

# DOUBLING ALGORITHMS WITH PERMUTED LAGRANGIAN GRAPH BASES

VOLKER MEHRMANN\* AND FEDERICO POLONI†

**Abstract.** We derive a new representation of Lagrangian subspaces in the form

$$\operatorname{Im} \Pi^T \begin{bmatrix} I \\ X \end{bmatrix},$$

where  $\Pi$  is a symplectic matrix which is the product of a permutation matrix and a real orthogonal diagonal matrix, and  $X$  satisfies

$$|X_{ij}| \leq \begin{cases} 1 & \text{if } i = j, \\ \sqrt{2} & \text{if } i \neq j. \end{cases}$$

This representation allows to limit element growth in the context of doubling algorithms for the computation of Lagrangian subspaces and the solution of Riccati equations. It is shown that a simple doubling algorithm using this representation can reach full machine accuracy on a wide range of problems, obtaining invariant subspaces of the same quality as those computed by the state-of-the-art algorithms based on orthogonal transformations.

The same idea carries over to representations of arbitrary subspaces and can be used for other types of structured pencils.

**Key words.** Lagrangian subspace, optimal control, structure-preserving doubling algorithm, symplectic matrix, Hamiltonian matrix, matrix pencil, graph subspace

**AMS subject classifications.** 65F30, 49N10

**1. Introduction.** A *Lagrangian subspace*  $\mathcal{U}$  is an  $n$ -dimensional subspace of  $\mathbb{C}^{2n}$  such that  $u^* J v = 0$  for each  $u, v \in \mathcal{U}$ . Here  $u^*$  denotes the conjugate transpose of  $u$ , and the transpose of  $u$  in the real case, and we set

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}.$$

The computation of Lagrangian invariant subspaces of Hamiltonian matrices of the form

$$\mathcal{H} = \begin{bmatrix} F & G \\ H & -F^* \end{bmatrix}$$

with  $H = H^*, G = G^*$ , satisfying  $(HJ)^* = HJ$ , (as well as symplectic matrices  $S$ , satisfying  $S^* J S = J$ ), is an important task in many optimal control problems [16, 25, 31, 36].

Most current solution algorithms [5, 10, 31, 33] use a structured Schur form approach and represent these subspaces via orthogonal bases

$$\mathcal{U} = \operatorname{Im} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}. \quad (1.1)$$

---

\* Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, D-10623 Berlin, Fed. Rep. Germany. [mehrmann@math.tu-berlin.de](mailto:mehrmann@math.tu-berlin.de). Partially supported by DFG Research Center MATHEON ‘Mathematics for key technologies’ in Berlin.

† Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, D-10623 Berlin, Fed. Rep. Germany. [poloni@math.tu-berlin.de](mailto:poloni@math.tu-berlin.de). Supported by Alexander von Humboldt Foundation.

and then use these to compute the Hermitian solution  $X = X^* = Q_2 Q_1^{-1}$  of the associated algebraic Riccati equation,

$$0 = H + F^T X + X F - X G X.$$

The solution of the Riccati equation corresponds to the choice of representing a Lagrangian subspace as a *graph subspace* via

$$\mathcal{U} = \text{Im} \begin{bmatrix} I \\ X \end{bmatrix}. \quad (1.2)$$

However, some Lagrangian subspaces cannot be represented in this form, e.g., if in the orthogonal representation (1.1) the matrix  $Q_1$  is singular. But even if  $Q_1$  is invertible but ill-conditioned with respect to inversion, then the representation (1.2) is not a good representation. This happens e.g. in control theory for systems that are close to being uncontrollable or unstabilizable [31] or in robust control problems near the optimal solution, see e.g. [6].

A common alternative in the case that the representation (1.2) is ill-conditioned is to abandon the solution of the optimal control problem via the Riccati equation and instead use the invariant Lagrangian subspace to solve the optimality boundary value problem directly. On the other hand, in particular for large scale problems in the context of optimal control problems with constraints given by partial differential equations, the Riccati approach has some advantages, since it allows low rank approximations of the Riccati solution, and thus can be carried on with low storage requirements. Thus it is used together with low-rank variants of the Newton method [8, 35]. In this context, only the part of the solution that corresponds to large eigenvalues of the positive semidefinite solution of the Riccati equation is important, and it is usually sufficient to compute this dominant low rank component. Another advantage of the Riccati approach is that Lagrangian property of the subspace in (1.2) is equivalent to  $X = X^*$ , and thus it is easier to preserve or enforce explicitly during numerical computations. In contrast, with (1.1), the subspace is Lagrangian if and only if  $Q_1^* Q_2 = Q_2^* Q_1$ , a property which is easily lost along numerical computations and is difficult to enforce explicitly, for instance by projection. Deviation from Lagrangian property is a problem for numerical methods enforcing this representation, such as the *Laub trick* [26, 33].

Another class of methods that has recently received much attention is the class of doubling algorithms which are used in particular for discrete time problems or via a Cayley transformation also in the case of continuous time problems [1, 3, 4, 11, 12, 23, 24, 28]. They are based on a suitable representation of  $\mathcal{H}$  as a matrix pencil, and on the use of *pencil* (or *inverse-free*) *arithmetic*, [4] which is a tool to extend some basic linear algebra operations to matrix pencils.

As we see in the following, both the problems of representing a matrix  $\mathcal{H}$  with an equivalent matrix pencil and inverse-free arithmetic are intimately related with the problem of representing (Lagrangian) subspaces that we have mentioned above. Again, two main strategies exist for their implementation; we can either choose orthogonal representations along the algorithm, leading to the *inverse-free sign (and disc) method* [3, 4], or impose the presence of identities and zero blocks in specified locations, leading to the *structure-preserving doubling algorithm* [1, 11, 12, 24]. Similarly to the subspace setting, in the former case all the matrices are norm-bounded, but trouble arises from loss of structure in the pencil, while in the latter the structure is preserved exactly, but the price is the inversion of some matrices which may be ill-conditioned along the algorithm. The authors have suggested a hybrid approach in [32], which improves

slightly the performance of the structure-preserving algorithms, but still does not perform as well as the Schur form based algorithms [5, 10, 33, 31] on the harder benchmark problems.

In view of this, and in order to improve the performance of doubling algorithms, we suggest to use a modification of (1.2) as

$$\mathcal{U} = \text{Im } \Pi^T \begin{bmatrix} I \\ X \end{bmatrix}, \quad (1.3)$$

where  $\Pi$  is, up to sign changes, a permutation matrix. It is an easy corollary of a result of [15] that every Lagrangian subspace admits at least one representation as (1.3). We show that we can choose and compute such a representation in which the entries of  $X$  are bounded above by a small constant.

Introducing this approach in doubling algorithms shows that already the most simple doubling algorithm enhanced with this representation can reach full machine accuracy on a wide range of problems, obtaining invariant subspaces of the same quality as that for the Schur type methods.

In the following, we denote by  $e_k$  the  $k$ -th column of the identity matrix and with  $0$  and  $e$  the vectors whose elements are all zeros and all ones, respectively. The sizes of said vectors can usually be inferred by the context, and are specified explicitly when needed. We denote by  $M_{i,:}$  the  $i$ -th row of a matrix  $M$  and by  $M_{:,j}$  its  $j$ -th column.

The paper is organized as follows. In Section 2 we introduce some of the basic concepts. In Section 3 we describe how to obtain theoretically a bounded representation of Lagrangian subspaces, while in Section 4 we describe how to compute it numerically in practice. In Sections 5 and 6 we apply this result to the representation of structured matrix pencils and to doubling algorithms, respectively. In Section 7 we discuss the convergence and numerical stability of this approach, and in Section 8 we test it with several numerical experiments. Finally, some conclusions and open problems are presented in Section 9.

**2. Permutations, Plücker coordinates and minors.** In this section we introduce some of the basic concepts that are needed to develop our new approach.

Let  $U \in \mathbb{C}^{m,n}$ , with  $m \geq n$  and let  $\Pi$  be a permutation matrix, then we define  $Y^\Pi \in \mathbb{C}^{n,n}$ ,  $Z^\Pi \in \mathbb{C}^{m-m,n}$ , and, whenever  $Y^\Pi$  is nonsingular,  $X^\Pi = [x_{i,j}^\Pi] \in \mathbb{C}^{n-m,n}$ , as

$$\Pi U = \begin{bmatrix} Y^\Pi \\ Z^\Pi \end{bmatrix}, \quad X^\Pi = Z^\Pi (Y^\Pi)^{-1}. \quad (2.1)$$

We then have the following characterization for the minors of  $X^\Pi$ .

