SIAM J. MATRIX ANAL. APPL.
Vol. 33, No. 3, pp. 780–805

DOUBLING ALGORITHMS WITH PERMUTED LAGRANGIAN GRAPH BASES*

VOLKER MEHRMANN† AND FEDERICO POLONI†

Abstract. We derive a new representation of Lagrangian subspaces in the form $\mathrm{Im}\Pi^T \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$, where Π 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 us 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

DOI. 10.1137/110850773

1. Introduction.

A *Lagrangian subspace* $\mathcal{U}$ is an N-dimensional subspace of $\mathbb{C}^{2N}$ such that $u^*Jv = 0$ 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 $(\mathcal{H}J)^* = \mathcal{H}J$ (as well as symplectic matrices S, satisfying $S^*JS = J$), is an important task in many optimal control problems [20, 31, 37, 43]. Traditionally, this computation takes the form of a matrix equation. If we impose that the invariant subspace is represented through a *graph basis*, i.e.,

$$\mathcal{U} = \mathrm{Im} \begin{bmatrix} I \\ X \end{bmatrix}, \tag{1.1}$$

then the subspace is Lagrangian if and only if X is Hermitian. If this is the case, then the problem of computing a Lagrangian invariant subspace can be transformed into an *algebraic Riccati equation*

$$0 = H + F^T X + XF - XGX, \tag{1.2}$$

*Received by the editors October 10, 2011; accepted for publication (in revised form) by C.-H. Guo May 15, 2012; published electronically July 26, 2012.
http://www.siam.org/journals/simax/33-3/85077.html

†Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany (mehrmann@math.tu-berlin.de, poloni@math.tu-berlin.de). The first author's work was partially supported by DFG Research Center Matheon "Mathematics for key technologies" in Berlin. The second author's work was supported by the Alexander von Humboldt Foundation.

which can be solved with iterative methods such as the Newton method; see, e.g., [31, 37]. However, the Riccati equation approach may be a source of problems. In general, given a basis for the Lagrangian subspace $\mathcal{U} \subseteq \mathbb{C}^{2N}$,

$$\mathcal{U} = \operatorname{Im} Q, \quad Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \quad Q_1, Q_2 \in \mathbb{C}^{N,N}, \tag{1.3}$$

the solution X of the Riccati equation is found by computing

$$X = Q_2 Q_1^{-1}. \tag{1.4}$$

If Q_1 is singular, then the subspace cannot be represented at all in the form (1.1), and if Q_1 is ill-conditioned, then we end up with large errors in X. Furthermore, in this case typically X has a very large norm and the sensitivity of the problem (1.2) may be much larger than the conditioning of the underlying invariant subspace problem [45].

Therefore, most current algorithms [6, 13, 37, 39] adopt another approach and represent these subspaces via a basis of the kind (1.3) with $Q_1 \neq I$. From any basis Q we can recover the Riccati solution X using (1.4), so the subspace approach has a wider applicability than the Riccati approach, since it avoids performing explicitly the inversion of Q_1. This is beneficial in particular when the computation of X is ill-conditioned but the application (such as computing an optimal feedback control) is well-conditioned and does not require the Riccati solution but only requires the subspace. This is particularly relevant in H_∞ control [5, 7], where near the optimal solution the Riccati solution typically fails to exist, while the Lagrangian subspace is still well-defined.

Once we have decided to follow the subspace approach, the natural choice is to choose Q in (1.3) to have orthogonal columns. Orthogonal bases and matrices are ubiquitous in numerical analysis; their main advantage is that they offer perfect (normwise) numerical backward stability in the sense of Wilkinson; see, e.g., [22].

However, the representation via orthogonal bases may also have some important drawbacks.

- In the case of large-size control problems, we have robust and mature techniques to solve the associated algebraic Riccati equation, such as the Newton-ADI method [10, 42], relying on the strong singular value decay of X that is present in many applications, but Q typically does not have the same property and thus the subspace approach is not as effective.
- Working with orthogonal bases typically requires storing and updating the $2N \times N$ matrix Q rather than the $N \times N$ (Hermitian) matrix X and is thus more expensive. For instance, in a related setting that we describe in more detail below, there is roughly a factor 8 difference between the cost of an algorithm using orthogonal bases [4] and one using graph bases [14].
- Last but not least, the Lagrangian structure of the subspace may be hard to preserve when working with orthogonal matrices. A basis Q spans a Lagrangian subspace if and only if $Q_1^* Q_2 = Q_2^* Q_1$. This property is very hard to enforce exactly in finite precision arithmetic, and after accumulating many successive orthogonal transformations, the Lagrangian property equality may be heavily violated. This is a serious problem for numerical methods enforcing this representation, such as the so-called Laub trick and later algorithms relying on it [32, 39]. It is not clear how to enforce the Lagrangian property while keeping the orthogonality when it is lost in finite precision arithmetic, for instance with a projection after every step.

A strictly related problem arises in a class of methods that has recently received much attention, the so-called doubling-type algorithms [1, 3, 4, 14, 15, 28, 29, 34]. 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 subspaces that we have mentioned above. Again, two main strategies are used: we can either choose orthogonal representations, 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, 14, 15, 29]. Similar 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 [38], which improves slightly the performance of the structure-preserving algorithms but still does not perform as well as the Schur form based algorithms [6, 13, 39, 37] on the harder benchmark problems [8, 9].

In this paper we suggest a modification of (1.1) as

$$\mathcal{U} = \operatorname{Im} \Pi^T \begin{bmatrix} I \\ X \end{bmatrix}, \tag{1.5}$$

where Π is, up to sign changes, a permutation matrix, $X = X^*$, and the entries of X are bounded in modulus by a small constant. Relying on a result of [19], we prove that every Lagrangian subspace can be written as in (1.5). This representation preserves the Lagrangian structure as (1.1), is numerically stable, and can be computed efficiently.

