Balancing Sparse Hamiltonian Eigenproblems*

Peter Benner† Daniel Kressner‡

April 3, 2003

Abstract

Balancing a matrix by a simple and accurate similarity transformation can improve the performance of numerical methods for computing eigenvalues. We describe balancing strategies for a large and sparse Hamiltonian matrix H. It is first shown how to permute H to irreducible form while retaining its structure. This form can be used to decompose the Hamiltonian eigenproblem into smaller-sized problems. Next, we discuss the computation of a symplectic scaling matrix D so that the norm of $D^{-1}HD$ is reduced. The considered scaling algorithm is solely based on matrix-vector products and thus particularly suitable if the elements of H are not explicitly given. The merits of balancing for eigenvalue computations are illustrated by several practically relevant examples.

1 Introduction

Computing eigenvalues of *Hamiltonian matrices*

$$H = \begin{bmatrix} A & G \\ Q & -A^T \end{bmatrix}, \tag{1}$$

where $A, G, Q \in \mathbb{R}^{n \times n}$ and G, Q are symmetric, has been an active field of research during the last two decades. Nowadays, there are several efficient and structure-preserving eigensolvers for dense Hamiltonian matrices available [5, 7, 8, 24]. Combined with recently developed balancing and block reduction algorithms [4, 15] they form more accurate and more efficient alternatives to standard eigensolvers like the QR algorithm [13].

Algorithms which compute eigenvalues of *sparse* Hamiltonian matrices are comparably less developed. Although some structure-preserving Lanczos and Arnoldi like methods have been investigated [6, 12, 19] there is no implementation publicly available yet. This work should be understood as a step to proliferate the development of such an implementation. We show how to balance a sparse Hamiltonian matrix H by a simple and accurate similarity transformation $D^{-1}HD$, which reduces the norm of H. This preprocessing step often has positive effects on the performance and accuracy of eigensolvers and therefore has obtained early attention in the development of numerical linear algebra [20, 21]. Balancing sparse eigenproblems is of more recent interest. Although efficient algorithms for approximate sparse balancing were

*Supported by the DFG Research Center "Mathematics for key technologies" (FZT 86) in Berlin.

†Institut für Mathematik, MA 4-5, Technische Universität Berlin, 10623 Berlin, Germany; e-mail: benner@math.tu-berlin.de

‡Institut für Mathematik, MA 4-5, Technische Universität Berlin, 10623 Berlin, Germany; e-mail: kressner@math.tu-berlin.de

already proposed in [10] the resulting balanced matrix $D^{-1}HD$ is in general not Hamiltonian. However, to be able to apply structure-preserving eigensolvers, D must be chosen in a way so that the structure of H is not destroyed. In [4], a structure-preserving balancing technique for Hamiltonian eigenproblems is developed. This technique is suitable for dense eigenproblems but is not readily applicable to large-scale problems where the Hamiltonian matrix is not given explicitly. We therefore investigate the question how to perform an approximate balancing using only information obtained from matrix-vector products with the Hamiltonian matrix.

Permuting H to block upper triangular form with irreducible diagonal blocks is a preprocessing step in order to guarantee the convergence of the sequel balancing step. It may also reduce the time needed by an eigensolver. In Section 2 we show how to achieve a similar form retaining the Hamiltonian structure. Diagonal transformations that aim to reduce the norm of the irreducible blocks of H are the topic of Section 3. In particular, a structure-preserving Krylov-based balancing algorithm similar to one of the algorithms presented in [10] will be developed. Numerical examples that illustrate the use of the proposed balancing strategies for eigenvalue computation are given in Section 4.

2 Irreducible forms

A matrix $A \in \mathbb{R}^{n\times n}$ is called *reducible* if there is a permutation matrix $P \in \mathbb{R}^{n\times n}$ so that

$$P^T A P = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \tag{2}$$

where A_{11} and A_{22} are square matrices of order not less than one. If no such permutation exists, then A is called *irreducible*. The matrices A_{11} and A_{22} can be further reduced until A is permuted to block upper triangular form with irreducible diagonal blocks. Constructing this final *irreducible form* is equivalent to finding the strongly connected components of the incidence graph of A and numbering them in reverse topological order. These connections are well known and shall not be reviewed here, see e.g. [14].

We now describe the necessary graph theoretic tools that allow us to develop a structure-preserving irreducible form for Hamiltonian matrices.

Definition 1 *Let $H \in \mathbb{R}^{2n\times 2n}$ be a Hamiltonian matrix as defined in (1), then the* incidence graph *of H, denoted by $\mathcal{G}_H(V,E)$, is a directed graph with vertex and edge sets*

$$\begin{aligned} V &= \{v_1,\dots,v_n,w_1,\dots,w_n\}, \\ E &= \{(v_i,v_j):\ a_{ij}\neq 0\} \cup \{(v_i,w_j):\ g_{ij}\neq 0\} \cup \\ &\quad\ \{(w_i,v_j):\ q_{ij}\neq 0\} \cup \{(w_i,w_j):\ -a_{ji}\neq 0\}. \end{aligned}$$

Lemma 2 *The incidence graph $\mathcal{G}_H(V,E)$ of a Hamiltonian matrix H satisfies the following:*

a) there is a path from v_i to v_j if and only if there is one from w_j to w_i,

b) there is a path from v_i to w_j if and only if there is one from v_j to w_i,

c) there is a path from w_i to v_j if and only if there is one from w_j to v_i.