**LEMMA 2.1.** *Let  $U \in \mathbb{C}^{m,n}$  ( $m \geq n$ ) and let  $\Pi$  be a permutation such that  $Y^\Pi$  is nonsingular. Let  $X_{I,J}^\Pi$  be the square submatrix of  $X^\Pi$  corresponding to rows  $I = (i_1, i_2, \dots, i_k)$  and columns  $J = (j_1, j_2, \dots, j_k)$ . Then,  $\det X_{I,J}^\Pi = \pm \det Y^{\Pi' \Pi} / \det Y^\Pi$ , where  $\Pi'$  is any permutation satisfying*

$$\Pi'(j) = \begin{cases} n + i_\ell & \text{if } j = j_\ell \text{ for some } \ell, \\ j & \text{otherwise,} \end{cases}$$

for  $j = 1, \dots, n$ .

*Proof.* Due to the choice of  $\Pi'$ , we have

$$(Y^{\Pi' \Pi} (Y^\Pi)^{-1})_{j,:} = \begin{cases} X_{i,:} & \text{if } j = j_l \text{ for some } \ell, \\ e_j^T & \text{otherwise.} \end{cases}$$

Therefore,  $\det Y^{\Pi' \Pi} / \det Y^{\Pi} = \det Y^{\Pi' \Pi} (Y^{\Pi})^{-1} = \det X_{I,J}^{\Pi}$ .  $\square$

The quantities  $\det Y^{\Pi}$  enjoy the following properties.

**THEOREM 2.2.** *Let  $U \in \mathbb{C}^{m,n}$ , ( $m \geq n$ ) have linearly independent columns. Then,*

1.  $\det Y^{\Pi} \neq 0$  for at least one permutation  $\Pi$ .
2. If we replace  $U$  by  $UQ$  with a nonsingular matrix  $Q \in \mathbb{C}^{n,n}$ , then for all permutations  $\Pi$ , the values of  $\det Y^{\Pi}$  are multiplied by a common factor  $\det Q$ .
3. The values of  $\det Y^{\Pi}$  for all possible  $\Pi$  uniquely characterize the subspace  $\text{Im } U$ .

*Proof.* 1. Since  $U$  has full column rank, there must be at least one nonzero  $n \times n$  minor.

2. The claim follows from

$$\Pi U Q = \begin{bmatrix} Y^{\Pi} Q \\ Z^{\Pi} Q \end{bmatrix}.$$

3. Choose a permutation  $\Pi$  such that  $\det Y^{\Pi} \neq 0$ . Then all entries of  $X^{\Pi}$  are uniquely determined, as  $(X^{\Pi})_{i,j} = \det X_{(i),(j)}^{\Pi}$  by Lemma 2.1. Thus,

$$U = \Pi^T \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix} Y^{\Pi}, \quad \text{Im } U = \text{Im } \Pi^T \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix}.$$

$\square$  Up to row reordering, there are only  $\binom{m}{n}$  possible choices of  $Y^{\Pi}$ , corresponding to the possible subsets of  $n$  elements out of  $m$ . Their determinants form a set of projective coordinates for the subspace  $\text{Im } U$ , that is known in projective geometry as *Plücker coordinates* [20]. Note that a canonical row ordering is needed to obtain a well-defined set of Plücker coordinates, and that different such orderings differ only by a change of sign.

While Theorem 2.2 is a classical result in algebraic geometry [20], the following result is not typically of interest in that field, although it is crucial here.

**THEOREM 2.3.** *Let  $U \in \mathbb{C}^{m,n}$  ( $m \geq n$ ) have full column rank. Then there exists a permutation matrix  $\Pi$  such that  $Y^{\Pi}$  (as in (2.1)) is nonsingular and we have  $|x_{i,j}^{\Pi}| \leq 1$ .*

*Proof.* From part 1. of Theorem 2.2, it follows that  $|\det Y^{\Pi}| \neq 0$  for at least one  $\Pi$ . Choose any permutation  $\Pi$  for which  $|\det Y^{\Pi}|$  is maximal. Then, by Lemma 2.1,  $|x_{i,j}^{\Pi}| = |\det Y^{\Pi' \Pi}| / |\det Y^{\Pi}| \leq 1$ .  $\square$  This result can be recast in the context of representations of subspaces in the following way.

**COROLLARY 2.4.** *Let  $\mathcal{U}$  be an  $n$ -dimensional subspace of  $\mathbb{C}^{m+n}$ . Then, there exists a permutation matrix  $\Pi$ , and a square matrix  $X^{\Pi}$  such that*

$$\mathcal{U} = \text{Im } \Pi^T \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix} \quad (2.2)$$

where the entries of  $X^{\Pi}$  satisfy  $|x_{i,j}^{\Pi}| \leq 1$ . It follows that a subspace can be represented with a basis that has an identity in selected rows and norm-bounded (by 1) entries in the remaining ones. We call such a form a *permuted graph representation*.

**3. Permuted graph representations of Lagrangian subspaces.** In this section we adapt the ideas of the previous section to obtain norm-bounded structure-preserving representations of Lagrangian subspaces.

Let  $\mathcal{I}^n := \{0, 1\}^n$ . For each  $v \in \mathcal{I}^n$ , we define a *symplectic swap matrix* as an orthogonal symplectic matrix given by

$$\Pi_v := \begin{bmatrix} \text{diag}(\hat{v}) & \text{diag}(v) \\ -\text{diag}(v) & \text{diag}(\hat{v}) \end{bmatrix},$$

where  $\hat{v}$  is the vector with  $\hat{v}_i = 1 - v_i$ .

Multiplication with the matrices  $\Pi_v$  permutes (up to a sign) the entries of a vector, with the limitation that the  $i$ -th row of a vector may only be exchanged with the  $n + i$ -th. Notice that  $J = \Pi_e$  with  $e = [1 \ 1 \ \cdots \ 1]^T$ . We denote by  $S^{2n}$  the set of all  $2^n$  symplectic swap matrices of size  $2n$ . For  $U \in \mathbb{C}^{2n,n}$  and  $\Pi \in S^{2n}$ , we define  $Y^\Pi, Z^\Pi$  and (whenever  $Y^\Pi$  is nonsingular)  $X^\Pi$  by the formulas in (2.1).

In the following we will make frequent use of the following result.

**THEOREM 3.1** ([15]). *If the columns of  $U \in \mathbb{C}^{2n,n}$  span a Lagrangian subspace, then there exists  $\Pi \in S^{2n}$  such that  $Y^\Pi$  is nonsingular.*

In fact, in [15] an equivalent result is shown in the context of symplectic matrices; the version presented here follows from their result after observing that the first  $n$  columns of a symplectic matrix span a Lagrangian subspace.

**LEMMA 3.2.** *For  $U \in \mathbb{C}^{2n,n}$  the following are equivalent.*

1.  $\text{Im } U$  is Lagrangian;
2. there exists at least one  $\Pi \in S^{2n}$  such that  $Y^\Pi$  is nonsingular and  $X^\Pi$  is Hermitian;
3. there exists at least one  $\Pi \in S^{2n}$  such that  $Y^\Pi$  is nonsingular, and, for all such choices of  $\Pi$ ,  $X^\Pi$  is Hermitian.

*Proof.* We prove a closed chain of implications.

1.  $\Rightarrow$  3. This follows directly from Theorem 3.1.

3.  $\Rightarrow$  2. is obvious.

2.  $\Rightarrow$  1. If  $\text{Im} \begin{bmatrix} I \\ X^\Pi \end{bmatrix}$  is Lagrangian and  $\Pi^T$  is symplectic, then  $\text{Im } U = \text{Im } \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix}$  is Lagrangian as well.  $\square$  Therefore, every Lagrangian subspace admits at least one representation as  $\text{Im } \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix}$ . We call the pair  $(X^\Pi, \Pi)$  a *permuted Lagrangian graph representation*. Note that in [15] a related object was called complementary basis representation.

The  $2^n$  injective maps

$$f_\Pi : X \mapsto \Pi^T \begin{bmatrix} I \\ X \end{bmatrix}$$

form an atlas for the *Lagrangian Grassmannian*, i.e., the variety of Lagrangian subspaces, and are a means to obtain a structure-preserving parametrization of these subspaces.

A result similar to Lemma 2.1 holds for symplectic swap matrices, with an important restriction on the allowed index sets,  $I = J$ .

