} = 0.
        N = HN;   M = HM;   S = HS;
    end
    if j < n
        compute L such that (LM)_{j+1+n,j} = 0.
        Transformation might not exist!!
        N = LN;   M = LM;   S = LS;
    end
    compute \tilde{L} such that (\tilde{L}M)_{j+n,j} = 0.
    Transformation might not exist!!
    N = \tilde{L}N;   M = \tilde{L}M;   S = \tilde{L}S;
    if j < n
        for k = n : -1 : j + 1
            compute G such that (GN)_{k,j+n} = 0.
            N = GN;   M = GM;   S = GS;
        end
    end
    if j < n - 1
        compute H such that (HN)_{j+2+n:2n,j+n} = 0.
        N = HN;   M = HM;   S = HS;
    end
end
```

REMARK 5. a) A careful implementation of this process as a bulge chasing process will just work with the $4n-1$ parameters and some additional variables instead of with the matrices M and N.

b) It is possible to incorporate pivoting into the process in order to make it more stable. E.g., in the process as described the jth column of M will be brought into the desired form. Due to symplecticity, the $(n+j)$th column of M will then be of desired form as well. One could just as well attack the $(n+j)$th column of M, the jth column will then be of desired form due to symplecticity.

c) The use of symplectic transformations throughout the reduction process assures that the factors M and N remain symplectic separately. If the objective is only to preserve the symplectic property of the pencil ($MJM^T = NJN^T$), one has greater latitude in the choice of transformations. Only the right-hand (Z) transformations need to be symplectic; the left (S) transforms can be more general as long as they are regular.

In [3] an algorithm for reducing a symplectic matrix pencil $M - \lambda N$, where M and N are both symplectic, to the reduced matrix pencil of the form (10) is developed. That elimination process uses the same elementary symplectic transformations as the process described here. The algorithm can also be used to chase the bulge created in $MZ_1^{-1} - \lambda NZ_1^{-1}$. It turns out that in that setting, for a double or quadruple shift step, there are slightly more nonzero entries in the matrices M and N than there are in the setting discussed here. This implies that more elementary symplectic transformations have to be used. In particular, additional $n-1$ symplectic Gaussian transformation of type II have to be used, which are not needed in the above bulge chasing process.

By applying a sequence of double or quadruple SZ steps to the symplectic matrix pencil $M - \lambda N$ of the form (11) it is possible to reduce the symmetric tridiagonal matrix T in the lower right block of N to quasi-diagonal form with 1×1 and 2×2 blocks on the diagonal. The eigenproblem decouples into a number of simple 2×2 or 4×4 eigenproblems. In doing so, it is necessary to monitor T's subdiagonal in order to bring about decoupling whenever possible. The complete process is as follows.

Algorithm 2 (SZ algorithm for butterfly pencils):

Given a symplectic matrix pencil $M - \lambda N$ of the form (11), the following algorithm computes symplectic matrices Z and S such that for $\widehat{M} := S^{-1}MZ$ and $\widehat{N} := S^{-1}NZ$, $\widehat{B} := \widehat{M}^{-1}\widehat{N}$ is a symplectic matrix in which the $(1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$ blocks are each block-diagonal where all blocks are either 1×1 or 2×2. Moreover, the block structure for all four blocks of $\widehat{B}$ is the same. Thus the eigenproblem for $\widehat{B}$ decouples into 2×2 and 4×4 symplectic eigenproblems. M is overwritten by $S^{-1}MZ$ and N by $S^{-1}NZ$.

repeat until $q = n$

set all sub- and superdiagonal entries $t_{i,i-1} = t_{i-1,i}$ in T to zero that satisfy

$$|t_{i,i-1}| \leq \epsilon(|t_{i-1,i-1}| + |t_{ii}|)$$

find the largest nonnegative q and the smallest nonnegative p such that if

$$N = \left[\begin{array}{ccc|ccc} & & & -I_p & & \\ & & & & -I_{n-p-q} & \\ & & & & & -I_q \\ \hline I_p & & & T_{11} & & \\ & I_{n-p-q} & & & T_{22} & \\ & & I_q & & & T_{33} \end{array}\right],$$

where $T_{11} \in \mathbb{R}^{p \times p}, T_{22} \in \mathbb{R}^{(n-p-q)\times(n-p-q)}$, and $T_{33} \in \mathbb{R}^{q \times q}$, then T_{33} is block diagonal with 2×2 and 4×4 blocks and T_{22} is unreduced symmetric tridiagonal.