Proof. By induction on the path length k. For $k = 2$, the conclusions are direct consequences of the definition of the incidence graph and the symmetries of G and Q. Now, assume that a)–c) hold for any path of maximum length $\bar{k}$ and that there is a path of length $\bar{k}+1$ from v_i to w_j. This path is cut; we choose a vertex $p = v_l$ or $p = w_l$ so that there are paths of length not greater than $\bar{k}$ from v_i to p and from p to w_j. In the case $p = v_l$, the induction hypothesis shows the existence of paths from w_l to w_i and from v_j to w_l. In the other case, we obtain paths from v_l to w_i and from v_j to v_l. In any case, there is a path from v_j to w_i. The converse as well as assertions a) and c) are proven analogously. $\square$

A directed graph is called *strongly connected* if for any pair of vertices v and w there is a path from v to w and one from w to v. The *strongly connected components* of a directed graph are its maximal strongly connected subgraphs. Lemma 2 implies consequences for the structure of the strongly connected components of $\mathcal{G}_H(V,E)$.

Corollary 3 *Let $\mathcal{G}_H(V,E)$ be the incidence graph of a Hamiltonian matrix H.*

$$V_1 := \{v_{i_1}, \ldots, v_{i_k}, w_{j_1}, \ldots, w_{j_l}\}$$

is the vertex set of a strongly connected component if and only if

$$\check{V}_1 := \{v_{j_1}, \ldots, v_{j_l}, w_{i_1}, \ldots, w_{i_k}\}$$

is the vertex set of a strongly connected component.

If $V_1 \cap \check{V}_1 \neq \emptyset$ then the strong connectivity property and Corollary 3 enforce $V_1 = \check{V}_1$. The corresponding component will be called of *type (II)*. In the other case, $V_1 \cap \check{V}_1 = \emptyset$, we say that the corresponding component is of *type (I)*. Further information is available about edges between components.

Lemma 4 *Let $\mathcal{G}_H(V,E)$ be the incidence graph of a Hamiltonian matrix H and let V_1 and V_2 correspond to strongly connected components of type (I). Then, there is an edge from a vertex in V_1 to a vertex in V_2 if and only if there is one from a vertex in $\check{V}_2$ to a vertex in $\check{V}_1$, where $\check{V}_1$ and $\check{V}_2$ are defined as in Corollary 3. Moreover, there are no edges between components of type (II).*

Proof. Let $(v_i, v_j) \in E$ with $v_i \in V_1$ and $v_j \in V_2$. Then, by definition, $w_i \in \check{V}_1$, $w_j \in \check{V}_2$ and by Lemma 2a), $(w_j, w_i) \in E$. For the second part, let V_3 and V_4 correspond to two distinct strongly connected components of type (II) and assume that there are vertices $v_i \in V_3$, $v_j \in V_4$ with $(v_i, v_j) \in E$. This implies $w_i \in V_3$ and $w_j \in V_4$. Again, by Lemma 2a), $(w_j, w_i) \in E$, which means that $V_3 \cup V_4$ is a strongly connected component contradicting the assumption. The proof is analogous for edges of the form (v_i, w_j), (w_i, v_j) or (w_i, w_j). $\square$

Let us denote $V_1 \preceq V_2$ if there is no edge from a vertex in V_2 to a vertex in V_1. This relation defines a preorder on the set of strongly connected components, the so called *topological order* [14]. Lemma 4 implies that there is always a numbering of the strongly connected components so that their reverse topological order takes the form

$$V_1 \succeq \cdots \succeq V_r \succeq V_{r+1} \succeq \cdots \succeq V_{r+s} \succeq \check{V}_r \succeq \cdots \succeq \check{V}_1, \tag{3}$$

where $V_1, \ldots, V_r$, $\check{V}_1, \ldots, \check{V}_r$ correspond to type (I) components and $V_{r+1}, \ldots, V_{r+s}$ correspond to type (II) components of $\mathcal{G}_H(V,E)$.

This shows, together with Corollary 3, the existence of a permutation matrix P so that

$$P^T H P = \begin{bmatrix} A_1 & \star & \dots & \star & \dots & \dots & \star & \dots & \star \\ & \ddots & \ddots & \vdots & & & \vdots & & \vdots \\ & & A_r & \star & \dots & \dots & \star & \dots & \star \\ & & & H_{r+1} & 0 & 0 & \vdots & & \vdots \\ & & & & \ddots & 0 & \vdots & & \vdots \\ & & & & & H_{r+s} & \star & \dots & \star \\ & & & & & & -A_r^T & \ddots & \vdots \\ & & & & & & & \ddots & \star \\ & & & & & & & & -A_1^T \end{bmatrix}, \tag{4}$$

where all diagonal blocks are irreducible and each matrix $H_{r+i} = \left[\begin{smallmatrix} A_{r+i} & G_{r+i} \\ Q_{r+i} & -A_{r+i}^T \end{smallmatrix}\right]$ is Hamiltonian. Thus, the spectrum of H contains the eigenvalues of the unstructured matrices $A_i, -A_i^T$ and those of the Hamiltonian matrices H_{r+i}. Unfortunately, the irreducible form (4) does not respect the Hamiltonian structure of H. Therefore, it is now of interest to construct a permutation matrix $\tilde{P}$ which is structure-preserving and leads to a form $\tilde{P}^T H \tilde{P}$ where the essential information of (4) can be easily read off. First, let us recall some well-known results about symplectic matrices.