**LEMMA 3.3.** *Let  $U$  be given, and  $\Pi \in S^{2n}$ , constructed from a vector  $v \in \mathcal{I}^n$ , such that  $Y^\Pi$  is nonsingular. Let  $X_{I,I}^\Pi$  be the principal submatrix of  $X^\Pi$  corresponding to rows and columns  $I = (i_1, i_2, \dots, i_k)$ . Let  $\Pi' \in S^{2n}$  be constructed from a vector  $v'$  that differs from  $v$  only in positions  $i_1, i_2, \dots, i_k$ . Then,  $\det X_{I,I}^\Pi = \pm \det Y^{\Pi'} / \det Y^\Pi$ .*

*Proof.* Due to the choice of  $\Pi'$ , we have

$$(Y^{\Pi'} (Y^\Pi)^{-1})_{i,:} = \begin{cases} \pm X_{i,:} & \text{if } i = i_\ell \text{ for some } \ell, \\ e_i^T & \text{otherwise.} \end{cases}$$

Therefore,  $\det Y^{\Pi'} / \det Y^{\Pi} = \pm \det Y^{\Pi'} (Y^{\Pi})^{-1} = \pm \det X_{I,I}^{\Pi}$ .  $\square$

With these preliminaries we are able to obtain a bound on the elements of a particular  $X^{\Pi}$ .

**THEOREM 3.4.** *For every Lagrangian subspace  $\mathcal{U} = \text{Im } U$ , there exists  $\Pi \in S^{2n}$  such that  $Y^{\Pi}$  is nonsingular and*

$$|x_{i,j}^{\Pi}| \leq \begin{cases} 1 & \text{if } i = j, \\ \sqrt{2} & \text{if } i \neq j \end{cases}. \quad (3.1)$$

*Proof.* Choose  $\Pi \in S^{2n}$  such that  $|\det Y^{\Pi}|$  is maximal then this determinant is nonzero, because of Theorem 3.1. For diagonal entries, we have directly from Lemma 3.3 that

$$|x_{i,i}^{\Pi}| = \pm \det Y^{\Pi'} / \det Y^{\Pi} \leq 1.$$

For off-diagonal entries, we obtain

$$\left| \det \begin{bmatrix} x_{i,i}^{\Pi} & x_{i,j}^{\Pi} \\ x_{j,i}^{\Pi} & x_{j,j}^{\Pi} \end{bmatrix} \right| = \pm \det Y^{\Pi'} / \det Y^{\Pi} \leq 1.$$

Using the triangle inequality, we then obtain  $|x_{i,j}^{\Pi}|^2 = |x_{i,j}^{\Pi} x_{j,i}^{\Pi}| \leq 1 + |x_{i,i}^{\Pi}| |x_{j,j}^{\Pi}| \leq 2$ .  $\square$

The bound (3.1) is sharp, as shown by the Lagrangian subspace spanned by the columns of

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix}.$$

#### 4. Numerical computation of bounded permuted graph representations.

In this section we discuss the numerical computation of bounded permuted graph representations of both Lagrangian and generic subspaces.

While it is not apparent how to compute a representation that maximizes  $|\det Y^{\Pi}|$ , there is a simple descent algorithm that guarantees the weaker condition  $|x_{i,j}^{\Pi}| \leq 1$  for all  $i, j$ .

The underlying idea is that, whenever we have an entry  $x_{i,j}^{\Pi}$  of modulus larger than 1, this means that there exists a permutation  $\Pi'$  such that  $|\det Y^{\Pi' \Pi}| \geq |\det Y^{\Pi}|$ , and thus we can update  $\Pi$  to  $\Pi' \Pi$ . We present the unstructured version as Algorithm 1 and then discuss the changes that are needed to perform the algorithm in the Lagrangian case.

Algorithm 1 has the following properties.

- The value of  $|\det Y^{\Pi}|$  increases monotonically during the iteration, thus the procedure always terminates, in exact arithmetic, after going through all  $n!$  permutations. With the proposed heuristic of always picking the entry of  $X^{\Pi}$  with largest modulus, in practice we observe that only a moderate number of steps is needed.
- When a subspace representation is computed in a converging iterative process, i.e., if  $U$  is updated during an iteration with a small modification, then the algorithm can be warm-started with the previous value of  $\Pi$ , thus greatly reducing the number of required steps.

---

**Algorithm 1:** Computation of a permuted graph representation satisfying  $|x_{i,j}^\Pi| < T$

---

**Input:**  $U \in \mathbb{C}^{n+m,n}$ , optionally an initial  $\Pi$  such that  $\det Y^\Pi \neq 0$ , and a threshold value  $T \geq 1$ .

**Output:** A permuted graph representation  $(\Pi, X^\Pi)$  with  $|x_{i,j}^\Pi| \leq T$  for all  $i, j$ .  
 If an initial  $\Pi$  is not supplied, choose an initial  $\Pi$  such that  $Y^\Pi$  is not exactly singular (e.g., by computing a QR decomposition with column pivoting,  $U^T = QRP$  and use the permutation  $\Pi = P$ );  
 Compute  $X^\Pi$ ;

**repeat**

  let  $M = \max |x_{i,j}^\Pi|$ , attained for  $i = \hat{i}, j = \hat{j}$ ;

**if**  $M > T$  **then**

    update the permutation  $\Pi$  by swapping  $\Pi(n + \hat{i})$  and  $\Pi(\hat{j})$ ;

    update  $X^\Pi$  (see Algorithm 2 below);

**end**

**until**  $M \leq T$  ;

---

- The algorithm may also be used with a maximum threshold  $T > 1$  for the modulus of the entries of  $X^\Pi$ , thus requiring less steps. In order to minimize the impact of numerical errors, we recommend using at least  $T = 1 + \sqrt{\mathbf{u}}$ , where  $\mathbf{u}$  is the machine precision.
- If  $X^\Pi$  has to be updated after a change of the permutation  $\Pi$  which involves only a small number of entries, then this can be done using the Sherman-Morrison updating formula [18], since the new  $Y^\Pi$  differs from the identity only by a small number of rows. Indeed, we change only one row at a time in Algorithm 1, but we present here an update formula in higher generality to allow an easier generalization to the Lagrangian case and to blocked versions. Namely, when entries  $(n + i_1, \dots, n + i_k)$  are exchanged with entries  $J = (j_1, \dots, j_k)$ , then we update  $X$  via

$$\begin{aligned} X &\leftarrow (X - e_I(e_I^T X - e_J^T))(I - e_J(e_J^T - e_I^T X))^{-1} \\ &= X + (e_I + X e_J)(e_I^T X e_J)^{-1}(e_J^T - e_I^T X), \end{aligned}$$

where  $e_I$  is obtained by stacking horizontally the columns  $e_i$  for each  $i \in I$ , and analogously for  $e_J$ . However, this formula may lead to numerical instabilities when the pivot-block  $X_{I,J}$  has an inverse with large entries; a better update strategy can be obtained by expanding four different versions of the formula separately, according to whether the row (resp. column) indices belong to  $I$  (resp.  $J$ ) or not. We describe this procedure in Algorithm 2.

- To avoid accumulation of errors along successive low-rank corrections, it is advisable to recompute  $X^\Pi$  from scratch from the original entries of  $U$  at the end of the procedure and, optionally, after a fixed number of steps.
- We expect the algorithm to be robust even when started with an ill-conditioned  $Y^\Pi$ , since we only need to estimate correctly the magnitude of the elements in order to get a good  $\Pi$ . In this case, after obtaining  $\Pi$ , the exact value of  $X^\Pi$  can be recomputed from the original  $U$ , as suggested in the previous point.
- The procedure in Algorithm 1 resembles the basic “complementary tableaux” implementation of the simplex method [13].

---

**Algorithm 2:** Updating a permuted graph representation  $X^\Pi$ 

---

**Input:**  $X = X^\Pi$  and index sets  $I = (i_1, \dots, i_k)$ ,  $J = (j_1, \dots, j_k)$  such that  $\det X_{I,J}^\Pi$  is large

**Output:**  $X = X^{\Pi' \Pi}$ , where  $\Pi'$  swaps all indices  $n + i_k$  with  $j_k$

. Let  $\bar{I}$ ,  $\bar{J}$  denote the entries not appearing in  $I$  and  $J$  respectively;

$S \leftarrow X_{I,J}^{-1}$ ;

$X_{\bar{I},\bar{J}} \leftarrow X_{\bar{I},\bar{J}} - X_{\bar{I},J} S X_{I,\bar{J}}$ ;

$X_{\bar{I},J} \leftarrow X_{\bar{I},J} S$ ;

$X_{I,\bar{J}} \leftarrow -S X_{I,\bar{J}}$ ;

$X_{I,J} \leftarrow S$ ;

---

When the subspace is Lagrangian, a modification of Algorithm 1 can be used to obtain a bounded permuted Lagrangian graph representation. This procedure is given in Algorithm 3.