Making use of this representation in doubling algorithms improves the numerical accuracy of even the most simple algorithm of this family, allowing it to reach full machine precision on a wide range of problems, obtaining invariant subspaces that are both backward stable and exactly Lagrangian. Without this representation, in contrast, the currently available doubling algorithms do not achieve this on all test problems.

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, generalizing similar results on unstructured subspaces. In section 4 we describe an optimization procedure to compute in practice such representation for general, unstructured subspaces; this procedure is generalized in section 5 to Lagrangian subspaces. In sections 6 and 7 we apply this result to the representation of structured matrix pencils and to doubling algorithms, respectively. In section 8 we discuss the convergence and numerical stability of this approach, and in section 9 we test it with several numerical experiments. Finally, some conclusions and open problems are presented in section 10.

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

First, we introduce some notation. We denote by e_k the kth column of the identity matrix and by 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 $A_{i,:}$ the ith row of a matrix A and by

$A_{:,j}$ its jth column. Given $A \in \mathbb{C}^{M,N}$ and two ordered subsets $\mathcal{I}$ of $\{1, 2, \ldots, M\}$ and $\mathcal{J}$ of $\{1, 2, \ldots, N\}$, we denote by $A_{\mathcal{I},\mathcal{J}}$ the submatrix obtained by taking the rows and columns of A specified by $\mathcal{I}$ and $\mathcal{J}$, respectively. The notation $\bar{\mathcal{I}}$ (resp., $\bar{\mathcal{J}}$) represents the subset of all indices that do not belong to $\mathcal{I}$ (resp., $\mathcal{J}$), taken in an unspecified order.

Let $U \in \mathbb{C}^{N+M,N}$, and let Π be a permutation matrix. Then we define $Y^\Pi \in \mathbb{C}^{N,N}$, $Z^\Pi \in \mathbb{C}^{M,N}$, and whenever Y^Π is nonsingular $X^\Pi = [x^\Pi_{i,j}] \in \mathbb{C}^{M,N}$, as

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

We then have the following characterization for the minors of X^Π.

LEMMA 2.1. *Let $U \in \mathbb{C}^{N+M,N}$, and let Π be a permutation such that Y^Π is nonsingular. Let $X^\Pi_{\mathcal{I},\mathcal{J}}$ be the square submatrix of $X^\Pi = Z^\Pi (Y^\Pi)^{-1}$ corresponding to rows $\mathcal{I} = (i_1, i_2, \ldots, i_k)$ and columns $\mathcal{J} = (j_1, j_2, \ldots, j_k)$. Then, $\det X^\Pi_{\mathcal{I},\mathcal{J}} = \det Y^P / \det Y^\Pi$, where P is the permutation such that*

$$\begin{cases} P(j_\ell) = \Pi(N + i_\ell), \\ P(N + i_\ell) = \Pi(j_\ell), \\ P(k) = \Pi(k) \quad \text{for all the other values of } k = 1, 2, \ldots, N + M. \end{cases} \tag{2.2}$$

Proof. Due to the specific choice of P, we have

$$(Y^P (Y^\Pi)^{-1})_{j,:} = \begin{cases} X^\Pi_{i_l,:} & \text{if } j = j_l \text{ for some } \ell, \\ e_j^T & \text{otherwise.} \end{cases}$$

Therefore, $\det Y^P / \det Y^\Pi = \det Y^P (Y^\Pi)^{-1} = \det X^\Pi_{\mathcal{I},\mathcal{J}}$. ∎

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

THEOREM 2.2. *Let $U \in \mathbb{C}^{N+M,N}$ have full column rank. Then, the following assertions hold:*

1. *There exists a permutation Π such that $\det Y^\Pi \neq 0$.*
2. *If we replace U by UQ with a nonsingular matrix $Q \in \mathbb{C}^{N,N}$, then for all permutations Π, the values of $\det Y^\Pi$ are multiplied by a common factor $\det Q$.*
3. *The values of $\det Y^\Pi$ for all possible Π uniquely characterize the subspace $\operatorname{Im} U$.*

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

$$U = \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix} Y^\Pi, \; \operatorname{Im} U = \operatorname{Im} \Pi^T \begin{bmatrix} I \\ X^\Pi \end{bmatrix}. \qquad \blacksquare$$

Up to row reordering, there are only $\binom{N+M}{N}$ possible choices of Y^Π, corresponding to the possible subsets of N elements out of $N + M$. Their determinants form a set

of projective coordinates for the subspace $\operatorname{Im} U$, known in projective geometry as *Plücker coordinates* [25]. 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 [25], the following result is not typically of interest in that field, although it is crucial here.

THEOREM 2.3. *Let $U \in \mathbb{C}^{N+M,N}$ have full column rank. Then there exists a permutation matrix Π such that Y^{Π} (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 Π. Choose any permutation Π for which $|\det Y^{\Pi}|$ is maximal. Then, by Lemma 2.1, $|x_{i,j}^{\Pi}| = |\det Y^{P}|/|\det Y^{\Pi}| \leq 1$. ∎

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 $\mathbb{C}^{N+M}$. Then, there exists a permutation matrix Π and a square matrix X^{Π} such that*

$$\mathcal{U} = \operatorname{Im} \Pi^T \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix}, \tag{2.3}$$

where the entries of X^{Π} 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* (PGR).

3. PGRs 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

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

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

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

In the following we will make frequent use of the next result, which is a direct consequence of a theorem in [19].

