Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Linear Algebra and its Applications 385 (2004) 23–45
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

On Hamiltonian and symplectic Lanczos processes

David S. Watkins

Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA

Received 21 April 2002; accepted 20 November 2002

Submitted by N.J. Higham

Dedicated to my thesis advisor, Peter Lancaster, on the occasion of his 75th birthday

Abstract

Large sparse Hamiltonian eigenvalue problems arise in a variety of contexts. These problems can be attacked directly, or they can first be transformed to problems having some related structure, such as symplectic or skew-Hamiltonian. In the interest of efficiency, stability, and accuracy, such problems should be solved by methods that preserve the structure, whether it be Hamiltonian, skew-Hamiltonian, or symplectic. The present work outlines Krylov subspace methods for computing partial eigensystems of skew-Hamiltonian, Hamiltonian, and symplectic matrices and records some of the relationships between them. The ordinary unsymmetric Lanczos process is a structure-preserving method for skew-Hamiltonian matrices. The Hamiltonian and symplectic Lanczos processes developed here are condensed versions of the processes that have been published previously. The condensed Hamiltonian Lanczos process applied to H is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian H^2 but costs half as much to execute. The condensed symplectic Lanczos process applied to S is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian matrix $S + S^{-1}$ but also costs half as much to execute. Implicit restarts of the Hamiltonian and symplectic Lanczos processes can be effected by the SR algorithm. Because of the known relationship between the SR and HR algorithms, the much simpler HR algorithm can be used to restart the condensed symplectic and Hamiltonian Lanczos processes. Each of these restart procedures is equivalent to restarting the unsymmetric Lanczos process using the HR algorithm. The HR algorithm is most effectively implemented as an implicit (bulge-chasing) HZ algorithm on a symmetric, tridiagonal–diagonal pencil $T - \lambda D$.

Keywords: Eigenvalue; Lanczos process; Hamiltonian; Symplectic; Implicit restarts

E-mail address: watkins@math.wsu.edu (D.S. Watkins).

0024-3795/$ - see front matter
doi:10.1016/j.laa.2002.11.001

1. Introduction

Hamiltonian eigenvalue problems arise in a variety of contexts, including linear-quadratic optimal control problems [18], gyroscopic systems [14,15], and the analysis of corner singularities in anisotropic structures [1,19].

In problems of the latter type, only a few of the smallest eigenvalues of a real Hamiltonian matrix H are needed. Since these are also the largest eigenvalues of H^{-1}, a Krylov subspace method can be applied to H^{-1} to find them, provided H^{-1} is accessible. Since H^{-1} inherits the Hamiltonian structure of H, we prefer to use a procedure that respects the Hamiltonian structure, in the interest of efficiency, stability, and accuracy. Eigenvalues of real Hamiltonian matrices occur in pairs $\{\lambda, -\lambda\}$ or quadruples $\{\lambda, -\lambda, \overline{\lambda}, -\overline{\lambda}\}$. A structure-preserving method will extract entire pairs and quadruples intact.

In situations where we have some prior information, we might prefer to shift before we invert. Specifically, if we know that the eigenvalues of interest lie near τ, we might prefer to work with $(H - \tau I)^{-1}$. Unfortunately, the shift destroys the structure, so we are lead to think of ways of effecting shifts without destroying the structure. One simple remedy is to work with the matrix

$$(H - \tau I)^{-1}(H + \tau I)^{-1},$$

which is not Hamiltonian but skew-Hamiltonian. It makes perfect sense that the shifts τ and $-\tau$ should be used together, in light of the symmetry of the spectrum of H. If τ is neither real nor purely imaginary, we prefer to work with

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

in order to stay within the real number system.

If we wish to use a shift *and* keep the Hamiltonian structure, we can work with the Hamiltonian matrix

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

or

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

for example.

Another possibility is to work with the Cayley transform

$$(H - \tau I)^{-1}(H + \tau I),$$

which is symplectic. To attack this one, we need a Krylov subspace method that preserves symplectic structure. We note in passing that symplectic eigenvalue problems also arise in linear-quadratic optimal control problems for discrete-time systems [18].

In this paper we will outline Krylov subspace methods, variants of the unsymmetric Lanczos process, that preserve skew-Hamiltonian, Hamiltonian, and symplectic

structures. For each of these we describe a structure-preserving implicit restart procedure in the spirit of Sorensen's implicitly-restarted Arnoldi process [13,17,20]. In the interest of accuracy and efficiency, short Lanczos runs with frequent implicit restarts are preferable to long Lanczos runs.

Benner and Fassbender have proposed families of implicitly-restarted Lanczos methods for both Hamiltonian [4] and symplectic [6] problems. The methods outlined in this paper are special cases of these, obtained by making choices of parameters that condense the processes as much as possible. The main focus of this paper is not on the introduction of new methods but on illuminating the relationships between the various methods. We will show that the ordinary unsymmetric Lanczos process is a structure-preserving method for skew-Hamiltonian matrices. Furthermore, the condensed Hamiltonian and symplectic Lanczos processes developed here are closely related to the unsymmetric Lanczos process: The condensed Hamiltonian Lanczos process applied to H is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian H^2 (but costs half as much to execute). The condensed symplectic Lanczos process applied to a symplectic matrix S is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian matrix $S + S^{-1}$ (and also costs half as much to execute).

A structure-preserving implicit restart procedure for the unsymmetric Lanczos process was published by Grimme et al. [13]. In that procedure, the filtering for the restart is done by the HR algorithm instead of the QR algorithm, in the interest of preserving structure. For the Hamiltonian and symplectic Lanczos processes, structure-preserving implicit restarts can be effected using the SR algorithm, as shown in [4,6]. For the condensed versions presented here, we show how to effect the restarts using the HR algorithm instead of SR. This is a major conceptual simplification that makes implementation much easier, especially in the symplectic case. This simplification is possible because of the relationships between the SR and HR algorithms that were demonstrated in [7]. Both the Hamiltonian and the symplectic restarts are equivalent to restarting the unsymmetric Lanczos process using the HR algorithm.

The HR algorithm is most effectively implemented as an implicit (bulge-chasing) HZ algorithm [21] on a symmetric, tridiagonal–diagonal pencil $T - \lambda D$.

MATLAB test versions of all of the algorithms described here have been implemented and found to work well. Fortran production versions for the corner singularity problems [1,19] are now being written and tested.

2. Terminology, notation, and some basic facts

The matrices appearing in this paper are real of dimension $2n \times 2n$. We define

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

Then $B \in \mathbb{R}^{2n\times 2n}$ is *skew-Hamiltonian* if $(JB)^{\mathrm{T}} = -JB$, *Hamiltonian* if $(JB)^{\mathrm{T}} = JB$, and *symplectic* if $B^{\mathrm{T}}JB = J$. Symplectic matrices are non-singular. If S is symplectic and A is skew-Hamiltonian, Hamiltonian, or symplectic, then $S^{-1}AS$ is skew-Hamiltonian, Hamiltonian, or symplectic, respectively. Thus all of the structures considered in this paper are preserved by symplectic similarity transformations. Any algorithm that effects (or partially effects) a symplectic similarity transformation is thus a structure-preserving method.

One easily checks the following basic facts: if H is Hamiltonian and S is symplectic, then H^2 and $S + S^{-1}$ are skew-Hamiltonian.

A subspace $\mathscr{S}$ of $\mathbb{R}^{2n}$ is called *isotropic* if $x^{\mathrm{T}}Jy = 0$ for all $x, y \in \mathscr{S}$. Every symplectic matrix S has isotropic subspaces associated with it. Writing $S = [S_1 \quad S_2]$ and writing the condition $S^{\mathrm{T}}JS = J$ in partitioned form, we find that

$$\begin{bmatrix} S_1^{\mathrm{T}} \\ S_2^{\mathrm{T}} \end{bmatrix} J \begin{bmatrix} S_1 & S_2 \end{bmatrix} = \begin{bmatrix} S_1^{\mathrm{T}}JS_1 & S_1^{\mathrm{T}}JS_2 \\ S_2^{\mathrm{T}}JS_1 & S_2^{\mathrm{T}}JS_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix},$$

so, in particular, $S_1^{\mathrm{T}}JS_1 = 0$ and $S_2^{\mathrm{T}}JS_2 = 0$. Thus the space spanned by the first n columns of a symplectic matrix is isotropic, as is the space spanned by the last n columns.