---

**Algorithm 3:** Computation of a permuted Lagrangian graph representation of a Lagrangian subspace satisfying (3.1)

---

**Input:**  $U \in \mathbb{C}^{2n,n}$  of full rank, spanning a Lagrangian subspace, and optionally an initial  $\Pi_v \in \mathbb{S}^{2n}$  such that  $\det Y^\Pi \neq 0$  and two thresholds  $S \geq 1$ ,  $T \geq \sqrt{1 + S^2}$

. **Output:** A permuted Lagrangian graph representation  $(\Pi, X^\Pi)$  satisfying (3.1)

. If an initial  $\Pi$  is not supplied, choose an initial  $\Pi \in \mathbb{S}^{2n}$  such that  $Y^\Pi$  is not exactly singular (e.g., compute a QR decomposition with column pivoting  $U^T = QRP$  and take  $v_i = 1$  if  $n + i$  comes first than  $i$  in the permutation  $P$ ); compute  $X^\Pi$ ;

**repeat**

  let  $M = \max |x_{i,j}^\Pi|$ ,  $i \neq j$ , attained for  $i = \hat{i}$ ,  $j = \hat{j}$ ;

  let  $N = \max |x_{k,k}^\Pi|$ , attained for  $k = \hat{k}$ ;

**if**  $N > S$  **then**

    update the swap matrix  $\Pi = \Pi_v$  with  $v_{\hat{k}} \leftarrow 1 - v_{\hat{k}}$ ;

**else if**  $M > T$  **then**

    update the swap matrix  $\Pi = \Pi_v$  with  $v_{\hat{i}} \leftarrow 1 - v_{\hat{i}}$  and  $v_{\hat{j}} \leftarrow 1 - v_{\hat{j}}$ ;

**end**

  update  $X^\Pi$ ;

**until**  $M \leq T$ ,  $N \leq S$  ;

---

Algorithm 3 has the following properties

- If  $U$  spans a Lagrangian subspace, then the first computed value of  $X^\Pi$  should be Hermitian; this property can fail only due to numerical errors in the given  $U$  or in the computation, so we can safely enforce it by projecting it on the nearest Hermitian matrix with  $X \leftarrow \frac{X + X^*}{2}$ .
- After computing the first permuted graph representation, all subsequent updates can be performed using a variant of Algorithm 2, since swap matrices differ from permutations only by a change of sign. The updates can be performed using operations that preserve the Hermitian property of  $X^\Pi$



exactly.

- It is important to check the optimality on diagonal elements first: We need to have  $|x_{k,k}^H| \leq S$  for each  $k$ , in order to prove that for any  $i, j$  such that  $|x_{i,j}^H| > T$  we have  $|\det X_{(i,j)(i,j)}| = |x_{i,i}^H x_{j,j}^H - x_{i,j}^H x_{j,i}^H| > T^2 - S^2 \geq 1$ , and thus the swapping operation increases the value of  $Y^H$ . This is in turn needed to prove the termination of the algorithm.

**5. Permuted graph representations of matrix pencils.** In the context of eigenvalues and invariant subspace computation, matrix pencils are usually considered up to *right equivalence*, i.e., up to the equivalence relation defined by

$$sE_1 - A_1 \equiv sE_2 - A_2$$

with  $E_1 = ME_2$ ,  $A_1 = MA_2$  for a nonsingular square matrix  $M$ . We may interpret this equivalence in terms of subspaces, by saying that we are not interested in the matrix  $\begin{bmatrix} E^T \\ A^T \end{bmatrix}$ , but rather in the subspace  $\text{Im} \begin{bmatrix} E^T \\ A^T \end{bmatrix}$ . Therefore, our results on the representation of subspaces may be adapted to representations of pencils up to right equivalence. Our main motivation stems from the representation of *regular symplectic pencils*, i.e., regular pencils  $sE - A$  satisfying  $EJE^* = AJA^*$  for which we have the following theorem.

**THEOREM 5.1.** *Let  $sE - A$ , with  $E, A \in \mathbb{C}^{2n, 2n}$ , be a regular symplectic pencil. Then, there exist  $\Pi_1, \Pi_2 \in S^{2n}$  such that*

$$sE - A \equiv s \begin{bmatrix} I & X_{11} \\ 0 & X_{21} \end{bmatrix} \Pi_1 - \begin{bmatrix} X_{12} & 0 \\ X_{22} & I \end{bmatrix} \Pi_2^T, \quad (5.1)$$

where

$$X^H = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$$

is Hermitian and satisfies (3.1).

*Proof.* Partitioning the pencil as  $E = \begin{bmatrix} E_1 & E_2 \end{bmatrix}$ ,  $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ , where all blocks are  $2n \times n$ , we can rewrite the condition  $EJE^* = AJA^*$  as

$$\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} E_1^* \\ A_2^* \\ E_2^* \\ A_1^* \end{bmatrix} = 0, \quad (5.2)$$

Moreover,  $\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix}$  is of full row rank, since otherwise one could find a vector  $w$  such that  $\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix}^* w = 0$ , i.e.,  $w^* E = w^* A = 0$ , which contradicts the regularity assumption.

Therefore, the columns of  $\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix}^*$  span a Lagrangian subspace of  $\mathbb{C}^{4n}$ , and Theorem 3.4 gives us a permuted graph representation

$$\Pi_v \begin{bmatrix} E_1^* \\ A_2^* \\ E_2^* \\ A_1^* \end{bmatrix} = \begin{bmatrix} I \\ X^H \end{bmatrix} (Y^H)^{-1}.$$

Note that  $\Pi_v$  acts separately on the blocks columns (1, 3) and (2, 4); these actions are given by  $\Pi_1 = \Pi_{v_1}$  and  $\Pi_2 = \Pi_{v_2}$ , where  $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ . After reshuffling the blocks, we obtain (5.1).  $\square$  The representation (5.1) with  $\Pi_1 = \Pi_2 = I$  is well-known, see e.g. [30, 31, 34], where the representation

$$sE - A = s \begin{bmatrix} I & G \\ 0 & F^* \end{bmatrix} - \begin{bmatrix} F & 0 \\ H & I \end{bmatrix}$$

with  $F = F^*$ ,  $G = G^*$  is used, but without the permutations the boundedness of the matrices cannot be guaranteed and this may lead to ill-conditioning in numerical methods.

Similarly for *Hamiltonian pencils*, i.e., pencils satisfying  $EJA^* + AJE^* = 0$ , or, equivalently,

$$\begin{bmatrix} E_1 & E_2 & A_2 & -A_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} E_1^* \\ E_2^* \\ A_2^* \\ -A_1^* \end{bmatrix} = 0 \quad (5.3)$$

we have that  $\text{Im} \begin{bmatrix} E_1 & E_2 & A_2 & -A_1 \end{bmatrix}^*$  is Lagrangian and  $sE - A$  is equivalent to  $s\tilde{E} - \tilde{A}$  with

$$\begin{bmatrix} \tilde{E}_1 & \tilde{A}_2 \end{bmatrix} = \begin{bmatrix} I & X_{11} \\ 0 & X_{12} \end{bmatrix} \Pi_1, \begin{bmatrix} -\tilde{A}_1 & \tilde{E}_2 \end{bmatrix} = \begin{bmatrix} X_{21} & 0 \\ X_{22} & I \end{bmatrix} \Pi_2, X^H = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix},$$

where  $X^H$  is Hermitian and elementwise bounded as in (3.1). Again, the case  $\Pi = I$  gives the well-known representation  $sI - \mathcal{H}$ , where  $\mathcal{H}J$  is Hermitian (i.e.,  $\mathcal{H}$  is a *Hamiltonian matrix*).

**6. Permuted graph representations and doubling algorithms.** In this section we discuss doubling algorithms for the computation of the stable deflating subspace of a symplectic pencil. These methods are based on the following result.

**THEOREM 6.1 ([3]).** *Let  $sE - A$  with  $E, A \in \mathbb{C}^{n,n}$  be a regular pencil, and let  $\tilde{E}, \tilde{A} \in \mathbb{C}^{n,n}$  be such that*

$$\tilde{E}A = \tilde{A}E, \quad \text{Rank} \begin{bmatrix} \tilde{E} & \tilde{A} \end{bmatrix} = n. \quad (6.1)$$

*Then, the pencil  $s\tilde{E}\tilde{E} - \tilde{A}\tilde{A}$  has the same deflating subspaces as the original pencil, and its eigenvalues are the squares of the corresponding eigenvalues.* If in Theorem 6.1 the matrices  $E$  and  $\tilde{E}$  are invertible, this result is simple, as  $(E^{-1}A)^2 = (\tilde{E}E)^{-1}(\tilde{A}A)$ . However, it provides an extension of the squaring operation to matrix pencils that is well-defined and can be applied also when  $E$  is singular or ill-conditioned. By iterating the above transformation and scaling to avoid element growth, the eigenvalues of the pencil are squared at each iteration and thus the stable eigenvalues converge to 0 and the unstable ones to  $\infty$ . After a sufficient (not too large) number of steps, it is easy to recover the stable and unstable invariant subspace as kernels of the two coefficients of the pencil.