Lemma 5 *Let $U \in \mathbb{R}^{2n\times 2n}$ be symplectic, i.e., $U^T J U = J$ with $J = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$, then the following holds.*

1. *If $H \in \mathbb{R}^{2n\times 2n}$ is a Hamiltonian matrix then $U^{-1}HU$ is again Hamiltonian.*
2. *If U is moreover orthogonal then it has the block representation $U = \left[\begin{smallmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{smallmatrix}\right]$ with $U_1, U_2 \in \mathbb{R}^{n\times n}$.*

It was already noted in [4] that the group of symplectic permutation matrices is too restrictive to obtain useful classifications for $P^T H P$. Hence, we propose to broaden the range of similarity transformations to $\tilde{P}^T H \tilde{P}$, where $\tilde{P} = DP$ is symplectic, $D = \mathrm{diag}\{\pm 1, \dots, \pm 1\}$ and P is a permutation matrix. These *symplectic generalized permutation matrices* clearly form a group and thus $\tilde{P}^T H \tilde{P}$ generates an equivalence relation, which shall now be classified.

Theorem 6 *For any Hamiltonian matrix H there exists a symplectic generalized permutation matrix $\tilde{P}$ so that*

$$\tilde{H} := \tilde{P}^T H \tilde{P} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{21} & \tilde{G}_{11} & \tilde{G}_{12} \\ 0 & \tilde{A}_{22} & \tilde{G}_{12}^T & \tilde{G}_{22} \\ 0 & 0 & -\tilde{A}_{11}^T & 0 \\ 0 & \tilde{Q}_{22} & -\tilde{A}_{21}^T & -\tilde{A}_{22}^T \end{bmatrix}, \tag{5}$$

where

$$\tilde{A}_{11} = \begin{bmatrix} A_1 & \star & \dots & \star \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \star \\ 0 & \dots & 0 & A_r \end{bmatrix}, \qquad \tilde{A}_{22} = \begin{bmatrix} A_{r+1} & 0 & \dots & 0 \\ 0 & A_{r+2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & A_{r+s} \end{bmatrix},$$

$$\tilde{G}_{22} = \begin{bmatrix} G_{r+1} & 0 & \dots & 0 \\ 0 & G_{r+2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & G_{r+s} \end{bmatrix}, \quad \tilde{Q}_{22} = \begin{bmatrix} Q_{r+1} & 0 & \dots & 0 \\ 0 & Q_{r+2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & Q_{r+s} \end{bmatrix},$$

and all matrices A_i, $i = 1, \dots, r$, and $\left[\begin{smallmatrix} A_{r+i} & G_{r+i} \\ Q_{r+i} & -A_{r+i}^T \end{smallmatrix}\right]$, $i = 1, \dots, s$, are irreducible.

Proof. First, note that the elementary symplectic matrix

$$E_j = \begin{bmatrix} I_{j-1} & & & & \\ & 0 & & 1 & \\ & & I_{n-1} & & \\ & -1 & & 0 & \\ & & & & I_{n-j} \end{bmatrix}$$

swaps, when applied to H, the vertices v_j and w_j of the incidence graph. This allows us to construct a product of elementary matrices, $\tilde{P}_1 = E_{j_1} E_{j_2} \dots E_{j_k}$, so that in all type (I) components of $\mathcal{G}_{\tilde{P}_1^T H \tilde{P}_1}(V, E)$ the vertex sets V_i contain only v-vertices, $\check{V}_i$ only w-vertices and $V_i \succeq \check{V}_i$ for $i = 1, \dots, r$. By a simultaneous reordering of the v- and w-vertices, there is a permutation matrix P_2 such that

$$\tilde{H} = \begin{bmatrix} P_2^T & 0 \\ 0 & P_2^T \end{bmatrix} (\tilde{P}_1^T H \tilde{P}_1) \begin{bmatrix} P_2 & 0 \\ 0 & P_2 \end{bmatrix}$$

has an incidence graph whose strongly connected components correspond to the vertex sets

$$V_1 = \{v_1, \dots, v_{k_1}\},\ V_2 = \{v_{k_1+1}, \dots, v_{k_2}\},\ \dots,\ V_r = \{v_{k_{r-1}+1}, \dots, v_{k_r}\},$$

$$V_{r+1} = \{v_{k_r+1}, \dots, v_{k_{r+1}}, w_1, \dots, w_{l_1}\},\ \dots,\ V_{r+s} = \{v_{k_{r+s-1}+1}, \dots, v_{k_{r+s}}, w_{l_{s-1}+1}, \dots, w_{l_s}\},$$

$$\check{V}_1 = \{w_1, \dots, w_{k_1}\},\ \check{V}_2 = \{w_{k_1+1}, \dots, w_{k_2}\},\ \dots,\ \check{V}_r = \{w_{k_{r-1}+1}, \dots, w_{k_r}\},$$

where the topological order is given by (3). Now, the structure of $\tilde{H}$ is a direct consequence of Lemma 4. □

A procedure to transform a Hamiltonian matrix H to the form (5) is given by the following algorithm. It calls a subroutine topsort which is discussed below.