The algorithms that will be discussed in this paper are Krylov subspace methods. The mth *Krylov subspace* associated with $u \in \mathbb{R}^{2n}$ and $B \in \mathbb{R}^{2n\times 2n}$ is defined by

$$\mathscr{K}_m(B, u) = \operatorname{span}\{u, Bu, B^2u, \ldots, B^{m-1}u\}.$$

3. The unsymmetric Lanczos process

We derive the well-known unsymmetric Lanczos process [16] using slightly non-standard notation that is suitable for our purposes. Let $B \in \mathbb{R}^{2n\times 2n}$. For now we are not assuming that B is skew-Hamiltonian. Later on we will see what happens in the skew-Hamiltonian case. The Lanczos process is based on the fact that B can be transformed to tridiagonal form by a similarity transformation: $BU_{2n} = U_{2n}\hat{T}_{2n}$, where U_{2n} is non-singular and $\hat{T}_{2n}$ is tridiagonal. Taking transposes, we also have $B^{\mathrm{T}}W_{2n} = W_{2n}\hat{T}_{2n}^{\mathrm{T}}$, where $W_{2n} = U_{2n}^{-\mathrm{T}}$. The transforming matrices U_{2n} and W_{2n} are far from unique; their first columns, u_1 and w_1 can be chosen almost arbitrarily, subject to the normalization $w_1^{\mathrm{T}}u_1 = 1$, which is forced by the condition $W_{2n}^{\mathrm{T}}U_{2n} = I$. We can also (almost always) construct $\hat{T}_{2n}$ in such a way that $|\hat{T}_{2n}|$ is symmetric; that is, $|\hat{t}_{k+1,k}| = |\hat{t}_{k,k+1}|$ for all k. Assuming $\hat{T}_{2n}$ has this form, we can write $\hat{T}_{2n} = T_{2n}D_{2n}$, where T_{2n} is symmetric and tridiagonal with non-negative off-diagonal entries, and D_{2n} is diagonal with entries ± 1 on the main diagonal.

The Lanczos process builds a similarity transformation $BU_{2n} = U_{2n}(T_{2n}D_{2n})$ one column at a time. We will use the following notation: $u_1, u_2, \ldots, u_{2n}$ and $w_1, w_2, \ldots, w_{2n}$ will denote the columns of U_{2n} and W_{2n}, respectively,

$$T_{2n} = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{2n-1} \\ & & & b_{2n-1} & a_{2n} \end{bmatrix}, \quad D_{2n} = \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & d_3 & & \\ & & & \ddots & \\ & & & & d_{2n} \end{bmatrix}.$$

Writing down the kth columns of the equations $BU_{2n} = U_{2n}(T_{2n}D_{2n})$ and $B^{\mathrm{T}}W_{2n} = W_{2n}(D_{2n}T_{2n})$, we obtain the recursions

$$u_{k+1}b_kd_k = Bu_k - u_ka_kd_k - u_{k-1}b_{k-1}d_k \tag{1}$$

and

$$w_{k+1}d_{k+1}b_k = B^{\mathrm{T}}w_k - w_kd_ka_k - w_{k-1}d_{k-1}b_{k-1}, \tag{2}$$

which form the basis of the Lanczos process. These are valid even when $k = 1$, provided we set $u_0 = w_0 = 0$. They give formulas for computing u_{k+1} and w_{k+1}, once we have the previous columns, provided we have a way of computing the coefficients a_j, b_j, d_j. As it turns out, the coefficients are determined (essentially) uniquely by the biorthogonality conditions $w_i^{\mathrm{T}}u_j = \delta_{ij}$, which come from $W_{2n}^{\mathrm{T}}U_{2n} = I$. For example, suppose (inductively) we have u_{k-1}, u_k, w_{k-1}, w_k, d_k, d_{k-1}, and b_{k-1}. Then, multiplying (1) by w_k^{T} on the left, and applying the condition $w_k^{\mathrm{T}}u_{k+1} = 0$, we find that $a_k = d_k^{-1}w_k^{\mathrm{T}}Bu_k$. Using this formula to compute a_k, we now have enough information to compute the right-hand sides of (1) and (2). Let $\hat{u}_{k+1}$ and $\hat{w}_{k+1}$ denote the vectors that result from these two computations. If either of these is zero, we have a benign breakdown; we have discovered an invariant subspace for either B or B^{T}. Assume this is not the case. Then the condition $w_{k+1}^{\mathrm{T}}u_{k+1} = 1$ implies $\hat{w}_{k+1}^{\mathrm{T}}\hat{u}_{k+1} = b_k^2d_kd_{k+1}$, which can be used to determine b_k and d_{k+1}. Let $\pi = \hat{w}_{k+1}^{\mathrm{T}}\hat{u}_{k+1}$. If $\pi = 0$, we have a serious breakdown resulting from an unlucky choice of starting vectors u_1 and w_1. If this does not happen, we have either $\pi > 0$, which implies $b_k = +\sqrt{\pi}$ and $d_{k+1} = d_k$, or $\pi < 0$, which implies $b_k = +\sqrt{-\pi}$ and $d_{k+1} = -d_k$. We then have $u_{k+1} = \hat{u}_{k+1}/(b_kd_k)$ and $w_{k+1} = \hat{w}_{k+1}/(b_kd_{k+1})$. This is all we need to move on to the next step. These considerations give the following algorithm. In practice we never run the algorithm to completion. Let us suppose we are going to take m steps.

Algorithm 1 (*Unsymmetric Lanczos Process*)

Start with initial vectors u_1 and w_1 satisfying $w_1^{\mathrm{T}}u_1 = 1$. The integer m denotes the desired number of steps.

$d_1 \leftarrow \pm 1,\ d_0 \leftarrow 0,\ b_0 \leftarrow 0$
$u_0 \leftarrow 0,\ w_0 \leftarrow 0$
for $k = 1, 2, \ldots, m$
  $y \leftarrow Bu_k$
  $a_k \leftarrow d_k w_k^{\mathrm{T}} y$
  $u_{k+1} \leftarrow y - u_k a_k d_k - u_{k-1} b_{k-1} d_k$
  $w_{k+1} \leftarrow B^{\mathrm{T}} w_k - w_k d_k a_k - w_{k-1} d_{k-1} b_{k-1}$
  $\pi \leftarrow w_{k+1}^{\mathrm{T}} u_{k+1}$
  if $\pi = 0$ then
    [stop (breakdown)
  if $\pi > 0$ then
    $[b_k \leftarrow \sqrt{\pi},\ d_{k+1} \leftarrow d_k$
  if $\pi < 0$ then
    $[b_k \leftarrow \sqrt{-\pi},\ d_{k+1} \leftarrow -d_k$
  $u_{k+1} \leftarrow u_{k+1}/(b_k d_k)$
  $w_{k+1} \leftarrow w_{k+1}/(d_{k+1} b_k)$

It does not matter whether d_1 is taken to be +1 or −1. Reversing d_1 has no essential effect on the algorithm: All computed quantities are the same as before, except that minus signs are introduced in a systematic way.

One easily checks that Algorithm 1 generates Krylov subspaces. We have

$$\operatorname{span}\{u_1, \ldots, u_k\} = \mathscr{K}_k(B, u_1)$$

and

$$\operatorname{span}\{w_1, \ldots, w_k\} = \mathscr{K}_k(B^{\mathrm{T}}, w_1)$$

for $k = 1, 2, 3, \ldots$

The following proposition can be proved by induction on k.

