Electronic Transactions on Numerical Analysis.
Volume 29, pp. 212-229, 2008.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

SOLVING LARGE-SCALE QUADRATIC EIGENVALUE PROBLEMS WITH HAMILTONIAN EIGENSTRUCTURE USING A STRUCTURE-PRESERVING KRYLOV SUBSPACE METHOD*

PETER BENNER†, HEIKE FASSBENDER‡, AND MARTIN STOLL§

Abstract. We consider the numerical solution of quadratic eigenproblems with spectra that exhibit Hamiltonian symmetry. We propose to solve such problems by applying a Krylov-Schur-type method based on the symplectic Lanczos process to a structured linearization of the quadratic matrix polynomial. In order to compute interior eigenvalues, we discuss several shift-and-invert operators with Hamiltonian structure. Our approach is tested for several examples from structural analysis and gyroscopic systems.

Key words. quadratic eigenvalue problem, Hamiltonian symmetry, Krylov subspace method, symplectic Lanczos process, gyroscopic systems

AMS subject classifications. 65F15, 15A24, 47A75, 47H60

1. Introduction. This paper is concerned with the solution of the quadratic eigenvalue problem (QEP)

$$(\lambda^2 M + \lambda G + K)x = 0, \quad M = M^T,\ G = -G^T,\ K = K^T, \tag{1.1}$$

where $M, G, K \in \mathbb{R}^{n\times n}$ are large and sparse. The task is to compute $\lambda \in \mathbb{C}$ and $x \in \mathbb{C}^n \setminus \{0\}$ such that (1.1) holds. It can be shown [1, 22, 33] that the eigenproblem (1.1) has a Hamiltonian eigenstructure, that is, the eigenvalues are symmetric with respect to both axes. In other words, if $\lambda \in \mathbb{C}$ is an eigenvalue with $\mathrm{Re}(\lambda) \neq 0$, then so are $-\lambda, \overline{\lambda}, -\overline{\lambda}$, while if $\lambda \in \mathbb{R}$ or $\lambda \in i\mathbb{R}$ is an eigenvalue, then so is $-\lambda$.

QEPs with Hamiltonian eigenstructure arise, for example, in the finite-element analysis of elastic deformation of anisotropic materials or when computing corner singularities; see, e.g., [1, 25] and the references therein. In these applications M and $-K$ are positive definite mass and stiffness matrices, respectively. Gyroscopic systems are another source of quadratic eigenproblems (1.1). Here, M and K are positive definite mass and stiffness matrices and G is the gyroscopic matrix resulting from the Coriolis force. Such systems arise when modeling vibrations of spinning structures, such as the simulation of tire noise, helicopter rotor blades, or spin-stabilized satellites with appended solar panels or antennas; see [10, 18, 33] and references therein. In the simulation of vibro-acoustics in flexible piping systems, the coupling of the linear wave equation without internal flow and the structural Lamé-Navier equations at fluid-structure interfaces also leads to a QEP with Hamiltonian symmetry [21]. A summary of conditions under which quadratic operator eigenvalue problems have a spectrum with Hamiltonian symmetry is given in [26]. Depending on the application, different parts of the spectrum are of interest. Typically, one is interested in the eigenvalues with smallest real part or the eigenvalues smallest or largest in modulus. The usual approach is to first linearize the quadratic eigenproblem to a generalized eigenproblem, transform this into a standard eigenproblem, and solve the latter by using a (shift-and-invert) Krylov subspace method. There are also different approaches such as the SOAR (second-order Arnoldi) algorithm [2], a Krylov

*Received October 31, 2006. Accepted for publication December 19, 2008. Published online on February 24, 2009. Recommended by B. Lang.

†TU Chemnitz, Fakultät für Mathematik, Mathematik in Industrie und Technik, 09107 Chemnitz, Germany (benner@mathematik.tu-chemnitz.de).

‡AG Numerik, Institut Computational Mathematics, TU Braunschweig, D-38023 Braunschweig, Germany (h.fassbender@tu-bs.de).

§Mathematical Institute, 24-29 St Giles', Oxford, OX1 3LB, England (stoll@maths.ox.ac.uk).

ETNA
Kent State University
etna@mcs.kent.edu

subspace based method for the solution of the QEP, or the Jacobi-Davidson algorithm applied to polynomial eigenvalue problems [28]. Both approaches can be applied to the QEP directly without any linearization.

Here we will apply the usual approach of first linearizing the QEP (1.1) by a suitable linearization. If a structure-preserving linearization is used, then the resulting eigenproblem exhibits the same Hamiltonian eigenstructure as the QEP (1.1); for a general discussion, see, e.g., [20]. In [1, 22, 33] several linearizations for (1.1) have been proposed. With $y = \lambda M x$ the skew-Hamiltonian/Hamiltonian eigenproblem

$$\lambda \mathcal{N} z - \mathcal{H} z = \lambda \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} 0 & -K \\ M^{-1} & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0$$

is obtained. Here $\mathcal{H}$ is a Hamiltonian matrix, that is

$$(J\mathcal{H})^T = J\mathcal{H},$$

where

$$J \equiv J^n = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \tag{1.2}$$

and $\mathcal{N}$ is a skew-Hamiltonian matrix, that is

$$(J\mathcal{N})^T = -J\mathcal{N}.$$

Since $\mathcal{N}$ is invertible, the pencil $\lambda\mathcal{N} - \mathcal{H}$ is regular. The skew-Hamiltonian matrix $\mathcal{N}$ can be factorized as

$$\mathcal{N} = \mathcal{Z}\mathcal{Z} = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix}.$$

Thus,

$$\lambda\mathcal{N} - \mathcal{H} = \mathcal{Z}(\lambda I - \mathcal{Z}^{-1}\mathcal{H}\mathcal{Z}^{-1})\mathcal{Z} = \mathcal{Z}(\lambda I - H)\mathcal{Z},$$

with the Hamiltonian matrix $H = \mathcal{Z}^{-1}\mathcal{H}\mathcal{Z}^{-1}$. Since

$$\mathcal{Z}^{-1} = \begin{bmatrix} I & -\frac{1}{2}G \\ 0 & I \end{bmatrix},$$

we have

$$H = \begin{bmatrix} I & -\frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -K \\ M^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & -\frac{1}{2}G \\ 0 & I \end{bmatrix}. \tag{1.3}$$

There are other linearizations which yield a standard eigenvalue problem for a Hamiltonian matrix [22].

Here we are concerned with algorithms for solving the quadratic eigenvalue problem which preserve and exploit its Hamiltonian eigenstructure. It is well-known that symplectic similarity transformations preserve the Hamiltonian structure. That is, if $S \in \mathbb{R}^{2n \times 2n}$ is a symplectic matrix, i.e.,

$$S^T J S = J$$

and $H \in \mathbb{R}^{2n \times 2n}$ is Hamiltonian, then $S^{-1}HS$ is Hamiltonian as well. Hence, algorithms based on symplectic transformations are structure-preserving. The payoffs are, in general,

ETNA
Kent State University
etna@mcs.kent.edu

more efficient and often more accurate algorithms. In some cases, preservation of the structure is crucial to obtain a physically meaningful solution; see [7, 11, 17].

Typical applications require a few eigenvalues that are largest or smallest in magnitude or closest to the imaginary axis. Computing the ones of largest magnitude can be achieved efficiently by Krylov subspace methods, e.g., Arnoldi or Lanczos processes, possibly combined with implicit restarting or a Krylov-Schur-type technique [31]. To compute other eigenvalues, first transformations must be applied to the matrix H which have the effect of shifting the desired eigenvalues to the periphery of the spectrum. In light of the symmetry of the spectrum, one might think of working with $(H-\tau I)^{-1}(H+\tau I)^{-1}$, in case τ is real or purely imaginary. All eigenvalues near to $\pm\tau$ are mapped simultaneously to values of large modulus. But this matrix is not Hamiltonian, but skew-Hamiltonian. The standard (implicitly restarted) Arnoldi method automatically preserves this structure. This led to the development of the SHIRA method as a structure-preserving (shift-and-invert) Arnoldi method for Hamiltonian matrices [22].