Algorithm 7

Input:	*Matrices $A, G, Q \in \mathbb{R}^{n \times n}$, with $Q = Q^T, G = G^T$, defining a Hamiltonian matrix $H \in \mathbb{R}^{2n \times 2n}$.*
Output:	*A symplectic generalized permutation matrix $\tilde{P}$; matrices $A, G, Q \in \mathbb{R}^{n \times n}$, with $Q = Q^T, G = G^T$, defining a Hamiltonian matrix $\tilde{H} = \tilde{P}^T H \tilde{P}$ which has the form (5).*

$\tilde{P} \leftarrow I_{2n}$
$(V_1, \ldots, V_k) \leftarrow \text{topsort}\left(\begin{bmatrix} A & G \\ Q & -A^T \end{bmatrix} \right)$
FOR $i = 1, \ldots, k$
  IF $\exists \{v_j, w_j\} \in V_k$ THEN $\text{type}(i) \leftarrow 2$ ELSE $\text{type}(i) \leftarrow 1$ END IF
END FOR
$l \leftarrow 0$
FOR $i = 1, \ldots, k$
  IF $\text{type}(i) = 1$ AND $l < \#(\text{type}(i) = 1)/2$ THEN
    $l \leftarrow l + 1$
    FOR EACH $w_j \in V_k$
      $P_1 \leftarrow I_n - e_j e_j^T, \quad P_2 \leftarrow I_n - P_1$
      $A \leftarrow P_1 A P_1 - P_2 A^T P_2 - P_1 G P_2 - P_2 Q P_1$
      $G \leftarrow P_1 G P_1 + P_1 A P_2 + P_2 A^T P_1 - P_2 Q P_2$
      $Q \leftarrow P_1 Q P_1 + P_2 A P_1 + P_1 A^T P_2 - P_2 G P_2$
      $\tilde{P} \leftarrow \tilde{P} \begin{bmatrix} P_1 & P_2 \\ -P_2 & P_1 \end{bmatrix}$
      $\text{delete}(V_k, w_j), \quad \text{insert}(V_k, v_j)$
    END FOR
  END IF
END FOR
$p \leftarrow 0$
FOR $i = 1, \ldots, k$
  IF $p < n$ THEN
    FOR EACH $v_j \in V_k$
      $p \leftarrow p + 1$
      $P_1 \leftarrow I_n + e_j e_p^T + e_p e_j^T - e_j e_j^T - e_p e_p^T$
      $A \leftarrow P_1 A P_1, \quad G \leftarrow P_1 G P_1, \quad Q \leftarrow P_1 Q P_1$
      $\tilde{P} \leftarrow \tilde{P} \begin{bmatrix} P_1 & 0 \\ 0 & P_1 \end{bmatrix}$
    END FOR
  END IF
END FOR

The called subroutine topsort accepts an arbitrary square matrix as input and returns the vertex sets of the strongly connected components of its incidence graph in reverse topological order. This can be done with Tarjan's algorithm [23] which uses two depth first searches and has found an excellent implementation in Fortran 77 [11]. In an object-oriented programming environment, it is preferable to use an implementation which is able to handle arbitrarily defined graphs. In this case, no information about the incidence graph has to be stored. Such subroutines are for example provided by the C++ library LEDA [17]. The complete Algorithm 7 runs in $O(n + nz)$ time, where nz is the number of nonzero in H.

The permutation algorithm proposed in [4, Alg. 3.4] is a special case of Algorithm 7. It is a structure-preserving version of the permutation algorithm from [21] that is used, e.g., in LAPACK [2] for solving nonsymmetric eigenproblems. It aims at deflating isolating eigenvalues. In graph-theoretic terms, it works on sink and source nodes of $\mathcal{G}_H(V, E)$ or subgraphs thereof. A sink (source) node s corresponds to a strongly connected component of $\mathcal{G}_H(V, E)$ that satisfies $s \preceq V_j$ ($s \succeq V_j$) for any other strongly connected component V_j. In the first phase, the algorithm seeks a sink node s of $\mathcal{G}_H(V, E)$ and permutes the corresponding diagonal element to the (n, n) position of H by means of a symplectic generalized permutation matrix $\tilde{P}$. For the rest only the matrix $\tilde{P}^T H \tilde{P}$ with the first and n-th rows and columns expunged is considered. The algorithm iterates until no sink node can be found. In the second phase, the procedure is repeated for source nodes, which are permuted to the $(1, 1)$ position. The worst case complexity of this algorithm is $O(n \cdot nz)$ which compares unfavorable with the complexity of Algorithm 7. However, from the experiences made with general matrices [10] it can be concluded that Algorithm 7 requires more computing time if the density of H becomes greater than a certain value, say 1/10.

3 Scaling

Suppose we have transformed the Hamiltonian matrix to the form (5). The Hamiltonian eigenproblem now decomposes into eigenproblems for matrices of the form

$$H_I = \begin{bmatrix} A_I & G_I \\ 0 & -A_I^T \end{bmatrix}, \quad H_{II} = \begin{bmatrix} A_{II} & G_{II} \\ Q_{II} & -A_{II}^T \end{bmatrix},$$

where A_I and H_{II} are irreducible matrices. In this section we describe scaling algorithms that aim to reduce the norms of these matrices while preserving their Hamiltonian structure.

Dense matrices are easily handled; the standard balancing procedure described in [20, 21] can be applied to H_I to construct a diagonal matrix D such that the rows and columns of $D^{-1} A_I D$ are as close in norm as possible. If the matrix H_I is transformed to

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} A_I & G_I \\ 0 & -A_I^T \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & D^{-1} \end{bmatrix} = \begin{bmatrix} D^{-1} A_I D & D^{-1} G_I D^{-1} \\ 0 & -(D^{-1} A_I D)^T \end{bmatrix},$$

then also the matrix $-(D^{-1} A_I D)^T$ is balanced in this sense. Such a scaling of a (not necessarily block-upper triangular) Hamiltonian matrix based on balancing just A is also used in [5].[1] How to equilibrate row and column norms of H_{II} by a symplectic scaling matrix is described in [4, Alg. 4.4].