THEOREM 3.1. *If the columns of $U \in \mathbb{C}^{2N,N}$ span a Lagrangian subspace, then there exists $\Pi \in \mathfrak{S}^{2N}$ such that Y^{Π} (as in* (2.1)*) is nonsingular.*

Proof. Let $U = QR$ be an economy-sized QR factorization (see, e.g., [22]) of U with Q partitioned as in (1.3). Since U has full column rank, R is nonsingular. The columns of Q still span a Lagrangian subspace; therefore $Q_1^* Q_2 = Q_2^* Q_1$. This implies that

$$\begin{bmatrix} Q_1 & -Q_2 \\ Q_2 & Q_1 \end{bmatrix}$$

is orthogonal and symplectic. Let $\mathcal{I} \subseteq \{1, 2, \ldots, N\}$ be a maximally independent set of rows of Q_1. Then [19, Theorem 3.1] implies that

$$\hat{Q} = \begin{bmatrix} (Q_1)_{\mathcal{I},:} \\ (Q_2)_{\bar{\mathcal{I}},:} \end{bmatrix}$$

is a nonsingular $N \times N$ matrix. If we take $v \in \mathcal{I}^N$ with

$$v_i = \begin{cases} 0, & i \in \mathcal{I}, \\ 1, & i \notin \mathcal{I}, \end{cases}$$

then with the associated swap matrix Π_v it follows that $Y^{\Pi_v} = \hat{Q}R$, which is nonsingular. ∎

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

1. *The subspace* $\operatorname{Im} U$ *is Lagrangian.*
2. *There exists $\Pi \in \mathfrak{S}^{2N}$ such that Y^Π is nonsingular and X^Π is Hermitian.*
3. *There exists $\Pi \in \mathfrak{S}^{2N}$ such that Y^Π is nonsingular, and for all swap matrices Π with this property, X^Π is Hermitian.*

Proof. The implication $1 \Rightarrow 3$ follows directly from Theorem 3.1, and $3 \Rightarrow 2$ is obvious. $2 \Rightarrow 1$. If $\operatorname{Im} \left[\begin{smallmatrix} I \\ X^\Pi \end{smallmatrix}\right]$ is Lagrangian and Π^T is symplectic, then it follows that $\operatorname{Im} U = \operatorname{Im} \Pi^T \left[\begin{smallmatrix} I \\ X^\Pi \end{smallmatrix}\right]$ is Lagrangian as well. ∎

Therefore, every Lagrangian subspace admits at least one representation as $\operatorname{Im} \Pi^T \left[\begin{smallmatrix} I \\ X^\Pi \end{smallmatrix}\right]$. We call the pair (X^Π, Π) a *permuted Lagrangian graph representation*. Note that in [19] 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, $\mathcal{I} = \mathcal{J}$.

LEMMA 3.3. *Let $U \in \mathbb{C}^{2N,N}$ be given and let $\Pi \in \mathfrak{S}^{2N}$, constructed from a vector $v \in \mathcal{I}^N$, be such that Y^Π is nonsingular. Moreover, let $X^\Pi_{\mathcal{I},\mathcal{I}}$ be the* principal *submatrix of X^Π corresponding to rows and columns $\mathcal{I} = (i_1, i_2, \ldots, i_k)$ and let $P \in \mathfrak{S}^{2n}$ be constructed from a vector w that differs from v only in positions $i_1, i_2, \ldots, i_k$. Then,* $\det X^\Pi_{\mathcal{I},\mathcal{I}} = \pm \det Y^P / \det Y^\Pi$.

Proof. Due to the choice of P, we have

$$(Y^P (Y^\Pi)^{-1})_{i,:} = \begin{cases} \pm X^\Pi_{i_l,:} & \text{if } i = i_l \text{ for some } \ell, \\ e_i^T & \text{otherwise.} \end{cases}$$

Therefore, $\det Y^P / \det Y^\Pi = \det Y^P (Y^\Pi)^{-1} = \pm \det X^\Pi_{\mathcal{I},\mathcal{I}}$. ∎

With these preliminaries we are able to obtain a bound on the elements of a particular X^Π.

THEOREM 3.4. *For every Lagrangian subspace $\mathcal{U} = \operatorname{Im} U \subset \mathbb{C}^{2N}$, there exists $\Pi \in \mathfrak{S}^{2N}$ such that Y^Π is nonsingular and*

$$|x^\Pi_{i,j}| \leq \begin{cases} 1 & \text{if } i = j, \\ \sqrt{2} & \text{if } i \neq j. \end{cases} \tag{3.1}$$

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

$$|x^\Pi_{i,i}| = \pm \det Y^P / \det Y^\Pi \leq 1.$$

For the off-diagonal entries, we obtain

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

Using the triangle inequality, we then have $|x^\Pi_{i,j}|^2 = |x^\Pi_{i,j} x^\Pi_{j,i}| \leq 1 + |x^\Pi_{i,i}||x^\Pi_{j,j}| \leq 2$. ∎

The bound (3.1) is sharp, as is 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. Computing bounded PGRs of unstructured subspaces. In this section we discuss the numerical computation of bounded PGRs of unstructured subspaces. The problem has been widely studied in the past, especially in connection with rank-revealing factorizations [16, 23, 30, 40]. We report here some results with the goal of making the generalization to Lagrangian subspaces in the next section easier to understand.

First, we describe how to convert different representations of the form (2.3) one into another.