Here we discuss an alternative based on the symplectic Lanczos algorithm [4], which preserves the Hamiltonian structure. In order to stay within the Hamiltonian structure, for a real shift τ, we can work with the Hamiltonian matrices

$$H_1(\tau) = H^{-1}(H-\tau I)^{-1}(H+\tau I)^{-1} = (H^3-\tau^2 H)^{-1}, \tag{1.4}$$

or

$$H_2(\tau) = H(H-\tau I)^{-1}(H+\tau I)^{-1} = (H-\tau^2 H^{-1})^{-1}, \tag{1.5}$$

for example. In case a complex shift τ is used, we can work with the Hamiltonian matrices

$$\begin{aligned} H_3(\tau) &= H^{-1}(H-\tau I)^{-1}(H+\tau I)^{-1}(H-\overline{\tau} I)^{-1}(H+\overline{\tau} I)^{-1} \\ &= (H^5-(\overline{\tau}^2+\tau^2)H^3+|\tau|^4 H)^{-1}, \end{aligned} \tag{1.6}$$

or

$$\begin{aligned} H_4(\tau) &= H(H-\tau I)^{-1}(H+\tau I)^{-1}(H-\overline{\tau} I)^{-1}(H+\overline{\tau} I)^{-1} \\ &= (H^3-(\overline{\tau}^2+\tau^2)H+|\tau|^4 H^{-1})^{-1}. \end{aligned} \tag{1.7}$$

The shift-and-invert operators $H_1(\tau), H_2(\tau), H_3(\tau)$ have first been considered in [13, 25, 34], while $H_4(\tau)$ is examined in [6]. We will apply a recently developed variant of the restarted symplectic Lanczos algorithm, the Hamiltonian Krylov-Schur-type method.

The next section briefly describes how to solve the large sparse Hamiltonian eigenvalue problem via a structure-preserving symplectic Lanczos algorithm. In Section 3 we give some details on the application of the symplectic Lanczos process to the QEP. Numerical examples are presented in Section 4.

2. The symplectic Lanczos method. The usual nonsymmetric Lanczos algorithm generates two sequences of vectors; see, e.g., [16]. Due to the Hamiltonian structure of H it is easily seen that one of the two sequences can be eliminated here and thus work and storage can essentially be halved. (This property is valid for a broader class of matrices; see [15].)

The structure-preserving symplectic Lanczos method [4, 13] generates a sequence of matrices that satisfy the Lanczos recursion

$$HS^{2n,2k} = S^{2n,2k}\widetilde{H}^{2k,2k} + \zeta_{k+1} v_{k+1} e_{2k}^T. \tag{2.1}$$

Here, $\widetilde{H}^{2k,2k}$ is a Hamiltonian J–Hessenberg matrix, that is, a Hamiltonian matrix of the form

$$\widetilde{H}^{2k,2k} = \left[\begin{array}{cccc|cccc} \delta_1 & & & & \beta_1 & \zeta_2 & & \\ & \delta_2 & & & \zeta_2 & \beta_2 & \ddots & \\ & & \ddots & & & \ddots & \ddots & \zeta_k \\ & & & \delta_k & & & \zeta_k & \beta_k \\ \hline \nu_1 & & & & -\delta_1 & & & \\ & \nu_2 & & & & -\delta_2 & & \\ & & \ddots & & & & \ddots & \\ & & & \nu_k & & & & -\delta_k \end{array}\right],$$

where $\delta_j, \beta_j, \nu_j, \zeta_j \in \mathbb{R}$ for $j = 1, \ldots, k$. The space spanned by the columns of $S^{2n,2k}$ is symplectic since $(S^{2n,2k})^T J^n S^{2n,2k} = J^k$, where J^j is a $2j \times 2j$ matrix of the form (1.2). Thus, the columns of $S^{2n,2k}$ are J–orthogonal. The vector $r_{k+1} := \zeta_{k+1} v_{k+1}$ is the residual vector and is J–orthogonal to the columns of $S^{2n,2k}$, the Lanczos vectors. The matrix $\widetilde{H}^{2k,2k}$ is the J–orthogonal projection of H onto the range of $S^{2n,2k}$,

$$\widetilde{H}^{2k,2k} = (J^k)^T (S^{2n,2k})^T J^n H S^{2n,2k}.$$

Hence, $\widetilde{H}^{2k,2k}$ is a Rayleigh quotient of H; as a J–orthogonal projection it is of Hamiltonian J–Hessenberg form.

Equation (2.1) defines a length $2k$ Lanczos factorization of H. If the residual vector r_{k+1} is the zero vector, then equation (2.1) is called a truncated Lanczos factorization when $k < n$. Note that r_{n+1} must vanish since $(S^{2n,2n})^T J r_{n+1} = 0$ and the columns of $S^{2n,2n}$ form a J–orthogonal basis for $\mathbb{R}^{2n}$. In this case the symplectic Lanczos method computes a reduction to J–Hessenberg form.

If the norm of the residual vector $r_{k+1}$is small, the $2k$ eigenvalues of $\widetilde{H}^{2k,2k}$ are approximations to the eigenvalues of H. Numerical experiments indicate that the norm of the residual rarely becomes small by itself. Nevertheless, some eigenvalues of $\widetilde{H}^{2k,2k}$ may be good approximations to eigenvalues of H. Let $\widetilde{\lambda}$ be an eigenvalue of $\widetilde{H}^{2k,2k}$ with the corresponding eigenvector $\widetilde{y}$. Then the vector $x = S^{2n,2k}\widetilde{y}$ is referred to as Ritz vector and $\widetilde{\lambda}$ as Ritz value of H. Note that the Ritz values exhibit Hamiltonian symmetry as the Rayleigh quotient $\widetilde{H}^{2k,2k}$ is Hamiltonian.

A symplectic Lanczos factorization exists for almost all starting vectors $S^{2n,2n}e_1 = v_1$. Moreover, the symplectic Lanczos factorization is, up to multiplication by a trivial matrix, specified by the starting vector v_1. Hence, as this reduction is strongly dependent on the first column of the transformation matrix that carries out the reduction, breakdown or near–breakdown in the Lanczos process is possible. Assume that no such breakdowns occur, and let $S^{2n,2n} = [v_1, v_2, \ldots, v_n, w_1, w_2, \ldots, w_n]$. For a given v_1, a Lanczos method constructs the matrix $S^{2n,2n}$ columnwise from the equations

$$HS^{2n,2n}e_j = S^{2n,2n}\widetilde{H}^{2n,2n}e_j, \qquad j = n+1, 2, n+2, 3, \ldots 2n.$$

This yields Algorithm 1, where $\langle \cdot, \cdot \rangle_J$ denotes the J–inner product

$$\langle v, w \rangle_J = v^T J w.$$

Eigenvalues and eigenvectors of Hamiltonian J–Hessenberg matrices such as $\widetilde{H}^{2k,2k}$ can be computed efficiently by the SR algorithm. This has been discussed to some extent in

Algorithm 1 Algorithm for a symplectic Lanczos method

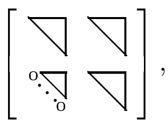
INPUT : $H \in \mathbb{R}^{2n\times 2n}$ and $k \in \mathbb{N}$.
OUTPUT : $S \in \mathbb{R}^{2n\times 2k}$, $\delta_1,\ldots,\delta_k$, $\beta_1,\ldots,\beta_k$, $\nu_1,\ldots,\nu_k$, $\zeta_2,\ldots,\zeta_{k+1}$ and v_{k+1} (assuming no break down).
Choose start vector $\widetilde{v}_1 \neq 0 \in \mathbb{R}^{2n}$.
$v_0 = 0 \in \mathbb{R}^{2n}$
$\zeta_1 = \|\widetilde{v}_1\|_2$
$v_1 = \frac{1}{\zeta_1}\widetilde{v}_1$
for $m = 1,2,\ldots,k$ **do**
  `% Computation of matrix-vector products`
  $u_1 = Hv_m$
  $u_2 = Hw_m$
  `% Computation of` δ_m
  $\delta_m = v_m^T u_1$
  `% Computation of` w_m
  $\widetilde{w}_m = u_1 - \delta_m v_m$
  $\nu_m = \langle v_m, u_1\rangle_J$
  $w_m = \frac{1}{\nu_m}\widetilde{w}_m$
  `% Computation of` β_m
  $\beta_m = -\langle w_m, u_2\rangle_J$
  `% Computation of` v_{m+1}
  $\widetilde{v}_{m+1} = u_2 - \zeta_m v_{m-1} - \beta_m v_m + \delta_m w_m$
  $\zeta_{m+1} = \|\widetilde{v}_{m+1}\|_2$
  $v_{m+1} = \frac{1}{\zeta_{m+1}}\widetilde{v}_{m+1}$
end for