partition M conformably

$$M = \left[\begin{array}{ccc|ccc} X_{11} & & & Y_{11} & & \\ & X_{22} & & & Y_{22} & \\ & & X_{33} & & & Y_{33} \\ \hline & & & X_{11}^{-1} & & \\ & & & & X_{22}^{-1} & \\ & & & & & X_{33}^{-1} \end{array}\right],$$

where $X_{11}, Y_{11} \in \mathbb{R}^{p \times p}, X_{22}, Y_{22} \in \mathbb{R}^{(n-p-q)\times(n-p-q)}$, and $X_{33}, Y_{33} \in \mathbb{R}^{q \times q}$.

if $q < n$

perform a double or quadruple shift SZ step using Algorithm 1 on

$$\left[\begin{array}{c|c} X_{22} & Y_{22} \\ \hline & X_{22}^{-1} \end{array}\right] - \lambda \left[\begin{array}{c|c} & -I_{n-p-q} \\ \hline I_{n-p-q} & T_{22} \end{array}\right]$$

update M and N accordingly

end

end

REMARK 6. Example 2 in Section 5 indicates that eigenvalues of symplectic butterfly pencils computed by this algorithm are significantly more accurate than those computed by the SR algorithm and often competitive to those computed by the QR algorithm. Hence if a symplectic matrix/matrix pencil is given in parameterized form as in the context of the symplectic Lanczos algorithm [7] one should not form the corresponding butterfly matrix, but compute the eigenvalues via the SZ algorithm.

5. Numerical Examples

The SR and SZ algorithms for computing the eigenvalues of symplectic matrices/matrix pencils as discussed in Sections 3 and 4 were implemented in MATLAB Version 5.1. Numerical experiments were performed on a SPARC Ultra 1 creator workstation.

In order to detect deflation, subdiagonal elements were declared to be zero during the iterations when a condition of the form

$$|h_{p+1,p}| \leq 10 \cdot n \cdot eps(|h_{pp}| + |h_{p+1,p+1}|)$$

was fulfilled, where the dimension of the problem is $2n \times 2n$ and $eps \approx 2.2204 * 10^{-16}$ is MATLAB's floating point relative accuracy.

The experiments presented here will illustrate the typical behavior of the proposed algorithms. For a general symplectic matrix or a symplectic matrix pencil with both matrices symplectic, our implementation first reduces the matrix/matrix pencil to butterfly form/a pencil of the form (11) and then iterates using only quadruple shift steps. The shifts are chosen according to the generalized Rayleigh strategy discussed in Section 3. Tests were run using

- randomly generated symplectic matrices/matrix pencils;
- randomly generated parameters $a_1, \ldots, a_n$, $b_1, \ldots, b_n$, $c_1, \ldots, c_n$, $d_2, \ldots, d_n \in \mathbb{R}$ from which a butterfly matrix and the corresponding symplectic matrix pencil were constructed;
- examples from the benchmark collection [8];
- the examples discussed in [19].

Our observations have been the following.

- The methods did always converge; not once did we encounter an example where an exceptional SR/SZ step with a random shift was necessary (although, no doubt, such an example can be constructed).
- Cubic convergence can be observed.
- The SZ algorithm is considerably better than the SR algorithm in computing the eigenvalues of a parameterized symplectic matrix/matrix pencil.
- The number of (quadruple-shift) iterations needed for convergence for each eigenvalue is about 2/3.

Example 1: For the first set of tests, 100 symplectic matrices for each of the dimensions $2n \times 2n$ for $n = 5 : 5 : 50$ were generated by computing the SR decomposition of random $2n \times 2n$ matrices:

```
A = rand(2*n);  [M,R] = sr(A);
```

where M is symplectic and R is J-triangular such that $A = MR$. Some of the results we obtained are summarized in Tables 1 and 2. In each table, the first column indicates the size of the problem.