LEMMA 4.1. *Let (X^Π, Π) be a PGR of the subspace $\mathcal{U} = \operatorname{Im} U$, let $\mathcal{I} = (i_1, i_2, \ldots, i_k)$, $\mathcal{J} = (j_1, j_2, \ldots, j_k)$ be given, and define the permutation P as in* (2.2)*. Set for brevity $X := X^\Pi$. Then, Y^P (as in* (2.1)*) is nonsingular whenever $X_{\mathcal{I},\mathcal{J}}$ is nonsingular, and when this property holds, then X^P is given by*

$$\begin{cases} (X^P)_{\bar{\mathcal{I}},\bar{\mathcal{J}}} = X_{\bar{\mathcal{I}},\bar{\mathcal{J}}} - X_{\bar{\mathcal{I}},\mathcal{J}} X^{-1}_{\mathcal{I},\mathcal{J}} X_{\mathcal{I},\bar{\mathcal{J}}}, \\ (X^P)_{\bar{\mathcal{I}},\mathcal{J}} = X_{\bar{\mathcal{I}},\mathcal{J}} X^{-1}_{\mathcal{I},\mathcal{J}}, \\ (X^P)_{\mathcal{I},\bar{\mathcal{J}}} = -X^{-1}_{\mathcal{I},\mathcal{J}} X_{\mathcal{I},\bar{\mathcal{J}}}, \\ (X^P)_{\mathcal{I},\mathcal{J}} = X^{-1}_{\mathcal{I},\mathcal{J}}. \end{cases} \tag{4.1}$$

Proof. Let $e_\mathcal{I} = \begin{bmatrix} e_{i_1} & e_{i_2} & \cdots & e_{i_k} \end{bmatrix}$ and $e_\mathcal{J} = \begin{bmatrix} e_{j_1} & e_{j_2} & \cdots & e_{j_k} \end{bmatrix}$. By permuting rows according to the definition of P, we get the identity

$$U(Y^\Pi)^{-1} = \Pi^T \begin{bmatrix} I \\ X \end{bmatrix} = P^T \begin{bmatrix} I - e_\mathcal{J} e_\mathcal{J}^T + e_\mathcal{J} e_\mathcal{I}^T X \\ X - e_\mathcal{I} e_\mathcal{I}^T X + e_\mathcal{I} e_\mathcal{J}^T \end{bmatrix}.$$

Therefore,

$$\begin{aligned} X^P &= (X - e_\mathcal{I}(e_\mathcal{I}^T X - e_\mathcal{J}^T))(I - e_\mathcal{J}(e_\mathcal{J}^T - e_\mathcal{I}^T X))^{-1} \\ &= X + (e_\mathcal{I} + X e_\mathcal{J}) X^{-1}_{\mathcal{I},\mathcal{J}} (e_\mathcal{J}^T - e_\mathcal{I}^T X), \end{aligned} \tag{4.2}$$

where the second equality follows from the application of the Sherman–Morrison–Woodbury (SMW) formula for the inversion. Using this representation, we can verify

(4.1) for each entry (i, j) by considering separately the four cases according to whether $i \in \mathcal{I}$ and whether $j \in \mathcal{J}$. ∎

Using the SMW formula as in (4.2) is suggested in [23]. Note, however, that when $\|X_{\mathcal{I},\mathcal{J}}\|$ is large, then (4.2) suffers from subtractive cancellation, while (4.1) only relies on $X^{-1}_{\mathcal{I},\mathcal{J}}$ and thus is expected to be more stable. This case is especially relevant, since our typical use for this result will be with $X^{-1}_{\mathcal{I},\mathcal{J}}$ as the 1×1 matrix containing the largest (in modulus) element of X. The computational cost of (4.1) is about $2N^2k + o(N^2)$ floating point operations.

We wish to compute an elementwise-bounded permuted graph basis for a given subspace using these cheap updating formulas. In the existence proofs, we have considered the permutation Π_* that maximizes $|\det Y^{\Pi}|$. However, computing this Π_* is not feasible in the general case, as it is an NP-hard problem [16]. On the other hand, the condition $|x^{\Pi}_{ij}| \le 1$ is weaker. From the proof of Theorem 2.3, we see that it is sufficient for Π to correspond to a *local* maximum of the determinant, i.e., by restricting to the permutations P that differ from Π by at most one transposition. Moreover, the argument used there can be easily transformed into a monotonic ascent algorithm. If $|x^{\Pi}_{ij}| > 1$, then this means that $|\det Y^{P}| > |\det Y^{\Pi}|$; thus we can find a permutation P yielding a larger objective function than Π. This procedure will necessarily terminate in a local maximum. A second modification that will prove beneficial is relaxing the condition $|x^{\Pi}_{ij}| \le 1$ to $|x^{\Pi}_{ij}| \le T$ for some $T > 1$. These ideas lead to Algorithm 1.

ALGORITHM 1. Computation of a bounded PGR.

Input: $U \in \mathbb{C}^{N+M,N}$ of full rank, a threshold value $T > 1$, an initial guess for Π such that $\det Y^{\Pi} \neq 0$ (where Y^{Π} is as in (2.1)).
Output: A PGR (Π, X^{Π}) with $|x^{\Pi}_{i,j}| \le T$ for all i, j.

```
Compute X^Π = Z^Π (Y^Π)^{-1};
repeat
    let M = max |x^Π_{i,j}|, attained at (î, ĵ);
    if M > T then
        compute P using (2.2) with I = (î), J = (ĵ);
                (this amounts to exchanging the values of Π(N + î) and Π(ĵ))
        compute X^P using (4.1);
        (X^Π, Π) ← (X^P, P);
    end
until M ≤ T;
Optionally (for increased accuracy): keep Π, but recompute X^Π from U if the
above loop took many steps;
```

The algorithm costs $8/3N^3 + 2N^2\xi + o(N^3)$, where ξ is the number of optimization steps to be performed, plus an additional $8/3N^3$ if X^{Π} is recomputed at the end. To evaluate the cost, it is thus important to estimate the number of optimization steps and to provide a good initial guess Π. For the following well-known result we present a proof that we can later generalize to the Lagrangian case.

THEOREM 4.2 (see [30]). *Let $U \in \mathbb{C}^{N+M,N}$ be of full rank, and let Π_* be such that $|\det Y^{\Pi_*}| = \max_{\Pi} |\det Y^{\Pi}|$. Moreover, let Π_0 be the initial guess used in Algorithm* 1.