[3, 4, 9, 36]. Assume that we have obtained (2.1). If $\widetilde{H}$ is the current iterate, then a spectral transformation function q is chosen (such that $q(\widetilde{H}) \in \mathbb{R}^{2k\times 2k}$) and the SR decomposition of $q(\widetilde{H})$ is formed, if possible:

$$q(\widetilde{H}) = SR,$$

where $S \in \mathbb{R}^{2k\times 2k}$ is symplectic and R is a J–triangular matrix,

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



i.e., R_{11}, R_{12}, R_{22} are upper triangular matrices and R_{21} is a strictly upper triangular matrix. Then the symplectic factor S is used to perform a similarity transformation on $\widetilde{H}$ to yield the next iterate, which is called $\widehat{H}$:

$$\widehat{H} = S^{-1}\widetilde{H}S. \tag{2.2}$$

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

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

or

$$q_4(\widetilde{H}) = (\widetilde{H} - \mu I)(\widetilde{H} + \mu I)(\widetilde{H} - \overline{\mu} I)(\widetilde{H} + \overline{\mu} I), \qquad \mu \in \mathbb{C}, \mathrm{Re}(\mu) \neq 0.$$

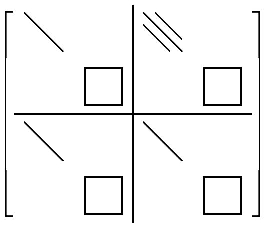
ETNA
Kent State University
etna@mcs.kent.edu

As proposed in [9], 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 4×4 Hamiltonian J–Hessenberg submatrix

$$\widetilde{H}_{4\times 4} = \left[\begin{array}{cc|cc} \delta_{k-1} & & \beta_{k-1} & \zeta_k \\ & \delta_k & \zeta_k & \beta_k \\ \hline \nu_{k-1} & & -\delta_{k-1} & \\ & \nu_k & & -\delta_k \end{array}\right].$$

This is exactly the generalized Rayleigh-quotient strategy for choosing the shifts proposed by Watkins and Elsner in [35]. Hence the convergence theorems 6.2, 6.3 and 6.5 from [35] can be applied here. In particular, the Hamiltonian SR algorithm is typically cubically convergent.

If $\operatorname{rank}(q(\widetilde{H})) = 2k$ and H (and hence, $\widetilde{H}$) is a Hamiltonian J–Hessenberg matrix, then so is $\widehat{H}$ in (2.2). If $\operatorname{rank}(q(\widetilde{H})) = 2k - \nu =: 2\ell$ and $\widetilde{H}$ is an unreduced Hamiltonian J–Hessenberg matrix, then $\widehat{H}$ is of the form

(2.3) $\widehat{H} =$  $= \left[\begin{array}{cc|cc} \widehat{H}_{11} & & \widehat{H}_{13} & \\ & \widehat{H}_{22} & & \widehat{H}_{24} \\ \hline \widehat{H}_{31} & & -\widehat{H}_{11}^T & \\ & \widehat{H}_{42} & & -\widehat{H}_{22}^T \end{array}\right],$

where $\widehat{H}_{11}, \widehat{H}_{13}, \widehat{H}_{31} \in \mathbb{R}^{\ell\times\ell}$ and $\widehat{H}_{22}, \widehat{H}_{24}, \widehat{H}_{42} \in \mathbb{R}^{k-\ell\times k-\ell}$, and

- $\begin{bmatrix} \widehat{H}_{11} & \widehat{H}_{13} \\ \widehat{H}_{31} & -\widehat{H}_{11}^T \end{bmatrix}$ is a Hamiltonian J–Hessenberg matrix,
- the eigenvalues of $\begin{bmatrix} \widehat{H}_{22} & \widehat{H}_{24} \\ \widehat{H}_{42} & -\widehat{H}_{22}^T \end{bmatrix}$ are the ν roots of $q(H)$ that are eigenvalues of $\widetilde{H}$.

An algorithm for computing S and R explicitly is presented in [9]. As with explicit QR steps, the expense of explicit SR steps comes from the fact that $q(\widetilde{H})$ has to be computed explicitly. A preferred alternative is the implicit SR step, an analog to the Francis QR step [14]. The first implicit transformation S_1 is selected in order to introduce a bulge into the J–Hessenberg matrix $\widetilde{H}$. Hence, a symplectic matrix S_1 is determined, such that

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

where $q(\widetilde{H})$ is an appropriately chosen spectral transformation function, e.g., $q_4(\widetilde{H})$, where the shift is chosen according to the generalized Rayleigh-quotient strategy as an eigenvalue of $\widetilde{H}_{4\times 4}$. Note that there is no need to compute the eigenvalues of $\widetilde{H}_{4\times 4}$ directly. Comparing $q_4(\widetilde{H})$ with the characteristic polynomial of $\widetilde{H}_{4\times 4}$ gives the first column of $q_4(\widetilde{H})$ which is needed to start the implicit SR step explicitly. The same can be done for the double shift case.

Applying the first transformation S_1 to the J–Hessenberg matrix yields a Hamiltonian matrix $S_1^{-1}\widetilde{H}S_1$ 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. Hence, a symplectic matrix S_2 is determined, such that $S_2^{-1}S_1^{-1}\widetilde{H}S_1S_2$ is of J–Hessenberg form again. If $\widetilde{H}$ is an unreduced J–Hessenberg matrix and $\operatorname{rank}(q(\widetilde{H})) =$

$2k$, then $\breve{H} = S_2^{-1}S_1^{-1}\widetilde{H}S_1S_2$ is also an unreduced J–Hessenberg matrix. Hence, there will be parameters $\breve{\delta}_1,\ldots,\breve{\delta}_k,\breve{\beta}_1,\ldots,\breve{\beta}_k$, $\breve{\zeta}_1,\ldots,\breve{\zeta}_k$, $\breve{\nu}_2,\ldots,\breve{\nu}_k$ which determine $\breve{H}$. An efficient implementation of the SR step for Hamiltonian J–Hessenberg matrices involves $\mathcal{O}(n)$ arithmetic operations [9, 12].

The symplectic Lanczos method described above inherits all numerical difficulties of Lanczos-like methods for nonsymmetric matrices, in particular serious breakdown is possible. One approach to deal with the numerical difficulties of Lanczos-like algorithms is to implicitly restart the Lanczos factorization. This approach was introduced by Sorensen [29] in the context of nonsymmetric matrices and the Arnoldi process. Usually only a small subset of the eigenvalues is desired. As the eigenvalues of the Hamiltonian J–Hessenberg matrices $\widetilde{H}^{2k,2k}$ are estimates for the eigenvalues of H, the length $2k$ symplectic Lanczos factorization (2.1) may suffice if the residual vector r_{k+1} is small. The idea of restarted Lanczos algorithms is to fix the number of steps in the Lanczos process at a prescribed value k which is dependent on the required number of approximate eigenvalues. The purpose of the implicit restart is to determine initial vectors such that the associated residual vectors are tiny. Given (2.1), an implicit Lanczos restart computes the Lanczos factorization

$$H\breve{S}^{2n,2k} = \breve{S}^{2n,2k}\breve{H}^{2k,2k} + \breve{r}_{k+1}e_{2k}^T, \tag{2.4}$$

which corresponds to the starting vector

$$\breve{s}_1 = p(H)s_1,$$

(where p is a polynomial such that $p(H)$ is real) without having to explicitly restart the Lanczos process with the vector $\breve{s}_1$. This process is iterated until the residual vector r_{k+1} is tiny. J–orthogonality of the k Lanczos vectors is secured by re–J–orthogonalizing these vectors when necessary. As the iteration progresses, some of the Ritz values may converge to eigenvalues of H long before the entire set of wanted eigenvalues have. These converged Ritz values may be part of the wanted or unwanted portion of the spectrum. In either case it is desirable to deflate the converged Ritz values and corresponding Ritz vectors from the unconverged portion of the factorization. If the converged Ritz value is wanted then it is necessary to keep it in the subsequent factorizations; if it is unwanted then it must be removed from the current and the subsequent factorizations. Unfortunately, such a locking and purging technique is quite difficult to accomplish for the symplectic Lanczos method. Most of the complications in the purging and deflating algorithms come from the need to preserve the structure of the decomposition, in particular, to preserve the J–Hessenberg form and the zero structure of the vector e_{2k}^T.