The inverse-free disc method [3] performs this doubling iteration by choosing  $\begin{bmatrix} \tilde{E} & \tilde{A} \end{bmatrix}^T$  with orthonormal rows; namely, it computes a QR decomposition

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} A \\ E \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (6.2)$$

and takes  $\tilde{E} = -Q_{21}$ ,  $\tilde{A} = Q_{22}$ . The Lagrangian property of the resulting subspace is not enforced and may be lost in finite precision arithmetic during the iteration. In other words, the algorithm is not structure-preserving with respect to the Lagrangian structure. The structure-preserving doubling algorithm (SDA) [11] is based instead on the version  $\Pi = I$  of the representation (5.1). At each step, the method uses a pencil of the form

$$E = \begin{bmatrix} I & X_{11} \\ 0 & X_{21} \end{bmatrix}, A = \begin{bmatrix} X_{12} & 0 \\ X_{22} & I \end{bmatrix}. \quad (6.3)$$

and chooses  $\tilde{E}$  and  $\tilde{A}$  having blocks  $I$  and  $0$  in the same position, and thus this structure is maintained in the products  $\tilde{E}E$  and  $\tilde{A}A$ . The resulting pencil is then symplectic if and only if the matrix  $X$  is Hermitian, and this can be easily enforced at every step. Matrices  $\tilde{E}$  and  $\tilde{A}$  with the required block structure can be found by inverting a suitable matrix, which is often well-conditioned but may approach singularity in some cases [23]. As an additional advantage of having these prescribed identity blocks, these methods have a lower computational cost than the ones in the inverse-free methods, as the latter require building and factorizing a  $4n \times 4n$  matrix rather than working directly with its  $n \times n$  blocks.

A doubling variant that enforces a hybrid representation is presented in [32], in order to deal with the cases in which the representation (6.3) is a poor choice. A block structure similar to (6.3) is used, but the identities are replaced by general matrices in order to maintain orthonormal bases for the first block row of  $E$  and the second of  $A$ . The algorithm works better than the classical SDA for those problems in which the representation (1.2) is a poor choice, but this new variant is not structure-preserving and still needs the inversion of a matrix at each step that may be ill-conditioned.

In view of these observations it seems natural to study the combination of the ideas in the structure-preserving doubling algorithm with the idea of enforcing a bounded permuted graph representation at every step to achieve added stability. Given a pencil  $sE - A$  and a bounded permuted graph representation

$$\begin{bmatrix} A \\ E \end{bmatrix} = \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix},$$

we can compute  $\tilde{E}, \tilde{A}$  with bounded entries thanks to the relation

$$([-X^\Pi \quad I] \Pi) \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix} = 0. \quad (6.4)$$

The resulting complete procedure is presented in Algorithm 4.

Note that Algorithm 4 in this form is still unsatisfactory, because it does not manage to go from a permuted graph representation matrix  $X^\Pi$  for  $sE - A$  to one for  $s\tilde{E}E - \tilde{A}A$  using only matrix operations that map exactly between Hermitian matrices, but one has to enforce the Hermitian property explicitly in the last instruction in the **while** cycle. The task of finding a symmetry-preserving version of the update formulas is currently under our investigation. Nevertheless, we show below that this preliminary version gives very good computational results.

**7. Convergence and stability issues.** From our derivation it is not clear at all that doubling algorithms converge when eigenvalues on the unit circle are present. A positive answer to this question was first given in [23], and the proof was later adapted to different types of doubling algorithms [9, 32]. We use the same technique

---

**Algorithm 4:** Structure-preserving doubling algorithm with bounded permuted graph representation

---

**Input:** A starting symplectic pencil  $sE - A$ .

**Output:** A basis for its stable invariant subspace.

Compute an optimal  $(\Pi, X^\Pi)$  for the subspace  $\begin{bmatrix} A \\ E \end{bmatrix}$  using Algorithm 1 (from the second step on, we warm-start with the  $\Pi$  from the previous iteration);

**while** a suitable stopping criterion is not satisfied **do**

    Compute  $\begin{bmatrix} \tilde{A} & -\tilde{E} \end{bmatrix} = [-X^\Pi & I] \Pi$ ;

    Form the products  $\tilde{E}E, \tilde{A}A$ ;

    Compute an optimal  $(\Pi, X^\Pi)$  for the symplectic pencil  $sE - A$  with Algorithm 3 (warm-started);

    Symmetrize  $X \leftarrow \frac{X+X^*}{2}$  to reduce the impact of numerical errors;

**end**

**return**  $U = \ker \begin{bmatrix} X_{22} & I \end{bmatrix} \Pi_2^T = \Pi_2 \begin{bmatrix} I \\ -X_{22} \end{bmatrix}$ ;

---

here, based on the Kronecker canonical form [17], to prove convergence of this new doubling variant; note that the proof is easier in our setting since we do not have to worry about boundedness, and that we need no nonsingularity assumption.

First of all, we introduce some notation. Let  $A_0 - sE_0$  be a regular  $2n \times 2n$  matrix pencil, and denote its Kronecker chains with  $(w_1^i, w_2^i, \dots, w_{k_i}^i)$ , and the associated eigenvalues with  $\lambda_i$  (here  $w_1^i$  are the eigenvectors; the  $\lambda_i$  are possibly infinite, and may be repeated if there are multiple chains with the same eigenvalue, and  $\sum_i k_i = 2n$ ). Let  $\mathcal{S} = \{w_j^i : |\lambda_i| < 1\}$ ,  $\mathcal{U} = \{w_j^i : |\lambda_i| > 1\}$ ,  $\mathcal{C}_1 = \{w_j^i : |\lambda_i| = 1, j \leq k_1/2\}$ ,  $\mathcal{C}_2 = \{w_j^i : |\lambda_i| = 1, j > k_1/2\}$ . Notice that  $|\mathcal{S}| + |\mathcal{U}| + |\mathcal{C}_1| + |\mathcal{C}_2| = 2n$  (where  $|\mathcal{X}|$  denotes the cardinality of a set  $\mathcal{X}$ ), and in fact their union is a basis of  $\mathbb{C}^{2n}$  composed of Kronecker chains.

With this notation in effect, we have the following convergence theorem.

**THEOREM 7.1.** *Let  $A_0 - sE_0$  be a regular  $2n \times 2n$  matrix pencil, such that for each  $i$  such that  $|\lambda_i| = 1$ ,  $k_i$  is even and  $|\mathcal{S}| + |\mathcal{C}_1| = n$ .*

*Let  $A_{k+1}, E_{k+1}$  be the sequence of matrix pencils generated by Algorithm 4. Then,*

$$\Pi_2 \begin{bmatrix} I \\ -X_{2,2} \end{bmatrix}, \Pi_1^T \begin{bmatrix} -X_{1,1} \\ I \end{bmatrix}$$

*converge to  $\text{span } S \cup C_1$  and  $\text{span } U \cup C_1$  respectively. The convergence is quadratic with rate  $l_{\max}/l_{\min}$ , where*

$$l_{\max} := \max_{|\lambda_i| < 1} |\lambda_i|, \quad l_{\min} := \min_{|\lambda_i| > 1} |\lambda_i|,$$

*if  $C_1$  (and thus  $C_2$ ) is empty, and linear with rate 1/2 otherwise.*

*Proof.* After some easy manipulations of the Kronecker form, we obtain

$$W(A_1 - sE_1)Z = \begin{bmatrix} J_S & & & \\ & J_{C_1} & H & \\ & & J_{C_1} & \\ & & & I \end{bmatrix} - s \begin{bmatrix} I & & & \\ & I & & \\ & & I & \\ & & & J_U \end{bmatrix}, \quad Z = [S \quad C_1 \quad C_2 \quad U]$$

where the columns of  $S, C_1, C_2, U$  are the elements of the corresponding sets  $\mathcal{S}, \mathcal{C}_1, \mathcal{C}_2, \mathcal{U}$ , and  $J_S - sI, J_{C_1} - sI, I - sJ_U$  are the Kronecker forms of the pencil restricted to the corresponding spaces. As in [23], from this equality we obtain

$$A_k Z \begin{bmatrix} I & & & \\ & I & & \\ & & I & \\ & & & J_U^{2^k} \end{bmatrix} = E_k Z \begin{bmatrix} J_S & & & \\ & J_{C_1} & H & \\ & & J_{C_1} & \\ & & & I \end{bmatrix}^{2^k} = E_k Z \begin{bmatrix} J_S^{2^k} & & & \\ & J_{C_1}^{2^k} & H_k & \\ & & J_{C_1}^{2^k} & \\ & & & I \end{bmatrix},$$

where  $H_k$  is defined so that the rightmost equality holds. Clearly,  $J_S^{2^k} = \mathcal{O}(s^{2^k})$ ,  $J_U^{2^k} = \mathcal{O}(u^{-2^k})$ , and it is proved in [23, Lemma 4.4] that  $H_k$  is invertible for sufficiently large  $k$ , and  $H_k^{-1} J_{C_1}^{2^k} = \mathcal{O}(2^{-k})$ ,  $J_{C_1}^{2^k} H_k^{-1} J_{C_1}^{2^k} = \mathcal{O}(2^{-k})$ . Multiplying both sides with

$$\begin{bmatrix} I & & \\ & I & \\ & & -H_k^{-1} J_{C_1} \\ I & & \end{bmatrix}$$

from the right, we obtain

$$A_k Z \left( \begin{bmatrix} I & & \\ & I & \\ 0 & & \\ & & 0 \end{bmatrix} + \mathcal{O}(2^{-k}) \right) = E_k Z \mathcal{O}(2^{-k}).$$