Both procedures require the calculation of row and column norms, implying that matrix elements must be given explicitly. This requirement is sometimes not satisfied, a large and sparse matrix might only be defined through its action on a vector. For these cases, only balancing algorithms which are solely based on a few matrix-vector multiplications, and possibly matrix-transpose-vector multiplications, can be used. Such algorithms were developed by Chen and Demmel [10] for general matrices. One of the algorithms, the so called KrylovAtz, is based on the following fact.

Lemma 8 *Let $A \in \mathbb{R}^{n \times n}$ be an irreducible matrix with non-negative entries and spectral radius $\rho(A)$. Let x and y be the normalized right and left Perron vectors of A, i.e., $Ax = \rho(A)x$ and $A^T y = \rho(A) y$ with $\|x\|_2 = \|y\|_2 = 1$. If*

$$D = \operatorname{diag}(\sqrt{x_1/y_1}, \sqrt{x_2/y_2}, \ldots, \sqrt{x_n/y_n}), \tag{6}$$

then $\|D^{-1} A D\|_2 = \rho(A)$.

[1]Note: if using the corresponding implementation (Algorithm 800 from the ACM Collected Algorithms, CALGO, see `http://www.acm.org/calgo`), it is beneficial to replace the balancing part by either one of the strategies developed here or in [4].

Proof. See e.g. [10]. ∎

This scaling achieves minimal 2-norm as $\|X^{-1}AX\|_2 \geq \rho(A)$ for any nonsingular matrix X. Also, the right and left Perron vectors of $D^{-1}AD$ equal $D^{-1}x = Dy$, thus the condition number of the spectral radius becomes minimal. If A contains negative entries we can apply Lemma 8 to $|A| := [|a_{ij}|]_{i,j=1}^n$ to construct a (possibly suboptimal) diagonal scaling matrix D. It was observed in [10] that this choice of scaling improves the accuracy of the computed eigenvalues for almost all considered examples. Nevertheless, it is not clear how to predict the potential gain in accuracy. Applying the proposed scaling to a Hamiltonian matrix yields a symplectic scaling matrix as the following lemma shows.

Lemma 9 *Let $H \in \mathbb{R}^{2n\times 2n}$ be an irreducible Hamiltonian matrix and $S = |H|$. If x and y are the normalized right and left Perron vectors of S, then the diagonal matrix D defined in (6) is symplectic.*

Proof. Let $P = \left[\begin{smallmatrix} 0 & I_n \\ I_n & 0 \end{smallmatrix}\right]$, then $PS = S^TP$ and thus $x = Py$ which implies that D is symplectic. ∎

It remains to compute the Perron vectors x and y of $|H|$. In principle, one could apply the power method to $|H|$ to approximate these vectors. However, if H is not explicitly defined then also the action of $|H|$ on a vector must be approximated by matrix-vector products which involves only the matrix H itself. For a general matrix A, a statistically motivated procedure based on products with a random vector z, where the entries z_i equal 1 or -1 with probability 1/2, was presented in [10]. It makes use of the fact that multiplying A by z approximates one step of the power method applied to $|A|$ with starting vector $[1, 1, \ldots, 1]^T$. For Hamiltonian matrices, the following lemma shows how to guarantee that these approximations yield symplectic scaling matrices.

Lemma 10 *Let $H \in \mathbb{R}^{2n\times 2n}$ be a Hamiltonian matrix partitioned as in (1), and let $z \in \mathbb{R}^{2n}$. If*

$$\bar{H} = \begin{bmatrix} A & G \\ Q & A^T \end{bmatrix}, \quad P = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}, \tag{7}$$

then $p = \bar{H}z$ and $r = \bar{H}^TPz$ satisfy $r = Pp$ implying that

$$\begin{aligned} D &= \operatorname{diag}(\sqrt{r_1/p_1}, \ldots, \sqrt{r_{2n}/p_{2n}}) \\ &= \operatorname{diag}(\sqrt{p_{n+1}/p_1}, \ldots, \sqrt{p_{2n}/p_n}, \sqrt{p_1/p_{n+1}}, \ldots, \sqrt{p_n/p_{2n}}) \end{aligned}$$

is symplectic.

Proof. The statement is a direct consequence of the fact that $P\bar{H}$ is symmetric. ∎

This leads us to the following adaption of KrylovAtz to Hamiltonian matrices.

Algorithm 11

Input: *An irreducible Hamiltonian matrix $H \in \mathbb{R}^{2n\times 2n}$.*

Output: *A symplectic diagonal matrix D so that $D^{-1}HD$ is approximately balanced in the sense of Lemma 8.*

$D \leftarrow I_{2n}$
FOR $k = 1, 2, \ldots$
  $z \leftarrow$ vector of length $2n$ with random ± 1 entries
  $z \leftarrow Dz, \quad p \leftarrow \bar{H}z, \quad p \leftarrow D^{-1}p$

```
    FOR i = 1,...,n
      IF p_i ≠ 0 AND p_{n+i} ≠ 0 THEN
        d_ii ← d_ii · sqrt(p_i / p_{n+i})
        d_{n+i,n+i} ← d_{n+i,n+i} · sqrt(p_{n+i} / p_i)
      END IF
    END FOR
  END FOR
```

Two remarks are in order.

