

# THE PARAMETERIZED $SR$ ALGORITHM FOR HAMILTONIAN MATRICES\*

H. FASSBENDER†

**Abstract.** The heart of the implicitly restarted symplectic Lanczos method for Hamiltonian matrices consists of the  $SR$  algorithm, a structure-preserving algorithm for computing the spectrum of Hamiltonian matrices. The symplectic Lanczos method projects the large, sparse  $2n \times 2n$  Hamiltonian matrix  $H$  onto a small, dense  $2k \times 2k$  Hamiltonian  $J$ -Hessenberg matrix  $\tilde{H}$ ,  $k \ll n$ . This  $2k \times 2k$  Hamiltonian matrix is uniquely determined by  $4k - 1$  parameters. Using these  $4k - 1$  parameters, one step of the  $SR$  algorithm can be carried out in  $\mathcal{O}(k)$  arithmetic operations (compared to  $\mathcal{O}(k^3)$  arithmetic operations when working on the actual Hamiltonian matrix). As in the context of the implicitly restarted symplectic Lanczos method the usual assumption, that the Hamiltonian eigenproblem to be solved is stable, does not hold, the case of purely imaginary eigenvalues in the  $SR$  algorithm is treated here.

**Key words.** Hamiltonian Matrix; Eigenvalue Problem;  $SR$  Algorithm.

**AMS(MOS) subject classifications.** 65F15.

**1. Introduction.** Renewed interest [36, 49, 10] in the implicitly restarted symplectic Lanczos method for computing a few eigenvalues of a large, sparse Hamiltonian matrix [9] led us to reconsider that algorithm. It projects the large, sparse  $2n \times 2n$  Hamiltonian matrix  $H$  onto a small, dense  $2k \times 2k$  Hamiltonian  $J$ -Hessenberg matrix  $\tilde{H}$ ,  $k \ll n$ . This matrix  $\tilde{H}$  is of the form

$$\left[ \begin{array}{c|c} \diagdown & \equiv \\ \hline \diagdown & \diagdown \end{array} \right]$$

that is, due to the Hamiltonian structure, it can be represented by  $4k - 1$  parameters instead of the usual  $k^2$  matrix entries. As observed in [18], the  $SR$  algorithm preserves the Hamiltonian  $J$ -Hessenberg form. A standard implementation of the  $SR$  algorithm will require  $\mathcal{O}(k^3)$  flops in each iteration step. As pointed out in [18], using the  $4k - 1$  parameters one step of the  $SR$  algorithm for  $H$  can be carried out in  $\mathcal{O}(k)$  flops. Usually, this algorithm is implemented such that it works on the Hamiltonian  $J$ -Hessenberg matrix itself, working in a narrow band around the diagonals of the  $J$ -Hessenberg form.

In [18], the  $SR$  algorithm is discussed only for the case that the Hamiltonian matrix is stable, that is, it has no eigenvalues on the imaginary axis. While this a reasonable assumption in a number of applications, in the context of a restarted symplectic Lanczos method in which small eigenproblems of Hamiltonian  $J$ -Hessenberg form have to be solved, this cannot be assumed for the small eigenproblems even if the original problem is stable.

Therefore, in this paper, we will develop an implementation of the  $SR$  algorithm which can deal with eigenvalues on the imaginary axis. Moreover, our goal will be to derive a description of an implicit  $SR$  step such that the parameters which represent the resulting matrix are computed directly from the original ones without ever forming the actual Hamiltonian  $J$ -Hessenberg matrix or the bulge which is chased in the  $SR$

---

\*SUBMITTED ON NOVEMBER, 6, 2006.

†Technische Universität Braunschweig, Institut *Computational Mathematics*, 38106 Braunschweig, Germany, email: h.fassbender@tu-bs.de.

step. Numerical experiments indicate that this extra care might make a positive difference in the accuracy of the computed results.

A Hamiltonian matrix  $H \in \mathbb{R}^{2n \times 2n}$  has the form

$$(1.1) \quad H = \begin{bmatrix} A & G \\ Q & -A^T \end{bmatrix}, \quad G = G^T, \quad Q = Q^T,$$

where  $A, G$  and  $Q$  are real  $n \times n$  matrices. A number of applications from control theory and related areas lead to eigenvalue problems: stability radius and  $H_\infty$  norm computation [23, 16], linear quadratic optimal control problems and the solution of continuous-time algebraic Riccati equations [5, 37, 42],  $H - \infty$  control [6, 7], passivity preserving model reduction [4, 43, 10], quadratic eigenvalue problems [39, 46], computation of pseudo spectra [20].

An ubiquitous matrix when dealing with Hamiltonian eigenvalue problems is the skew-symmetric matrix

$$(1.2) \quad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix},$$

where  $I$  denotes the  $n \times n$  identity matrix. By straightforward algebraic manipulation one can show that a Hamiltonian matrix  $H$  is equivalently defined by the property

$$(1.3) \quad HJ = (HJ)^T.$$

Any matrix  $S \in \mathbb{R}^{2n \times 2n}$  satisfying

$$S^T J S = S J S^T = J$$

is called symplectic, and since

$$(1.4) \quad (S^{-1} H S) J = S^{-1} H J S^{-T} = S^{-1} J^T H^T S^{-T} = [(S^{-1} H S) J]^T,$$

we see that symplectic similarity transformations preserve Hamiltonian structure. There are relevant cases, however, where both  $H$  and  $S^{-1} H S$  are Hamiltonian, but  $S$  is not a symplectic matrix [28].

One of the most remarkable properties of a Hamiltonian matrix is that its eigenvalues always occur in pairs  $\{\lambda, -\lambda\}$  if  $\lambda$  is real or purely imaginary, or in quadruples  $\{\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda}\}$  otherwise. Hence, the spectrum of any Hamiltonian matrix is symmetric with respect to the real and imaginary axis. Numerical methods that take this structure into account are capable of preserving the eigenvalue pairings despite the presence of roundoff errors. Besides the preservation of such eigenvalue symmetries, there are several other benefits to be gained from using structure-preserving algorithms in place of general-purpose algorithms for computing eigenvalues. These benefits include reduced computational time and improved eigenvalue/-eigenvector accuracy.

The eigenvalues and invariant subspaces of Hamiltonian matrices  $H$  may be computed by the  $QR$  algorithm [29]. But the  $QR$  method cannot take advantage of the Hamiltonian structure of  $H$ , it will treat  $H$  like any arbitrary  $2n \times 2n$  matrix. The computed eigenvalues will in general not come in quadruple  $\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda}$ , although the exact eigenvalues have this property. Even worse, small perturbations may cause eigenvalues close to the imaginary axis to cross the axis such that the number of true and computed eigenvalues in the right half plane may differ.

To preserve the Hamiltonian structure of  $H$ , we would have to employ similarity transformations with symplectic matrices instead of the transformations with the usual unitary matrices in the  $QR$  algorithm. A vast number of approaches for deriving a suitable algorithm can be found in the literature, see, e.g., [1, 2, 5, 17, 21, 22, 24, 27, 30, 31, 33, 34, 35, 38, 37, 40, 42, 45, 47].

In [14], Benner, Mehrmann and Xu propose a numerically backward stable method to compute the eigenvalues (but not the invariant subspaces) of real Hamiltonian matrices. It is not yet clear whether the method is strongly backward stable. Based on this algorithm, in [13], the authors develop a backward stable, structure preserving  $O(n^3)$  method for computing all eigenvalues and invariant subspaces of  $H$ . The extension of these algorithms to the complex case is considered in [15]. Implementations of the algorithms in [14] and [13] are freely available from the HAPACK package<sup>1</sup>[11]. Using the ideas of [13, 14], Chu, Liu and Mehrmann suggest in [25] a numerically strongly backward stable  $O(n^3)$  method for computing the Hamiltonian Schur form of a Hamiltonian matrix that has no purely imaginary eigenvalues.

Algorithms based on symplectic but non-orthogonal transformations include the  $SR$  algorithm [18, 37] and related methods [19, 41]. The  $SR$  method is a  $QR$ -like method based on the  $SR$  decomposition. In an initial step, the  $2n \times 2n$  Hamiltonian matrix is reduced to a more condensed form, the Hamiltonian  $J$ -Hessenberg form, which in general contains  $6n - 2$  nonzero entries. A Hamiltonian matrix in  $J$ -Hessenberg form is determined by  $4n - 1$  parameters. Our interest in the  $SR$  algorithm stems from the fact that the implicitly symplectic Lanczos algorithm for Hamiltonian matrices [9] employs the  $SR$  algorithm in order to solve small Hamiltonian  $J$ -Hessenberg eigenproblems. It is not recommended to use the  $SR$  algorithm just by itself for solving a Hamiltonian eigenproblem, as it is potentially unstable. In case, one would like to use the  $SR$  algorithm for this purpose, it should be accompanied by a defect-correction method like the Newton method in order to improve the accuracy of the computed results.

The  $SR$  algorithm as discussed in [18] is reviewed in Section 2. As will be shown in Section 3 of this paper, 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. Thus only  $O(n)$  arithmetic operations per  $SR$  step are needed compared to  $O(n^3)$  arithmetic operations when working on the actual Hamiltonian matrix. Moreover, the Hamiltonian structure, which will be destroyed in the numerical process due to roundoff errors when working with a Hamiltonian matrix, will be forced by working just with the parameters. The  $SR$  iteration proceeds until the problem has completely decoupled into Hamiltonian  $J$ -Hessenberg subproblems of size  $2 \times 2$  or  $4 \times 4$ . In a final step each of these subproblems has to be transformed into a form from which the eigenvalues can be read off. In [18], this was considered for the case that the small subproblems have no purely imaginary eigenvalues. As this cannot be assumed in the context of the implicitly restarted symplectic Lanczos method, the solution of these subproblems in the presence of purely imaginary eigenvalues is discussed in Section 4.