In [30], Stewart shows how to relax the definition of an Arnoldi decomposition such that the purging and deflating problems can be solved in a natural and efficient way. Since the method is centered about the Schur decomposition of the Hessenberg matrix, the method is called the Krylov-Schur method. A Krylov-Schur-like method for the symplectic Lanczos process has been developed in [6, 32]. Let us assume that we have constructed a symplectic Lanczos factorization of order $2(k+p) = 2m$ of the form (2.1),

$$HS^{2n,2m} = S^{2n,2m}\widetilde{H}^{2m,2m} + \zeta_{m+1}v_{m+1}e_{2m}^T. \tag{2.5}$$

Applying the SR algorithm to $\widetilde{H}^{2m,2m}$ yields a symplectic matrix $\breve{S}$, such that

$$\breve{S}^{-1}\widetilde{H}^{2m,2m}\breve{S} = \begin{bmatrix} \widetilde{A} & \widetilde{G} \\ \widetilde{Q} & -\widetilde{A}^T \end{bmatrix} = \breve{H}^{2m,2m}$$

decouples into 1×1 or 2×2 blocks on the diagonals of each of the four subblocks $\widetilde{A}, \widetilde{G}$ and $\widetilde{Q}$;

$$\left[\begin{array}{cccc|cccc} \widetilde{A}_{11} & & & & \widetilde{G}_{11} & & & \\ & \widetilde{A}_{22} & & & & \widetilde{G}_{22} & & \\ & & \ddots & & & & \ddots & \\ & & & \widetilde{A}_{mm} & & & & \widetilde{G}_{mm} \\ \hline \widetilde{Q}_{11} & & & & -\widetilde{A}_{11}^T & & & \\ & \widetilde{Q}_{22} & & & & -\widetilde{A}_{22}^T & & \\ & & \ddots & & & & \ddots & \\ & & & \widetilde{Q}_{mm} & & & & -\widetilde{A}_{mm}^T \end{array}\right]. \tag{2.6}$$

Assume furthermore, that $\breve{S}$ has been constructed such that the desired eigenvalues of $\widetilde{H}^{2m,2m}$ have been moved to the leading parts of the four submatrices, such that

$$\breve{H}^{2m,2m} = \left[\begin{array}{cc|cc} \widetilde{A}_1 & & \widetilde{G}_1 & \\ & \widetilde{A}_2 & & \widetilde{G}_2 \\ \hline \widetilde{Q}_1 & & -\widetilde{A}_1^T & \\ & \widetilde{Q}_2 & & -\widetilde{A}_2^T \end{array}\right] \tag{2.7}$$

and

$$\breve{H} = \begin{bmatrix} \widetilde{A}_1 & \widetilde{G}_1 \\ \widetilde{Q}_1 & -\widetilde{A}_1^T \end{bmatrix} \in \mathbb{R}^{2\ell \times 2\ell}$$

contains the desired eigenvalues. Then post-multiplying (2.5) by $\breve{S}$, the formula

$$HS^{2n,2m}\breve{S} = S^{2n,2m}\breve{S}\breve{S}^{-1}\widetilde{H}^{2m,2m}\breve{S} + \zeta_{m+1}v_{m+1}e_{2m}^T\breve{S}$$

yields a Hamiltonian Krylov-Schur-type decomposition

$$H\breve{S}^{2n,2m} = \breve{S}^{2n,2m}\breve{H}^{2m,2m} + \zeta_{m+1}v_{m+1}\breve{s}_{2m}^T \tag{2.8}$$

similar to the symplectic Lanczos factorization (2.5) in the sense that the matrices $\widetilde{H}^{2m,2m}$ and $\breve{H}^{2m,2m}$ are similar and hence (2.5) and (2.8) essentially give the same eigeninformation; see [6] for a proper definition. Due to the special form of $\breve{H}^{2m,2m}$, the Hamiltonian Krylov-Schur-type decomposition can be partitioned in the form

$$H[\breve{S}^1\ \breve{S}^2\ \breve{S}^3\ \breve{S}^4] = [\breve{S}^1\ \breve{S}^2\ \breve{S}^3\ \breve{S}^4]\left[\begin{array}{cc|cc} \widetilde{A}_1 & & \widetilde{G}_1 & \\ & \widetilde{A}_2 & & \widetilde{G}_2 \\ \hline \widetilde{Q}_1 & & -\widetilde{A}_1^T & \\ & \widetilde{Q}_2 & & -\widetilde{A}_2^T \end{array}\right] + \zeta_{m+1}v_{m+1}\breve{s}_{2m}^T,$$

where $\breve{S}^1 = [v_1, \ldots, v_\ell]$, $\breve{S}^2 = [v_{\ell+1}, \ldots, v_m]$, $\breve{S}^3 = [w_1, \ldots, w_\ell]$, and $\breve{S}^4 = [w_{\ell+1}, \ldots, w_m]$ if $\widetilde{A}_1, \widetilde{G}_1, \widetilde{Q}_1 \in \mathbb{R}^{\ell \times \ell}$. Then, with $\breve{s}_{2m}^T = [\breve{s}_{2m,1}, \ldots, \breve{s}_{2m,2m}]^T$, $\breve{s}_{2\ell}^T = [\breve{s}_{2m,1}, \ldots, \breve{s}_{2m,\ell}, \breve{s}_{2m,m+1}, \ldots, \breve{s}_{2m,m+\ell}]^T$, and $\breve{S}^{2n,2\ell} = [\breve{S}^1, \breve{S}^3]$, we have that

$$H\breve{S}^{2n,2\ell} = \breve{S}^{2n,2\ell}\breve{H} + \zeta_{m+1}v_{m+1}\breve{s}_{2\ell}^T \tag{2.9}$$

is also a Hamiltonian Krylov-Schur-type decomposition. In other words, a Hamiltonian Krylov-Schur-type decomposition splits at any point where its Rayleigh quotient is block diagonal. There is a similar symplectic Lanczos factorization

$$H\check{S}^{2n,2\ell} = \check{S}^{2n,2\ell}\check{H}^{2\ell,2\ell} + \check{v}_{2\ell+1}e_{2\ell}^T,$$

where $\check{H}^{2\ell,2\ell}$ is in Hamiltonian J–Hessenberg form, and the columns of $\check{S}^{2n,2\ell}$ are J–orthogonal. Thus, the purging problem can be solved by applying the SR algorithm to $\widetilde{H}^{2m,2m}$, moving the unwanted Ritz values into the Hamiltonian matrix defined by $\widetilde{A}_2, \widetilde{G}_2, \widetilde{Q}_2$, truncating the decomposition, and returning to a symplectic Lanczos factorization.

The restarting algorithm then consists of expanding this symplectic Lanczos factorization, computing the decomposition (2.6), moving the desired eigenvalues to the beginning, throwing away the rest of the decomposition, and transforming the decomposition back to a symplectic Lanczos factorization. As the iteration progresses, the Ritz estimates will converge at different rates. When a Ritz estimate is small enough, the corresponding Ritz value is said to have converged. The converged Ritz value may be wanted or unwanted. Unwanted ones can be purged from the current factorization using the above procedure. Wanted ones should be deflated to speed up convergence. For a thorough discussion of this issue; see [6, 12, 32].

While the implicitly restarted symplectic Lanczos factorization (2.4) can restart with an arbitrary filter polynomial, the Krylov-Schur-like method discussed here cannot do that. When exact shifts are used for the implicit restarting, then the symplectic Lanczos factorization achieved is equivalent to the one achieved with the above method if the same Ritz values are discarded in both methods. When it comes to exact shifts, the Krylov-Schur-like method is to be preferred because exchanging eigenvalues in the form (2.6) is a more reliable process than using implicit SR steps to deflate.