1. *Then in Algorithm* 1 *at most $\xi = \log_T |\frac{\det Y^{\Pi_*}}{\det Y^{\Pi_0}}|$ steps are needed.*
2. *If Π_0 is the permutation returned by the QR factorization with column pivoting* [22, section 5.5.6] *of U^*, then $\xi \le \frac{N}{2} \log_T N$.*

Proof. 1. At every step, $|\det Y^P|/|\det Y^{\Pi}| = |x^{\Pi}_{i,j}| > T$, so in each step the determinant increases by at least a factor T.

2. For the QR factorization $\Pi_0 U = R^* Q^*$, by construction $|R_{i,j}| \leq |R_{i,i}|$ for each j. (Otherwise, the jth column rather than the ith would be selected by the pivoting procedure at the ith factorization step.) Therefore, if $R = [r_{ij}]$ and if we set $D = \operatorname{diag}(r_{1,1}, r_{2,2}, \ldots, r_{N,N})$, then $D^{-1}R$ has all its entries bounded by 1. In particular, every $N \times N$ submatrix S of $D^{-1}R$ has entries bounded in modulus by 1. Hence, by the Hadamard determinant bound [11], we have $|\det S| \leq N^{N/2}$. Since $(Y^{\Pi_0} Q)^*$ is the upper triangular matrix formed by the first N columns of R, it has determinant $\prod_{i=1}^{N} r_{i,i} = \det D$. On the other hand, we can choose the submatrix S in the above argument so that $(Y^{\Pi_*} Q)^* = DS$, since Y^{Π_*} is a submatrix obtained by choosing a suitable subset of rows of U. Thus we obtain $|\frac{\det Y^{\Pi_*}}{\det Y^{\Pi_0}}| \leq N^{N/2}$, and the assertion follows. ∎

The estimate in Theorem 4.2 is often fairly pessimistic, but nevertheless it shows that we can obtain a worst-case complexity of $O(N^3 \log N)$ if T is chosen to be constant and $O(N^3)$ if we allow T to grow moderately with N (e.g., $T = N^{1/3}$). Moreover, when we use this procedure at every step of a doubling algorithm as in the presented algorithms below, then a good starting guess for Π will be available in every iteration after the first. Note that the QRP factorization can be reused as a method to invert Y^{Π} in line 1 of the algorithm and thus does not increase the total cost of the algorithm. Note further that the procedure in Algorithm 1 resembles the basic "complementary tableaux" implementation of the simplex method [17].

5. Computing bounded PGRs of Lagrangian subspaces. Since symplectic swap matrices are essentially permutations (up to some sign changes), Lemma 4.1 needs only minor changes for the Lagrangian case.

LEMMA 5.1. *Let (X^{Π}, Π) be a PGR of the Lagrangian subspace $\mathcal{U} = \operatorname{Im} U$, let $\mathcal{I} = (i_1, i_2, \ldots, i_k)$ be given, and let P be the symplectic swap matrix defined in Lemma* 3.3. *Set for brevity $X := X^{\Pi}$. Then, Y^P (as in* (2.1)*) is nonsingular whenever $X_{\mathcal{I},\mathcal{I}}$ is nonsingular. Furthermore, when this property holds, then X^P is given by*

$$
\begin{cases}
(X^P)_{\bar{\mathcal{I}},\bar{\mathcal{I}}} = X_{\bar{\mathcal{I}},\bar{\mathcal{I}}} - X_{\bar{\mathcal{I}},\mathcal{I}} X^{-1}_{\mathcal{I},\mathcal{I}} X_{\mathcal{I},\bar{\mathcal{I}}}, \\
(X^P)_{\bar{\mathcal{I}},\mathcal{I}} = X_{\bar{\mathcal{I}},\mathcal{I}} X^{-1}_{\mathcal{I},\mathcal{I}}, \\
(X^P)_{\mathcal{I},\bar{\mathcal{I}}} = X^{-1}_{\mathcal{I},\mathcal{I}} X_{\mathcal{I},\bar{\mathcal{I}}}, \\
(X^P)_{\mathcal{I},\mathcal{I}} = - X^{-1}_{\mathcal{I},\mathcal{I}}.
\end{cases} \tag{5.1}
$$

The computational cost of this formula is about $N^2 k + o(N^2)$ floating point operations, since we can exploit that X and X^P are Hermitian.

The analogue of Algorithm 1 in this setting is Algorithm 2, which is slightly more complicated, due to the fact that we need to consider a threshold T_D for the diagonal entries and another T_O for the off-diagonal ones.

If U spans a Lagrangian subspace, then the first computed value of X^{Π} 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 to the nearest Hermitian matrix via $X \leftarrow \frac{X+X^*}{2}$. The cost of the algorithm is about $5/3N^3 + N^2\xi + o(N^3)$, where ξ is the sum of all values of $|\mathcal{I}|$ along the iteration—essentially, we add 1 for each step in which $\mathcal{I} = (\hat{k})$ and 2 for each step with $\mathcal{I} = (\hat{\imath}, \hat{\jmath})$.

ALGORITHM 2. Computation of a bounded permuted Lagrangian graph representation of a Lagrangian subspace.

Input: $U \in \mathbb{C}^{2N,N}$ of full rank spanning a Lagrangian subspace, thresholds $T_D > 1$, $T_O > \sqrt{1+T_D^2}$, an initial guess for Π such that $\det Y^{\Pi} \neq 0$.

Output: A permuted Lagrangian graph representation (Π, X^{Π}) satisfying $|x^{\Pi}_{k,k}| \leq T_D$ and $|x^{\Pi}_{i,j}| \leq T_O$ for all $i \neq j$.