Remark 12 *Algorithm 11 works with $\bar{H}$, see (7), instead of H implying that the action of the matrices A, G and Q on a vector must be known. Alternatively, one could make use of the relation*

$$\bar{H}\begin{bmatrix} z_u \\ z_l \end{bmatrix} = H\begin{bmatrix} z_u \\ 0 \end{bmatrix} + \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} H \begin{bmatrix} 0 \\ z_l \end{bmatrix}.$$

Remark 13 *Based on the experimental results presented in [10] we propose to replace the condition $p_i \neq 0$ in the inner loop of Algorithm 11 by $|p_i| > \delta \|H\|_F$. Although there is little theoretical justification for adding such a* cutoff value *δ it turns out that the choice $\delta = 10^{-8}$ often results in smaller norms for the scaled matrices.*

4 Numerical Examples

Since most applications of Hamiltonian eigenproblems have their background in control theory we used for our numerical experiments two benchmark collections from this area. The first collection [1] contains examples of continuous-time algebraic Riccati equations (CAREs) of the form

$$Q + A^T X + XA - XGX = 0.$$

Computing eigenvalues and invariant subspaces of the corresponding Hamiltonian matrix $H = \left[\begin{smallmatrix} A & G \\ Q & -A^T \end{smallmatrix}\right]$ plays a fundamental role in most algorithms to solve CAREs; see, e.g., [3, 18, 22]. The second collection by Chahlaoui and Van Dooren [9] aims at model reduction problems for linear time-invariant systems. Each example provides matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. Here, the corresponding Hamiltonian matrix $H = \left[\begin{smallmatrix} A & BB^T \\ C^T C & -A^T \end{smallmatrix}\right]$ can be used to determine the closed-loop poles of the system which in turn can help to evaluate the quality of the closed-loop performance of the reduced-order model.

4.1 Permutation algorithms

We compared the permutation algorithm in [4, Alg. 3.4], in the following denoted by HPER, with the proposed Algorithm 7, denoted by HIRR. Both algorithms attempt to decouple the Hamiltonian eigenproblem into smaller-sized problems. HIRR is potentially more successful, as explained at the end of Section 2. Indeed, we observed this phenomena in the four examples that are listed in Table 1. For instance, HPER applied to Example 2.9 [1] isolates four eigenvalues, which means that the other eigenvalues can be computed from a 106×106 Hamiltonian matrix. HIRR isolates ten eigenvalues. The other eigenvalues are contained in two 2×2 matrices and a 96×96 Hamiltonian matrix. As costs for computing eigenvalues crucially depend on the size of the largest decoupled eigenproblem, we may conclude that it

Example	n	HPER	HIRR
Ex. 1.6 [1]	30	$8 \times (1 \times 1)$	$8 \times (1 \times 1)$
			$2 \times (2 \times 2)$
		$1 \times (52 \times 52)$	$1 \times (48 \times 48)$
Ex. 2.4 [1] ($\varepsilon = 0$)	4	$1 \times (4 \times 4)$	$2 \times (2 \times 2)$
Ex. 2.9 [1]	55	$4 \times (1 \times 1)$	$10 \times (1 \times 1)$
			$2 \times (2 \times 2)$
		$1 \times (106 \times 106)$	$1 \times (96 \times 96)$
Ex. 4.3 [1]	60	$56 \times (1 \times 1)$	$56 \times (1 \times 1)$
($\mu = 4, \delta = 0, \kappa = 0$)		$1 \times (8 \times 8)$	$2 \times (4 \times 4)$

Table 1: Sizes of the decoupled eigenproblems after application of HPER and HIRR.

will be beneficial to use HIRR as a cheap preliminary reduction step. However, it should be noted that all examples of [9] correspond to irreducible Hamiltonian matrices, showing the limitation of such an approach.

4.2 Matrix norm reduction

We examined the capability of Algorithm 11, in the following denoted by HTZ, to reduce the norms of Hamiltonian matrices. If a cutoff value δ, see Remark 13, was used then Algorithm 11 is denoted by CUT. We let the number of iterations in HTZ and CUT vary from 1 to 10, the cutoff δ from 0.1 to 10^{-10} by powers of 10, and measured the minimal Frobenius norm of the scaled matrices. Those norms were compared with the Frobenius norms of the scaled matrices returned by Algorithm 4.4 [4]), or for short BAL. All tests were done in MATLAB. Table 2 summarizes the results we obtained with the two benchmark collections. For the examples not listed either scaling strategy makes no or little difference to the matrix norm. In most cases, BAL, HTZ and CUT give very similar results. A notable exception is Example

Example	n	$\|H\|_F$	BAL	HTZ	CUT
Ex. 1.6 [1]	30	1.4×10^{08}	1.2×10^{03}	1.3×10^{03}	1.3×10^{03}
Ex. 2.2 [1]	2	1.0×10^{06}	2.9×10^{05}	5.9×10^{05}	2.7×10^{05}
Ex. 2.3 [1]	2	1.4×10^{06}	2.0×10^{04}	1.4×10^{06}	1.8×10^{05}
Ex. 2.7 [1]	4	1.0×10^{12}	2.1×10^{06}	1.9×10^{06}	1.9×10^{06}
Ex. 2.9 [1]	55	4.4×10^{10}	4.0×10^{03}	4.1×10^{03}	2.7×10^{03}
Ex. 4.4 [1]	421	8.6×10^{11}	2.5×10^{06}	7.2×10^{09}	3.5×10^{06}
Beam [9]	348	1.0×10^{05}	5.0×10^{03}	6.3×10^{03}	5.7×10^{03}
Build [9]	48	2.2×10^{04}	8.0×10^{02}	5.4×10^{03}	2.9×10^{03}
CDPlayer [9]	120	1.5×10^{06}	3.3×10^{05}	3.6×10^{05}	3.4×10^{05}
ISS [9]	270	2.9×10^{04}	8.8×10^{02}	3.4×10^{04}	3.4×10^{04}

Table 2: Norms of Hamiltonian matrices with and without scaling.