$2*n$	$\max(\|M^TJM - J\|_2)$	$\|S^{-1}MS - B\|_2$	condmax	iter
10	$\mathcal{O}(10^{-12})$	$\mathcal{O}(10^{-14})$ – $\mathcal{O}(10^{-7})$	$\mathcal{O}(10^{3})$	0.698
20	$\mathcal{O}(10^{-12})$	$\mathcal{O}(10^{-13})$ – $\mathcal{O}(10^{-6})$	$\mathcal{O}(10^{4})$	0.716
30	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-12})$ – $\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{4})$	0.716
40	$\mathcal{O}(10^{-12})$	$\mathcal{O}(10^{-10})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{5})$	0.683
50	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-10})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{4})$	0.678
60	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-10})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{6})$	0.658
70	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-9})$ – $\mathcal{O}(10^{-3})$	$\mathcal{O}(10^{5})$	0.661
80	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-9})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{5})$	0.653
90	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-9})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{6})$	0.656
100	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-9})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{6})$	0.656

Table 1: first set of tests: SR algorithm

As the generated matrices M are only symplectic modulo roundoff errors, symplecticity was tested via $\|M^TJM - J\|$ for all examples. The second column of Table 1 reports the maximal norm observed for each dimension. It is obvious that for increasing dimension, symplecticity is more and more lost. Hence, we may expect our algorithm to have some difficulties performing well, as its theoretical foundation is the symplecticity of the matrix/matrix pencil treated. The SR algorithm computes a symplectic matrix S and a symplectic matrix B such that in exact arithmetic, $S^{-1}MS = B$ is of butterfly-like form and B decouples into a number of 2×2 and 4×4 subproblems. In order to see how well the computed S and B obey this relation, $\|S^{-1}MS - B\|_2$ was computed for each example, and the maximal and minimal value of these norms for each dimension is reported in the third column of Table 1. In the course of the iterations, symplectic Gaussian transformations have to be used. All other involved transformations are orthogonal. These are known to be numerically stable. Hence, the Gaussian transformations are the only source for instability. The column 'condmax' of the table displays the maximal condition number of all Gaussian transformations applied during all 100 examples of each dimension. The condition number of the Gaussian transformations were never too large (i.e., exceeding the tolerance threshold, chosen here as $1/eps$), hence no exceptional SR step with a random shift was required. The last column of Table 1 gives the average number of iterations needed for convergence of each eigenvalue. This number tends to be around 2/3 iterations per eigenvalue.

$2*n$	$\max(relerr)$	$\min(relerr)$	average$(relerr)$	average
10	$\mathcal{O}(10^{-9})$	$\mathcal{O}(10^{-12})$	$\mathcal{O}(10^{-15})$ – $\mathcal{O}(10^{-9})$	$2.4 * 10^{-11}$
20	$\mathcal{O}(10^{-6})$	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-14})$ – $\mathcal{O}(10^{-7})$	$2.8 * 10^{-9}$
30	$\mathcal{O}(10^{-7})$	$\mathcal{O}(10^{-10})$	$\mathcal{O}(10^{-13})$ – $\mathcal{O}(10^{-8})$	$2.8 * 10^{-9}$
40	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-12})$ – $\mathcal{O}(10^{-6})$	$2.8 * 10^{-8}$
50	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-12})$ – $\mathcal{O}(10^{-6})$	$1.6 * 10^{-8}$
60	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-12})$ – $\mathcal{O}(10^{-6})$	$4.1 * 10^{-8}$
70	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-6})$	$1.0 * 10^{-7}$
80	$\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-6})$	$2.5 * 10^{-8}$
90	$\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-6})$	$5.1 * 10^{-8}$
100	$\mathcal{O}(10^{-6})$	$\mathcal{O}(10^{-13})$	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-7})$	$2.2 * 10^{-8}$

Table 2: first set of tests — SR algorithm

Table 2 reports on the accuracy of the computed eigenvalues. For this purpose, the MATLAB function `eig` was called in order to solve the 2×2 and 4×4 subproblems of B to generate a list of eigenvalues computed via the SR algorithm. These eigenvalues were compared to the eigenvalues of M obtained via `eig`; the latter eigenvalues were considered to be the 'exact' eigenvalues. This assumption is justified for the randomly generated examples using as a criterion $\sigma_{\min}(M - \lambda I_{2n})$ which turns out to be of order eps for eigenvalues computed via `eig` while for the eigenvalues computed via the SR algorithm, this 'residual' is larger by an order $\mathcal{O}(10^d)$ where d is the number of digits lost as indicated by our relative error measure.

The column $\max(relerr)$ reports the maximal relative error so obtained, the column $\min(relerr)$ the minimal relative error. In order to get an idea about the average relative accuracy obtained, we computed for each example the arithmetic mean; the range in which these values were found is given in column 'average$(relerr)$'. Finally, in order to compare our results with those given in [3], we computed the average relative accuracy for all examples of each dimension using the arithmetic mean of all examples for each dimension. In [3], these averages are given for dimensions $10, 20$, and 40; our results confirm those results.