Compute $X^{\Pi} = Z^{\Pi}(Y^{\Pi})^{-1}$;
repeat
  let $M_O = \max |x^{\Pi}_{i,j}|$, $i \neq j$, be attained at $(\hat{\imath}, \hat{\jmath})$;
  let $M_D = \max |x^{\Pi}_{k,k}|$, be attained at $(\hat{k}, \hat{k})$;
  if $M_D > T_D$ **then** Set $\mathcal{I} = (\hat{k})$;
  else if $M_O > T_O$ **then** Set $\mathcal{I} = (\hat{\imath}, \hat{\jmath})$;
  compute P as in Lemma 3.3;
    (this amounts to exchanging one or two entries in v such that $\Pi = \Pi_v$)
  compute X^P using (5.1);
  $(X^{\Pi}, \Pi) \leftarrow (X^P, P)$;
until $M_O \leq T_O$, $M_D \leq T_D$;
Optionally (for increased accuracy): keep Π, but recompute X^{Π} from U if the above loop took many steps;

In order to obtain a good starting guess for Π, we propose here a modification of the QR factorization with column pivoting, where we use a symplectic swap matrix instead of a permutation. The factorization is described in Algorithm 3.

ALGORITHM 3. $QR\Pi_v$ factorization of a $N \times 2N$ matrix.

Input: $M \in \mathbb{C}^{N,2N}$.

Output: $Q \in \mathbb{C}^{N,N}$ orthogonal, $R^{(1)}, R^{(2)} \in \mathbb{C}^{N,N}$ such that $R^{(1)}$ is a column permutation of an upper triangular matrix, $\Pi_v \in \mathfrak{S}^{2N}$ such that $U^* = Q \begin{bmatrix} R^{(1)} & R^{(2)} \end{bmatrix} \Pi_v$.

Set $C = \{1, 2, \ldots, 2N\}$ (it will be the set of "available" column indices at each step);
for $k = 1, 2, \ldots, N$ **do**
  compute $p \in C$ such that $\|M_{k:N,p}\|$ is maximal;
  $M \leftarrow Q_k M$, where Q_k is a Householder matrix that zeroes out $M_{(k+1):N,p}$
  if $p > N$ **then** $M \leftarrow M\Pi_k$, where $\Pi_k \in \mathfrak{S}^{2N}$ swaps the pth and $(p-N)$th columns);
  $C \leftarrow C \setminus \{p, p \pm N\}$;
end
Set $\begin{bmatrix} R^{(1)} & R^{(2)} \end{bmatrix} = M$, $Q = \prod Q_k$, $\Pi_v = \prod \Pi_k$ (the two products can be accumulated along the algorithm);

Since $\mathfrak{S}^{2N}$ does not contain all permutations, we have to settle for a slightly more general form in R, namely, that its first N columns $R^{(1)}$ can be permuted to form an upper triangular matrix. At each step, we choose the "available" column of largest norm, permute it to the first N columns if necessary, and then apply a usual

Householder transformation that zeroes out its bottom entries. After each step, we have to remove from the set of "available" columns not only the used one p, but also the column $p+N$ or $p-N$, since this column cannot end up in $R^{(1)}$ at the same time as p due to the special structure of symplectic swap matrices.

With this algorithm, we can prove a symplectic analogue of Theorem 4.2.

THEOREM 5.2. *Let the columns of $U \in \mathbb{C}^{2N,N}$ span a Lagrangian subspace, let Π_* be such that $|\det Y^{\Pi_*}| = \max_\Pi |\det Y^\Pi|$, and let Π_0 be the initial guess used in Algorithm* 2.

1. *The number ξ of optimization steps in Algorithm* 2 *satisfies $\xi \le \log_\tau \frac{\det Y^{\Pi_*}}{\det Y^{\Pi_0}}$, where $\tau = \min\{T_D, \sqrt{T_O^2 - T_D^2}\}$.*
2. *If Π_0 is the permutation returned by applying Algorithm* 3 *to U^*, then $\xi \le 3N \log_\tau N + N \log_\tau 18$.*

Proof. 1. In each step when $M_D > T_D$, $|\det Y^\Pi|$ increases by a factor T_D, and in each step when $M_D \le T_D, M_O > T_O$ by a factor

$$|x_{i,i}x_{j,j} - x_{i,j}x_{j,i}| \ge |x_{i,j}x_{j,i}| - |x_{i,i}x_{j,j}| = |x_{i,j}|^2 - |x_{i,i}||x_{j,j}| \ge T_O^2 - T_D^2.$$

(Here we use the facts that $x_{j,i} = \overline{x_{i,j}}$ and $|x_{i,i}| \le T_D$ for all i, since we check the condition on the diagonal entries first.) Since each step dealing with off-diagonal elements is counted as an increase of ξ by 2, we get the square root.

2. We use the same strategy as in point 2 of Theorem 4.2; namely, we prove that the determinant of every $N \times N$ submatrix of U (and in particular Y^{Π_*}) is bounded by an exponential term times $|\det Y^{\Pi_0}|$.

Denote by $\operatorname{triu}(A)$ the upper triangular part of a matrix A (*including* its diagonal) and by $\operatorname{tril}(A)$ the lower triangular part (*excluding* the diagonal, so that the identity $A = \operatorname{triu}(A) + \operatorname{tril}(A)$ holds).

We may assume without loss of generality that the pivots chosen in the algorithm are exactly $(1, 2, \dots, N)$ (otherwise, we reorder columns in U^*). With $R^{(k)} = [r_{ij}^{(k)}]$, $k = 1, 2$, by setting

$$\operatorname{diag}(r_{1,1}^{(1)}, r_{2,2}^{(1)}, \dots, r_{N,N}^{(1)})^{-1} \begin{bmatrix} R^{(1)} & R^{(2)} \end{bmatrix} =: \begin{bmatrix} T^{(1)} & T^{(2)} \end{bmatrix},$$

we obtain that $T^{(1)}$ is unit upper triangular. The pivoting procedure ensures that the elements of both $T^{(1)}$ and $M := \operatorname{triu}(T^{(2)})$ are bounded in modulus by 1. No elementwise bound can be inferred directly on $L := \operatorname{tril}(T^{(2)})$, though, and this makes the proof more involved than the unstructured case of Theorem 4.2.