## 2. Preliminaries. A Hamiltonian matrix

$$(2.1) \quad \tilde{H} = \begin{bmatrix} D & B \\ V & -V \end{bmatrix}$$

<sup>1</sup>see the HAPACK homepage <http://www.tu-chemnitz.de/mathematik/hapack/>

with  $D = \text{diag}(\delta_1, \dots, \delta_n)$ ,  $V = \text{diag}(\nu_1, \dots, \nu_n)$ , and a symmetric tridiagonal  $B$  is called a Hamiltonian  $J$ -Hessenberg matrix. The diagonal entries of  $B$  will be denoted by  $\beta_1, \dots, \beta_n$ , the subdiagonal entries by  $\zeta_2, \dots, \zeta_n$ . Bunse-Gerstner and Mehrmann [18] showed that for every Hamiltonian matrix  $H$  there exist numerous symplectic matrices  $S$  such that  $S^{-1}HS = \tilde{H}$  is a Hamiltonian  $J$ -Hessenberg matrix. Hence, any  $2n \times 2n$  Hamiltonian matrix is symplectically similar to a matrix that is determined by  $4n - 1$  parameters. A Hamiltonian  $J$ -Hessenberg matrix will be called unreduced, if  $\nu_j \neq 0$  for all  $j = 1, \dots, n$  and  $\zeta_k \neq 0$  for all  $k = 2, \dots, n$ . Unreduced  $J$ -Hessenberg matrices have similar properties as unreduced Hessenberg matrices. For the  $SR$  theory, the unreduced Hamiltonian  $J$ -Hessenberg matrices play a role analogous to that of unreduced Hessenberg matrices in the standard  $QR$  theory.

The structure-preserving symplectic Lanczos algorithm for Hamiltonian matrices [9] projects a large  $2n \times 2n$  Hamiltonian matrix  $H$  onto a small  $2k \times 2k$  Hamiltonian  $J$ -Hessenberg matrix  $\tilde{H}_{2k}$

$$HS_{2k} = S_{2k}\tilde{H}_{2k} + \zeta_{k+1}v_{k+1}e_{2k}^T$$

where the columns of the  $2n \times 2k$  matrix  $S_{2k}$  are symplectic. In order to compute the eigenvalues and, if desired, eigenvectors of  $\tilde{H}_{2k}$  the  $SR$  algorithm is the method of choice as it makes full use of the special Hamiltonian  $J$ -Hessenberg form. This has already been discussed to some extent in [18, 9, 8, 49]. The  $SR$  algorithm 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. A matrix

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} \nabla & \nabla \\ \circ \nabla & \nabla \\ & \circ \end{bmatrix}$$

is said to be  $J$ -triangular if the  $n \times n$  submatrices  $R_{ij}$  are all upper triangular, and  $R_{21}$  is strictly upper triangular. Analogous to the general practice in the  $QR$  iteration [29], the  $SR$  iteration is implemented in an implicit fashion. The first implicit transformation  $S_1$  is selected in order to introduce a bulge into the  $J$ -Hessenberg matrix  $H$ . That is, a symplectic matrix  $S_1$  is determined such that

$$S_1^{-1}q(H)e_1 = \alpha e_1, \quad \alpha \in \mathbb{R},$$

where  $q(H)$  is an appropriately chosen spectral transformation function. Applying this first transformation to the  $J$ -Hessenberg matrix yields a Hamiltonian matrix  $S_1^{-1}HS_1$  with almost  $J$ -Hessenberg form having a small bulge. The remaining implicit transformations perform a bulge-chasing sweep down the subdiagonals to restore the  $J$ -Hessenberg form. That is, a symplectic matrix  $S_2$  is determined such that  $S_2^{-1}S_1^{-1}HS_1S_2$  is of  $J$ -Hessenberg form again. If  $H$  is an unreduced  $J$ -Hessenberg matrix and  $\text{rank}(q(H)) = 2n$ , then

$$(2.2) \quad \tilde{H} = S_2^{-1}S_1^{-1}HS_1S_2$$

is also an unreduced  $J$ -Hessenberg matrix. Hence, there will be parameters  $\tilde{\delta}_1, \dots, \tilde{\delta}_n$ ,  $\tilde{\beta}_1, \dots, \tilde{\beta}_n$ ,  $\tilde{\zeta}_1, \dots, \tilde{\zeta}_n$ ,  $\tilde{\nu}_2, \dots, \tilde{\nu}_n$  which determine  $\tilde{H}$ . The algorithm for reducing a Hamiltonian matrix to  $J$ -Hessenberg form as given in [18] can be used as a building block for the implicit  $SR$  step. The algorithm uses the following elementary symplectic transformations:

- symplectic Givens transformation

$$G(k, c, s) = \left[ \begin{array}{cc|cc} I_{k-1} & & & \\ & c & & s \\ & & I_{n-k} & \\ \hline & -s & & I_{k-1} \\ & & & c \\ & & & & I_{n-k} \end{array} \right], \text{ where } c^2 + s^2 = 1;$$

- symplectic Householder transformation

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

- symplectic Gauss transformation

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

The symplectic Givens and Householder transformations are orthogonal, while the symplectic Gauss transformations are nonorthogonal. Algorithms to compute the entries of the above mentioned transformations can be found, e.g., in [40] and [19]. Here we will use  $[c, s] = \mathbf{givens}(a, b)$  and  $[c, d] = \mathbf{gauss1}(a, b)$  to denote the computation of the parameters of a symplectic Givens/Gauss transformation such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \text{ where } c^2 + s^2 = 1,$$

or, resp.,

$$\begin{bmatrix} c & & & d \\ & c & & \\ & & c^{-1} & \\ & & & c^{-1} \end{bmatrix} \begin{bmatrix} \star \\ a \\ b \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ r \\ 0 \end{bmatrix}.$$

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.

Bunse-Gerstner and Mehrmann present in [18] an algorithm for reducing an arbitrary matrix to *J*-Hessenberg form, that is to the  $2 \times 2$  block form where  $A_{11}, A_{21}, A_{22}$  are upper triangular matrices and  $A_{12}$  is an upper Hessenberg matrix,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} \nabla & \nabla \\ \nabla & \nabla \end{bmatrix}.$$

The basic idea of the algorithm can be summarized as follows:

for  $j = 1$  to  $n$   
 determine a symplectic matrix  $S$  such that the  $j$ th column of  
 $S^{-1}H$  is of the desired form  
 set  $H = S^{-1}HS$   
 determine a symplectic matrix  $S$  such that the  $(n + j)$ th column  
 of  $S^{-1}H$  is of the desired form  
 set  $H = S^{-1}HS$

The remaining rows and columns of  $H$  that are not touched explicitly during the process will be in the desired form due to the Hamiltonian structure. The algorithm uses the symplectic Givens transformations  $G_k$ , the symplectic Householder transformations  $H_k$ , and the symplectic Gauss transformations  $L_k$ .

In each step of the  $SR$  iteration, due to the special Hamiltonian eigenstructure, the spectral transformation function will be chosen either as

$$q_2(H) = (H - \mu I)(H + \mu I), \quad \mu \in \mathbb{R} \text{ or } \mu = i\omega, \omega \in \mathbb{R},$$

or

$$q_4(H) = (H - \mu I)(H + \mu I)(H - \bar{\mu} I)(H + \bar{\mu} I), \quad \mu \in \mathbb{C}, \operatorname{Re}(\mu) \neq 0.$$

If the chosen shifts are good approximate eigenvalues, we expect deflation at the end of the  $SR$  step as indicated in

$$(2.3) \quad \hat{H} = \left[ \begin{array}{cc|cc} \diagdown & & \equiv & \\ & \square & & \square \\ \hline \diagdown & & \diagdown & \\ & \square & & \square \end{array} \right]$$

As proposed in [18], a shift strategy similar to that used in the standard  $QR$  algorithm should be used. For example, for a quadruple shift, we choose the 4 eigenvalues of the  $4 \times 4$  Hamiltonian  $J$ -Hessenberg submatrix

$$H_{4 \times 4} = \left[ \begin{array}{cc|cc} \delta_{n-1} & & \beta_{n-1} & \zeta_n \\ & \delta_n & \zeta_n & \beta_n \\ \hline \nu_{n-1} & & -\delta_{n-1} & \\ & \nu_n & & -\delta_n \end{array} \right].$$

There is no need to compute the eigenvalues of  $H_{4 \times 4}$  directly. Comparing  $q_4(H)$  with the characteristic polynomial of  $H_{4 \times 4}$  gives the first column of  $q_4(H)$  which is needed to start the implicit  $SR$  step explicitly. The same can be done for the double shift case. This is exactly the generalized Rayleigh-quotient strategy for choosing the shifts proposed by Watkins and Elsner in [48]. Hence the convergence theorems Theorem 6.2, 6.3 and 6.5 from [48] can be applied here. In particular the Hamiltonian  $SR$  algorithm is typically cubically convergent.

**3. The parameterized  $SR$  step.** As will be shown in this section, the  $SR$  algorithm for a Hamiltonian  $J$ -Hessenberg matrix  $H$  can be rewritten in a parameterized form that will work only with the  $4n - 1$  parameters which determine  $\tilde{H}$  instead of the entire matrix in each iteration step. The parameters which define  $\tilde{H}$  (2.2) will be

computed directly from those of  $H$  without ever forming  $H$  or  $\tilde{H}$ . Implementations existing so far usually set up  $H$ , and work in a narrow band around the diagonals. This is completely avoided here.