Proposition 2. *The vectors u_k and w_k produced by Algorithm* 1 *are biorthonormal, i.e. $w_i^{\mathrm{T}} u_j = \delta_{ij}$ for all i and j.*

Remark. This proposition holds in exact arithmetic. In a practical computational setting, roundoff errors will destroy the biorthogonality after a few steps. If we want to get biorthogonality in practice, we must apply a re-biorthogonalization process. This requires biorthogonalization against all previous vectors, which means that we must store all of the vectors that have been generated to a given point. As a consequence, we cannot plan to do long Lanczos runs. Instead we can do short runs with repeated implicit restarts [13,20].

After m steps we have only generated part of the similarity transformations $BU_{2n} = U_{2n}(T_{2n} D_{2n})$ and $B^{\mathrm{T}} W_{2n} = W_{2n}(D_{2n} T_{2n})$. Let U_m and W_m denote the lead-

ing m columns of U_{2n} and W_{2n}, respectively, and let T_m and D_m denote the leading $m \times m$ submatrices of T_{2n} and D_{2n}, respectively. Then the objects that have been generated by the first m steps of the unsymmetric Lanczos process are related by

$$BU_m = U_m T_m D_m + u_{m+1} b_m d_m e_m^{\mathrm{T}}, \tag{3}$$

$$B^{\mathrm{T}} W_m = W_m D_m T_m + w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}, \tag{4}$$

where e_m is the mth standard basis vector in $\mathbb{R}^{2n}$, and

$$W_m^{\mathrm{T}} U_m = I_m. \tag{5}$$

4. Unsymmetric Lanczos process for skew-Hamiltonian matrices

Now suppose B is skew-Hamiltonian ($(JB)^{\mathrm{T}} = -JB$). Then it is impossible to run the Lanczos process to completion, as the following lemma [19] shows.

Lemma 3. *Let $B \in \mathbb{R}^{2n\times 2n}$ be skew-Hamiltonian, and let $u \in \mathbb{R}^{2n}$ be an arbitrary non-zero vector. Then the Krylov subspace $\mathcal{K}_m(B, u)$ is isotropic for all m. That is, for all $x, y \in \mathcal{K}_m(B, u)$, $y^{\mathrm{T}} J x = 0$.*

Proof. Since B is skew-Hamiltonian, one easily shows that all of its powers B^i, $i = 0, 1, 2, 3, \ldots$, are also skew-Hamiltonian. This means that JB^i is skew-symmetric. To establish isotropy of $\mathcal{K}_m(B, u)$, it suffices to prove that $(B^i u)^{\mathrm{T}} J B^k u = 0$ for all $i \geqslant 0$ and $k \geqslant 0$. Since $B^{\mathrm{T}} J = JB$, we have $(B^i u)^{\mathrm{T}} J B^k u = u^{\mathrm{T}} J B^{i+k} u$, which equals zero because JB^{i+k} is skew-symmetric. □

Since an isotropic subspace cannot have dimension greater than n, we conclude that the Krylov space $\mathcal{K}_{n+1}(B, u_1)$ has dimension at most n. This implies that $\mathcal{K}_{n+1}(B, u_1) = \mathcal{K}_n(B, u_1)$ and $\mathcal{K}_n(B, u_1)$ is invariant under B. Since B^{T} is also skew-Hamiltonian, we also have that $\mathcal{K}_n(B^{\mathrm{T}}, w_1)$ is invariant under B^{T}. Thus the Lanczos process must terminate at the nth step, if not sooner.

Suppose the algorithm does not terminate before the nth step. Then we are able to generate U_n and W_n, whose columns span Krylov subspaces that are invariant under B and B^{T}, respectively. Eqs. (3) and (4) become $BU_n = U_n T_n D_n$ and $B^{\mathrm{T}} W_n = W_n D_n T_n$, respectively, since the remainder terms are zero. Since we can go no further than this, let us make the following abbreviations: $U = U_n \in \mathbb{R}^{2n\times n}$, $W = W_n \in \mathbb{R}^{2n\times n}$, $T = T_n \in \mathbb{R}^{n\times n}$, and $D = D_n \in \mathbb{R}^{n\times n}$. Now we have

$$BU = U(TD), \tag{6}$$

$$B^{\mathrm{T}} W = W(DT), \tag{7}$$

$$W^{\mathrm{T}} U = I, \tag{8}$$

and by isotropy,

$$U^{\mathrm{T}}JU = 0 \quad \text{and} \quad W^{\mathrm{T}}JW = 0. \tag{9}$$

These equations represent only partial similarities. We can build full similarity transformations by exploiting the structure. Since B is skew-Hamiltonian, we have $B^{\mathrm{T}}J = JB$ and $BJ = JB^{\mathrm{T}}$. Multiplying each of (6) and (7) by $J^{\mathrm{T}} = -J$ on the left and defining

$$X = J^{\mathrm{T}}U = -JU \quad \text{and} \quad V = J^{\mathrm{T}}W = -JW,$$

we obtain the new equations

$$B^{\mathrm{T}}X = X(TD), \tag{10}$$

$$BV = V(DT), \tag{11}$$

$$V^{\mathrm{T}}X = I, \tag{12}$$

$$X^{\mathrm{T}}JX = 0 \quad \text{and} \quad V^{\mathrm{T}}JV = 0. \tag{13}$$

Combining (6) and (11) we obtain the full similarity transformation

$$B\begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix}\begin{bmatrix} TD & \\ & DT \end{bmatrix}. \tag{14}$$

Likewise, combining (7) and (10) we obtain the equivalent equation

$$B^{\mathrm{T}}\begin{bmatrix} W & X \end{bmatrix} = \begin{bmatrix} W & X \end{bmatrix}\begin{bmatrix} DT & \\ & TD \end{bmatrix}. \tag{15}$$

These are indeed similarity transformations, since $[U \quad V]$ and $[W \quad X]$ are both non-singular; in fact $[W \quad X] = [U \quad V]^{-\mathrm{T}}$. Moreover, these transforming matrices are symplectic ($S^{\mathrm{T}}JS = J$), as one easily infers from (8), (9), (12), and (13). Thus (14) and the equivalent (15) are symplectic similarity transformations to tridiagonal, block-diagonal form. The condensed form is also skew-Hamiltonian, since the symplectic similarity transformation preserves this form. In other words, this algorithm preserves the skew-Hamiltonian structure.

Eqs. (14) and (15) hold only if we are able to reach the nth step with no breakdowns. Even if we do not get that far, we still obtain partial results. If we multiply (3) and (4) each by $-J$ on the left, we obtain

$$B^{\mathrm{T}}X_m = X_m T_m D_m + x_{m+1} b_m d_m e_m^{\mathrm{T}}, \tag{16}$$

$$BV_m = V_m D_m T_m + v_{m+1} d_{m+1} b_m e_m^{\mathrm{T}} \tag{17}$$

and

$$V_m^{\mathrm{T}}X_m = I_m. \tag{18}$$

These equations show that applying the Lanczos process to the skew-Hamiltonian matrix B with starting vectors u_1 and w_1 is equivalent to applying the Lanczos process to B^{T} with starting vectors $x_1 = -Ju_1$ and $v_1 = -Jw_1$.

5. The condensed Hamiltonian Lanczos process

The Hamiltonian Lanczos process was discussed in [4,12]. Every Hamiltonian matrix is symplectically similar to a matrix in the condensed form

$$\begin{bmatrix} E & T \\ D & -E \end{bmatrix},$$

where T is tridiagonal and symmetric, and E and D are diagonal [11]. The transformation is far from unique; the first column of the symplectic transforming matrix can be chosen almost arbitrarily. Furthermore, there is a great deal of freedom in the choice of the entries of T, E, and D. The diagonal matrix E can be chosen arbitrarily; for example, we can take $E = 0$ if we so desire. D can be scaled so that its main-diagonal entries are all ± 1. The off-diagonal entries of T can be taken to be non-negative. If we make these normalizations, we have a similarity transformation

$$H\begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix}\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}, \tag{19}$$

where $[U \quad V]$ is symplectic, D is diagonal with main-diagonal entries ± 1, and T is symmetric and tridiagonal with non-negative off-diagonal entries.