Thus, using the definition of  $Z$  and boundedness of  $A_k$  and  $E_k$ ,

$$\mathcal{O}(2^{-k}) = A_k \begin{bmatrix} S & C_1 \end{bmatrix} = \begin{bmatrix} X_{12} \\ X_{22} & I \end{bmatrix} \Pi_2^T \begin{bmatrix} S & C_1 \end{bmatrix},$$

from which we see that  $\Pi_2 \begin{bmatrix} I \\ -X_{22} \end{bmatrix}$  converges to a permuted graph representation of  $\begin{bmatrix} S & C_1 \end{bmatrix}$ . The analogous result for the semi-unstable subspace follows with a similar argument by considering a Kronecker canonical form with  $\tilde{Z} = \begin{bmatrix} U & C_1 & C_2 & S \end{bmatrix}$ .  $\square$

To examine the stability and conditioning, for a given matrix  $U$ , we define its condition number  $\kappa(U) = \sigma_{\min}(U)^{-1} \sigma_{\max}(U)$ , where  $\sigma_{\min}(U)$  and  $\sigma_{\max}(U)$  denote, respectively, the smallest and largest singular value. This condition number can be regarded as a measure of how good  $U$  is as a representation of its column space. This quantity plays a central role when computing projectors, for which we need to form  $(U^T U)^{-1}$ , and when extracting an orthonormal basis, as the condition number of the  $Q$  factor in the QR factorization of  $U$  depends on it [21]. In this sense, we show that, when  $X^\Pi$  has bounded entries, a permuted graph representation is a good representation of the subspace, and it can be computed stably from another given good representation.

**THEOREM 7.2.** *Let  $\Pi^T \begin{bmatrix} I \\ X \end{bmatrix} = U \in \mathbb{C}^{m+n,n}$ , where  $|X_{ij}^\Pi| \leq T$  for each  $i, j$ . Then,  $\kappa(U) \leq \sqrt{mnT^2 + 1}$ .*

*Proof.* We have  $U^* U = I + X^* X \geq I$  in the Loewner ordering of symmetric matrices, thus  $\sigma_{\min}(U) = \lambda_{\min}(U^* U) \geq 1$ . To assess  $\sigma_{\max}(U)$ , we estimate directly

$\|U\|_2$ . Given a vector  $w$  with  $\|w\|_2 = 1$ ,  $(X^\Pi w)_i \leq T\sqrt{n}$  for each  $i$ , by the Cauchy-Schwarz inequality, and thus  $\sigma_{\max}(U) = \|U\|_2 \leq \sqrt{mnT^2 + 1}$ .  $\square$

**THEOREM 7.3.** *Let  $U$ ,  $\Pi$ ,  $Y^\Pi$ ,  $Z^\Pi$ ,  $X^\Pi$  be as in (2.1), and let  $|x_{i,j}^\Pi| \leq T$  for each  $i, j$ . Then,*

$$\kappa(Y^\Pi) \leq \kappa(U) \sqrt{mnT^2 + 1}.$$

*Proof.* Since multiplying by the orthogonal matrix  $\Pi$  has no effect on the conditioning, we may safely assume  $\Pi = I$  and drop the superscripts  $\Pi$  for ease of notation. Let  $QDQ^* = Y^*Y + Z^*Z$  and  $PEP^* = I + X^*X$  be spectral decompositions of the symmetric positive definite matrices in the right-hand side, so that  $P, Q$  are unitary and  $D, E$  are diagonal. Then,

$$QDQ^* = Y^*Y + Z^*Z = Y^*(I + X^*X)Y = Y^*PEP^*Y,$$

and

$$I = D^{1/2}Q^*Y^*PE^{1/2}E^{1/2}P^*YQD^{1/2},$$

from which we infer that  $L = E^{1/2}P^*YQD^{1/2}$  is unitary. Then, we have the inequalities

$$\|Y\|_2 = \|PE^{-1/2}LD^{-1/2}Q\|_2 \leq \|D^{-1/2}\|_2 \|E^{-1/2}\|_2,$$

and

$$\|Y^{-1}\|_2 = \|Q^*D^{1/2}L^*E^{1/2}P^*\|_2 \leq \|D^{1/2}\|_2 \|E^{1/2}\|_2.$$

Multiplying the two bounds and noticing that

$$\|D^{-1/2}\|_2 \|D^{1/2}\|_2 = \kappa(U), \quad \|E^{-1/2}\|_2 \|E^{1/2}\|_2 = \kappa\left(\Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix}\right) \leq \sqrt{mnT^2 + 1}$$

the assertion follows.  $\square$

Another interesting observation is the following. Given a choice of  $(\tilde{E}, \tilde{A})$  satisfying (6.1), all other possible choices can be expressed as  $(M\tilde{E}, M\tilde{A})$  for a suitable nonsingular  $M$ . Note that all such  $M$  lead to the same  $s\tilde{E}E - \tilde{A}A$  up to right-handed equivalence. However, not all choices of  $M$ , i.e., of the pair satisfying (6.1), are equally good from a numerical point of view, since some might give rise to large errors in the resulting pencil. For instance, it is clear that in the two pencils

$$s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad s \begin{bmatrix} 1 & 0 \\ 1 & \varepsilon \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1 + \varepsilon & 1 + \varepsilon \end{bmatrix},$$

the first is to be preferred, since the second is close to a singular pencil since the two matrices almost have a common left nullspace. Extending our analogy between matrix pencils up to right-handed equivalence and subspaces, we may argue that  $\kappa\left(\begin{bmatrix} E^T \\ A^T \end{bmatrix}\right)$  measures how well-conditioned our choice of the representative is in the equivalence class of pencils up to right-handed equivalence. If we are looking for the best possible representation of  $s\tilde{E} - \tilde{A}$ , then it is clear that an orthogonal basis of  $\kappa\left(\begin{bmatrix} \tilde{E}^T \\ \tilde{A}^T \end{bmatrix}\right)$  is

the best choice, and this is precisely what is computed by the inverse-free doubling algorithms. However, a more meaningful goal is stability of the final result of the doubling step, i.e.,

$$\kappa \left( \begin{bmatrix} E^T \tilde{E}^T \\ A^T \tilde{A}^T \end{bmatrix} \right). \quad (7.1)$$

In this view, it is not clear that the path chosen in the inverse-free disc algorithm is the best choice; in fact, we argue that for very small matrices the graph subspace strategy is equivalent. We compared the magnitude of (7.1) when  $(\tilde{E}, \tilde{A})$  are computed via a QR decomposition as in (6.2) or with a permuted graph representation and (6.4). We chose 1000 random pencils with entries extracted from a Gaussian distribution of mean zero and variance one. In all cases, the condition numbers given by the two techniques are comparable. In 551 cases the conditioning of the doubled pencil computed with (6.2) is lower, and in the other 449 (6.4) gave a lower condition number. This shows that, despite the intuition that using an orthonormal basis should always give more stable results, in fact the two strategies are comparable for small matrices. For larger matrices, we may lose (on average) a factor  $n$  with respect to the orthogonal approach, as predicted by Theorem 7.3.

The next natural step in a complete stability analysis would be to show that a single step of doubling performed with the strategy of (6.4) is backward stable. However, this result cannot be obtained, not because the error bounds are unsatisfactory, but rather because of the fact that the backward stability setting cannot be adapted meaningfully to doubling algorithms. Consider for instance the matrix pencil

$$sE - A = s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

for which all known doubling methods give

$$s\tilde{E}E - \tilde{A}A = s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that this is a perfectly good problem, far from the critical and ill-conditioned cases, from the point of view of computing the invariant subspace associated with the eigenvalues inside the unit circle. A backward stability result would give us, for a special choice of the perturbation, a pair  $(E_c, A_c)$  that is very close to  $(E, A)$  and for which

$$s\tilde{E}_c E_c - \tilde{A}_c A_c = s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

holds in exact arithmetic. However, this would imply that  $E_c^{-1}A_c$  is a matrix square root of  $\begin{bmatrix} 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$ , but it is well known that this matrix does not admit a square root [22]. Therefore, a backward stability result for *a single step* of doubling is impossible. If we focus on the full algorithm as a way to compute the stable and anti-stable invariant subspace, a backward stability analysis may still be possible, although a challenging task.