The same kind of test runs was performed for randomly generated symplectic matrix pencils $M - \lambda N$ where M and N are both symplectic using the SZ algorithm. M and N were generated analogous to M as above. Note that this introduces more difficulties here than above; our SZ algorithm makes use of the fact that $M^T JM = N^T JN = J$; all of these equalities are violated. But despite this, the SZ algorithm performs as well as the SR algorithm. Our implementation of the SZ algorithm first reduces M and N to the pencil form (11) and than iterates using only quadruple shift steps where the shifts are chosen according to the generalized Rayleigh strategy.

In the following two tables we report the same information as in the two tables presented for the SR algorithm. This time we give the data only for dimensions 30 and 50, in order to save some space but to support our claim that the SZ algorithm works as well as the SR algorithm. The SZ algorithm computes symplectic matrices S, Q, $\widetilde{M}$ and $\widetilde{N}$ such that $S(M - \lambda N)Q = \widetilde{M} - \lambda\widetilde{N}$ and the pencil $\widetilde{M} - \lambda\widetilde{N}$ decouples into a number of 2×2 and 4×4 subproblems. The eigenvalues of these small subproblems were computed using the MATLAB function `eig` and compared to the eigenvalues obtained via `eig(M,N)`.

$2 * n$	$\lVert SMQ - \widetilde{M} \rVert_2$	$\lVert SNQ - \widetilde{N} \rVert_2$	condmax	iter
30	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-6})$	$\mathcal{O}(10^{-11})$ – $\mathcal{O}(10^{-7})$	$\mathcal{O}(10^{5})$	0.574
50	$\mathcal{O}(10^{-10})$ – $\mathcal{O}(10^{-4})$	$\mathcal{O}(10^{-10})$ – $\mathcal{O}(10^{-5})$	$\mathcal{O}(10^{5})$	0.546

Table 3: first set of tests — SZ algorithm

$2 * n$	max($relerr$)	min($relerr$)	aver($relerr$)	average
30	$\mathcal{O}(10^{-7})$	$\mathcal{O}(10^{-9})$	$\mathcal{O}(10^{-13})$ – $\mathcal{O}(10^{-8})$	$9.7 * 10^{-10}$
50	$\mathcal{O}(10^{-6})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-12})$ – $\mathcal{O}(10^{-7})$	$1.0 * 10^{-8}$

Table 4: first set of tests — SZ algorithm

Example 2: A second set of tests was performed to see whether the SR or the SZ algorithm performs better once the symplectic matrix/matrix pencil is reduced to parameterized form. For this purpose, parameters $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n, d_2, \ldots, d_n \in \mathbb{R}$ were generated, from which a symplectic pencil $L - \lambda N$ and the corresponding butterfly matrix M were constructed as in (4), (9), and (17), respectively.

The examples generated this way do not suffer from loss of symplecticity, any matrix pencil $L - \lambda N$ of the above form is symplectic. Furthermore no initial reduction to butterfly form is necessary here; L, N, and M are already in parameterized form. For each $n = 5 : 5 : 50$, one hundred sets of parameters were generated, L, N, and M were constructed, and the SR/SZ algorithm was used to compute the eigenvalues. As before, the 2×2 and 4×4 subproblems were solved using `eig`. The eigenvalues so obtained were compared to eigenvalues computed via `eig(M)`. Table 5 reports some of the results so obtained, using the same notation as above.