4.4, where HTZ reduces the norm of H only by two orders of magnitude while BAL and CUT reduce it by more than five orders of magnitude. Furthermore, only BAL is capable to reduce the norm of the ISS example from [9].

It was proposed in [10] to use 5 as the default number of iterations and $\delta = 10^{-8}$ as the default cutoff value. Using those instead of optimal values, the norms of the scaled matrices returned by CUT are usually no more than by a factor of ten larger. The only exception in the benchmark collections is Example 1.6 where the norm of the scaled matrix, using CUT with default values, is 1.7×10^{05}.

4.3 Eigenvalue computation

Balancing may have a strong positive impact on the accuracy of eigencomputations. The first point we want to illuminate is the merits of decoupling. Let us consider Example 2.9 [1]. We applied a MATLAB implementation of the square-reduced method [5], (SQRED), to the corresponding 110×110 Hamiltonian matrix. The relative errors of seven selected eigenvalues are displayed in the second column of Table 3. The 'exact' eigenvalues used to obtain these errors were computed with the QR algorithm in quadruple precision. Next, we used HPER as a preliminary reduction step, which isolates $\pm\lambda_1, \pm\lambda_2$. Consequently, these eigenvalues are computed without any round-off error. All the other eigenvalues were computed using SQRED applied to the remaining 106×106 block. The third column of Table 3 contains the resulting relative errors. With HIRR ten eigenvalues, $\pm\lambda_1, \pm\lambda_2, \ldots, \pm\lambda_5$, are isolated and four eigenvalues $\pm\lambda_6, \pm\lambda_7$ are contained in two 2×2 blocks. The latter eigenvalues were computed applying the QR algorithm to the 2×2 blocks which yields, as can be seen in the last column of Table 3, relatively small errors. In fact, they are by almost 10 orders more accurate than the eigenvalues obtained by SQRED with and without HPER.

Eigenvalue	SQRED	HPER+SQRED	HIRR
$\lambda_1 = -20$	1.7×10^{-05}	0	0
$\lambda_2 = -20$	1.7×10^{-05}	0	0
$\lambda_3 \approx -5.30$	1.2×10^{-10}	4.5×10^{-11}	0
$\lambda_4 \approx -33.3$	1.2×10^{-12}	7.7×10^{-11}	0
$\lambda_5 \approx -221$	3.8×10^{-13}	4.2×10^{-12}	0
$\lambda_6 \approx -5.16 + 5.26\imath$	1.9×10^{-06}	2.6×10^{-05}	5.5×10^{-15}
$\lambda_7 \approx -5.16 - 5.26\imath$	1.9×10^{-06}	2.6×10^{-05}	5.5×10^{-15}

Table 3: Relative errors of eigenvalues computed by the square-reduced method with and without permuting.

We also investigated the influence of scaling on the accuracy of sparse eigensolvers. For this purpose, we applied the Fortran implementation of ARPACK [16] to the 96×96 irreducible Hamiltonian matrix $\tilde{H}$ obtained after HIRR had been applied to Example 2.9 [1]. The parameter `tol` in the stopping criterion, see [16, Sec. 2.3.5], was set to machine precision and the dimension of the Arnoldi basis was limited to 40. All computations were performed in a Compaq Visual Fortran environment. ARPACK computed the 20 eigenvalues of largest magnitude, the relative errors of those eigenvalues which have negative real part are displayed in the left graph of Figure 1. Also displayed are the relative errors when ARPACK is applied to the operators $D_{\mathrm{BAL}}^{-1}\tilde{H}D_{\mathrm{BAL}}$ and $D_{\mathrm{CUT}}^{-1}\tilde{H}D_{\mathrm{CUT}}$, where D_{BAL} and D_{CUT} are the symplectic scaling matrices computed by BAL and CUT, respectively. The right graph contains the same quantities, but now a Fortran implementation of SHIRA [19] instead of ARPACK is used. SHIRA is basically ARPACK applied to the operators $\tilde{H}^2$, $D_{\mathrm{BAL}}^{-1}\tilde{H}^2D_{\mathrm{BAL}}$ and $D_{\mathrm{CUT}}^{-1}\tilde{H}^2D_{\mathrm{CUT}}$

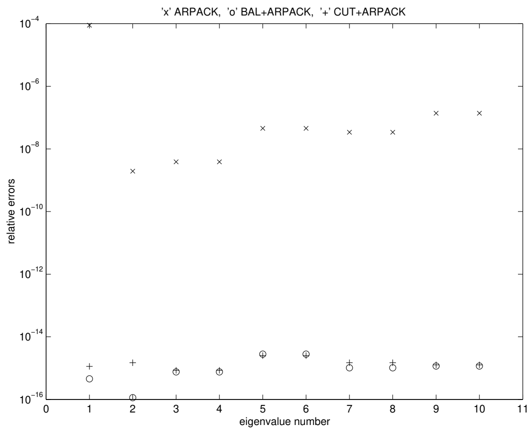

Figure 1: Relative errors of eigenvalues computed by ARPACK and SHIRA with and without scaling for Example 2.9.