**8. Numerical results.** We implemented a MATLAB version of a permuted graph representation doubling algorithm (PGR-SDA) as in Algorithm 4. We ran the method

on the 33 test examples in [10], which are created from the standard `carex` test suite, and on the corresponding problems with Hamiltonian  $-\mathcal{H}$  instead of  $\mathcal{H}$ , as suggested in [32] in order to obtain problems for which vanilla structure-preserving doubling algorithm (SDA) runs into trouble as a method for computing invariant subspaces. We transformed the pencil  $sI - \mathcal{H}$  to a symplectic pencil using a Cayley transform with parameter  $\gamma = \|\mathcal{H}\|_2$ . Notice that this differs from the usual heuristic for  $\gamma$  in the standard SDA. The reason is that the usual heuristic aims to reduce the value of  $\kappa(Y^{\Pi})$ , with  $\Pi = I$ , in the first step of the algorithm. Since we do not restrict ourselves to  $\Pi = I$  in the new algorithms, it makes no sense to use a heuristic aimed at this case. In the optimization, Algorithm 1 was run with a threshold  $T = 2$  and Algorithm 3 with  $S = 2$ ,  $T = 3$ .

We compared the results with the original SDA algorithm [11], with the MATLAB command `care`, and with the palindromic doubling algorithm (PDA) of [27]. The PDA method is a new type of doubling algorithm, which enforces palindromic rather than symplectic structure. It still relies on the inversion of a possibly ill-conditioned matrix at each step, but the condition number of this matrix does not seem to be related to the one of the matrix to be inverted in SDA; there are problems for which it is unstable, but they are in general different from those for which SDA is unstable. The `care` command from MATLAB can handle satisfactorily most of the original problems, but for four of the modified problems (namely, problems 7, 8, 32, 33) it stops reporting that the Riccati equation has no finite solution (incidentally, this does not seem to agree with the results from the column  $\kappa(X)$  in [7, Table 1]). This reflects a problem implicit in the approach of using a Riccati equation solver for computing subspaces, since when the Riccati solution does not exist or is ill-conditioned we cannot use it to recover accurately the associated graph subspace as an invariant subspace; the same problem could in principle appear with the original SDA.

The numerical results are reported in Figure 8.1 for the original problems and Figure 8.2 for the problems with Hamiltonian  $-\mathcal{H}$ . We measured the residual of the computed stable invariant subspace as suggested in [10], with the formula

$$r_S = \frac{\|\mathcal{H}U - UU^T\mathcal{H}U\|_2}{\|\mathcal{H}\|_2}. \quad (8.1)$$

This allows an immediate comparison of Figure 8.1 with [10, Figure 4], in which the same residuals lie among  $10^{-14}$  and  $10^{-18}$ .

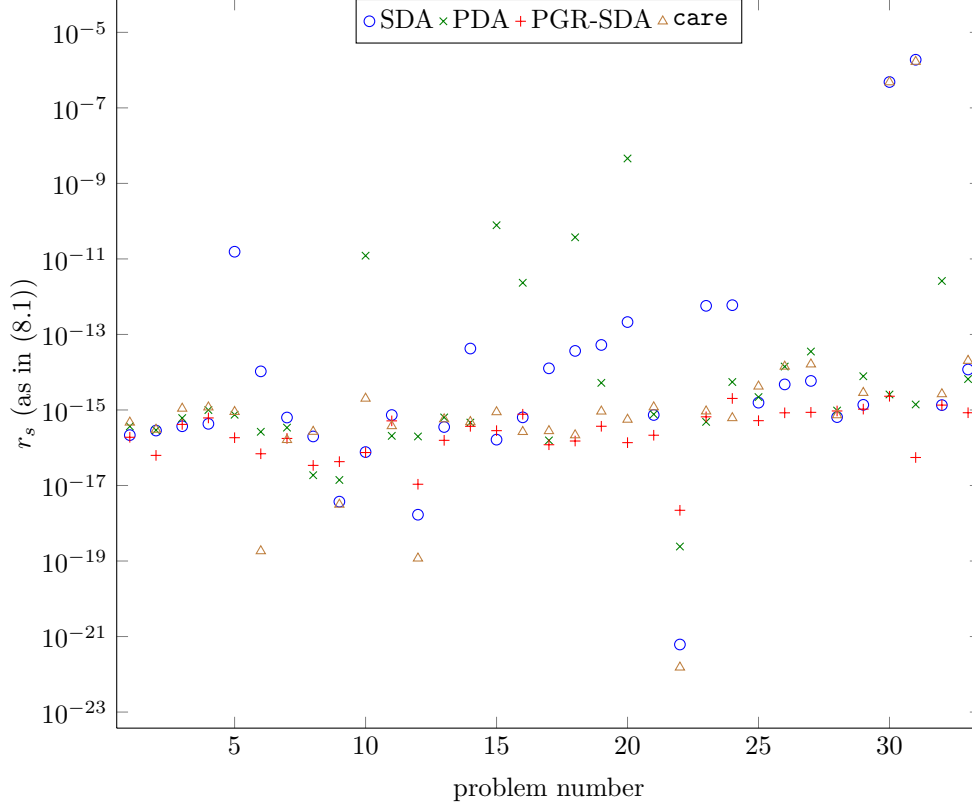
The results show that doubling algorithms with permuted graph representations can compute invariant subspaces of the same quality as the ones provided by the direct methods based on orthogonal transformations. The method does not run into the same numerical troubles as classical SDA methods when the Riccati solution does not exist or has very large entries. All the other tested algorithms, on the contrary, do have convergence problems for some of the test cases.

The number of rank-1 and rank-2 updates to be performed using Algorithm 2 is very small in all test cases; in all of them, no more than  $\max(7, 2n)$  updates (counting 1 for a rank-1 update and 2 for a rank-2 one) are needed as a whole along all doubling steps.

**9. Conclusions and challenges.** As main result of this paper we have shown that doubling algorithms can be performed in a structure-preserving fashion, without the need of inverting ill-conditioned matrices, and that the accuracy of the computed invariant subspaces is of equal quality as that of the modern algorithms based on



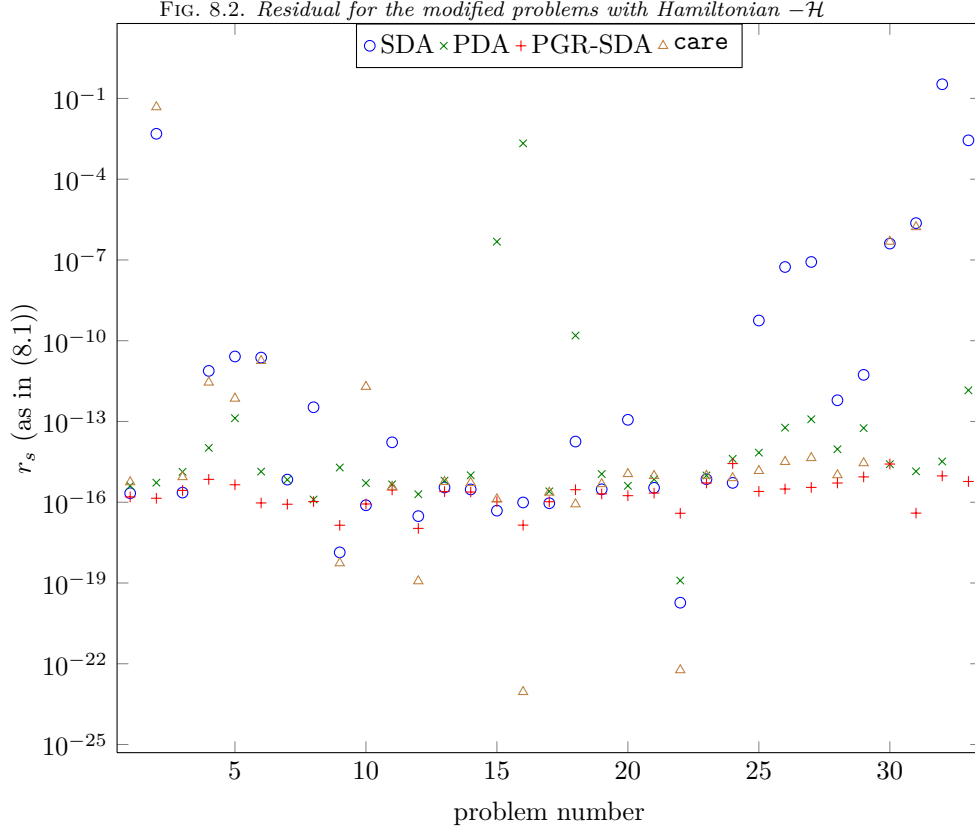
FIG. 8.1. Subspace (relative) residual for the 33 problems in [10]



orthogonal transformations. We have formulated all results for complex matrices, but all the results hold in a similar way for real matrices.