The condensed Hamiltonian Lanczos process is a procedure that builds the similarity transformation (19) one column at a time. Perhaps we should say two columns at a time, since each step will produce a column of U and a column of V. Writing down the kth and $(n+k)$th columns of (19), we have

$$Hu_k = v_k d_k \tag{20}$$

and

$$Hv_k = u_{k-1}b_{k-1} + u_k a_k + u_{k+1}b_k, \tag{21}$$

using notation consistent with that introduced in the previous sections. From these we obtain

$$u_{k+1}b_k = Hv_k - u_k a_k - u_{k-1}b_{k-1} \tag{22}$$

and

$$v_{k+1}d_{k+1} = Hu_{k+1}, \tag{23}$$

which can be used to obtain u_{k+1} and v_{k+1} from previously constructed vectors, provided we can determine the coefficients a_k, b_k, and d_{k+1}. As we shall see, these coefficients are determined uniquely from the conditions $u_i^{\mathrm{T}} J v_j = \delta_{ij} = -v_j^{\mathrm{T}} J u_i$, which are necessary if the matrix $[U \quad V]$ is to be symplectic. First of all, multiplying (22) on the left by $v_k^{\mathrm{T}} J$, we find that $a_k = -v_k^{\mathrm{T}} J H v_k$. Since we will know b_{k-1}

from the previous step, we now have enough to compute the right-hand side of (22). Let

$$\hat{u}_{k+1} = Hv_k - u_k a_k - u_{k-1} b_{k-1}$$

and

$$\hat{v}_{k+1} = H\hat{u}_{k+1}.$$

Then $\hat{u}_{k+1} = u_{k+1} b_k$, and $\hat{v}_{k+1} = Hu_{k+1} b_k = v_{k+1} d_{k+1} b_k$. We can now determine b_k and d_{k+1} uniquely from the condition $u_{k+1}^{\mathrm{T}} J v_{k+1} = 1$. Let $\pi = \hat{u}_{k+1}^{\mathrm{T}} J \hat{v}_{k+1}$. Then we must have $\pi = d_{k+1} b_k^2$. If $\pi = 0$, we have a breakdown. If $\pi > 0$ we must have $d_{k+1} = 1$ and $b_k = \sqrt{\pi}$, and if $\pi < 0$ we must have $d_{k+1} = -1$ and $b_k = \sqrt{-\pi}$. We then have $u_{k+1} = \hat{u}_{k+1}/b_k$ and $v_{k+1} = \hat{v}_{k+1}/(d_{k+1} b_k)$. Now we have all we need for the next step.

Before we can write down the algorithm, we need to say a few words about picking starting vectors. By (23) with $k = 0$, u_1 and v_1 must be related by $v_1 d_1 = Hu_1$, and they must also satisfy $u_1^{\mathrm{T}} J v_1 = 1$. We can obtain vectors satisfying these conditions by picking a more or less arbitrary starting vector $\hat{u}_1$ and using the normalization process described in the previous paragraph. Thus we define $\hat{v}_1 = H\hat{u}_1$, $\pi = \hat{u}_1^{\mathrm{T}} J \hat{v}_1$, and so on.

Now we are ready to write down the algorithm.

Algorithm 4 (*Condensed Hamiltonian Lanczos Process*)

Start with initial vector u_1. The integer m denotes the desired number of steps.

$b_0 \leftarrow 0,\ u_0 \leftarrow 0$
for $k = 0, 1, 2, \ldots, m$
  if $k > 0$ then
    $y \leftarrow Hv_k$
    $a_k \leftarrow -v_k^{\mathrm{T}} J y$
    $u_{k+1} \leftarrow y - u_k a_k - u_{k-1} b_{k-1}$
  $v_{k+1} \leftarrow Hu_{k+1}$
  $\pi \leftarrow u_{k+1}^{\mathrm{T}} J v_{k+1}$
  if $\pi = 0$ then
    stop (breakdown)
  if $\pi > 0$ then
    $s \leftarrow \sqrt{\pi},\ d_{k+1} \leftarrow 1$
  if $\pi < 0$ then
    $s \leftarrow \sqrt{-\pi},\ d_{k+1} \leftarrow -1$
  $u_{k+1} \leftarrow u_{k+1}/s$
  $v_{k+1} \leftarrow v_{k+1}/(d_{k+1} s)$
  if $k > 0$ then
    $b_k \leftarrow s$

The following result can be verified by induction on k.

Proposition 5. *The vectors u_k and v_k produced by Algorithm* 4 *are columns of a symplectic matrix, i.e.* $u_i^{\mathrm{T}} J v_j = \delta_{ij}$, $u_i^{\mathrm{T}} J u_j = 0$, *and* $v_i^{\mathrm{T}} J v_j = 0$ *for all* i *and* j.

Remark. Like the previous proposition, this result holds only in exact arithmetic. In practice, roundoff errors will destroy the J-orthogonality after a few steps. If we really want to get a symplectic basis, we must apply a J-reorthogonalization process, which requires saving all of the vectors. Therefore we must again think in terms of doing short runs with repeated implicit restarts, rather than doing long Lanczos runs.

After m steps we have only generated part of the similarity transformation

$$H \begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}.$$

Using notation consistent with what we introduced earlier, the relationships between the objects that have been generated by the first m steps of the condensed Hamiltonian Lanczos process are given by

$$H \begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix} \begin{bmatrix} 0 & T_m \\ D_m & 0 \end{bmatrix} + u_{m+1} b_m e_{2m}^{\mathrm{T}} \tag{24}$$

and

$$\begin{bmatrix} U_m^{\mathrm{T}} \\ V_m^{\mathrm{T}} \end{bmatrix} J \begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix}. \tag{25}$$

Let $W_m = JV_m$ and $X_m = -JU_m$ (which is consistent with our earlier definitions). Then $J[U_m \quad V_m]J_{2m}^{\mathrm{T}} = [W_m \quad X_m]$. If we multiply (24) on the left by J and on the right by J_{2m}^{T} and apply the identity $H^{\mathrm{T}}J = -JH$ (valid because H is Hamiltonian), we obtain

$$H^{\mathrm{T}} \begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} W_m & X_m \end{bmatrix} \begin{bmatrix} 0 & D_m \\ T_m & 0 \end{bmatrix} + x_{m+1} b_m e_m^{\mathrm{T}}, \tag{26}$$

along with

$$\begin{bmatrix} W_m^{\mathrm{T}} \\ X_m^{\mathrm{T}} \end{bmatrix} J \begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix} \tag{27}$$

and

$$\begin{bmatrix} W_m^{\mathrm{T}} \\ X_m^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} U_m & V_m \end{bmatrix} = I_{2m}. \tag{28}$$

5.1. Connection with the unsymmetric Lanczos process

The equations that we have just derived can be used to established a connection between the condensed Hamiltonian Lanczos process applied to H and the unsymmetric Lanczos process applied to the skew-Hamiltonian matrix H^2. If we multiply (24) by H on the left and simplify, we obtain

$$H^2\begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix}\begin{bmatrix} T_m D_m & 0 \\ 0 & D_m T_m \end{bmatrix} + u_{m+1} b_m d_m e_m^{\mathrm{T}} + v_{m+1} d_{m+1} b_m e_{2m}^{\mathrm{T}}. \tag{29}$$

Applying the analogous procedure to (26), we obtain

$$(H^2)^{\mathrm{T}}\begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} W_m & X_m \end{bmatrix}\begin{bmatrix} D_m T_m & 0 \\ 0 & T_m D_m \end{bmatrix} + w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}} + x_{m+1} b_m d_m e_{2m}^{\mathrm{T}}. \tag{30}$$

If we now draw from (29), (30), and (28) the parts that pertain to U_m and W_m, we find that

$$H^2 U_m = U_m (T_m D_m) + u_{m+1} b_m d_m e_m^{\mathrm{T}},$$

$$(H^2)^{\mathrm{T}} W_m = W_m (D_m T_m) + w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}$$

and

$$W_m^{\mathrm{T}} U_m = I_m.$$

Comparing these with (3)–(5), we see that our Hamiltonian Lanczos process implicitly effects an unsymmetric Lanczos process with the skew-Hamiltonian matrix H^2.