$2*n$	SR max($relerr$)	SR average	SZ max($relerr$)	SZ average	SR/SZ iter
10	$\mathcal{O}(10^{-11})$	$1.7*10^{-13}$	$\mathcal{O}(10^{-13})$	$1.6*10^{-15}$	0.60
20	$\mathcal{O}(10^{-9})$	$8.4*10^{-12}$	$\mathcal{O}(10^{-12})$	$5.5*10^{-15}$	0.64
30	$\mathcal{O}(10^{-12})$	$3.5*10^{-14}$	$\mathcal{O}(10^{-13})$	$2.3*10^{-15}$	0.65
40	$\mathcal{O}(10^{-7})$	$6.9*10^{-11}$	$\mathcal{O}(10^{-13})$	$2.7*10^{-15}$	0.65
50	$\mathcal{O}(10^{-11})$	$2.5*10^{-14}$	$\mathcal{O}(10^{-14})$	$2.7*10^{-15}$	0.64
60	$\mathcal{O}(10^{-8})$	$3.6*10^{-12}$	$\mathcal{O}(10^{-11})$	$1.8*10^{-14}$	0.64
70	$\mathcal{O}(10^{-9})$	$2.1*10^{-12}$	$\mathcal{O}(10^{-13})$	$3.4*10^{-15}$	0.63
80	$\mathcal{O}(10^{-9})$	$8.1*10^{-13}$	$\mathcal{O}(10^{-13})$	$3.5*10^{-15}$	0.64
90	$\mathcal{O}(10^{-8})$	$5.4*10^{-12}$	$\mathcal{O}(10^{-13})$	$3.6*10^{-15}$	0.63
100	$\mathcal{O}(10^{-8})$	$2.5*10^{-11}$	$\mathcal{O}(10^{-12})$	$5.3*10^{-15}$	0.63

Table 5: second set of tests

As expected, the examples showed the same convergence behavior no matter which algorithm was used. That is, the number of iterations needed for convergence was almost the same, the maximal condition number of the Gaussian transformations were the same. The maximal relative error observed for the different examples was bigger for the SR algorithm than for the SZ algorithm. These results indicate that the SZ algorithm computes more accurate eigenvalues than the SR algorithm.

Example 3: Tests with examples from the benchmark collection [8] were performed. None of these examples result in a symplectic pencil $L - \lambda N$ with symplectic L and N matrices. Hence, whenever possible, a symplectic matrix M was formed from the given data. Table 6 presents the results obtained applying the SR algorithm to M. Again, the relative error in the eigenvalues was computed by comparing the eigenvalues computed via the SR algorithm with those computed via `eig`. The first column of the table gives the number of the example as given in [8]. The next columns display the dimension of the problem, the maximal and minimal relative errors for the computed eigenvalues, the maximal condition number used, and the total number of iterations needed to achieve convergence.

Example Number	$2*n$	$\max(relerr)$	$\min(relerr)$	condmax	number of iterations
1	4			2.5	0
2	4			25.3	0
6	8	$1.0*10^{-14}$	$4.0*10^{-15}$	6.4	4
7	8	$2.2*10^{-12}$	$6.8*10^{-14}$	$5.4*10^{2}$	3
8	8	$2.4*10^{-11}$	$8.5*10^{-16}$	8.4	3
9	10	$9.3*10^{-13}$	$4.1*10^{-16}$	20.4	5
11	18	$1.2*10^{-2}$	$7.1*10^{-12}$	$7.9*10^{3}$	6

Table 6: Example 3

For the first two examples, no SR iteration was necessary: after the initial reduction to butterfly form, the problem either decoupled into two 2×2 subproblems or the eigenvalues could be read off directly. The relative error of the so computed eigenvalues is of order $\mathcal{O}(eps)$. For Example 8 of [8] the eigenvalues computed via the SR algorithm were better than those computed via `eig`. This was checked via the smallest singular value $\sigma_{\min}$ of $(M-\lambda I)$ for the eigenvalues λ computed via `eig` as well as via the SR algorithm. It turns out that $\sigma_{min}(M-\lambda_{SR}I)$ is smaller then $\sigma_{min}(M-\lambda_{\texttt{eig}}I)$. For Example 11 from [8] one should note that the matrix M there is only almost symplectic, that is, $\|M^TJM-J\|_2\approx 1*10^{-10}$ and the condition number of M is given by $\kappa(M)\approx 1.6*10^6$.

Example 4: Flaschka, Mehrmann, and Zywietz report in [19] that the SR algorithm for symplectic J-Hessenberg matrices does not perform satisfactory due to roundoff errors. They present two examples to demonstrate the behavior of the SR algorithm for symplectic J-Hessenberg matrices. The first example presented is a symplectic matrix with the eigenvalues $5, 1/5, 3\pm 4i, 0.12\pm 0.16i$; the matrix itself is given in [19]. It is reported in [19] that complete deflation was observed after 19 iteration, but the final iteration matrix was far from being symplectic. The maximal condition number used during the iterations was $6.4*10^3$.

Our algorithm first reduced the symplectic matrix to butterfly form (this is denoted here as iteration step 0), then two iterations were needed for convergence. Moreover, cubic convergence can be observed by monitoring the parameters d_j during the course of the iteration, as they indicate deflation. Table 7 reports the values for the d_j's after each iteration.