Nevertheless, since the starting subspace $\operatorname{Im} U$ is Lagrangian, it follows that $T^{(2)}T^{(1)*}$ is symmetric, and this can be translated into a different bound for L. Using

$$LT^{(1)*} + \operatorname{tril}(MT^{(1)*}) = \operatorname{tril}((L+M)T^{(1)*}) = \operatorname{tril}(T^{(1)}(L+M)^*) = \operatorname{tril}(T^{(1)}M^*)$$

we get

$$LT^{(1)*} = \operatorname{tril}(T^{(1)}M^* - MT^{(1)*})$$

and thus the matrix

$$T^{(2)}T^{(1)*} = \operatorname{tril}(T^{(1)}M^* - MT^{(1)*}) + MT^{(1)*}$$

must have all its elements smaller than $3N$ in modulus. Let us now consider any $N \times N$ submatrix S of $\begin{bmatrix} T^{(1)} & T^{(2)} \end{bmatrix}$. We may choose $\begin{bmatrix} T^{(3)} & T^{(4)} \end{bmatrix} \in \mathbb{C}^{N,2N}$ as a

suitable column permutation of $\begin{bmatrix} I_N & 0_N \end{bmatrix}$, so that the matrix

$$T = \begin{bmatrix} T^{(1)} & T^{(2)} \\ T^{(3)} & T^{(4)} \end{bmatrix}$$

has determinant equal to $\pm \det S$. The $2N \times 2N$ matrix

$$\tilde{T} = \begin{bmatrix} T^{(1)} & T^{(2)} \\ T^{(3)} & T^{(4)} \end{bmatrix} \begin{bmatrix} T^{(1)*} & 0 \\ 0 & T^{(1)*} \end{bmatrix}$$

has every entry smaller in modulus than $3N$. Thus, by the Hadamard bound [11], $|\det \tilde{T}| \leq (3N)^{2N}(2N)^{2N/2} = 18^N N^{3N}$. Since $\det T^{(1)} = 1$, we get $|\det S| = |\det T| = |\det \tilde{T}| \leq 18^N N^{3N}$. One of the possible choices for S is $\operatorname{diag}(r^{(1)}_{1,1}, r^{(1)}_{2,2}, \ldots, r^{(1)}_{N,N})^{-1} Q^* (Y^{\Pi_*})^*$; in this case,

$$18^N N^{3N} \geq |\det S| = \frac{|\det Y^{\Pi_*}|}{|\prod r_{i,i}|} = \frac{|\det Y^{\Pi_*}|}{|\det Y^{\Pi_0}|}. \qquad \square$$

We stress once again that these bounds are usually pessimistic, and in practice the number of iterations that we encountered was always low, in particular due to the initial guess for Π available in many situations; see section 9 for some numerical examples.

6. PGRs of matrix pencils. In the context of computing eigenvalues and invariant subspaces, matrix pencils are usually considered up to *right equivalence*, i.e., up to the equivalence relation defined by

$$sE_1 - A_1 \sim sE_2 - A_2$$

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 interested not in the matrix $\left[\begin{smallmatrix} E^T \\ A^T \end{smallmatrix}\right]$ but rather in the subspace $\operatorname{Im}\left[\begin{smallmatrix} E^T \\ A^T \end{smallmatrix}\right]$. Therefore, our results on the representation of subspaces may be adapted to the representation 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 6.1. *Let $sE - A$ with $E, A \in \mathbb{C}^{2n,2n}$ be a regular symplectic pencil. Then, there exist $\Pi_1, \Pi_2 \in \mathfrak{S}^{2n}$ such that*

$$sE - A \sim s \begin{bmatrix} I & X_{11} \\ 0 & X_{21} \end{bmatrix} \Pi_1 - \begin{bmatrix} X_{12} & 0 \\ X_{22} & I \end{bmatrix} \Pi_2^T, \tag{6.1}$$

where

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

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

$$\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} E_1^* \\ A_2^* \\ E_2^* \\ A_1^* \end{bmatrix} = 0. \tag{6.2}$$

Moreover, $\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix}$ is of full row rank, since otherwise one could find a nonzero 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 would contradict the regularity assumption.

Therefore, the columns of $\begin{bmatrix} E_1 & A_2 & E_2 & A_1 \end{bmatrix}^*$ span a Lagrangian subspace of $\mathbb{C}^{2N}$ with $N = 2n$, and from Theorem 3.4 we obtain a PGR

$$\Pi_v \begin{bmatrix} E_1^* \\ A_2^* \\ E_2^* \\ A_1^* \end{bmatrix} = \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix} (Y^{\Pi})^{-1}. \tag{6.3}$$

Note that Π_v acts separately on the block columns $(1,3)$ as well as $(2,4)$, so these actions are given by $\Pi_1 = \Pi_{v_1}$ and $\Pi_2 = \Pi_{v_2}$, where $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$. After reshuffling the blocks, we obtain (6.1). ∎

The representation (6.1) with $\Pi_1 = \Pi_2 = I$ is well-known; see, e.g., [36, 37, 41], where the representation

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

with $H = H^*$, $G = G^*$ is used. However, without the further 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,

$$\begin{bmatrix} E_1 & E_2 & A_2 & -A_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} E_1^* \\ E_2^* \\ A_2^* \\ -A_1^* \end{bmatrix} = 0, \tag{6.4}$$

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

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

where X^{Π} 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*).

7. PGRs 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 7.1 (see [3]). *Let $sE - A$ with $E, A \in \mathbb{C}^{N,N}$ be a regular pencil, and let $\widetilde{E}, \widetilde{A} \in \mathbb{C}^{N,N}$ be such that*

$$\widetilde{E}A = \widetilde{A}E, \quad \operatorname{Rank} \begin{bmatrix} \widetilde{E} & \widetilde{A} \end{bmatrix} = N. \tag{7.1}$$

Then, the pencil $s\widetilde{E}E - \widetilde{A}A$ has the same deflating subspaces as $sE - A$, and its eigenvalues are the squares of the corresponding eigenvalues.