As already mentioned, in order to compute eigenvalues other than the ones largest in modulus, a suitable transformation has to be applied to H. In order to stay within the Hamiltonian structure, we can work with the Hamiltonian matrix $H_1(\tau) = H^{-1}(H - \tau I)^{-1}(H + \tau I)^{-1}$ from (1.4), or $H_2(\tau) = H(H - \tau I)^{-1}(H + \tau I)^{-1}$ from (1.5), for a real or purely imaginary shift τ. In order to obtain the eigenvalues λ of H from the eigenvalues ϖ of these shifted Hamiltonian matrices, a cubic polynomial equation

$$\lambda^3 - \tau^2\lambda - \frac{1}{\varpi} = 0$$

has to be solved in case $H_1(\tau)$ is used, while a quadratic polynomial equation

$$\lambda^2 - \frac{1}{\varpi}\lambda - \tau^2 = 0 \tag{2.10}$$

has to be solved in case $H_2(\tau)$ is used. In case a complex shift τ is used, we can work with the Hamiltonian matrix

$$H_3(\tau) = H^{-1}(H - \tau I)^{-1}(H + \tau I)^{-1}(H - \overline{\tau} I)^{-1}(H + \overline{\tau} I)^{-1}$$

from (1.6) or

$$H_4(\tau) = H(H - \tau I)^{-1}(H + \tau I)^{-1}(H - \overline{\tau} I)^{-1}(H + \overline{\tau} I)^{-1}$$

from (1.7). Similarly as before, in order to obtain the eigenvalues λ of H from the eigenvalues of the shifted matrices, polynomial equations of order five or four have to be solved: in case $H_3(\tau)$ is used,

$$\lambda^5 - (\overline{\tau}^2 + \tau^2)\lambda^3 + |\tau|^4\lambda - \frac{1}{\varpi} = 0$$

ETNA
Kent State University
etna@mcs.kent.edu

has to be solved, while in case $H_4(\tau)$ is employed,

$$\lambda^4 - (\overline{\tau}^2 + \tau^2)\lambda^2 - \frac{1}{\varpi}\lambda + |\tau|^4 = 0.$$

Hence, working with $H_2(\tau)$ or $H_4(\tau)$ makes the back transformation of the eigenvalues slightly simpler than when working with $H_1(\tau)$ or $H_3(\tau)$.

3. Solving the QEP by the symplectic Lanczos method. In order to apply the symplectic Lanczos algorithm to any of the matrices $H_j(\tau)$, $j = 1, \dots, 4$, formed from a structured linearization of the QEP (1.1), we need to be able to multiply the matrix $H_j(\tau)$ by an arbitrary vector at reasonable cost, since this operation is performed repeatedly by the algorithm. Thus, we need to be able to apply operators of the form $(H - \tau I)^{-1}$ inexpensively. The inverse of $H - \tau I$ is given by

$$(H - \tau I)^{-1} = \begin{bmatrix} I & \frac{1}{2}G + \tau M \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & M \\ -Q(\tau)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G + \tau M \\ 0 & I \end{bmatrix},$$

with $Q(\tau) = \tau^2 M + \tau G + K$. Once an LU decomposition of $Q(\tau)$ is known, $(H - \tau I)^{-1}$ can be applied to a vector in an efficient way. It is easy to see that the same LU decomposition is needed for applying $(H + \tau I)^{-1}$, $(H - \overline{\tau} I)^{-1}$, and $(H + \overline{\tau} I)^{-1}$ to a vector [22]: Since $Q(\tau)^T = Q(-\tau)$, we have

$$(H + \tau I)^{-1} = \begin{bmatrix} I & \frac{1}{2}G - \tau M \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & M \\ -Q(\tau)^{-T} & 0 \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G - \tau M \\ 0 & I \end{bmatrix},$$

and since H is a real matrix, we have $(H - \overline{\tau} I)^{-1} = \overline{(H - \tau I)}^{-1}$ and $(H + \overline{\tau} I)^{-1} = \overline{(H + \tau I)}^{-1}$ for a complex τ. Hence, once the LU decomposition of $Q(\tau)$ has been determined, all four factors can be applied to any vector in an efficient way. We would like to note here that in case the problem is so large that a sparse LU decomposition of $Q(\tau)$ is not feasible, Krylov subspace methods have to be employed in this step, as in any shift-and-invert algorithm of very large size. For brevity of exposition this is not elaborated on here any further.

In case one would like to set up $H_2(\tau)$ or $H_4(\tau)$, we have to multiply by H from (1.3) as well. On first glimpse, one might think that for this the inverse of M is needed. In the following we will see that multiplication by H essentially comes for free. For this, first consider the skew-Hamiltonian operator

$$R_2(\tau) = (H - \tau I)^{-1}(H + \tau I)^{-1}, \quad \tau \in \mathbb{R}, i\mathbb{R}, \tag{3.1}$$

which is a suitable shift-and-invert operator for the skew-Hamiltonian Arnoldi algorithm SHIRA [22]. It is observed in [22] that this operator can be expressed as

$$\begin{aligned} R_2(\tau) = {} & \begin{bmatrix} M & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\tau)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \\ & \times \begin{bmatrix} 0 & I \\ -Q(\tau)^{-T} & 0 \end{bmatrix} \begin{bmatrix} I & -\tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}. \end{aligned}$$

A detailed analysis of the cost for applying $R_2(\tau)$ to a vector is provided in [22]. It can be summarized as follows, assuming that a sparse LU decomposition of $Q(\tau)$ is available:

- 4 sparse triangular solves,
- 2 symmetric sparse matrix-vector products Mz,

ETNA
Kent State University
etna@mcs.kent.edu

- 3 skew-symmetric sparse matrix-vector products Gz,
- 5 saxpy operations.

When using $H_2(\tau)$, we have to multiply $R_2(\tau)$ from the left (or the right) by H as given in (1.3). Carefully checking the resulting expression, we obtain that

$$H_2(\tau) = \begin{bmatrix} -\frac{1}{2}G & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & I \\ -Q(\tau)^{-1} & 0 \end{bmatrix} \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \times \begin{bmatrix} 0 & I \\ -Q(\tau)^{-T} & 0 \end{bmatrix} \begin{bmatrix} I & -\tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & M \end{bmatrix}.$$

Thus, the only difference in the application of $H_2(\tau)$ as compared to $R_2(\tau)$ is that one multiplication by the mass matrix M is replaced by a multiplication with the stiffness matrix K. As in the applications considered here, the sparsity patterns of M and K are usually the same, the cost for applying $H_2(\tau)$ is the same as for $R_2(\tau)$. Hence, in this respect, none of the algorithms is to be preferred. In contrast, applying $H_1(\tau)$ requires additionally the inversion of K which makes its application less efficient (as observed in [25]). Similar observations hold for $H_4(\tau)$ compared to $H_3(\tau)$.

When eigenvalues of the largest or smallest modulus are required, the Hamiltonian Krylov-Schur-type algorithm is applied to H as in (1.3) or

$$H^{-1} = \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -K^{-1} \\ M & 0 \end{bmatrix} \begin{bmatrix} I & \frac{1}{2}G \\ 0 & I \end{bmatrix}. \tag{3.2}$$

As SHIRA requires a skew-Hamiltonian operator, H^2 or H^{-2} have to be used in these situations so that the Hamiltonian Krylov-Schur-type approach is more efficient in these cases.

Another advantage of the Hamiltonian Krylov-Schur-type method is that the lower half of the computed eigenvectors of H and H^{-1} yields the corresponding eigenvector of the quadratic matrix polynomial $Q(\lambda)$ while the eigenvectors computed by SHIRA can not be used to deduce eigenvectors of $Q(\lambda)$; see [1, 22, 25] for more details on this. In any case, no matter how approximate eigenvalues $\widetilde{\lambda}$ are computed, eigenvectors also can be obtained by applying the inverse iteration

$$Q(\widetilde{\lambda})x_k = x_{k-1}, \quad x_k = \frac{1}{\|x_k\|}x_k, \quad k = 1, 2, \ldots, \tag{3.3}$$

where x_0 of unit norm is chosen arbitrarily; see [25, 32].

In case H_2 is used to compute some eigenvalues of H, the quadratic polynomial (2.10) has to be solved. As the computation of roots of polynomials is in general very sensitive to small perturbations in the coefficients, one might want to use Rayleigh quotient iteration in order to improve the accuracy of the computed eigenvalues. As the examples in the next section show, this is not necessary for the examples considered in this paper.