The key to the development of a  $SR$  algorithm working only on the parameters is the observation that at any point in the implicit  $SR$  step only a certain, limited number of rows and columns of the Hamiltonian  $J$ -Hessenberg matrix is worked on. In the leading part of the intermediate matrices the Hamiltonian  $J$ -Hessenberg form is already retained and is not changed any longer, while the trailing part has not been changed yet. Hence, from the leading part the first parameters of the resulting  $J$ -Hessenberg matrix can be read off, while from the trailing part the last parameters of the original  $J$ -Hessenberg matrix can still be read off.

Due to the special Hamiltonian eigenstructure, the spectral transformation function will be chosen either as

$$q_2(H) = (H - \mu I)(H + \mu I), \quad \mu \in \mathbb{R} \text{ or } \mu = i\omega, \omega \in \mathbb{R},$$

or

$$q_4(H) = (H - \mu I)(H + \mu I)(H - \bar{\mu} I)(H + \bar{\mu} I), \quad \mu \in \mathbb{C}, \operatorname{Re}(\mu) \neq 0.$$

By applying a sequence of quadruple or double shift  $SR$  steps to a Hamiltonian  $J$ -Hessenberg matrix  $H$  it is possible to reduce the tridiagonal block in  $H$  to quasi-diagonal form with  $1 \times 1$  and  $2 \times 2$  blocks on the diagonal. The eigenproblem decouples into a number of simple Hamiltonian  $2 \times 2$  or  $4 \times 4$  eigenproblems. Their treatment is covered in Section 4.

In case an exceptional shift step is needed in the  $SR$  algorithm, one might want to use a single shift

$$q_1(H) = H - \mu I, \quad \mu \in \mathbb{R}.$$

For illustration purposes, we will consider here only the single shift case, for a derivation of the double and quadruple shift case see [26].

**3.1. A single shift implicit  $SR$  step.** Consider a single shift implicit  $SR$  step, which might be used as an exceptional shift step in the  $SR$  iteration. As  $H$  is a real matrix, for a single shift the shift polynomial  $q_1(H) = H - \mu I$  should be chosen for  $\mu \in \mathbb{R}$ . The first column of  $q_1$  is of the form

$$x = q_1(H)e_1 = (\delta_1 - \mu)e_1 + \nu_1 e_{n+1}.$$

This vector can be transformed into a multiple of  $e_1$  by a symplectic Givens transformation  $G_1$  where the parameters  $c_1, s_1$  are given by  $[c_1, s_1] = \mathbf{givens}(\delta_1 - \mu, \nu_1)$ . Hence, the first step of the implicit  $SR$  step introduces a bulge by a similarity transformation of  $H$  with

$$(3.1) \quad G_1 = \left[ \begin{array}{c|c} c_1 & s_1 \\ \hline I_{n-1} & \\ \hline -s_1 & \\ \hline & c_1 \\ & I_{n-1} \end{array} \right].$$



where

$$\begin{aligned}\tilde{\beta}_1 &= -d_2^2\nu_2 + c_2^2\beta_1', & \tilde{\nu}_1 &= c_2^{-2}\nu_1', \\ \beta_2' &= c_2^2\beta_2 - c_2d_2b_1, & \nu_2' &= c_2^{-2}\nu_2, \\ \zeta_2'' &= -d_2c_2(\delta_2 + \tilde{\delta}_1) + c_2^2\zeta_2', & b_2 &= d_2c_2^{-1}\nu_2, \\ \zeta_3' &= c_2\zeta_3.\end{aligned}$$

The bulge has moved from the  $(2, 1)$  position to the  $(1, 2)$  position. Due to the Hamiltonian structure, simultaneously the  $((n+1, n+2))$  position moved to the  $((n+2, n+1))$  position. Hence, the first column of  $H_2$  is in the desired form. Next the  $(n+1)$ st column is treated. A symplectic Givens transformation  $G_2$  where the parameters  $c_3, s_3$  are given by  $[c_3, s_3] = \mathbf{givens}(\zeta_2'', b_2)$

$$G_2 = \left[ \begin{array}{ccc|ccc} 1 & & & & & \\ & c_3 & & & s_3 & \\ & & I_{n-2} & & & \\ \hline & & & 1 & & \\ & -s_3 & & & c_3 & \\ & & & & & I_{n-2} \end{array} \right]$$

is used to eliminate the  $(n+2, n+1)$  entry (and simultaneously the  $(1, 2)$  entry). This gives  $H_3 = G_2H_2G_2^T$

$$H_3 = \left[ \begin{array}{cccc|cccc} \tilde{\delta}_1 & & & & \tilde{\beta}_1 & \zeta_2''' & & \\ & \tilde{\delta}_2 & & & \zeta_2''' & \beta_2'' & \zeta_3 & \\ & b_3 & \delta_3 & & \zeta_3'' & \beta_3 & \zeta_4 & \\ & & & \delta_4 & & & \zeta_4 & \ddots \\ & & & & & & & \ddots \\ & & & & & & & \ddots \\ & & & & & & & \zeta_n \\ & & & & & & & \zeta_n \\ & & & \delta_n & & & & \delta_n \\ \hline \tilde{\nu}_1 & & & & -\tilde{\delta}_1 & & & \\ & \nu_2'' & & & & -\tilde{\delta}_2 & -b_3 & \\ & & \nu_3 & & & & -\delta_3 & \\ & & & \nu_4 & & & & -\delta_4 \\ & & & & & & & \ddots \\ & & & & & & & \nu_n \\ & & & & & & & -\delta_n \end{array} \right],$$

where

$$\begin{aligned}\tilde{\delta}_2 &= (c_3^2 - s_3^2)\tilde{\delta}_2 - c_3s_3(\nu_2' + \beta_2'), & \zeta_2''' &= c_3\zeta_2'' - s_3b_2, \\ \beta_2'' &= c_3^2\beta_2' - 2c_3s_3\tilde{\delta}_2 - s_3^2\nu_2', & \zeta_3'' &= c_3\zeta_3', \\ \nu_2'' &= c_3^2\nu_2' - 2c_3s_3\tilde{\delta}_2 - s_3^2\beta_2', & b_3 &= s_2\zeta_3'.\end{aligned}$$

The first and the  $(n+1)$ st column of  $H_3$  are in the desired form. The bulge has been chased from position  $(2, 1)$   $((n+1, n+2))$  to position  $(3, 2)$   $((n+2, n+3))$ . Continuing in the same fashion, it can be chased until the  $J$ -Hessenberg form has been restored. In order to derive an algorithm that works only on the parameters which determine the Hamiltonian matrix  $H$ , let us consider the next step of the process as well. We will see that  $\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\nu}_1$  and  $\tilde{\beta}_1$  will not be changed in the following calculations. They belong to the set of parameters which determine the matrix  $\tilde{H}$  (3.2).





plications of a Gauss and a Givens transformation which requires  $(n-1) \times 39$  multiplications, 17 additions and 3 square roots (minus 2 multiplications in the last step). Hence, the algorithm requires  $39n - 22$  multiplications,  $17n - 9$  additions and  $3n - 2$  square roots. In other words, the implicit single *SR* step working on the parameters only requires  $\mathcal{O}(n)$  flops. As the entire process works only on the parameters which determine the Hamiltonian matrix, the Hamiltonian structure is forced in every step of the algorithm.

**3.2. A double shift implicit *SR* step.** As  $H$  is a Hamiltonian matrix, for a double shift the shift polynomial  $q_2(H) = (H - \mu I)(H + \mu I)$  should be chosen for  $\mu \in \mathbb{R}$  or  $\mu = i\omega, \omega \in \mathbb{R}$ . In the first case, the first column of  $q_2$  is of the form

$$x = q_2(H)e_1 = (H^2 - \mu^2 I)e_1 = (\delta_1^2 + \nu_1 \beta_1 - \mu^2)e_1 + \nu_1 \zeta_2 e_2.$$

while in the second case the first column of  $q_2$  is of the form

$$x = q_2(H)e_1 = (H^2 + \omega^2 I)e_1 = (\delta_1^2 + \nu_1 \beta_1 + \omega^2)e_1 + \nu_1 \zeta_2 e_2.$$

In case, a Rayleigh-quotient strategy is used, the shifts will be chosen as the eigenvalues of

$$H_{2 \times 2} = \left[ \begin{array}{c|c} \delta_n & \beta_n \\ \hline \nu_n & -\delta_n \end{array} \right].$$

There is no need to compute the eigenvalues of  $H_{2 \times 2}$  directly. Comparing  $q_2(H) = (H - \mu I)(H + \mu I) = H^2 - \mu^2 I$  with the characteristic polynomial of  $H_{2 \times 2}$

$$\det(H_{2 \times 2} - \lambda I) = \lambda^2 - (\delta_n^2 + \beta_n \nu_n)$$

yields that  $q_2$  should be chosen as

$$q_2(H) = H^2 - (\delta_n^2 + \beta_n \nu_n)I.$$

The first column is then given by

$$q_2(H)e_1 = (\delta_1^2 + \nu_1 \beta_1 - \delta_n^2 - \beta_n \nu_n)e_1 + \nu_1 \zeta_2 e_2.$$

Hence, in any case, the first column of the spectral transformation has the same form. A parameterized version of the implicit double shift *SR* step can be derived in complete analogy to the implicit single step. For a derivation see [26].