We can equally well draw from (29), (30), and (28) the parts that pertain to X_m and V_m to obtain

$$(H^2)^{\mathrm{T}} X_m = X_m (T_m D_m) + x_{m+1} b_m d_m e_m^{\mathrm{T}},$$

$$H^2 V_m = V_m (D_m T_m) + v_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}$$

and

$$V_m^{\mathrm{T}} X_m = I_m,$$

which we can compare with (16)–(18).

We have established the following result.

Theorem 6. *The condensed Hamiltonian Lanczos process applied to the Hamiltonian matrix H, with starting vectors u_1 and $v_1 = Hu_1 d_1$ is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian matrix H^2, with starting vectors u_1 and $w_1 = Jv_1$.*

To compare the cost of the two procedures, let us assume that the dominant cost is that of applying the matrices. This is a reasonable assumption, even if reorthogonalization is used, assuming the size of the matrix is much larger than the number of steps that are taken. We will also assume that we apply H^2 to a vector by applying H

twice. Since, $(H^2)^{\mathrm{T}} = JH^2J^{\mathrm{T}}$, we can also apply $(H^2)^{\mathrm{T}}$ by applying H twice. Thus the cost of a step of the unsymmetric Lanczos process applied to H^2 is four evaluations of H. In contrast, the Hamiltonian Lanczos process costs only two evaluations of H per step, so it is only about half as expensive.

6. The condensed symplectic Lanczos process

The symplectic Lanczos process was discussed in [2,5,6]. Every symplectic matrix $S \in \mathbb{R}^{2n\times 2n}$ is symplectically similar to a matrix in the *symplectic butterfly form*

$$\begin{bmatrix} D_1 & T_1 \\ D_2 & T_2 \end{bmatrix},$$

where T_1 and T_2 are tridiagonal, and D_1 and D_2 are diagonal [2,3]. The transformation is far from unique; the first column of the symplectic transforming matrix can be chosen almost arbitrarily. Beyond that, there is a much additional freedom in the choice of the entries. If T_2 is irreducible, the matrix is called an *unreduced symplectic butterfly matrix*. The diagonal matrix D_2 is then forced to be non-singular and can be scaled so that its main-diagonal entries are all ± 1. The diagonal matrix D_1 can take arbitrary values, so let us take $D_1 = 0$. Once we have done this, the symplectic structure of the matrix implies that $T_1 = -D_2$ and $D_2^{-1}T_2$ is symmetric. Let $D = D_2$ and $T = D^{-1}T_2$. Then our condensed, unreduced butterfly matrix has the form

$$\begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix},$$

where T is symmetric and tridiagonal, and D is diagonal with main diagonal entries ± 1. Finally, we can arrange for the off-diagonal entries of T to be positive. This canonical form was presented in [7]. It is a simple matter to check that the inverse is

$$\begin{bmatrix} TD & D \\ -D & 0 \end{bmatrix}.$$

The condensed symplectic Lanczos process builds the similarity transformation

$$S\begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix}\begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix}, \tag{31}$$

a column at a time. Writing down the kth and $(n+k)$th columns of (31), we have

$$Su_k = v_k d_k \tag{32}$$

and

$$Sv_k = -u_k d_k + v_{k-1}d_{k-1}b_{k-1} + v_k d_k a_k + v_{k+1}d_{k+1}b_k. \tag{33}$$

From these we obtain

$$v_{k+1}d_{k+1}b_k = Sv_k - v_k d_k a_k - v_{k-1}d_{k-1}b_{k-1} + u_k d_k \tag{34}$$

and

$$u_{k+1}d_{k+1} = S^{-1}v_{k+1}, \tag{35}$$

which can be used to obtain u_{k+1} and v_{k+1} from previously constructed data. The appearance of S^{-1} poses no problem; since S is symplectic, $S^{-1} = J^{\mathrm{T}}S^{\mathrm{T}}J$.

If we multiply (34) by S^{-1} and d_k and make a few simple substitutions, we find that we can rewrite (34) and (35) as

$$u_{k+1}b_k d_k = S^{-1}u_k - u_k a_k d_k - u_{k-1}b_{k-1}d_k + v_k d_k \tag{36}$$

and

$$v_{k+1}d_{k+1} = Su_{k+1}, \tag{37}$$

which shows that the u_k and v_k vectors play a symmetric role in this algorithm. We can equally well work with the pair (34) and (35) or the pair (36) and (37). We will work with the former.

As in the Hamiltonian case, the coefficients that are needed in (34) and (35) can be determined from the conditions $u_i^{\mathrm{T}}Jv_j = \delta_{ij} = -v_j^{\mathrm{T}}Ju_i$. Multiplying (34) on the left by $u_k^{\mathrm{T}}J$, we find that $a_k = u_k^{\mathrm{T}}JSv_k d_k$. Since we will know b_{k-1}, d_k, and d_{k-1} from previous steps, we now have enough to compute the right-hand side of (34). Let

$$\hat{v}_{k+1} = Sv_k - v_k d_k a_k - v_{k-1}d_{k-1}b_{k-1} + u_k d_k$$

and

$$\hat{u}_{k+1} = S^{-1}\hat{v}_{k+1}.$$

Then $\hat{v}_{k+1} = v_{k+1}d_{k+1}b_k$, and $\hat{u}_{k+1} = S^{-1}v_{k+1}d_{k+1}b_k = u_{k+1}b_k$. We can now determine b_k and d_{k+1} exactly as in the Hamiltonian algorithm. Let $\pi = \hat{u}_{k+1}^{\mathrm{T}}J\hat{v}_{k+1}$. Then we must have $\pi = d_{k+1}b_k^2$. If $\pi = 0$, we have a breakdown. If $\pi > 0$ we must have $d_{k+1} = 1$ and $b_k = \sqrt{\pi}$, and if $\pi < 0$ we must have $d_{k+1} = -1$ and $b_k = \sqrt{-\pi}$. We then have $u_{k+1} = \hat{u}_{k+1}/b_k$ and $v_{k+1} = \hat{v}_{k+1}/(d_{k+1}b_k)$. Now we have all we need for the next step.

The procedure for choosing starting vectors is similar to the procedure for the Hamiltonian algorithm. If we pick a more or less arbitrary starting vector $\hat{v}_1$ and let $\hat{u}_1 = S^{-1}\hat{v}_1$, then we can use the procedure outlined at the end of the previous paragraph to produce vectors u_1 and v_1 and a scalar $d_1 = \pm 1$ such that $u_1^{\mathrm{T}}Jv_1 = 1$ and $Su_1 = v_1 d_1$.

Putting these ideas together, we have the following algorithm.

Algorithm 7 (*Condensed Symplectic Lanczos Process*)
Start with initial vector v_1. The integer m denotes the desired number of steps.