4. Numerical results. In this section, we report the results of numerical experiments obtained with the Krylov-Schur-type method for Hamiltonian eigenproblems applied to the QEP (1.1). All experiments are performed in MATLAB R2006a using double precision on a Pentium M notebook with 512 MB main memory or a HP compute server with 2 Xeon 3,06 GHz processors, 533 MHz 512-KB level 2 cache, 1 MB level 3 cache, 9 GB main memory (of which only 2 GB can be used due to limitations in addressing more than 2 GB of memory in the available 32-Bit version of MATLAB).

The accuracy of computed eigenvalues and eigenvectors is compared using relative residuals

$$\frac{\|Q(\widetilde{\lambda})\widetilde{x}\|_1}{\|Q(\widetilde{\lambda})\|_1\|\widetilde{x}\|_1},$$

ETNA
Kent State University
etna@mcs.kent.edu

where $(\widetilde{\lambda}, \widetilde{x})$ is a computed Ritz pair.

4.1. Computing corner singularities. The solutions of elliptic boundary value problems, such as the Laplace and linear elasticity (Lamé) equations in domains with polyhedral corners often exhibit singularities in the neighborhood of the corners. The singularities can be quantified if this neighborhood is intersected with the unit ball centered at the corner and parameterized with spherical coordinates (r, ϕ, θ). Then the singular part of the solution can be expanded in a series with terms of the form $r^\alpha u(\phi, \theta)$, where α is the singularity exponent. It turns out that $\alpha = \lambda - \frac{1}{2}$ and u can be computed as eigenpairs of quadratic operator eigenvalue problems of the form

$$\lambda^2 m(u, v) + \lambda g(u, v) = k(u, v), \tag{4.1}$$

where $m(.,.), k(.,.)$ are Hermitian positive definite sesquilinear forms and $g(.,.)$ is a skew-Hermitian sesquilinear form. Finite-element discretization of the operator eigenvalue problem (4.1) leads to a QEP as in (1.1), where M and $-K$ are positive definite. For the numerical solution of (4.1), the software package *CoCoS* [25] can be used. Note that *CoCoS* includes Fortran implementations of SHIRA as well as a solver based on applying the implicitly restarted Hamiltonian Lanczos process [4, 34] to H^{-1} from (3.2). Here we use *CoCoS* only for the assembly of the matrices M, G, K.

In the following example we consider brittle elastic bodies where the environment of crack peaks is sufficiently well approximated by the linear material law (Lamé equation). A more detailed discussion can be found in [27]. The 3D elasticity problem is considered for the Fichera corner which results from cutting the cube $[0, 1] \times [0, 1] \times [0, 1]$ out of the cube $(-1, 1) \times (-1, 1) \times (-1, 1)$. The problem is defined by the Lamé constants μ, ν derived from Young's modulus of elasticity, and the opening angle ξ of the corner. In the following computations for the Fichera corner, the Lamé's constants are $\mu = 0.5$ and $\nu = 0.3$ and the opening angle is $90°$. We compare the Hamiltonian Krylov-Schur-type algorithm, a MATLAB implementation of SHIRA and the MATLAB function `eigs`.

For the first test an example of size $n = 5139$ is chosen. All algorithms use the same starting vector. We chose the shift $\tau = 1$ and set up an operator to apply $H_2(\tau)$ to a vector. In order to compare the three algorithms considered here, the Hamiltonian Krylov-Schur-type algorithm and MATLAB's `eigs` are applied to $H_2(1)$, while SHIRA is used with the skew-Hamiltonian operator $R_2(1)$ as in (3.1). Note that the cost for one step of each of the algorithms is about the same as `eigs` and HKS are applied to the same operator while the application of $R_2(\tau)$ to a vector comes at the same cost as the application of $H_2(\tau)$; cf. Section 3. We allowed for a search space of size 24 and asked for 12 eigenvalues in the sense that HKS was asked to compute 6 pairs of eigenvalues, while `eigs` was asked to compute 12 eigenvalues. The results are shown in Tables 4.1 and 4.2. Here we only show results for the positive eigenvalues; results for their negative counterparts are similar. In both cases the residuals are computed using eigenvectors obtained from inverse iteration as in (3.3). The results are comparable: SHIRA needs the least iterations to meet its stopping criterion, but some of the eigenvalues are less accurate than for the other two methods. On the other hand, the Hamiltonian Krylov-Schur-type method needs two iterations less than `eigs` at similar accuracy.

The SR algorithm employed for computing the Schur-type form (2.7) of the Rayleigh quotient $\widetilde{H}^{2k \times 2k}$ is using nonorthogonal transformations. Therefore we give for completeness the maximal condition number of all nonorthogonal transformations used. In contrast, SHIRA is using only orthogonal transformations. As can be seen, even a condition number of the order of 10^5 yields an accuracy of the eigenvalues computed with HKS which is at

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.1
Fichera corner ($n = 5139$): SHIRA with shift $\tau = 1$.

SHIRA 3 iterations	
Eigenvalue	Residual
0.90592878886121	$3.585 \cdot 10^{-15}$
0.90634686035000	$6.867 \cdot 10^{-15}$
1.07560224930036	$1.927 \cdot 10^{-15}$
1.60332758477231	$9.932 \cdot 10^{-14}$
1.65786577099000	$3.383 \cdot 10^{-14}$
1.66121735256292	$1.657 \cdot 10^{-14}$

TABLE 4.2
Fichera corner ($n = 5139$): `eigs` and Hamiltonian Krylov-Schur (HKS) applied to $H_2(1)$, 12 eigenvalues requested.

`eigs` 6 iterations		HKS 4 iterations max. condition number 26537	
Eigenvalue	Residual	Eigenvalue	Residual
0.90592878886137	$5.3 \cdot 10^{-17}$	0.90592878886208	$7.6 \cdot 10^{-17}$
0.90634686034990	$6.8 \cdot 10^{-17}$	0.90634686034995	$6.2 \cdot 10^{-17}$
1.07560224930040	$5.8 \cdot 10^{-17}$	1.07560224930041	$5.8 \cdot 10^{-17}$
1.60332758476368	$4.5 \cdot 10^{-17}$	1.60332758476305	$1.4 \cdot 10^{-16}$
1.65786577098522	$4.9 \cdot 10^{-17}$	1.65786577098420	$1.1 \cdot 10^{-16}$
1.66121735256605	$5.5 \cdot 10^{-17}$	1.66121735256562	$7.2 \cdot 10^{-17}$

least as good or better than the one obtained with SHIRA (when compared to the eigenvalues computed using `eigs`). Other choices of the shift τ give similar results.

Next we compare the Hamiltonian Krylov-Schur-type method and `eigs` for computing some eigenvalues of smallest magnitude; hence we apply them to H^{-1}. As before, we are interested in 12 eigenvalues and the search space has the size 24. Both algorithms use the same starting vector. The results are shown in Table 4.3, here we give the results for the negative eigenvalues to emphasize our claim that the results for the positive and negative eigenvalues are similar. Comparing the results it is easy to see that the number of iterations and the residuals are very similar. In general, the Hamiltonian Krylov-Schur-type method needs fewer iterations than `eigs` while yielding the same accuracy. Here we also compare residuals obtained for eigenvectors computed by the inverse iteration (3.3) and read off of the Ritz vectors of H^{-1} computed within the symplectic Lanczos process. The latter ones are clearly less accurate, but on the other hand they are obtained as by-products and do not require any factorization of $Q(\widetilde{\lambda})$, which is the most expensive part in all of the computations!

Next we choose an example in which the matrices M, G and K are all of size $n = 12828$. The same kind of test runs as above were performed. Very similar results were obtained, in particular the iteration counts were the same; see [5] for more details.

4.2. Gyroscopic systems. Such systems arise when modeling vibrations of spinning structures such as the simulation of tire noise, helicopter rotor blades, or spin-stabilized satellites with appended solar panels or antennas; see [10, 18, 33] and references therein. Here, M and K are positive definite mass and stiffness matrices and G is the skew-symmetric gyroscopic matrix resulting from the Coriolis force. It is known that under these conditions, all

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.3
Fichera corner ($n = 5139$): eigs and Hamiltonian Krylov-Schur (HKS) applied to H^{-1}, 12 eigenvalues requested.

eigs 8 iterations		HKS 6 iterations max. condition number 4298		
Eigenvalue	Residual	Eigenvalue	Residual inverse iteration	 Ritz vectors
-0.90592878886009	$1.7 \cdot 10^{-16}$	-0.90592878886360	$2.1 \cdot 10^{-16}$	$5.9 \cdot 10^{-14}$
-0.90634686034887	$2.1 \cdot 10^{-16}$	-0.90634686035032	$2.1 \cdot 10^{-16}$	$4.5 \cdot 10^{-14}$
-1.07560224929966	$1.6 \cdot 10^{-16}$	-1.07560224929977	$1.5 \cdot 10^{-16}$	$7.6 \cdot 10^{-15}$
-1.60332758476358	$7.6 \cdot 10^{-17}$	-1.60332758476361	$7.5 \cdot 10^{-17}$	$1.5 \cdot 10^{-14}$
-1.65786577098385	$1.5 \cdot 10^{-16}$	-1.65786577098420	$1.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-13}$
-1.66121735256510	$1.1 \cdot 10^{-16}$	-1.66121735256192	$4.8 \cdot 10^{-16}$	$3.1 \cdot 10^{-13}$

eigenvalues of the QEP are purely imaginary; see [19, 33].