**3.3. A quadruple shift implicit *SR* step.** For an implicitly shifted quadruple shift step, the shift  $\gamma \in \mathbb{C}$ ,  $\operatorname{Re}(\gamma) \neq 0$  defines the spectral transformation

$$(H - \gamma I)(H + \gamma I)(H - \bar{\gamma} I)(H + \bar{\gamma} I).$$

An implicitly shifted quadruple step can also be used to perform two double shift steps, in that case we have shifts  $\mu$  and  $\eta$  that are either real or purely imaginary. The spectral transformation is given by

$$(3.3) \quad X = (H - \mu I)(H + \mu I)(H - \eta I)(H + \eta I).$$

With  $\mu = \gamma$  and  $\eta = \bar{\gamma}$  this is just the spectral transformation given above. Hence, let us consider the slightly more general expression in our derivations.

First we need to compute the first column of  $X$  (3.3) in order to determine the first symplectic transformation which will introduce the bulge to be chased;

$$\begin{aligned} x &= (H - \mu I)(H + \mu I)(H - \eta I)(H + \eta I)e_1 \\ &= (H^2 - \mu^2 I)(H^2 - \eta^2 I)e_1 \\ &= (H^2 - \mu^2 I)[(\delta_1^2 + \nu_1 \beta_1 - \eta^2)e_1 + \nu_1 \zeta_2 e_2] \\ &= (H^2 - \mu^2 I)[(a_1 - \eta^2)e_1 + \nu_1 \zeta_2 e_2] \end{aligned}$$

where we used for notational convenience

$$a_j = \delta_j^2 + \nu_j \beta_j.$$

Hence,

$$\begin{aligned} x &= (a_1 - \eta^2)[(a_1 - \mu^2)e_1 + \nu_1 \zeta_2 e_2] + \nu_1 \zeta_2 [H(\delta_2 e_2 + \nu_2 e_{n+2} - \mu^2 e_2)] \\ &= (a_1 - \eta^2)[(a_1 - \mu^2)e_1 + \nu_1 \zeta_2 e_2] \\ &\quad + \nu_1 \zeta_2 [\delta_2 (\delta_2 e_2 + \nu_2 e_{n+2}) + \nu_2 (\zeta_2 e_1 + \beta_2 e_2 + \zeta_3 e_3 - \delta_2 e_{n+2}) - \mu^2 e_2] \\ &= [(a_1 - \eta^2)(a_1 - \mu^2) + \nu_1 \nu_2 \zeta_2^2]e_1 + \nu_1 \zeta_2 [(a_1 + a_2 - \eta^2 - \mu^2)]e_2 + \nu_1 \nu_2 \zeta_2 \zeta_3 e_3 \\ &= [a_1^2 + \nu_1 \nu_2 \zeta_2^2 - (\eta^2 + \mu^2)a_1 + \mu^2 \eta^2]e_1 + \nu_1 \zeta_2 [(a_1 + a_2) - (\eta^2 + \mu^2)]e_2 \\ &\quad + \nu_1 \nu_2 \zeta_2 \zeta_3 e_3 \end{aligned}$$

In case, a Raleigh-quotient like shift strategy is chosen, the spectral transformation function  $q_4$  has to be chosen as

$$\begin{aligned} q_4(H)e_1 &= (a_1^2 + \zeta_2^2 \nu_1 \nu_2 - (a_{n-1} + a_n)a_1 + a_{n-1}a_n - \nu_{n-1}\nu_n \zeta_n^2)e_1 \\ &\quad + \zeta_2 \nu_1 [(a_1 + a_2) - (a_{n-1} + a_n)]e_2 + \nu_1 \nu_2 \zeta_2 \zeta_3 e_3. \end{aligned}$$

Hence, in any case, the first column of the spectral transformation has the same form. A parameterized version of the implicit quadruple shift  $SR$  step can be derived in complete analogy to the implicit single step. For a derivation see [26].

**4. Solving the  $2 \times 2$  and  $4 \times 4$  subproblems.** We proceed with the  $SR$  iteration until the problem has completely decoupled into Hamiltonian  $J$ -Hessenberg subproblems of size  $2 \times 2$  or  $4 \times 4$ ; that is, at least every second  $\zeta_j$  in  $\widehat{H} = S^{-1}HS$  is neglectably small. In a final step we now have to transform each of these subproblems into a form from which the eigenvalues can be read off. In order to do so, each subproblem is transformed into Hamiltonian Schur form  $H_{Schur}$  by an orthogonal symplectic transformation

$$H_{Schur} = \begin{bmatrix} T & N \\ 0 & -T^T \end{bmatrix},$$

where  $N$  is symmetric and  $T$  is a block upper triangular matrix with  $1 \times 1$  or  $2 \times 2$  blocks on the diagonal. If possible, the eigenvalues of  $T$  will have negative real part. This has already been discussed in [18] for the case that the Hamiltonian matrix has no purely imaginary eigenvalues. In that paper, the  $SR$  algorithm is used to solve complete stable eigenproblems, this implies that the  $2 \times 2$  and  $4 \times 4$  subproblems have no purely imaginary eigenvalues. This cannot be assumed here. Even if the original Hamiltonian matrix  $H$  does not have purely imaginary eigenvalues, the projected much smaller Hamiltonian  $J$ -Hessenberg matrix computed by the symplectic Lanczos

method for Hamiltonian matrices may have some. When possible, the results from [18] are used. The cases not considered in [18] are discussed here in detail.

Let us start with the  $2 \times 2$  subproblems. They are of the form

$$H_{2 \times 2} = \begin{bmatrix} \delta_j & \beta_j \\ \nu_j & -\delta_j \end{bmatrix}.$$

The characteristic polynomial is given by

$$\det(H_{2 \times 2} - \lambda I) = \lambda^2 - (\delta_j^2 + \beta_j \nu_j).$$

Hence, the eigenvalues are

$$\pm \lambda = \pm \sqrt{\delta_j^2 + \beta_j \nu_j}.$$

In case the eigenvalues are real that is, if  $\delta_j^2 + \beta_j \nu_j > 0$ , a  $2 \times 2$  Givens transformation [18]

$$Q = \begin{cases} \sqrt{(\delta_j - \lambda)^2 + \nu_j^2}^{-1} \begin{bmatrix} \delta_j - \lambda & -\nu_j \\ \nu_j & \delta_j - \lambda \end{bmatrix} & \text{if } \lambda \neq \delta_j \text{ and } \nu_j \neq 0, \\ I & \text{if } \lambda = -\delta_j \text{ and } \nu_j = 0, \\ \sqrt{(4\delta_j^2 + \beta_j^2)}^{-1} \begin{bmatrix} \beta_j & -2\delta_j \\ 2\delta_j & \beta_j \end{bmatrix} & \text{if } \lambda = \delta_j \text{ and } \nu_j = 0. \end{cases}$$

is used in order to put the negative eigenvalue in the  $(1, 1)$  position.  $Q$  is orthogonal and symplectic and

$$Q^T H_{2 \times 2} Q = \begin{bmatrix} -\lambda & x \\ & \lambda \end{bmatrix}.$$

Lifting  $Q$  into the  $(j, n + j)$ th plane of a corresponding  $2n \times 2n$  orthogonal symplectic Givens matrix  $G$  and transforming  $\hat{H}$  with  $G$  will eliminate  $\nu_j$  and put  $-\lambda$  in the  $(j, j)$  position of  $G^T \hat{H} G$ . Therefore,  $SG$  contains in its  $j$ th column the eigenvector corresponding to  $-\lambda$  which belongs to the stable invariant subspace (if there exists one).

REMARK 4.1. In [49], it is suggested to further reduce the  $2 \times 2$  problem by using the symplectic transformation

$$Q' = \begin{bmatrix} \frac{1}{2\lambda} & x \\ & 2\lambda \end{bmatrix},$$

which transforms  $Q^T H_{2 \times 2} Q$  into diagonal form

$$Q'^{-1} Q^T H_{2 \times 2} Q Q' = \begin{bmatrix} -\lambda & \\ & \lambda \end{bmatrix}.$$

This allows to read off the eigenvectors to both eigenvalues, but, depending on  $\lambda$  this additional transformation may increase the numerical instability of the SR algorithm.

This additional transformation is not necessary as both eigenvectors can be read off directly from

$$\begin{bmatrix} -\lambda & x \\ & \lambda \end{bmatrix}.$$

The eigenvector corresponding to  $-\lambda$  is  $e_1 = [1, 0]^T$ , the eigenvector corresponding to  $\lambda$  is

$$\begin{bmatrix} x \\ 2\lambda \end{bmatrix}.$$

In case the eigenvalues are purely imaginary (that is, if  $\delta_j^2 + \beta_j \nu_j < 0$ ) nothing needs to be done. If one is interested in at least one eigenvector corresponding to the pair of imaginary eigenvalues, then  $H_{2 \times 2}$  has to be transformed into its canonical Schur form

$$H_{final} = \begin{bmatrix} 0 & \pm\beta \\ \mp\beta & 0 \end{bmatrix}.$$

From

$$H[x \ y] = [x \ y]H_{final}, \quad x, y \in \mathbb{R}^{2n}$$

we obtain with  $z = x + iy$  for the case  $Hx = -\beta y, Hy = \beta x$

$$Hz = i\beta z,$$

and just as for the case  $Hx = \beta y, Hy = -\beta x$

$$Hz = -i\beta z.$$

Hence, an eigenvector can be read off; the eigenvector to the other eigenvalue is  $\bar{z}$ .

There exists an orthogonal Givens transformation such that

$$(4.1) \quad \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \delta_j & \beta_j \\ \nu_j & -\delta_j \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} 0 & d \\ b & 0 \end{bmatrix},$$

where  $b, d \in \mathbb{R}, bd < 0$  and  $\lambda = \pm i\beta = \pm\sqrt{bd} = \pm\sqrt{\delta_j^2 + \beta_j \nu_j}$ . For a numerical sound implementation of this step, see, e.g., the LAPACK routine `lanev2f` [3]. In our actual implementation, we use MATLAB's `schur` function for solving the  $2 \times 2$  problem in case there are purely imaginary eigenvalues, as this will compute (4.1) right away.