Still, several questions remain open.

- Can we perform the doubling step in Algorithm 4 using a strategy that preserves the Hermitian structure explicitly? This would lead to a more efficient implementation, and allow to drop the final symmetrization at every step after the first.
- Doubling iterations for the matrix sign function can be accelerated with a suitable scaling. The same strategy could in principle be applied to this doubling variant. Note that choosing a suitable  $\gamma$  in the Cayley transform corresponds to scaling at the first step only. Moreover, as argued in the previous section, the value of  $\gamma$  is not usually chosen in order to minimize the number of iterations, but rather in order to obtain good conditioning in the matrix to invert at the first step. Since we have now overcome that problem, a different heuristic for the choice of  $\gamma$  can be sought, focusing on convergence speed.
- We are still missing theoretical results on the worst-case and mean-case number of swap steps needed during the optimization process (Algorithm 1 and 3).
- The presented results can be adapted to doubling algorithms for several nonsymmetric entry-wise-positive equations as studied in [19, 9]. It would be interesting to see if the entry-wise positive structure can be preserved



explicitly.

- Another possible application of doubling algorithms is spectral separation for some divide-and-conquer nonsymmetric eigenvalue calculation algorithms [2, 14, 29]. The goal of this class of algorithms is to move all the computational work into routines such as matrix multiplications and QR factorizations, as they can be parallelized and implemented on complex memory architectures with better performance than the usual Hessenberg QR-based algorithms. In order to make our new version of doubling suitable to this setting, more work needs to be done to restructure Algorithm 1 into a more high performance-computing version, with less communication costs and more use of BLAS level-3 arithmetic.

**10. Acknowledgments.** The authors are grateful to David Speyer [37] for pointing out the connection to Plücker coordinates, that led to a clearer presentation of the results in Section 2.

## REFERENCES

- [1] B.D.O. ANDERSON, *Second-order convergent algorithms for the steady-state Riccati equation*, Internat. J. Control, 28 (1978), pp. 295–306.
- [2] Z. BAI, J. DEMMEL, AND M. GU, *An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems*, Numer. Math., 76 (1997), pp. 279–308.

- [3] P. BENNER, *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems*, doctoral dissertation, Fakultät für Mathematik, TU Chemnitz-Zwickau, Chemnitz, 1997. Published by Logos-Verlag, Berlin.
- [4] P. BENNER AND R. BYERS, *An arithmetic for matrix pencils: theory and new algorithms*, Numer. Math., 103 (2006), pp. 539–573.
- [5] P. BENNER, R. BYERS, V. MEHRMANN, AND H. XU, *Numerical computation of deflating subspaces of skew Hamiltonian/Hamiltonian pencils*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 165–190.
- [6] ———, *A robust numerical method for the  $\gamma$ -iteration in  $H_\infty$ -control*, Linear Algebra Appl., 425 (2007), pp. 548–570.
- [7] P. BENNER, A. LAUB, AND V. MEHRMANN, *A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: the continuous-time case*, Tech. Report SPC 95-22, Forschergruppe ‘Scientific Parallel Computing’, Fakultät für Mathematik, TU Chemnitz-Zwickau, 1995.
- [8] P. BENNER, J.-R. LI, AND T. PENZL, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numer. Linear Algebra Appl., 15 (2008), pp. 755–777.
- [9] C.-Y. CHIANG, E.K.-W. CHU, C.-H. GUO, T.-M. HUANG, W.-W. LIN, AND S.-F. XU, *Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 227–247.
- [10] D. CHU, X. LIU, AND V. MEHRMANN, *A numerical method for computing the Hamiltonian Schur form*, Numer. Math., 105 (2007), pp. 375–412.
- [11] E. K.-W. CHU, H.-Y. FAN, AND W.-W. LIN, *A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations*, Linear Algebra Appl., 396 (2005), pp. 55–80.
- [12] E. K.-W. CHU, H.-Y. FAN, W.-W. LIN, AND C.-S. WANG, *Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations*, Internat. J. Control, 77 (2004), pp. 767–788.
- [13] G.B. DANTZIG AND M.N. THAPA, *Linear programming. 1*, Springer Series in Operations Research, Springer-Verlag, New York, 1997. Introduction, With 1 CD-ROM for Windows.
- [14] J.W. DEMMEL, I. DUMITRIU, AND O. HOLTZ, *Fast linear algebra is stable*, Numer. Math., 108 (2007), pp. 59–91.
- [15] F.M. DOPICO AND C.R. JOHNSON, *Complementary bases in symplectic matrices and a proof that their determinant is one*, Linear Algebra Appl., 419 (2006), pp. 772–778.
- [16] H. FASSBENDER, *Symplectic methods for the symplectic eigenproblem*, Kluwer Academic/Plenum Publishers, New York, 2000.
- [17] F. R. GANTMACHER, *The theory of matrices. Vols. 1, 2*, Translated by K. A. Hirsch, Chelsea Publishing Co., New York, 1959.
- [18] G.H. GOLUB AND C.F. VAN LOAN, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.
- [19] X.-X. GUO, W.-W. LIN, AND S.-F. XU, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 103 (2006), pp. 393–412.
- [20] J. HARRIS, *Algebraic geometry*, vol. 133 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1995. A first course, Corrected reprint of the 1992 original.
- [21] N.J. HIGHAM, *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 2002.
- [22] ———, *Functions of matrices. Theory and computation*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
- [23] T.-M. HUANG AND W.-W. LIN, *Structured doubling algorithms for weakly stabilizing Hermitian solutions of algebraic Riccati equations*, Linear Algebra Appl., 430 (2009), pp. 1452–1478.
- [24] M. KIMURA, *Convergence of the doubling algorithm for the discrete-time algebraic Riccati equation*, Internat. J. Systems Sci., 19 (1988), pp. 701–711.
- [25] P. LANCASTER AND L. RODMAN, *Algebraic Riccati equations*, Oxford University Press, Oxford, 1995.
- [26] A.J. LAUB, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.
- [27] T. LI, C.-Y. CHIANG, E.K.-E. CHU, AND W.-W. LIN, *The palindromic generalized eigenvalue problem  $A^*x = \lambda Ax$ : Numerical solution and applications*, Linear Algebra and its Applications, 434 (2011), pp. 2269 – 2284. Special Issue: Devoted to the 2nd NASC 08 Conference in Nanjing (NSC).
- [28] W.-W. LIN AND S.-F. XU, *Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 26–39 (electronic).

- [29] A. N. MALYSHEV, *Calculation of invariant subspaces of a regular linear matrix pencil*, Sibirsk. Mat. Zh., 30 (1989), pp. 76–86, 217.
- [30] V. MEHRMANN, *A symplectic orthogonal method for single input or single output discrete time optimal linear quadratic control problems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 221–248.
- [31] V.L. MEHRMANN, *The autonomous linear quadratic control problem*, vol. 163 of Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1991. Theory and numerical solution.
- [32] V. MEHRMANN AND F. POLONI, *A generalized structured doubling algorithm for optimal control problems*, Tech. Report 750, MATHEON, 2010. Available from <http://www.matheon.de>.
- [33] V. MEHRMANN, C. SCHRÖDER, AND D. S. WATKINS, *A new block method for computing the Hamiltonian Schur form*, Linear Algebra Appl., 431 (2009), pp. 350–368.
- [34] T. PAPPAS, A.J. LAUB, AND N.R. SANDELL, *On the numerical solution of the discrete-time algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 631–641.
- [35] T. PENZL, *LYAPACK users' guide: a MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear quadratic optimal control problems*, tech. report, Technical University Chemnitz, SFB 393, 2000. Version 1.0.
- [36] P.H. PETKOV, N.D. CHRISTOV, AND M.M. KONSTANTINOV, *Computational Methods for Linear Control Systems*, Prentice-Hall, Hertfordshire, UK, 1991.
- [37] D. SPEYER (MATHOVERFLOW.NET/USERS/297), *"best" local chart for an element of  $gr(n, 2n)$* . MathOverflow. <http://mathoverflow.net/questions/58766> (version: 2011-03-17).