As can be seen, it takes only two iterations for d_3 to become zero with respect to machine precision. Decoupling is possible and the problem splits

iteration	d_2	d_3
0	1.8576	$2.389 * 10^{-2}$
1	-0.2783	$-2.117 * 10^{-5}$
2	-4.3422	$2.242 * 10^{-16}$

Table 7: Example 4 – first test

into a 2×2 and a 4×4 subproblem. The observed maximal condition number was 57.39.

The second example discussed in [19] is a 12×12 symplectic matrix with the eigenvalues $1 \pm 1i, 0.5 \pm 0.5i, 2 \pm 2i, 0.25 \pm 0.25i, 3 \pm 4i, 0.12 \pm 0.16i$. Here, a symplectic diagonal matrix with these eigenvalues on the diagonal was constructed and a similarity transformation with a randomly generated orthogonal symplectic matrix was performed to obtain a symplectic matrix M. The implementation presented in [19] first reduces this matrix to J-Hessenberg form, then a double shift SR step with the perfect shift $3 \pm 4i$ is performed. This resulted in deflation and good approximation of these eigenvalues, but symplecticity was lost completely.

iteration	deflation?	size of deflated subproblem	d_3	d_5
0	no		$1.07 * 10^{0}$	$0.91 * 10^{0}$
1	no		$1.29 * 10^{-1}$	$-8.50 * 10^{-2}$
2	no		$-5.30 * 10^{-2}$	$1.37 * 10^{-4}$
3	no		$-1.49 * 10^{-3}$	$-1.26 * 10^{-12}$
4	yes	4×4	$2.18 * 10^{-3}$	$-3.40 * 10^{-24}$
5	no		$-4.36 * 10^{-10}$	
6	yes	4×4	$3.09 * 10^{-25}$	

Table 8: Example 4 – second test

Our algorithm again first reduced the symplectic matrix to butterfly form, then six iterations were needed for convergence. As before, cubic convergence can be observed by monitoring the parameters d_j during the course of the iteration. Table 8 reports the values for the d_j's after each iteration as well as whether deflation occurred and whether a 2×2 or a 4×4 subproblem was deflated.

The observed maximal condition number was 73.73.

Example 5: We also tested an implementation of the SR algorithm using a standard polynomial p instead of a Laurent polynomial q to drive the SR step. In cases where no trouble arose, both algorithms performed similarly. That is, although the version that uses the Laurent polynomial uses fewer arithmetic operations, both versions of the algorithm needed the same number of iterations for convergence, and the accuracy of the computed eigenvalues was similar. But, as indicated in our discussion in Section 3, using the standard polynomial might sometimes cause some problems. Using the Laurent polynomial to drive the SR step, the algorithm behaved as expected. Convergence of even-dimensional subspaces occurred, which resulted in the convergence of some of the d_k's to zero. But when working with standard polynomials to drive the SR step, one might observe convergence of a_1 to zero and stagnation of the algorithm afterwards. This will be illustrated here by the following example. We generated a 30×30 symplectic matrix using the parameters $a_1, \ldots, a_{15}, b_1, \ldots, b_{15}$, $c_1, \ldots, c_{15}$, and $d_2, \ldots, d_{15}$ as given in Table 9.

	a	b	c	d
1	0.76880950325	0.82064368228	0.06824661097	
2	0.96970170497	0.97047237460	0.96412426837	0.84800944806
3	0.71479723187	0.48692499554	0.20765658836	0.72860019101
4	0.78196184196	0.81746853554	0.16111822555	0.95509863327
5	0.23756508204	0.64157116784	0.63822138259	0.65635111059
6	0.19573076378	0.30634935951	0.00022817289	0.74230513350
7	0.26321391517	0.66093213223	0.33563294335	0.34496601390
8	0.71378506459	0.35801711338	0.27509982146	0.88402194967
9	0.97759973943	0.93819943010	0.04452752039	0.34724408649
10	0.63712194084	0.48766697476	0.09389649759	0.05947668054
11	0.54592415509	0.09099035774	0.40999739977	0.71841459107
12	0.84805722441	0.67383411686	0.81689231949	0.95821429290
13	0.80209765848	0.51488031898	0.87051707180	0.15683486507
14	0.66830641006	0.22157934638	0.02255512045	0.41635310614
15	0.67098263396	0.72500937095	0.72717698369	0.09403486897

Table 9: Example 5 – parameters

The resulting symplectic matrix M has only two real eigenvalues:

$$\mu = 1.97700698420 \qquad \text{and} \quad \mu^{-1} = 0.50581510737.$$

The twenty-eight complex eigenvalues occur in pairs $(\lambda, \overline{\lambda})$ where $\lambda \in \mathbb{C}, |\lambda| = 1$.