If in (4.1),  $b < 0$  and  $d > 0$ , then the transformation

$$(4.2) \quad \begin{bmatrix} 1/x & \\ & x \end{bmatrix} \begin{bmatrix} 0 & d \\ b & 0 \end{bmatrix} \begin{bmatrix} x & \\ & 1/x \end{bmatrix} = \begin{bmatrix} 0 & \beta \\ -\beta & 0 \end{bmatrix}, \quad x = \sqrt{-\beta/b}$$

puts (4.1) into its canonical Schur form. Note, that the transformation matrix is symplectic. If in (4.1),  $b > 0$  and  $d < 0$ , then the transformation

$$(4.3) \quad \begin{bmatrix} 1/x & \\ & x \end{bmatrix} \begin{bmatrix} 0 & d \\ b & 0 \end{bmatrix} \begin{bmatrix} x & \\ & 1/x \end{bmatrix} = \begin{bmatrix} 0 & -\beta \\ \beta & 0 \end{bmatrix}, \quad x = \sqrt{\beta/b}$$

puts (4.1) into its canonical Schur form.

REMARK 4.2. *In case, a different ordering of the eigenvalues on the diagonal is desired, a reordering of the eigenvalues is possible later using the idea presented in [29, Chapter 7.6.2].*

Now, let us turn to the  $4 \times 4$  subproblems. They are of the form

$$H_{4 \times 4} = \left[ \begin{array}{cc|cc} \delta_j & 0 & \beta_j & \zeta_{j+1} \\ 0 & \delta_{j+1} & \zeta_{j+1} & \beta_{j+1} \\ \nu_j & 0 & -\delta_j & 0 \\ 0 & \nu_{j+1} & 0 & -\delta_{j+1} \end{array} \right], \quad \zeta_{j+1} \neq 0.$$

The eigenvalues are given as

$$\begin{aligned} \lambda_{1/2/3/4} &= \pm \sqrt{-\frac{a_j + a_{j+1}}{2} \pm \sqrt{\left(\frac{a_j + a_{j+1}}{2}\right)^2 - a_j a_{j+1} + \nu_j \nu_{j+1} \zeta_{j+1}^2}} \\ &= \pm \sqrt{-\frac{a_j + a_{j+1}}{2} \pm \sqrt{\left(\frac{a_j - a_{j+1}}{2}\right)^2 + \nu_j \nu_{j+1} \zeta_{j+1}^2}}, \end{aligned}$$

where

$$a_j = \delta_j^2 + \nu_j \beta_j.$$

In case the term under the inner square root is nonnegative,

$$\theta_j := \left(\frac{a_j - a_{j+1}}{2}\right)^2 + \nu_j \nu_{j+1} \zeta_{j+1}^2 \geq 0,$$

there will be just real or purely imaginary eigenvalues and only if  $\theta_j$  is negative, there will be complex eigenvalues with nonzero real part. Obviously, in case  $\nu_j$  or  $\nu_{j+1}$  is zero (or both of them), the eigenvalues will be real or purely imaginary, but never complex with nonzero real part. As this is easily detected by inspecting the matrix entries, we will deal with both cases separately (in contrast to [18]).

First, let us assume that  $\theta_j < 0$ . This implies that  $\nu_j \nu_{j+1} \neq 0$  and complex eigenvalues with nonzero real part. Following the derivations in [18] an orthogonal symplectic matrix  $Q$  can be constructed such that

$$(4.4) \quad Q^T H_{4 \times 4} Q = \left[ \begin{array}{cc|cc} x_1 & x_2 & x & x \\ x_3 & x_4 & x & x \\ 0 & 0 & -x_1 & -x_3 \\ 0 & 0 & -x_2 & -x_4 \end{array} \right] = \left[ \begin{array}{cc} \Delta & X \\ 0 & -\Delta^T \end{array} \right],$$

where the  $2 \times 2$  matrix  $\Delta$  has two eigenvalues  $-\lambda, -\mu$  with negative real part. Using the same approach as in (4.1) an orthogonal matrix  $U_1$  can be found such that

$$U_1^T \left[ \begin{array}{cc} x_1 & x_2 \\ x_3 & x_4 \end{array} \right] U_1 = \left[ \begin{array}{cc} -\alpha & d \\ b & -\alpha \end{array} \right],$$

where  $\alpha, b, d \in \mathbb{R}$ , and  $-\lambda = -\alpha + i\beta, -\mu = -\alpha - i\beta$  with  $i\beta = \sqrt{bd}$ . Using

$$U = \text{diag}(U_1, U_1),$$

yields

$$U^T Q^T H_{4 \times 4} Q U = \left[ \begin{array}{cc|cc} -\alpha & d & x & x \\ b & -\alpha & x & x \\ 0 & 0 & \alpha & -b \\ 0 & 0 & -d & \alpha \end{array} \right].$$

In case eigenvectors have to be read off, transformations as in (4.2) or (4.3) have to be applied to achieve

$$\left[ \begin{array}{cc|cc} -\alpha & \beta & x & x \\ -\beta & -\alpha & x & x \\ \hline 0 & 0 & \alpha & -\beta \\ 0 & 0 & \beta & \alpha \end{array} \right].$$

If we again lift  $QU$  into the suitable  $2n \times 2n$  orthogonal symplectic  $G$ , then  $SG$  has columns  $j$  and  $j+1$  belonging to the (stable) invariant subspace of  $H$ . This completes the discussion for the case that the  $4 \times 4$  subproblem has complex eigenvalues with nonzero real part.

REMARK 4.3. *In case, a different ordering of the diagonal blocks in (4.4) is desired, a reordering is possible later using the ideas presented in, e.g., [32, Section 4.5.4]. An orthogonal symplectic matrix  $U$  can be found such that*

$$(4.5) \quad U^T \begin{bmatrix} \Delta & X \\ 0 & -\Delta^T \end{bmatrix} U = \begin{bmatrix} \tilde{\Delta} & X \\ 0 & -\tilde{\Delta}^T \end{bmatrix}$$

where  $\Delta$  has eigenvalue with negative real part and  $\tilde{\Delta}$  has eigenvalues with positive real part. If  $Y$  is the solution of the Lyapunov equation

$$\Delta Y - Y \Delta^T = X,$$

then  $Y$  is symmetric and consequently the columns of  $[-Y, I]^T$  span an isotropic subspace of the  $4 \times 4$  Hamiltonian matrix. Thus, there exists a symplectic  $QR$  decomposition

$$\begin{bmatrix} -Y \\ I \end{bmatrix} = U \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

By direct computation, it can be seen that  $U$  is an orthogonal symplectic matrix which produces a reordering of the form (4.5). In our implementation, the HAPACK [11] routine `haschord` is called to perform this computation.

Next let us consider the case that  $\theta_j \geq 0$ , that is case that the eigenvalues of  $H_{4 \times 4}$  are real or purely imaginary. In case  $\nu_j \nu_{j+1} \neq 0$ , we suggest to use a few implicit double shift *SR* steps in order to decouple the problem further into two  $2 \times 2$  subproblems. These can be solved as discussed at the beginning of this section. The same approach is used in the case  $\nu_j \nu_{j+1} = 0$  and either  $\nu_j = 0$  or  $\nu_{j+1} = 0$ , but not both of them. If  $\nu_{j+1} \neq 0$ ,  $H_{4 \times 4}$  should first be permuted using

$$(4.6) \quad J_P = \left[ \begin{array}{cc|cc} & & 1 & \\ & & & \\ \hline -1 & & & \\ & & & 1 \\ \hline & & & -1 \end{array} \right]$$

such that we have

$$H_{4 \times 4} = \left[ \begin{array}{cc|cc} \delta_j & 0 & \beta_j & \zeta_{j+1} \\ 0 & \delta_{j+1} & \zeta_{j+1} & \beta_{j+1} \\ \hline \nu_j & 0 & -\delta_j & 0 \\ 0 & 0 & 0 & -\delta_{j+1} \end{array} \right]$$

in both cases.