$b_0 \leftarrow 0,\ d_0 \leftarrow 0,\ v_0 \leftarrow 0$
for $k = 0, 1, 2, \ldots, m$
  if $k > 0$ then
    $y \leftarrow Sv_k$
    $a_k \leftarrow (u_k^{\mathrm{T}} J y) d_k$
    $v_{k+1} \leftarrow y - v_k d_k a_k - v_{k-1} d_{k-1} b_{k-1} + u_k d_k$
  $u_{k+1} \leftarrow S^{-1} v_{k+1}$
  $\pi \leftarrow u_{k+1}^{\mathrm{T}} J v_{k+1}$
  if $\pi = 0$ then
    stop (breakdown)
  if $\pi > 0$ then
    $s \leftarrow \sqrt{\pi},\ d_{k+1} \leftarrow 1$
  if $\pi < 0$ then
    $s \leftarrow \sqrt{-\pi},\ d_{k+1} \leftarrow -1$
  $u_{k+1} \leftarrow u_{k+1}/s$
  $v_{k+1} \leftarrow v_{k+1}/(d_{k+1} s)$
  if $k > 0$ then
    $b_k \leftarrow s$

The bottom half of Algorithm 7 is identical to the bottom half of Algorithm 4.
As in the Hamiltonian case, we can obtain the following result by induction on k.

Proposition 8. *The vectors u_k and v_k produced by Algorithm 7 are columns of a symplectic matrix, i.e. $u_i^{\mathrm{T}} J v_j = \delta_{ij}$, $u_i^{\mathrm{T}} J u_j = 0$, and $v_i^{\mathrm{T}} J v_j = 0$ for all i and j.*

Remark. Again we have a result that holds only in exact arithmetic. If we really want to get a symplectic basis, we must apply a J-reorthogonalization process.

After m steps we have only generated part of the similarity transformation

$$S \begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix}.$$

The relationships between the objects that have been generated by the first m steps of the condensed symplectic Lanczos process are given by

$$S \begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix} \begin{bmatrix} 0 & -D_m \\ D_m & D_m T_m \end{bmatrix} + v_{m+1} d_{m+1} b_m e_{2m}^{\mathrm{T}} \tag{38}$$

and

$$\begin{bmatrix} U_m^{\mathrm{T}} \\ V_m^{\mathrm{T}} \end{bmatrix} J \begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix}. \tag{39}$$

The small matrix $\begin{bmatrix} 0 & -D_m \\ D_m & D_m T_m \end{bmatrix}$ is symplectic, and its inverse is

$$\begin{bmatrix} T_m D_m & D_m \\ -D_m & 0 \end{bmatrix}.$$

Applying this inverse to (38) on the right, and multiplying on the left by S^{-1}, we obtain

$$S^{-1}\begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix}\begin{bmatrix} T_m D_m & D_m \\ -D_m & 0 \end{bmatrix} + u_{m+1} b_m d_m e_m^{\mathrm{T}}. \tag{40}$$

Equations for the transposed matrices are also easily obtained. Let $W_m = JV_m$ and $X_m = -JU_m$ as before. Then $J[U_m \quad V_m]J_{2m}^{\mathrm{T}} = [W_m \quad X_m]$. If we multiply (40) on the left by J and on the right by J_{2m}^{T} and apply the identity $JS^{-1} = S^{\mathrm{T}}J$ (valid because S is symplectic), we obtain

$$S^{\mathrm{T}}\begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} W_m & X_m \end{bmatrix}\begin{bmatrix} 0 & D_m \\ -D_m & T_m D_m \end{bmatrix} + x_{m+1} b_m d_m e_{2m}^{\mathrm{T}}. \tag{41}$$

Applying the same procedure to (38), we obtain

$$S^{-\mathrm{T}}\begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} W_m & X_m \end{bmatrix}\begin{bmatrix} D_m T_m & -D_m \\ D_m & 0 \end{bmatrix} + w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}. \tag{42}$$

6.1. Connection with the unsymmetric Lanczos process

Examining (38) and (40), we see that some simplification occurs if we add them together. We get

$$\begin{aligned}(S + S^{-1})\begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix}&\begin{bmatrix} T_m D_m & 0 \\ 0 & D_m T_m \end{bmatrix} \\ &+ u_{m+1} b_m d_m e_m^{\mathrm{T}} + v_{m+1} d_{m+1} b_m e_{2m}^{\mathrm{T}}.\end{aligned} \tag{43}$$

Similarly, the sum of (41) and (42) is

$$\begin{aligned}(S + S^{-1})^{\mathrm{T}}\begin{bmatrix} W_m & X_m \end{bmatrix} = \begin{bmatrix} W_m & X_m \end{bmatrix}&\begin{bmatrix} D_m T_m & 0 \\ 0 & T_m D_m \end{bmatrix} \\ &+ w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}} + x_{m+1} b_m d_m e_{2m}^{\mathrm{T}}.\end{aligned} \tag{44}$$

If we now draw from (43), (44), and (28) the parts that pertain to U_m and W_m, we find that

$$(S + S^{-1})U_m = U_m(T_m D_m) + u_{m+1} b_m d_m e_m^{\mathrm{T}},$$

$$(S + S^{-1})^{\mathrm{T}} W_m = W_m(D_m T_m) + w_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}$$

and

$$W_m^{\mathrm{T}} U_m = I_m.$$

Comparing these with (3)–(5), we see that condensed symplectic Lanczos process implicitly effects an unsymmetric Lanczos process with the skew-Hamiltonian matrix $(S + S^{-1})$.

We can equally well draw from (43), (44), and (28) the parts that pertain to X_m and V_m to obtain

$$(S + S^{-1})^{\mathrm{T}} X_m = X_m (T_m D_m) + x_{m+1} b_m d_m e_m^{\mathrm{T}},$$

$$(S + S^{-1}) V_m = V_m (D_m T_m) + v_{m+1} d_{m+1} b_m e_m^{\mathrm{T}}$$

and

$$V_m^{\mathrm{T}} X_m = I_m,$$

which we can compare with (16)–(18).

We have established the following result.

Theorem 9. *The condensed symplectic Lanczos process applied to the Symplectic matrix S, with starting vectors u_1 and $v_1 = Su_1 d_1$ is equivalent to the unsymmetric Lanczos process applied to the skew-Hamiltonian matrix $S + S^{-1}$, with starting vectors u_1 and $w_1 = Jv_1$.*

To compare the cost of the two procedures, let us assume that the dominant cost is that of applying the matrices. We will also assume that we apply $S + S^{-1}$ to a vector by applying S and S^{-1} separately, and adding the result. Thus the cost of a step of the unsymmetric Lanczos process applied to $S + S^{-1}$ is two applications of S and two applications of S^{-1}. In contrast, the symplectic Lanczos process costs only one application of S and one application of S^{-1} per step, so it is only about half as expensive. Since $S^{-1} = JS^{\mathrm{T}}J^{\mathrm{T}}$, we normally expect the costs of applying S and S^{-1} to be about the same.

7. Implicit restarts

As we have explained above, we intend to combine implicit restarts with each of the Lanczos processes that we have discussed, because we cannot afford long Lanczos runs. In each case we need to use a restart procedure that respects the underlying structure.

7.1. Restarting the unsymmetric Lanczos process

Suppose we have

$$BU_m = U_m T_m D_m + u_{m+1} b_m d_m e_m^{\mathrm{T}}$$

and

$$B^{\mathrm{T}}W_m = W_m D_m T_m + w_{m+1}d_{m+1}b_m e_m^{\mathrm{T}},$$

and we wish to restart with

$$\hat{u}_1 = p(B)u_1 \quad \text{and} \quad \hat{w}_1 = p(B^{\mathrm{T}})w_1,$$

where p is some filtering polynomial of degree j, with $j < m$. For simplicity we will drop the subscripts and write

$$BU = U(TD) + u_+\alpha e_m^{\mathrm{T}},$$

$$B^{\mathrm{T}}W = W(DT) + w_+\beta e_m^{\mathrm{T}}.$$

As was pointed out in [13], we can do implicit restarts with the HR algorithm [9,10], which preserves the special tridiagonal form of the small matrices TD and DT. We will describe briefly in our notation how this is done. An HR iteration driven by p has the form

$$\hat{T}\hat{D} = H^{-1}TDH,$$

where

$$p(TD) = HR,$$

H satisfies $H^{\mathrm{T}}DH = \hat{D}$, and R is upper triangular. (The transformation on T alone is $\hat{T} = H^{-1}TH^{-\mathrm{T}}$.)