The following table reports the values of a_1, b_1, and c_1 in the course of the iteration when the SR step is driven by a standard polynomial (the first column indicates the number of iterations). The choice of shifts is as before.

	a_1	b_1	c_1
1	$1.8 * 10^{-1}$	1.98410591	2.2791
2	$1.6 * 10^{-2}$	1.97948316	30.8977
3	$2.2 * 10^{-3}$	1.97726186	233.4375
4	$6.6 * 10^{-4}$	1.97714161	761.5576
5	$3.6 * 10^{-4}$	1.97718534	1397.3210
6	$4.4 * 10^{-4}$	1.97728897	1149.0454
7	$9.1 * 10^{-5}$	1.97706343	5575.8521
8	$1.0 * 10^{-5}$	1.97701113	49627.9530
9	$7.8 * 10^{-7}$	1.97700726	643154.9327

Table 10: Example 5 – standard polynomial

Already after the first iteration the largest eigenvalue μ is emerging as b_1. During the subsequent iterations, b_1 converges towards this eigenvalue while a_1 converges to zero. The growth of c_1 reflects the ill conditioning of the transforming matrices. At the bottom of the matrix, deflations take place: after iteration 5, a 2×2 subproblem is decoupled, after iteration 7 a 4×4, after iteration 11 and 12 a 2×2, and after iteration 16 another 4×4 subproblem is decoupled. At that point a_1 is less than eps so that a 2×2 subproblem can be deflated at the top which corresponds to the pair of real eigenvalues, $b_1 \approx \mu$ and the $(n+1, n+1)$ entry of the iteration matrix is approximately equal to μ^{-1}. The resulting 14×14 subproblem has only complex pairs of eigenvalues on the unit circle. Parametrising this subproblem, one observes that three of the six parameters d_j are of order $\sqrt{eps}$, the other three are of order 1. This does not change during subsequent iterations, no convergence is achieved (the required tolerance for deflation is of order eps).

Using a Laurent polynomial to drive the SR step, the process converges after 22 iterations, a_1 does not converge to zero. All eigenvalues are computed accurately ($\max(relerr) = \mathcal{O}(10^{-15})$).

6. Concluding remarks

In this paper we have presented SR and SZ algorithms for the symplectic butterfly eigenproblem. The SR algorithm works with the $8n-4$ nonzero entries of the butterfly matrix. Laurent polynomials are used to drive the SR step as this results in a smaller bulge and hence less arithmetic operations than using standard shift polynomials. Forcing the symplectic structure of the iterates whenever necessary, the algorithm works better than the SR algorithm for symplectic J-Hessenberg matrices proposed in [19]. The SZ algorithm works with the $4n-1$ parameters that determine the butterfly matrix. It reduces a symplectic matrix pencil of the form

$$M - \lambda N \;=\; \begin{bmatrix} \ddots & \ddots \\ 0 & \ddots \end{bmatrix} - \lambda \begin{bmatrix} 0 & -I \\ I & \ddots \end{bmatrix},$$

where M and N are both symplectic, to a form that decouples into a number of 2×2 and 4×4 symplectic eigenproblems. The algorithm ensures that the matrices M and N remain symplectic separately.

Future research will address the development of an SZ algorithm that works for symplectic matrix pencils $M-\lambda N$, where $MJM^T = NJN^T \neq J$. It is still an open research problem to which reduced form such a pencil can be reduced when M and N are singular.

REFERENCES

1 E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov and D. Sorensen. *LAPACK Users' Guide.* SIAM Philadelphia, PA, second edition, 1994.

2 G. Banse. Eigenwertverfahren für symplektische Matrizen zur Lösung zeitdiskreter optimaler Steuerungsprobleme. *Z. Angew. Math. Mech.*, 75(Suppl. 2):615–616, 1995.

3 G. Banse. *Symplektische Eigenwertverfahren zur Lösung zeitdiskreter optimaler Steuerungsprobleme.* Dissertation, Universität Bremen, Fachbereich 3 – Mathematik und Informatik, Bremen, Germany, 1995.

4 G. Banse. Condensed forms for symplectic matrices and symplectic pencils in optimal control. *Z. Angew. Math. Mech.*, 76(Suppl. 3):375–376, 1996.