4.2.1. Butterfly gyro. For our first set of experiments, we chose the model of a butterfly gyro as described in [8]. The butterfly gyro is a vibrating micro-mechanical system developed for use in inertial navigation applications.

The data matrices M, K of order $n = 17361$ (which are available from the Oberwolfach Benchmark Collection[1]) are obtained from a finite-element analysis performed with ANSYS using quadratic tetrahedral elements (SOLID187). As the gyroscopic matrix G is missing, we choose a randomly generated skew-symmetric matrix with the same sparsity pattern as K and with entries of considerably smaller magnitude as the influence of the Coriolis force is usually much smaller than that of the stiffness of the system.

In our first test run, we apply `eigs` and the Hamiltonian Krylov-Schur-type method to H^{-1} to obtain the smallest frequency modes of the butterfly gyro—these are usually the modes of interest in vibration analysis of gyroscopic systems. Both the Hamiltonian Krylov-Schur-type method and `eigs` need 3 iterations to compute 12 eigenvalues. The maximal condition number encountered in the SR algorithm is about $1.5 \cdot 10^3$. The accuracy of the computed eigenvalues as measured by their relative residuals is similar: for both methods, the residuals are smaller than the machine epsilon `eps`. Figure 4.1 shows the computed eigenvalues with positive imaginary parts. Obviously, the Hamiltonian Krylov-Schur-type method locates all eigenvalues on the imaginary axis as expected from theory while for the eigenvalues computed by `eigs`, it is rather difficult to decide whether or not these are purely imaginary even though the real parts are relatively small.

In the second test run for the butterfly gyro, we compute interior eigenvalues. For this purpose, we choose a shift $\tau = 10^6 i$ and apply `eigs` and the Hamiltonian Krylov-Schur-type method to $H_2(\tau)$. Here, `eigs` needs one iteration less than the Krylov-Schur-type method (2 as compared to 3). The maximum condition number encountered in the SR algorithm is about $3.15 \cdot 10^6$. The accuracy of the computed eigenvalues does not seem to be affected by this fairly large condition number; again, all residuals are smaller than 10^{-16} as for `eigs`. But moreover, the new approach yields physically meaningful results as all computed eigenvalues are located on the imaginary axis. In contrast, the eigenvalues computed by `eigs` have nonzero real parts as can be seen from Figure 4.2 except for the eigenvalue closest to the

[1] See `http://www.imtek.de/simulation/benchmark`.

ETNA
Kent State University
etna@mcs.kent.edu

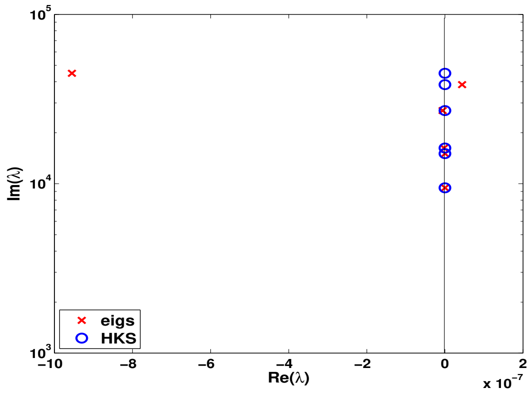


FIG. 4.1. *Butterfly gyro ($n = 17361$):* `eigs` *and Hamiltonian Krylov-Schur (HKS) applied to H^{-1}, 12 eigenvalues requested.*

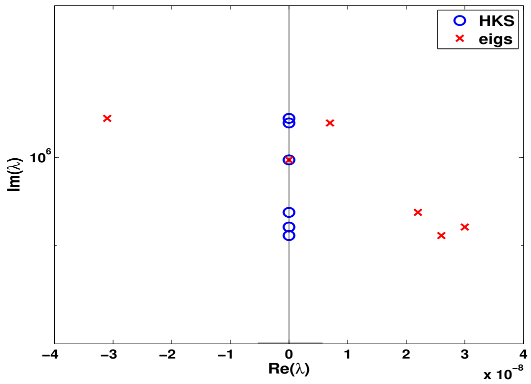


FIG. 4.2. *Butterfly gyro ($n = 17361$):* `eigs` *and Hamiltonian Krylov-Schur (HKS) applied to $H_2(10^6 i)$, 12 eigenvalues requested.*

target $\tau = 10^6 i$.

4.2.2. Rolling Tires. The quadratic eigenvalue problem considered in this example is related to modeling the noise resulting from rolling tires. One step in the simulation of tire noise as described in [24] consists of determining transient vibrations (which are then superimposed on the nonlinear tire deflections). These vibrations are computed from the eigenpairs of a gyroscopic eigenvalue problem. The data considered here are obtained from applying the algebraic multilevel substructuring (AMLS) approach discussed in [10] to a finite-element model of a deformable wheel rolling on a rigid plane surface. The original finite-element model from [23] is of size $n = 124,992$ and the sparse LU factorization of $Q(\tau)$ requires about 6 GByte. As this is beyond the reach of 32-bit MATLAB (whose address space can only deal with arrays up to 2 GByte), the problem is first reduced to a gyroscopic eigenproblem of size $n = 2,697$ using AMLS in [10]. Then it is suggested in [10] to compute the required eigenpairs by applying `eigs` to a standard linearization of this reduced second-order eigenproblem.

Here, we compare this approach with our approach applied to a Hamiltonian lineariza-

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.4
Rolling Tires ($n = 2697$)*:* `eigs` *(top) and Hamiltonian Krylov-Schur (HKS) (bottom) applied to* H^{-1}*,* 12 *eigenvalues requested.*

`eigs`
8 iterations

Eigenvalue	Residual
$4 \cdot 10^{-9} + 1.73705142673011 \cdot 10^{3}i$	$2.1 \cdot 10^{-14}$
$-3 \cdot 10^{-9} + 1.66795405953410 \cdot 10^{3}i$	$8.2 \cdot 10^{-15}$
$2 \cdot 10^{-10} + 1.66552788163842 \cdot 10^{3}i$	$2.2 \cdot 10^{-15}$
$4 \cdot 10^{-11} + 1.58209209803838 \cdot 10^{3}i$	$1.1 \cdot 10^{-16}$
$-1 \cdot 10^{-11} + 1.13657108577706 \cdot 10^{3}i$	$7.6 \cdot 10^{-17}$
$1 \cdot 10^{-11} + 0.80560062107003 \cdot 10^{3}i$	$1.1 \cdot 10^{-16}$

HKS
6 iterations
max. condition number 331

Eigenvalue	Residual
$1.73705142670608 \cdot 10^{3}i$	$4.7 \cdot 10^{-17}$
$1.66795405954599 \cdot 10^{3}i$	$1.7 \cdot 10^{-15}$
$1.66552788164134 \cdot 10^{3}i$	$1.4 \cdot 10^{-16}$
$1.58209209803815 \cdot 10^{3}i$	$5.1 \cdot 10^{-17}$
$1.13657108577691 \cdot 10^{3}i$	$7.4 \cdot 10^{-18}$
$0.80560062106969 \cdot 10^{3}i$	$6.0 \cdot 10^{-18}$

tion of the reduced gyroscopic eigenproblem. That is, we apply `eigs` and the Hamiltonian Krylov-Schur-type method to H^{-1} to obtain the smallest eigenvalues. The Hamiltonian Krylov-Schur-type method needs 6 iterations to compute 6 eigenvalue pairs, while `eigs` needs 8 iterations. The maximal condition number encountered in the SR algorithm is 331. The accuracy of the computed eigenvalues, as measured by their relative residuals, differs about an order of magnitude if both methods are compared. The residuals for the SR approach are smaller than those for `eigs`; see Table 4.4. Obviously, the Hamiltonian Krylov-Schur-type method locates all eigenvalues on the imaginary axis as expected from theory, while for the eigenvalues computed by `eigs`, it is basically impossible to decide whether or not these are purely imaginary even so the real parts are relatively small.