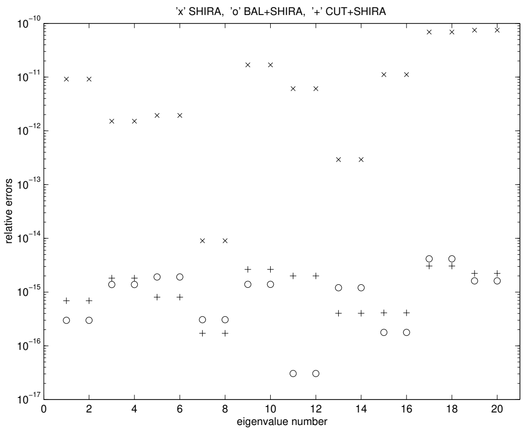

Figure 2: Relative errors of eigenvalues computed by SHIRA with and without scaling for Example 4.4.

with slight modifications to guarantee that the Arnoldi basis satisfies a certain relationship. Figure 1 shows that for both, ARPACK and SHIRA, either scaling strategy yields considerable improvements with respect to eigenvalue accuracies.

Example 4.4 from [1] demonstrates that balancing is a must in some applications. The QR algorithm applied to the corresponding Hamiltonian matrix without balancing does *not* converge. ARPACK encounters a similar error, the QR algorithm fails to compute the eigenvalues of some Ritz block during the Arnoldi iteration. Scaling resolves these problems. Both, BAL+ARPACK and CUT+ARPACK compute eigenvalues with a relative error close to machine precision. On the other hand, SHIRA runs to completion, even for the unscaled matrix. The relative errors of the 20 largest eigenvalues with negative real part computed with SHIRA, BAL+SHIRA and CUT+SHIRA are displayed in Figure 2. Again, Figure 2 shows that scaling leads to considerably more accurate eigenvalues.

5 Conclusions

We have introduced a new permutation strategy for Hamiltonian matrices based on graph-theoretic considerations. The corresponding structure-preserving similarity transformation allows to reduce Hamiltonian matrices to a block-upper triangular form with irreducible diagonal blocks, thereby improving earlier permutation strategies based on deflating isolated eigenvalues. For the subsequent step of balancing Hamiltonian matrices using diagonal scaling, we propose a method that requires only information provided by matrix-vector products. This method, which is closely related to an approximate balancing algorithm for general matrices, allows to (approximately) balance sparse Hamiltonian eigenproblems where the Hamiltonian matrix is not given explicitly. Numerical examples clearly exhibit the advantages of scaling. The implementation of this balancing strategy will hopefully help to develop robust numerical software for solving sparse Hamiltonian eigenproblems.

6 Acknowledgments

The authors thank Cornelia Pester for providing the Fortran implementation of SHIRA used in the numerical experiments. Parts of this work were completed while the second author was staying as a Marie Curie fellow (MCFI-2001-00403) at CESAME, Université catholique de Louvain. The hospitality of this institute is gratefully acknowledged.

References

[1] J. Abels and P. Benner. CAREX - a collection of benchmark examples for continuous-time algebraic Riccati equations (version 2.0). SLICOT working note 1999-14, WGS, 1999. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[3] P. Benner. Computational methods for linear-quadratic optimization. *Supplemento ai Rendiconti del Circolo Matematico di Palermo, Serie II*, No. 58:21–56, 1999.

[4] P. Benner. Symplectic balancing of Hamiltonian matrices. *SIAM J. Sci. Comput.*, 22(5):1885–1904, 2000.

[5] P. Benner, R. Byers, and E. Barth. Algorithm 800. Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices I: The square-reduced method. *ACM Trans. Math. Software*, 26(1):49–77, 2000.

[6] P. Benner and H. Faßbender. An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem. *Linear Algebra Appl.*, 263:75–111, 1997.

[7] P. Benner, V. Mehrmann, and H. Xu. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numerische Mathematik*, 78(3):329–358, 1998.

[8] R. Byers. A Hamiltonian QR algorithm. *SIAM J. Sci. Statist. Comput.*, 7(1):212–229, 1986.

[9] Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. SLICOT working note 2002-2, WGS, 2002. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[10] T.-Y. Chen and J. W. Demmel. Balancing sparse matrices for computing eigenvalues. *Linear Algebra Appl.*, 309:261–287, 2000.

[11] I. S. Duff and J. K. Reid. An implementation of Tarjan's algorithm for the block triangularization of a matrix. *ACM Trans. Math. Software*, 4:137–147, 1978.

[12] W.R. 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(10):23–40, 1997.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[14] D. E. Knuth. *The art of computer programming. Volume 3, Sorting and searching.* Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973.

[15] D. Kressner. Block algorithms for orthogonal symplectic factorizations, 2002. Submitted.

[16] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK Users' Guide.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[17] K. Mehlhorn and S. Näher. *LEDA. A platform for combinatorial and geometric computing.* Cambridge University Press, Cambridge, 1999.

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

[19] V. Mehrmann and D. Watkins. Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Sci. Comput.*, 22(6):1905–1925, 2000.

[20] E. E. Osborne. On preconditioning of matrices. *Journal of the ACM*, 7:338–345, 1960.

[21] B. N. Parlett and C. Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numerische Mathematik*, 13:293–304, 1969. Also in [25, pp.315–326].

[22] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, NY, 1996.

[23] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[24] C. F. Van Loan. A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix. *Linear Algebra Appl.*, 61:233–251, 1984.

[25] J. H. Wilkinson and C. Reinsch. *Handbook for Automatic Computation. Vol. II Linear Algebra.* Springer-Verlag, New York, 1971.