5 G. Banse and A. Bunse-Gerstner. A condensed form for the solution of the symplectic eigenvalue problem. In U. Helmke, R. Menniken, and J. Sauer, editors, *Systems and Networks: Mathematical Theory and Applications*, pages 613–616. Akademie Verlag, 1994.

6 P. Benner and H. Faßbender. The symplectic eigenvalue problem, the butterfly form, the SR algorithm, and the Lanczos method. *Linear Algebra Appl.*, to appear.

7 P. Benner and H. Faßbender. An implicitly restarted symplectic Lanczos method for the symplectic eigenvalue problem. Preprint, Universität Bremen, Fachbereich 3 – Mathematik und Informatik, Bremen, Germany, 1998.

8 P. Benner, A. Laub and V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations II: discrete-time case. Tech. Report SPC 95_23, Fak. f. Mathematik, TU Chemnitz-Zwickau, 09107 Chemnitz, FRG, 1995.

9 M. Bohner. Linear Hamiltonian difference systems: Disconjugacy and Jacobi–type conditions. *J. Math. Anal. Appl.*, 199:804–826, 1996.

10 J. R. Bunch. The weak and strong stability of algorithms in numerical algebra. *Linear Algebra Appl.*, 88:49–66, 1987.

11 A. Bunse-Gerstner. An analysis of the HR algorithm for computing the eigenvalues of a matrix. *Linear Algebra Appl.*, 35:155–178, 1981.

12 A. Bunse-Gerstner. Matrix factorization for symplectic QR-like methods. *Linear Algebra Appl.*, 83:49–77, 1986.

13 A. Bunse-Gerstner and L. Elsner. Schur parameter pencils for the solution of the unitary eigenvalue problem. *Linear Algebra Appl.*, 154–156:741–778, 1991.

14 A. Bunse-Gerstner and V. Mehrmann. A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation. *IEEE Trans. Automat. Control*, AC-31:1104–1113, 1986.

15 A. Bunse-Gerstner, V. Mehrmann, and D. S. Watkins. An SR algorithm for Hamiltonian matrices based on Gaussian elimination. *Methods for Operations Research*, 58:339–356, 1989.

16 J. Della-Dora. Numerical linear algorithms and group theory. *Linear Algebra Appl.*, 10:267–283, 1975.

17 L. Elsner. On some algebraic problems in connection with general eigenvalue problems. *Linear Algebra Appl.*, 26:123–138, 1979.

18 H. Faßbender. *NO TITLE YET*. Habilitationsschrift, Universität Bremen, Germany, 1998. In Preparation.

19 U. Flaschka, V. Mehrmann, and D. Zywietz. An analysis of structure preserving methods for symplectic eigenvalue problems. *RAIRO Automatique Productique Informatique Industrielle*, 25:165–190, 1991.

20 J. G. F. Francis. The QR transformation. Part I and Part II. *Comput. J.*, 4:265–271 and 332–345, 1961.

21 G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

22 V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *U.S.S.R. Comput. Math. and Math. Phys.*, 3:637–657, 1961.

23 P. Lancaster and L. Rodman. *The Algebraic Riccati Equation.* Oxford University Press, Oxford, 1995.

24 A. J. Laub. Algebraic aspects of generalized eigenvalue problems for solving Riccati equations. In C.I. Byrnes and A. Lindquist, editors, *Computational and Combinatorial Methods in Systems Theory*, pages 213–227. Elsevier (North-Holland), 1986.

25 V. Mehrmann. Der SR-Algorithmus zur Berechnung der Eigenwerte einer Matrix. Diplomarbeit, Universität Bielefeld, Bielefeld, FRG, 1979.

26 V. Mehrmann. *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution.* Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

27 C. C. Paige and C. F. Van Loan. A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.*, 14:11–32, 1981.

28 T. Pappas, A. J. Laub, and N. R. Sandell. On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Trans. Automat. Control*, AC-25:631–641, 1980.

29 B. N. Parlett and W. G. Poole, A geometric theory for the QR, LU, and power iterations. *SIAM J. Numer. Anal.*, 10:389–412, 1973.

30 D. S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Linear Algebra Appl.*, 143:19–47, 1991.

31 D. S. Watkins and L. Elsner. Chasing algorithms for the eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 12:374–384, 1991.

32 D. S. Watkins and L. Elsner. Theory of decomposition and bulge chasing algorithms for the generalized eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 15:943–967, 1994.