At the same time an HR step is effected on $DT = (TD)^{\mathrm{T}}$: Letting $\tilde{H} = H^{-\mathrm{T}} = DH\hat{D}$ and $\tilde{R} = \hat{D}RD$, we have

$$\hat{D}\hat{T} = \tilde{H}^{-1}DT\tilde{H},$$

where

$$p(DT) = \tilde{H}\tilde{R},$$

$\tilde{H}^{\mathrm{T}}D\tilde{H} = \hat{D}$ and $\tilde{R}$ is upper triangular.

Letting $\hat{U} = UH$ and $\hat{W} = WH^{-\mathrm{T}}$, we have

$$B\hat{U} = \hat{U}(\hat{T}\hat{D}) + u_+\alpha e_m^{\mathrm{T}}H$$

and

$$B^{\mathrm{T}}\hat{W} = \hat{W}(\hat{D}\hat{T}) + w_+\beta e_m^{\mathrm{T}}H^{-\mathrm{T}}.$$

Since the remainder terms are zero except in the last $j+1$ columns (j is the degree of p), we can restart with the first $m-j$ columns. Biorthogonality is preserved:

$$\hat{W}^{\mathrm{T}}\hat{U} = H^{-1}W^{\mathrm{T}}UH = H^{-1}IH = I.$$

Since $j < m$, we have $p(B)Ue_1 = Up(TD)e_1$ and $p(B^{\mathrm{T}})We_1 = Wp(DT)e_1$, so

$$\hat{u}_1 = UHe_1 = Up(TD)e_1\gamma = \gamma p(B)u_1$$

and

$$\hat{w}_1 = WH^{-\mathrm{T}}e_1 = Wp(DT)e_1\delta = \delta p(B^{\mathrm{T}})w_1.$$

This shows that we have achieved the desired filtering.

7.2. Restarting the condensed Hamiltonian Lanczos process

Since we have appropriated the symbol H for use in the HR decomposition, let us now use $\mathscr{H}$ to denote a Hamiltonian matrix. After m steps of the condensed Hamiltonian Lanczos process we have

$$\mathscr{H}\begin{bmatrix} U_m & V_m \end{bmatrix} = \begin{bmatrix} U_m & V_m \end{bmatrix}\begin{bmatrix} 0 & T_m \\ D_m & 0 \end{bmatrix} + u_{m+1}b_m e_{2m}^{\mathrm{T}},$$

which we can abbreviate as

$$\mathscr{H}\begin{bmatrix} U & V \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix}\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix} + u_+\alpha e_{2m}^{\mathrm{T}}.$$

Now suppose we wish to restart with $\hat{u}_1 = q(\mathscr{H})u_1$ and $\hat{v}_1 = q(\mathscr{H})v_1$, where q is a filtering polynomial of degree $2j$ of the form

$$q(z) = \prod_{i=1}^{j}(z-\mu_i)(z+\mu_i)$$

with $j < m$. Clearly $q(\mathscr{H}) = p(\mathscr{H}^2)$, where

$$p(w) = \prod_{i=1}^{j} w - \mu_i^2$$

has degree j.

In [4] it was shown that the Hamiltonian Lanczos process can be restarted implicitly using the SR algorithm [11], which preserves the J-tridiagonal form. Since we are using the even more condensed form $\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}$, we can exploit the connection between the SR and HR algorithms [7]. An SR step on $\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}$, driven by q has the special form

$$\begin{bmatrix} 0 & \hat{T} \\ \hat{D} & 0 \end{bmatrix} = \begin{bmatrix} H^{-1} & 0 \\ 0 & H^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}\begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix},$$

where

$$q\left(\begin{bmatrix} 0 & T \\ D & 0 \end{bmatrix}\right) = \begin{bmatrix} p(TD) & 0 \\ 0 & p(DT) \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix}\begin{bmatrix} R & 0 \\ 0 & \hat{D}RD \end{bmatrix}.$$

This is equivalent to an HR iteration on TD (and DT) driven by p. Thus we can effect the filtering process using the HR algorithm, which is simpler than the SR algorithm.

Letting

$$\begin{bmatrix}\hat{U} & \hat{V}\end{bmatrix} = \begin{bmatrix}U & V\end{bmatrix}\begin{bmatrix}H & 0\\ 0 & H^{-\mathrm{T}}\end{bmatrix}, \tag{45}$$

we have

$$\mathscr{H}\begin{bmatrix}\hat{U} & \hat{V}\end{bmatrix} = \begin{bmatrix}\hat{U} & \hat{V}\end{bmatrix}\begin{bmatrix}0 & \hat{T}\\ \hat{D} & 0\end{bmatrix} + u_+\alpha e_{2m}^{\mathrm{T}}\begin{bmatrix}H & 0\\ 0 & H^{-\mathrm{T}}\end{bmatrix}.$$

Since the remainder term is zero except in the last $j+1$ columns, we can restart with the first $m-j$ columns of $\hat{U}$ and of $\hat{V}$. The new basis vectors retain the symplectic property, because the transforming matrix in (45) is symplectic.

Since $j < m$, we have $q(\mathscr{H})Ue_1 = p(\mathscr{H}^2)Ue_1 = Up(TD)e_1$ and $q(\mathscr{H})Ve_1 = p(\mathscr{H}^2)Ve_1 = Vp(DT)e_1$, so

$$\hat{u}_1 = UHe_1 = Up(TD)e_1\gamma = \gamma p(\mathscr{H}^2)u_1 = \gamma q(\mathscr{H})u_1$$

and

$$\hat{v}_1 = VH^{-\mathrm{T}}e_1 = Vp(DT)e_1\delta = \delta p(\mathscr{H}^2)v_1 = \delta q(\mathscr{H})v_1.$$

This shows that we have achieved the desired filtering.

7.3. Restarting the condensed symplectic Lanczos process

After m steps of the condensed symplectic Lanczos process we have

$$S\begin{bmatrix}U_m & V_m\end{bmatrix} = \begin{bmatrix}U_m & V_m\end{bmatrix}\begin{bmatrix}0 & -D_m\\ D_m & D_mT_m\end{bmatrix} + v_{m+1}d_{m+1}b_m e_{2m}^{\mathrm{T}},$$

which we rewrite as

$$S\begin{bmatrix}U & V\end{bmatrix} = \begin{bmatrix}U & V\end{bmatrix}\begin{bmatrix}0 & -D\\ D & DT\end{bmatrix} + v_+\alpha e_{2m}^{\mathrm{T}}.$$

Now suppose we wish to restart with $\hat{u}_1 = q(S)u_1$ and $\hat{v}_1 = q(S)v_1$, where q is a filtering Laurent polynomial of degree $2j$ of the form

$$q(z) = \prod_{i=1}^{j} z^{-1}(z-\mu_i)(z-\mu_i^{-1})$$

with $j < m$. Clearly $q(S) = p(S+S^{-1})$, where

$$p(w) = \prod_{i=1}^{j} w - (\mu_i + \mu_i^{-1})$$

has degree j.

It was shown in [5] that the symplectic Lanczos process can be restarted implicitly using the symplectic SR algorithm. Since we are using a condensed version of the symplectic butterfly form, we can again exploit a connection between the SR and

HR algorithms [7] to simplify the process. An SR iteration on $\begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix}$ driven by q has the special form

$$\begin{bmatrix} 0 & -\hat{D} \\ \hat{D} & \hat{D}\hat{T} \end{bmatrix} = \begin{bmatrix} H^{-1} & 0 \\ 0 & H^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix},$$

where

$$q\left(\begin{bmatrix} 0 & -D \\ D & DT \end{bmatrix}\right) = \begin{bmatrix} p(TD) & 0 \\ 0 & p(DT) \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & \hat{D}RD \end{bmatrix}.$$

This is equivalent to an HR iteration on TD (and DT) driven by p. Thus we can effect the filtering by the HR algorithm, which is much simpler than the symplectic SR algorithm.