If  $\nu_j = \nu_{j+1} = 0$  we have

$$H_{4 \times 4} = \left[ \begin{array}{cc|cc} \delta_j & 0 & \beta_j & \zeta_{j+1} \\ 0 & \delta_{j+1} & \zeta_{j+1} & \beta_{j+1} \\ \hline 0 & 0 & -\delta_j & 0 \\ 0 & 0 & 0 & -\delta_{j+1} \end{array} \right],$$

and the eigenproblem decouples right away. In case the eigenvalues appear in the wrong order on the diagonal, they can be reordered as follows. In order to interchange  $\delta_{j+1}$  and  $-\delta_{j+1}$  a rotation of the form

$$G = \begin{bmatrix} 1 & & & \\ & c & & s \\ & & 1 & \\ & -s & & c \end{bmatrix}, \quad \begin{aligned} c &= \beta_{j+1} / \sqrt{\beta_{j+1} + 4\delta_{j+1}^2} \\ s &= 2\delta_{j+1} / \sqrt{\beta_{j+1} + 4\delta_{j+1}^2} \end{aligned}$$

needs to be applied

$$G^T H_{4 \times 4} G = \left[ \begin{array}{cc|cc} \delta_j & -s\zeta_{j+1} & \beta_j & c\zeta_{j+1} \\ 0 & -\delta_{j+1} & c\zeta_{j+1} & \beta_{j+1} \\ \hline 0 & 0 & -\delta_j & 0 \\ 0 & 0 & s\zeta_{j+1} & \delta_{j+1} \end{array} \right].$$

Similarly,  $\delta_j$  and  $-\delta_j$  can be interchanged;

$$G = \begin{bmatrix} & c & & s \\ & & 1 & \\ -s & & & c \\ & & & & 1 \end{bmatrix}, \quad \begin{aligned} c &= \beta_j / \sqrt{\beta_j + 4\delta_j^2} \\ s &= 2\delta_j / \sqrt{\beta_j + 4\delta_j^2} \end{aligned}$$

needs to be applied

$$G^T H_{4 \times 4} G = \left[ \begin{array}{cc|cc} -\delta_j & 0 & \beta_j & c\zeta_{j+1} \\ -s\zeta_{j+1} & \delta_{j+1} & c\zeta_{j+1} & \beta_{j+1} \\ \hline 0 & 0 & \delta_j & s\zeta_{j+1} \\ 0 & 0 & 0 & -\delta_{j+1} \end{array} \right].$$

An invariant subspace can be read off from this form. In case, eigenvectors are desired, a similarity transformation with  $J_P$  as in (4.6) will achieve

$$\left[ \begin{array}{cc|cc} \delta_{j+1} & s\zeta_{j+1} & \beta_{j+1} & -c\zeta_{j+1} \\ 0 & -\delta_j & -c\zeta_{j+1} & \beta_j \\ \hline 0 & 0 & -\delta_{j+1} & 0 \\ 0 & 0 & -s\zeta_{j+1} & \delta_j \end{array} \right].$$

In case both eigenvalues have to be interchanged, that is, the (1,1) and the (2,2) block have to be interchanged, this can be done as described in Remark 4.3.

REMARK 4.4. In [49], it is suggested to further reduce the  $4 \times 4$  problem

$$H'_{4 \times 4} = \begin{bmatrix} \Delta_j & X_j \\ 0 & -\Delta_j^T \end{bmatrix},$$

where  $\Delta_j$  contains the stable eigenvalues  $-\lambda, -\bar{\lambda}$ , to block diagonal form. The column range of  $U_j = (H'_{4 \times 4} + \lambda_j I)(H'_{4 \times 4} + \bar{\lambda} I)$  spans the unstable invariant subspace. This implies

$$\text{span}(U_j) = \text{span}\left(\begin{bmatrix} \Delta_j X_j - X_j \Delta_j^T + 2\text{Re}(\lambda) \Delta_j \\ -4\text{Re}(\lambda) \Delta_j^T \end{bmatrix}\right) = \text{span}\left(\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}\right).$$

As  $Z_2$  is regular, we can define

$$F_j = \begin{bmatrix} Z_2^{-T} & Z_1 \\ 0 & Z_2 \end{bmatrix}.$$

$F_j$  is symplectic and

$$F_j^{-1} H'_{4 \times 4} F_j = \begin{bmatrix} \Delta_j & \\ & -\Delta_j^T \end{bmatrix}.$$

This allows to read off the eigenvectors to all eigenvalues, but, this additional transformation may increase the numerical instability of the SR algorithm.

Such an additional transformation is not necessary as the eigenvectors can be read off directly from

$$\begin{bmatrix} \Delta & X \\ 0 & -\Delta^T \end{bmatrix}, \quad \text{with} \quad \Delta = \begin{bmatrix} -\alpha & \beta \\ -\beta & -\alpha \end{bmatrix}, \quad \alpha, \beta \in \mathbb{R}, \alpha > 0.$$

The vectors

$$z_1 = \begin{bmatrix} 1 \\ i \\ 0 \\ 0 \end{bmatrix}, \quad z_2 = \begin{bmatrix} i \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

are eigenvectors corresponding to the eigenvalues

$$\lambda_1 = -\alpha + i\beta, \lambda_2 = -\alpha - i\beta.$$

In order to obtain the eigenvectors corresponding to the eigenvalues

$$\lambda_3 = \alpha + i\beta, \lambda_4 = \alpha - i\beta$$

consider for  $j = 3, 4$

$$\begin{bmatrix} \Delta & X \\ 0 & -\Delta^T \end{bmatrix} z_j = \lambda_j z_j, \quad z_j = \begin{bmatrix} z_{j1} \\ z_{j2} \end{bmatrix}, \quad z_{j1}, z_{j2} \in \mathbb{C}^2.$$

This implies

$$-\Delta^T z_{j2} = \lambda_j z_{j2}, \quad \text{and} \quad \Delta z_{j1} + X z_{j2} = \lambda_j z_{j1}.$$

The first system is solved easily

$$z_{32} = \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \text{and} \quad z_{42} = \begin{bmatrix} i \\ 1 \end{bmatrix}.$$

This allows to rewrite the second system as

$$(\Delta - \lambda_j I)z_{j1} = -Xz_{j2}.$$

As  $\lambda_j$  is not an eigenvalue of  $\Delta$ , this  $2 \times 2$  systems of linear equations can easily be solved. This determines the eigenvectors

$$z_3 = \begin{bmatrix} z_{31} \\ 1 \\ i \end{bmatrix}, \quad z_4 = \begin{bmatrix} z_{41} \\ i \\ 1 \end{bmatrix}.$$

When all  $2 \times 2$  and  $4 \times 4$  subproblems are solved,  $H$  has been transformed into a matrix of the form

$$\left[ \begin{array}{ccc|ccc} H_{11} & & & H_{1,m+1} & & \\ & H_{22} & & & H_{2,m+2} & \\ & & \ddots & & & \ddots \\ & & & H_{mm} & & H_{m,2m} \\ \hline H_{m+1,1} & & & -H_{11}^T & & \\ & H_{m+2,2} & & & -H_{22}^T & \\ & & \ddots & & & \ddots \\ & & & H_{2m,m} & & -H_{mm}^T \end{array} \right]$$

where the blocks  $H_{ij}$  are either  $1 \times 1$  or  $2 \times 2$ . If the block  $H_{jj}$  is of order  $2 \times 2$ , then the block  $H_{m+j,j} = 0$  and  $H_{jj}$  has a pair of complex conjugate eigenvalues with negative real part. If the block  $H_{jj}$  is of order  $1 \times 1$  and  $H_{m+j,j} = 0$ , then  $H_{jj}$  is a negative real eigenvalue of  $H$ , otherwise (that is, if  $H_{m+j,j} \neq 0$ )  $H_{jj} = 0$  and the  $2 \times 2$  submatrix

$$\begin{bmatrix} H_{jj} & H_{j,m+j} \\ H_{m+j,j} & -H_{jj}^T \end{bmatrix} = \begin{bmatrix} 0 & H_{j,m+j} \\ H_{m+j,j} & 0 \end{bmatrix}$$

represents a pair of purely imaginary eigenvalue.

Any order of the eigenvalues on the diagonal is possible. As

$$\begin{bmatrix} P^T & 0 \\ 0 & P^T \end{bmatrix} \begin{bmatrix} A & G \\ Q & -A^T \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} = \begin{bmatrix} P^T A P & P^T G P \\ P^T Q P & -P^T A P \end{bmatrix},$$

we can easily rearrange the order of the eigenvalues by appropriate permutations. If, e.g., one wants to move the blocks corresponding to purely imaginary eigenvalues all the way to the end of the matrix, then this can be done as follows: Assume the purely imaginary eigenvalue is represented by the entries  $H_{jj}$ ,  $H_{m+j,j}$ ,  $H_{j,m+j}$  and  $H_{j+1,j+1}$  is a  $2 \times 2$  block, then a permutation

$$\begin{bmatrix} 0 & I_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} H_{xy} & 0 \\ 0 & H_{x+1,y+1} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ I_2 & 0 \end{bmatrix} = \begin{bmatrix} H_{x+1,y+1} & 0 \\ 0 & H_{xy} \end{bmatrix}, \quad x, y = j, m+j$$

will interchange the blocks  $H_{jj}$  and  $H_{j+1,j+1}$ , as well as the corresponding blocks  $H_{j,m+j}$  and  $H_{j+1,m+j+1}$ ,  $H_{m+j,j}$  and  $H_{m+j+1,j+1}$ , and  $H_{m+j,m+j}$  and  $H_{m+j+1,m+j+1}$ . If  $H_{j+1,j+1}$  is a  $1 \times 1$  block, then a permutation with

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

will interchange the blocks. Proceeding in this way, all purely imaginary eigenvalue can be moved to the end of the matrix such that

$$(4.7) \quad \left[ \begin{array}{c|c} \begin{array}{cccc} H_{11} & & & \\ & \ddots & & \\ & & H_{kk} & \\ & & & 0 \\ & & & & \ddots \\ 0 & & & & & 0 \end{array} & \begin{array}{cccc} H_{1,r} & & & \\ & \ddots & & \\ & & H_{k,\ell} & \\ & & & H_{q,p} \\ & & & & \ddots \\ & & & & & H_{m,2m} \end{array} \\ \hline \begin{array}{cccc} & & & \\ & & & \\ & & 0 & \\ & & & H_{p,q} \\ & & & & \ddots \\ & & & & & H_{2m,m} \end{array} & \begin{array}{cccc} -H_{11}^T & & & \\ & \ddots & & \\ & & -H_{kk}^T & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{array} \end{array} \right]$$

with  $\ell = m + k, p = \ell + 1, q = k + 1, r = m + 1$ , where the blocks  $H_{11}$  to  $H_{kk}$  represent the real or complex eigenvalues with negative real part. Eigenvectors can be read off the Schur-like form (4.7) obtained via the  $SR$  iteration (see Remarks 4.1, 4.4). The eigenvectors of the original Hamiltonian matrix  $H$  can be obtained from the eigenvectors  $z$  of the matrix  $\tilde{H}$  (4.7) if the transformation matrix  $S$  which transforms  $H$  into the form (4.7) is accumulated:

$$HSz = S\tilde{H}z = \lambda Sz.$$

In case  $S$  is badly conditioned or has not been accumulated, inverse iteration might be more advisable in order to compute an eigenvector of the original Hamiltonian  $J$ -Hessenberg matrix. This has been first discussed in [44], for a parameterized version see [26]. Due to the sparse structure of a Hamiltonian  $J$ -Hessenberg form inverse iteration can be implemented as an  $\mathcal{O}(n)$  process.