Finally, we asked for 180 pairs of eigenvalues as required in [10] for solving the application problem. The residuals for both methods were comparable. Figure 4.3 shows the computed eigenvalues, as before, one can clearly see that the eigenvalues computed via `eigs` do not necessarily lie on the imaginary axis.

5. Conclusions. We have discussed the application of the Hamiltonian Krylov-Schur-type method based on the symplectic Lanczos process to quadratic eigenvalue problems with Hamiltonian symmetry. The method is an alternative to unstructured methods, such as the implicitly restarted Arnoldi method as implemented in the MATLAB function `eigs` or its structure-preserving variant SHIRA, which can be applied to skew-Hamiltonian operators. Compared to `eigs` our method has the advantage of respecting the symmetry properties inherent in the problem and thus yields meaningful physical results. This is demonstrated for a stable gyroscopic system, where theoretically all eigenvalues are located on the imaginary axis. On the other hand, the Hamiltonian Krylov-Schur-type approach is about as efficient as SHIRA for computing eigenvalues, but sometimes more accurate. Eigenvectors are directly

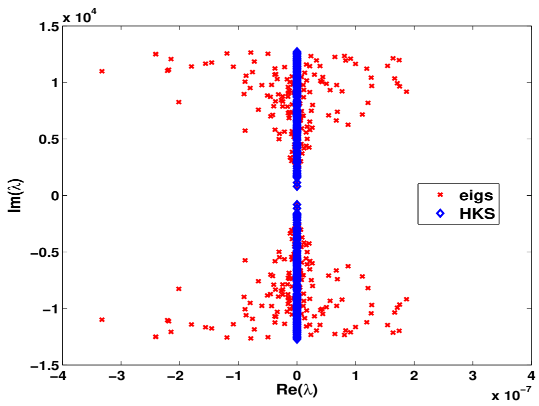


FIG. 4.3. *Rolling Tires ($n = 2697$): eigs and Hamiltonian Krylov-Schur (HKS) applied to H^{-1}, 180 eigenpairs requested.*

available in case no shifts are used while SHIRA only provides eigenvalues.

Acknowledgments. We thank Conny Pester for providing the data matrices for the Fichera corner examples. We thank Heinrich Voss and, especially, Kolja Elssel for providing the data for the rolling tires example.

REFERENCES

[1] T. APEL, V. MEHRMANN, AND D. WATKINS, *Structured eigenvalue methods for the computation of corner singularities in 3d anisotropic elastic structures*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4459–4473.

[2] Z. BAI AND Y. SU, *SOAR: A second-oder Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal., 26 (2005), pp. 640–659.

[3] 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), Fakultät für Mathematik, Germany, 1995.

[4] ———, *An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem*, Linear Algebra Appl., 263 (1997), pp. 75–111.

[5] P. BENNER, H. FASSBENDER, AND M. STOLL, *Solving large-scale quadratic eigenvalue problem with Hamiltonian eigenstructure using a structure-preserving Krylov subspace method*, Tech. Report 07/03, Oxford University Computing Labarotory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, England, 2007.

[6] ———, *A Hamiltonian Krylov-Schur-type method based on the symplectic Lanczos process*, unpublished manuscript.

[7] P. BENNER, D. KRESSNER, AND V. MEHRMANN, *Structure preservation: A challenge in computational control*, Future Generation Comp. Syst., 19 (2003), pp. 1243–1252.

[8] D. BILLGER, *The butterfly gyro*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. Sorensen, eds., Lecture Notes in Computational Science and Engineering, vol. 45, Springer-Verlag, Berlin/Heidelberg, Germany, 2005, pp. 349–352.

[9] 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.

[10] K. ELSSEL AND H. VOSS, *Reducing huge gyroscopic eigenproblems by automated multi-level substructuring*, Arch. Appl. Mech., 76 (2006), pp. 171–179.

[11] H. FASSBENDER AND D. KRESSNER, *Structured eigenvalue problems*, GAMM Mitt., 29 (2006), pp. 297–318.

[12] H. FASSBENDER, *A detailed derivation of the parameterized SR algorithm and the symplectic Lanczos method for Hamiltonian matrices*, internal report, TU Braunschweig, Institut *Computational Mathematics*, 2006. Available at http://www.icm.tu-bs.de/~hfassben/papers/ham_evp.pdf

[13] W. FERNG, W.-W. LIN, AND C.-S. WANG, *The shift-inverted J-Lanczos algorithm for the numerical solutions of large sparse algebraic Riccati equations*, Comput. Math. Appl., 33 (1997), pp. 23–40.

[14] J. FRANCIS, *The QR transformation, Part I and Part II*, Comput. J., 4 (1961), pp. 265–271 and 332–345.

[15] R. FREUND, *Lanczos-type algorithms for structured non-Hermitian eigenvalue problems*, in Proceedings of the Cornelius Lanczos International Centenary Conference, J. Brown, M. Chu, D. Ellison, and R. Plemmons, eds., SIAM, Philadelphia, 1994, pp. 243–245.

[16] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, 3rd ed., Baltimore, 1996.

[17] D. KRESSNER, *Numerical Methods for General and Structured Eigenvalue Problems*, Lecture Notes in Computational Science and Engineering, vol. 46, Springer, Berlin, 2005.

[18] P. LANCASTER, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966.

[19] ———, *Strongly stable gyroscopic systems*, Electr. J. Linear Algebra, 5 (1999), pp. 53–66.

[20] D. MACKEY, N. MACKEY, C. MEHL, AND V. MEHRMANN, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal., 28 (2006), pp. 971–1004.

[21] M. MAESS AND L. GAUL, *Simulation of vibro-acoustics in flexible piping systems*, GAMM Mitt., 28 (2005), pp. 37–55.

[22] V. MEHRMANN AND D. WATKINS, *Structure-preserving methods for computing pairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Statist. Comput., 22 (2001), pp. 1905–1925.

[23] U. NACKENHORST, *The ALE-formulation of bodies in rolling contact. Theoretical foundations and finite element approach*, Comput. Meth. Appl. Mech. Engrg., 193 (2004), pp. 4299–4322.

[24] U. NACKENHORST AND O. VON ESTORFF, *Numerical analysis of tire noise radiation — a state of the art review*, in Inter-noise 2001, The 2001 International Congress and Exhibition on Noise Control Engineering, The Hague, The Netherlands, 2001.

[25] C. PESTER, *CoCoS - computation of corner singularities*, Preprintreihe des Chemnitzer SFB 393, Preprint 05-03, TU Chemnitz, Germany, 2005. Available at http://www.tu-chemnitz.de/sfb393/preprints.html

[26] ———, *Hamiltonian eigenvalue symmetry for quadratic operator eigenvalue problems*, J. Integral Equations Appl., 17 (2005), pp. 71–89.

[27] J. ROSAM, *Berechnung der Rissgeometrie bei spröden elastischen Körpern*, Diplomarbeit, TU Chemnitz, 2004.

[28] G. SLEIJPEN, A. BOOTEN, D. FOKKEMA, AND H. VAN DER VORST, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.

[29] D. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[30] G. STEWART, *A Krylov-Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.

[31] ———, *Matrix Algorithms, Volume II: Eigensystems*, SIAM , Philadelphia, USA, 2001.

[32] M. STOLL, *Locking und Purging für den Hamiltonischen Lanczos-Prozess*, Diplomarbeit, TU Chemnitz, Fakultät für Mathematik, Germany, 2005.

[33] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[34] D. WATKINS, *On Hamiltonian and symplectic Lanczos processes*, Linear Algebra Appl., 385 (2004), pp. 23–45.

[35] D. WATKINS AND L. ELSNER, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.

[36] N. WONG, V. BALAKRISHNAN, AND C.-K. KOH, *Passivity-preserving model reduction via a computationally efficient project-and-balance scheme*, in Proceedings of Design Automation Conference, San Diego, 2004, pp. 369–374.