Letting

$$\begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix},$$

we have

$$S\begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} = \begin{bmatrix} \hat{U} & \hat{V} \end{bmatrix} \begin{bmatrix} 0 & -\hat{D} \\ \hat{D} & \hat{D}\hat{T} \end{bmatrix} + v_+ \alpha e_{2m}^{\mathrm{T}} \begin{bmatrix} H & 0 \\ 0 & H^{-\mathrm{T}} \end{bmatrix}.$$

Since the remainder term is zero except in the last $j+1$ columns, we can restart using the first $m-j$ columns of $\hat{U}$ and of $\hat{V}$. Since $j < m$, we have $q(S)Ue_1 = p(S+S^{-1})Ue_1 = Up(TD)e_1$ and $q(S)Ve_1 = p(S+S^{-1})Ve_1 = Vp(DT)e_1$, so

$$\hat{u}_1 = UHe_1 = Up(TD)e_1\gamma = \gamma q(S)u_1$$

and

$$\hat{v}_1 = VH^{-\mathrm{T}}e_1 = Vp(DT)e_1\delta = \delta q(S)v_1.$$

This shows that we have achieved the desired filtering.

8. Implementation of the HR algorithm

The algorithm that we have implemented is actually an HZ algorithm that operates on the matrix pencil $T - \lambda D$. An HZ iteration driven by p has the form [21]

$$\hat{T} = H^{-1}T\tilde{H} \qquad \hat{D} = H^{-1}D\tilde{H},$$

where

$$p(TD^{-1}) = HR \quad \text{and} \quad p(D^{-1}T) = \tilde{H}\tilde{R}$$

are both HR decompositions. As we have seen above, if $p(TD) = HR$, with $H^{\mathrm{T}}DH = \hat{D}$, then $p(DT) = H^{-\mathrm{T}}(\hat{D}RD)$, with $H^{-1}DH^{-\mathrm{T}} = \hat{D}$. Thus an HR decomposition of $p(TD)$ automatically yields an HR decomposition of $p(DT)$, and we can take $\tilde{H} = H^{-\mathrm{T}}$ and $\tilde{R} = \hat{D}RD$. Then our HZ step has the form

$$\hat{T} = H^{-1}TH^{-\mathrm{T}} \qquad \hat{D} = H^{-1}DH^{-\mathrm{T}}.$$

This transformation is implemented as a sequence of bulge chases, each driven by a quadratic real polynomial $r(z) = (z-\sigma)(z-\tau)$ that is a divisor of p. At each step of each bulge chase, the current T is tridiagonal, except for a bulge of degree 2, and the current D is diagonal with entries ± 1. Each bulge chase is a sequence of transformations $T \leftarrow \check{H}^{-1}T\check{H}^{-\mathrm{T}}$, $D \leftarrow \check{H}^{-1}D\check{H}^{-\mathrm{T}}$ acting on two adjacent rows, say j and $j+1$. The transformation $T \leftarrow \check{H}^{-1}T\check{H}^{-\mathrm{T}}$ eliminates one bulge entry and creates another lower down, and the transformation $D \leftarrow \check{H}^{-1}D\check{H}^{-\mathrm{T}}$ keeps D diagonal. To this end, H should be taken to be a rotator if d_j and d_{j+1} have the same sign; H should be a hyperbolic transformation if d_j and d_{j+1} have opposite sign.

9. Practical experience

The focus of this paper has been on relationships between the algorithms, not on their performance. Nevertheless we shall close with a few words on this subject. We have implemented the methods in MATLAB, using J-reorthogonalization to enforce the symplectic structure of the basis vectors. All of our experiments, run on examples of corner singularity problems [1,19], indicate that the methods work well.

Since each of the algorithms is subject to breakdown, it is natural to wonder whether breakdowns or near breakdowns pose a problem in practice. Our experience is that near breakdowns almost never occur. Consequently, we can deal with these rare occurrences simply by restarting (implicitly). This approach to breakdowns is much simpler than the look-ahead schemes that have been touted in recent years.

Because of the possibility of near breakdowns, all of the algorithms we have described are potentially unstable. Fortunately there is a simple a posteriori backward stability test: compute the residual. This is possible because all of our codes compute eigenvectors as well as eigenvalues. A tiny residual guarantees stability. This test should never be skipped. Since both right and left eigenvectors are obtained, it is also possible to compute condition numbers for the eigenvalues.

Acknowledgments

The author thanks Peter Benner and Heike Fassbender for several suggestions that improved the paper and for their indirect contributions as well. Their work [4–8] made this work possible.

References

[1] T. Apel, V. Mehrmann, D. Watkins. Structured eigenvalue methods for the computation of corner singularities in 3d anisotropic elastic structures, Comput. Methods Appl. Mech. Engrg. 191 (2002) 4459–4473.

[2] G. Banse, Symplektische Eigenwertverfahren zur Lösung zeitdiskreter optimaler Steuerungsprobleme, Ph.D. Thesis, University of Bremen, 1995.
[3] G. Banse, A. Bunse-Gerstner, A condensed form for the solution of the symplectic eigenvalue problem, in: U. Helmke, R. Menniken, J. Sauer (Eds.), Systems and Networks: Mathematical Theory and Applications, Akademie Verlag, 1994, pp. 613–616.
[4] P. Benner, H. Fassbender, An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem, Linear Algebra Appl. 263 (1997) 75–111.
[5] P. Benner, H. Fassbender, The symplectic eigenvalue problem, the butterfly form, the SR algorithm, and the Lanczos method, Linear Algebra Appl. 275–276 (1998) 19–47.
[6] P. Benner, H. Fassbender, An implicitly restarted symplectic Lanczos method for the symplectic eigenvalue problem, SIAM J. Matrix Anal. Appl. 22 (2000) 682–713.
[7] P. Benner, H. Fassbender, D.S. Watkins, Two connections between the SR and HR eigenvalue algorithms, Linear Algebra Appl. 272 (1998) 17–32.
[8] P. Benner, H. Fassbender, D.S. Watkins, SR and SZ algorithms for the symplectic (butterfly) eigenproblem, Linear Algebra Appl. 287 (1999) 41–76.
[9] M.A. Brebner, J. Grad, Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to pseudosymmetric form and the HR process, Linear Algebra Appl. 43 (1982) 99–118.
[10] A. Bunse Gerstner, An analysis of the HR algorithm for computing the eigenvalues of a matrix, Linear Algebra Appl. 35 (1981) 155–173.
[11] A. Bunse Gerstner, V. Mehrmann, A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation, IEEE Trans. Automat. Control AC-31 (1986) 1104–1113.
[12] R. Freund, V. Mehrmann, A symplectic look-ahead Lanczos algorithm for the Hamiltonian eigenvalue problem, Unpublished manuscript.
[13] E.J. Grimme, D.C. Sorensen, P. Van Dooren, Model reduction of state space systems via an implicitly restarted Lanczos method, Numer. Algorithms 12 (1996) 1–31.
[14] P. Lancaster, Lambda-Matrices and Vibrating Systems, Pergamon Press, Oxford, 1966.
[15] P. Lancaster, Strongly stable gyroscopic systems, Electron. J. Linear Algebra 5 (1999) 53–66.
[16] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand. 45 (1950) 255–281.
[17] R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, 1998.
[18] V. Mehrmann, The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution, Lecture Notes in Control and Information Sciences, vol. 163, Springer Verlag, Heidelberg, 1991.
[19] V. Mehrmann, D. Watkins, Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltoninan/Hamiltonian pencils, SIAM J. Sci. Comput. 22 (2001) 1905–1925.
[20] D. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1992) 357–385.
[21] D.S. Watkins, L. Elsner, Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem, SIAM J. Matrix Anal. Appl. 15 (1994) 943–967.