**5. Numerical Experiments.** The parameterized  $SR$  algorithm for Hamiltonian matrices as described in the previous sections has been implemented in MATLAB Release 2006a. Numerical tests were run on a Pentium M processor. The tolerance for declaring deflation was chosen as

$$|\zeta_j| \leq 2^{-52}(|\delta_{j-1}| + |\delta_j|).$$

A symplectic Gauss transformation was reject when its condition number was larger than  $10^8$ . This never happened for random test cases.

First numerical tests to determine the convergence properties of the algorithm have been performed. 100 random Hamiltonian  $J$ -Hessenberg matrices of the size  $2n \times 2n$ ,  $n = 3, \dots, 200$  were generated and the average number of iterations needed for computing all eigenvalues has been determined. Here each implicit quadruple shift  $SR$  step was counted as one iteration, no matter how small the problem has become due to deflation. Figure 5.1 displays the average number of iterations needed as well as the maximal and the minimal number of iterations needed within the test set of 100 matrices. This data shows that on average, 0.706 iterations per eigenvalue are required. When considering only smaller matrices ( $n = 3, \dots, 20$ ), on average only 0.67 iterations per eigenvalue are required. This is comparable to the number of

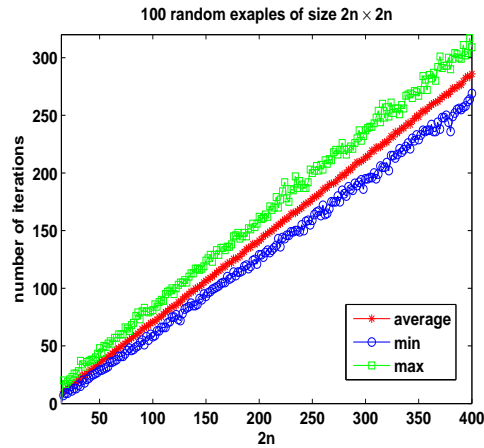


FIG. 5.1. Average number of iterations needed to compute all eigenvalues

iterations needed by the  $QR$  algorithm. Similar tests have been performed in [18]. There a different implementation (not parameterized) has been used, the code used only single precision. In [18], it was reported that 'The number of iterations needed to compute all eigenvalues of the  $2n \times 2n$  Hamiltonian  $J$ -Hessenberg matrix was between  $2n$  and  $4n$ . The average number of iterations for the computation of an eigenvalue was between 1 and 1.5.' In our tests with the non-parameterized version of the  $SR$  algorithm we also observed an increase in the number of iterations as compared to the number of iterations needed by the parameterized version.

As the implementation works on the parameters only, each iteration step requires  $\mathcal{O}(n)$  flops (not considering the update of the overall transformation matrix). Therefore, the overall work for the computation of the eigenvalues alone is  $\mathcal{O}(n^2)$  plus  $\mathcal{O}(n^3)$  for the initial reduction to  $J$ -Hessenberg form, which is comparable with the work for the  $QR$  algorithm for symmetric matrices.

In order to say something about the accuracy of the computed eigenvalues, let us consider the Hamiltonian  $J$ -Hessenberg matrix

$$H = \begin{bmatrix} 1 & & & & & & 19 & 2 \\ & 2 & & & & & 2 & 18 & 8 \\ & & 3 & & & & 8 & 17 & 5 \\ & & & 4 & & & 5 & 16 & 3 \\ & & & & 5 & & 3 & 15 & 6 \\ & & & & & 6 & 6 & 14 & \\ -3 & & & & & & -1 & & \\ & -5 & & & & & & -2 & \\ & & -7 & & & & & & -3 \\ & & & -9 & & & & & -4 \\ & & & & -11 & & & & -5 \\ & & & & & -13 & & & -6 \end{bmatrix}.$$

All eigenvalues are purely imaginary. The eigenvalues have been computed using the parameterized  $SR$  algorithm and the routine `hæig` from the HAPACK package, a different, structure-preserving eigensolvers for Hamiltonian eigenproblem. This routine does not make use of the Hamiltonian  $J$ -Hessenberg structure, but it is in contrast

to the  $SR$  algorithm backward stable. Hence we expect that it will perform slightly better than the (parameterized)  $SR$  algorithm. For each computed eigenvalue  $\lambda_j$ , the minimal singular value  $\sigma_j^{min}$  of  $H - \lambda_j I$  has been computed. In exact arithmetic, this value has to be zero.

$\sigma^{min}$ <b>haeig</b>	$\sigma^{min}$ $SR$	eigenpair
3.4843e-016	3.2871e-015	$0 \pm 6.1777e+000i$
3.5513e-015	3.6666e-015	$0 \pm 7.5082e+000i$
4.5359e-015	4.5359e-015	$0 \pm 8.1416e+000i$
7.1138e-016	6.8828e-015	$0 \pm 1.0691e+001i$
1.9564e-015	1.5470e-014	$0 \pm 1.3046e+001i$
2.0029e-015	3.0869e-015	$0 \pm 1.3046e+001i$

As expected, the eigenvalues computed by the  $SR$  algorithm are almost as accurate as those computed by **haeig**, only one digit might be lost.

Cubic convergence of the  $SR$  algorithm can clearly be seen for this example by considering the values  $\zeta_j$  during the iterations:

iteration	$\zeta_4$	$\zeta_5$
0	5.0000e+000	3.0000e+000
1	-5.1783e+000	-6.4104e-001
2	6.6055e+000	5.8846e-003
3	-4.6184e+000	-6.5287e-009
4	1.6696e+000	1.4244e-022
5	9.2538e-003	0
6	1.3824e-009	0
7	-7.6066e-024	0

Finally, consider the following  $4 \times 4$  problem

$$H = \left[ \begin{array}{cc|cc} 3 - \epsilon & 1 & -1 & -1 \\ 4 & 2 - \epsilon & -1 & -1 \\ \hline 4\epsilon - 11 & 2\epsilon - 5 & -3 + \epsilon & -4 \\ 2\epsilon - 5 & 2\epsilon - 2 & -1 & -2 + \epsilon \end{array} \right] = \begin{bmatrix} A & G \\ Q & -A^T \end{bmatrix}$$

is taken from the collection of benchmark examples for the numerical solution of algebraic Riccati equations [12, Example 11]. It represents a type of algebraic Riccati equations arising in  $H_\infty$ -control problems. The spectrum of  $H$  is  $\{\pm\epsilon \pm \sqrt{-1}\}$ ; the eigenvalues approach the imaginary axis as  $\epsilon \rightarrow 0$ . The solution of the corresponding Riccati equation

$$(5.1) \quad 0 = Q + A^T X + XA - XGX$$

can be computed using the stable invariant subspace of  $H$ ; i.e., the subspace corresponding to the eigenvalues of  $H$  in the open left half plane. If this subspace is spanned by  $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$  and  $U_1$  is invertible, then  $X = U_2 U_1^{-1}$  is the stabilizing solution of (5.1). For the example considered here, the matrix

$$X = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

solves (5.1) for arbitrary  $\epsilon$ . Computing the Hamiltonian  $J$ -Hessenberg form of  $H$  and using the direct approach of solving the resulting  $4 \times 4$  problem as discussed in Section 4, we obtain

$\epsilon$	$\ X - X_{computed}\ $	$ \lambda_{exact} - \lambda_{computed} $
$10^{-1}$	$6.5 \cdot 10^{-15}$	$5.2 \cdot 10^{-15}$
$10^{-2}$	$3.1 \cdot 10^{-14}$	$2.9 \cdot 10^{-14}$
$10^{-3}$	$1.2 \cdot 10^{-12}$	$5.8 \cdot 10^{-13}$
$10^{-4}$	$5.6 \cdot 10^{-12}$	$5.6 \cdot 10^{-12}$
$10^{-5}$	$1.3 \cdot 10^{-10}$	$1.3 \cdot 10^{-10}$
$10^{-6}$	$9.8 \cdot 10^{-10}$	$9.8 \cdot 10^{-10}$
$10^{-7}$	$9.7 \cdot 10^{-9}$	$5.1 \cdot 10^{-9}$
$10^{-8}$	$4.4 \cdot 10^{-8}$	$6.7 \cdot 10^{-9}$
$10^{-9}$	$4.3 \cdot 10^{-7}$	$4.9 \cdot 10^{-10}$
$10^{-10}$	$6.1 \cdot 10^{-7}$	$2.5 \cdot 10^{-10}$
$\vdots$	$\vdots$	$\vdots$
0	$1.2 \cdot 10^{-7}$	$4.4 \cdot 10^{-9}$

The accuracy of the computed eigenvalues degrades as  $\epsilon$  is chosen smaller and smaller, but from  $\epsilon = 10^{-7}$  on, the error in the computed eigenvalues is essentially of the order of  $10^{-9}$ , no matter how small  $\epsilon$  is chosen. Similarly, the accuracy of the computed solution  $X_{computed}$  of the Riccati equation degrades, until it finally stagnates by an error of the order of  $10^{-7}$ .

**6. Conclusions.** In this work, the parameterized implementation of the *SR* algorithm for  $2k \times 2k$  Hamiltonian *J*-Hessenberg matrices has been discussed in detail. In particular, the case of purely imaginary eigenvalues in the final step of the *SR* algorithm has been considered. Any  $2k \times 2k$  Hamiltonian (*J*-Hessenberg) matrix is uniquely determined by  $4k - 1$  parameters. Using these  $4k - 1$  parameters, each step of the *SR* algorithm can be carried out in  $\mathcal{O}(k)$  arithmetic operations (compared to  $\mathcal{O}(k^3)$  arithmetic operations when working on the actual Hamiltonian matrix). Numerical experiments show that this parameterized version works better than the non parameterized one. About 0.7 iterations are needed for the computation of one eigenvalue; cubic convergence can be observed.

**Acknowledgments** My thanks go to Peter Benner, Nil Mackey and Michael Overton for their hospitality in Chemnitz, Kalamazoo and New York during my sabbatical. I would like to point out that the implementation of the *SR* algorithm in [44] is nearly parameterized, although the implementation sets up and uses the Hamiltonian *J*-Hessenberg matrix as an  $2n \times 2n$  matrix.

#### REFERENCES

- [1] G.S. AMMAR, P. BENNER, AND V. MEHRMANN, *A multishift algorithm for the numerical solution of algebraic Riccati equations*, *Electr. Trans. Num. Anal.*, 1 (1993), pp. 33–48.
- [2] G.S. AMMAR AND V. MEHRMANN, *On Hamiltonian and symplectic Hessenberg forms*, *Linear Algebra Appl.*, 149 (1991), pp. 55–72.
- [3] 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 Publications, Philadelphia, PA, 2nd ed., 1995.
- [4] A.C. ANTOULAS, *A new result on positive real interpolation and model reduction*, *Systems & Control Letters*, 54 (2005), pp. 361–374.
- [5] P. BENNER, *Computational methods for linear-quadratic optimization*, *Supplementario ai Rendiconti del Circolo Matematico di Palermo, Serie II* (1997), pp. 21–56.
- [6] P. BENNER, R. BYERS, V. MEHRMANN, AND H. XU, *Numerical methods for linear quadratic and  $H_\infty$  control problems*, in *Dynamical systems, control, coding, computer vision: new trends, interfaces, and interplay*, G. Picci and D.S. Gilliam, eds., vol. 25 of *Progress in systems and control theory*, Birkhäuser, Basel, 1999, pp. 203–222.

- [7] ———, *Robust numerical methods for robust control*, Technical Report 06–2004, Institut für Mathematik, TU Berlin, Germany, 2004.
- [8] P. BENNER AND H. FASSBENDER, *A restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem*, Technical Report SPC 95.28, TU Chemnitz-Zwickau (now TU Chemnitz), TU Chemnitz, Fakultät für Mathematik, Germany, 1995.
- [9] ———, *An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem*, *Linear Algebra Appl.*, 263 (1997), pp. 75–111.
- [10] ———, *Computing passive reduced-order LTI models using structured Krylov subspace methods*, preprint, TU Braunschweig, Germany, 2006.
- [11] P. BENNER AND D. KRESSNER, *Algorithm 8xx: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices*, *ACM Transactions on Mathematical Software*, (to appear).
- [12] 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. Available from <http://www.tu-chemnitz.de/sfb393/spc95pr.html>.
- [13] P. BENNER, V. MEHRMANN, AND H. XU, *A new method for computing the stable invariant subspace of a real Hamiltonian matrix*, *J. Comput. Appl. Math.*, 86 (1997), pp. 17–43.
- [14] ———, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, *Numerische Mathematik*, 78 (1998), pp. 329–358.
- [15] ———, *A note on the numerical solution of complex Hamiltonian and skew-Hamiltonian eigenvalue problems*, *Electr. Trans. Num. Anal.*, 8 (1999), pp. 115–126.
- [16] S. BOYD, V. BALAKRISHNAN, AND P. KABAMBA, *A bisection method for computing the  $H_\infty$  norm of a transfer matrix and related problems*, *Math. Control, Signals, Sys.*, 2 (1989), pp. 207–219.
- [17] A. BUNSE-GERSTNER AND H. FASSBENDER, *A Jacobi-like method for solving algebraic Riccati equations on parallel computers*, *IEEE Trans. Automat. Control*, 42 (1997), pp. 1071–1087.
- [18] 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 (1986), pp. 1104–1113.
- [19] A. BUNSE-GERSTNER, V. MEHRMANN, AND D.S. WATKINS, *An SR algorithm for Hamiltonian matrices based on Gaussian elimination*, *Methods of Operations Research*, 58 (1989), pp. 339–356.
- [20] J.V. BURKE, A.S. LEWIS, AND M.L. OVERTON, *Robust stability and a criss-cross algorithm for pseudospectra*, *IMA J. Numer. Anal.*, 23 (2003), pp. 359–375.
- [21] R. BYERS, *Hamiltonian and symplectic algorithms for the algebraic Riccati equation*, PhDthesis, Cornell University, Dept. Comp. Sci., Ithaca, NY, 1983.
- [22] ———, *A Hamiltonian QR-algorithm*, *SIAM J. Sci. Statist. Comput.*, 7 (1986), pp. 212–229.
- [23] ———, *A bisection method for measuring the distance of a stable to unstable matrices*, *SIAM J. Sci. Statist. Comput.*, 9 (1988), pp. 875–881.
- [24] ———, *A Hamiltonian-Jacobi algorithm*, *IEEE Trans. Automat. Control*, 35 (1990), pp. 566–570.
- [25] D. CHU, X. LIU, AND V. MEHRMANN, *A numerical method for computing the Hamiltonian Schur form*, to appear in *Numer. Math.*, (2006).
- [26] H. FASSBENDER, *A detailed derivation of the parameterized SR algorithm and the symplectic Lanczos method for Hamiltonian matrices*, Preprint, (2006).
- [27] H. FASSBENDER, D.S. MACKAY, AND N. MACKAY, *Hamilton and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems*, *Linear Algebra Appl.*, 332-334 (2001), pp. 37–80.
- [28] G. FREILING, V. MEHRMANN, AND H. XU, *Existence, uniqueness and parameterization of Lagrangian invariant subspaces*, *SIAM J. Matrix Anal. Appl.*, 23 (2002), pp. 1045–1069.
- [29] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 3rd ed., 1996.
- [30] C-H. GUO AND P. LANCASTER, *Analysis and modification of Newton’s method for algebraic riccati equations*, *Math. Comp.*, 67 (1998), pp. 1089–1105.
- [31] M.M. KONSTANTINOV, V. MEHRMANN, AND P.HR. PETKOV, *Perturbation analysis for the Hamiltonian Schur and block-Schur forms*, *SIAM J. Matrix Anal. Appl.*, 23 (2001), pp. 387–424.
- [32] D. KRESSNER, *Numerical Methods for General and Structured Eigenvalue*, vol. 46 of *Lecture Notes in Computational Science and Engineering*, Springer, Berlin, 2005.
- [33] P. LANCASTER AND L. RODMAN, *The Algebraic Riccati Equation*, Oxford University Press, Oxford, 1995.
- [34] W.-W. LIN AND T.-C. HO, *On Schur type decompositions for Hamiltonian and symplectic*

- pencils*, tech. report, Institute of Applied Mathematics, National Tsing Hua University, 1990.
- [35] W.-W. LIN, V. MEHRMANN, AND H. XU, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 302-303 (1999), pp. 469–533.
  - [36] L. LOPEZ AND V. SIMONCINI, *Preserving geometric properties of the exponential matrix by block Krylov subspace methods*, Preprint, (2005).
  - [37] V. MEHRMANN, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, no. 163 in Lecture Notes in Control and Information Sciences, Springer-Verlag, Heidelberg, 1991.
  - [38] V. MEHRMANN AND E. TAN, *Defect correction methods for the solution of algebraic Riccati equations*, IEEE Trans. Automat. Control, 33 (1988), pp. 695–698.
  - [39] V. MEHRMANN AND D.S. WATKINS, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Statist. Comput., 22 (2000), pp. 1905–1925.
  - [40] C.C. PAIGE AND C.F. VAN LOAN, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 41 (1981), pp. 11–32.
  - [41] A.C. RAINES AND D.S. WATKINS, *A class of Hamiltonian-symplectic methods for solving the algebraic riccati equation*, Linear Algebra Appl., 205/206 (1994), pp. 1045–1060.
  - [42] V. SIMA, *Algorithms for Linear-Quadratic Optimization*, vol. 200 of Pure and Applied Mathematics, Marcel Dekker, Inc., New York, NY, 1996.
  - [43] D.C. SORENSEN, *Passivity preserving model reduction via interpolation of spectral zeros*, Systems & Control Letters, 54 (2005), pp. 347–360.
  - [44] M. STOLL, *Locking und Purging für den Hamiltonischen Lanczos-Prozess*, Diplomarbeit, TU Chemnitz, Fakultät für Mathematik, Germany, 2005.
  - [45] F. TISSEUR, *Stability of structured Hamiltonian eigensolvers*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 103–125.
  - [46] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Review, 43 (2001), pp. 235–286.
  - [47] C.F. VAN LOAN, *A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix*, Linear Algebra Appl., 16 (1984), pp. 233–251.
  - [48] D.S. WATKINS AND L. ELSNER, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.
  - [49] N. WONG, V. BALAKRISHNAN, AND C.-K. KOH, *Passivity-preserving model reduction via a computationally efficient project-and-balance scheme*, in Proc. Design Automation Conference, San Diego, CA, June 2004, pp. 369–374.