If in Theorem 7.1 the matrices E and $\widetilde{E}$ are invertible, this result is simple, as $(E^{-1}A)^2 = (\widetilde{E}E)^{-1}(\widetilde{A}A)$. 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 eigenvalues inside the unit disk converge to 0 and the ones outside the unit disk converge to ∞. If there are no eigenvalues of modulus 1, then after a sufficient (not too large) number of steps, it is easy to recover the corresponding invariant subspaces associated with the eigenvalues inside and outside the unit disk, respectively, as kernels of the two coefficients of the pencil.

The inverse-free disc function method [3] performs this doubling iteration by choosing $[\widetilde{E} \quad \widetilde{A}]$ with orthonormal rows, i.e., it computes a QR decomposition

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} A \\ E \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{7.2}$$

and takes $\widetilde{E} = -Q_{21}$, $\widetilde{A} = Q_{22}$. 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) [14] is based instead on the version $\Pi = I$ of the representation (6.1). At each step, the method uses a pencil of the form

$$E = \begin{bmatrix} I & X_{11} \\ 0 & X_{21} \end{bmatrix}, A = \begin{bmatrix} X_{12} & 0 \\ X_{22} & I \end{bmatrix} \tag{7.3}$$

and chooses $\widetilde{E}$ and $\widetilde{A}$ having blocks I and 0 in the same position, and thus this structure is maintained in the products $\widetilde{E}E$ and $\widetilde{A}A$. 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 $\widetilde{E}$ and $\widetilde{A}$ 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 [28]. 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 [38], in order to deal with the cases in which the representation (7.3) is a poor choice. A block structure similar to (7.3) 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 (1.1) 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 PGR at every step to achieve added stability. Given a pencil $sE - A$ and a bounded PGR

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

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

$$\left(\begin{bmatrix} -X^{\Pi} & I \end{bmatrix} \Pi\right) \Pi^T \begin{bmatrix} I \\ X^{\Pi} \end{bmatrix} = 0. \tag{7.4}$$

The resulting complete procedure is presented in Algorithm 4.

ALGORITHM 4. Doubling algorithm with bounded PGR.

Input: A symplectic pencil $sE - A$.
Output: A basis for its invariant subspace associated with the eigenvalues inside the unit disk.

1 **while** *a suitable stopping criterion is not satisfied* **do**
2 Compute an optimal (Π, X^{Π}) for the subspace $\begin{bmatrix} A \\ E \end{bmatrix}$ using Algorithm 1. At each step after the first, warm-start with the Π from the previous iteration;
3 Set $\begin{bmatrix} \widetilde{A} & -\widetilde{E} \end{bmatrix} = \begin{bmatrix} -X^{\Pi} & I \end{bmatrix} \Pi$;
4 Form the products $E \leftarrow \widetilde{E}E$, $A \leftarrow \widetilde{A}A$;
5 Compute an optimal (Π, X^{Π}) associated to the symplectic pencil $sE - A$ as in (6.3) with Algorithm 2 (warm-started);
6 Symmetrize $X^{\Pi} \leftarrow \frac{X^{\Pi} + (X^{\Pi})^*}{2}$ to reduce the impact of numerical errors;
7 Using the computed (Π, X^{Π}), replace the computed $sE - A$ with the right-equivalent pencil in (6.1);
8 **end**
9 **return** $U = \ker \begin{bmatrix} X_{22} & I \end{bmatrix} \Pi_2^T = \Pi_2 \begin{bmatrix} I \\ -X_{22} \end{bmatrix}$;

After each time lines 5 and 7 are executed, each single entry of X (and thus of E and A) is bounded above by the chosen threshold T_O, and after each time line 2 is executed, each entry of X^{Π} (and thus of $\tilde{E}$ and $\tilde{A}$) is bounded above by T. Each entry of the product in line 4 is therefore bounded as well by $2nTT_O$.

The computational cost of Algorithm 4 is $(\frac{8}{3}N^3 + 2N^2\xi_1) + (\frac{5}{3}N^3 + N^2\xi_2) + 2N^3 = \frac{19}{3}N^3 + N^2(2\xi_1 + \xi_2) + o(N^3)$ floating point operations per step, where ξ_1 and ξ_2 are the numbers of used optimization steps. This compares with $11N^3 + o(N^3)$ for QR-based doubling and $\frac{4}{3}N^3 + o(N^3)$ for a symmetry-preserving implementation of SDA. The number of outer steps needed is comparable, as the convergence speed is related to the eigenvalues of the pencil, which are the same for all three variants. As a stopping criterion in Algorithm 4, we can use the variation in X^{Π} after each step.

It is an interesting observation that we can recover SDA both for Riccati equations (called SDA-I in [34]) and for unilateral matrix equations (called SDA-II in [34] and cyclic reduction in the queuing theory and matrix equations literature) by choosing specific values of Π in lines 2 and 5, rather than running the optimization loop to obtain optimal ones. Thus, in some sense, we can now recognize them as two out of 2^N variants of the same algorithm, and we are free to switch among all these variants at every step to choose the most numerically stable one.

Algorithm 4 in this form is, however, still unsatisfactory, because it does not manage to go from a PGR matrix X^{Π} for $sE - A$ to one for $s\widetilde{E}E - \widetilde{A}A$ 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 open. Such a method could lower the computational cost to match that of the similar SDA. Nevertheless, we show below that this preliminary version gives very good computational results.

8. 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 [28], and the proof was later adapted to different types of doubling algorithms [12, 38]. We use the same technique based on the Kronecker canonical form [21] here 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.

Let us introduce some notation. Let $A_0 - sE_0$ be a regular $N \times N$ matrix pencil, and denote its Kronecker chains [21] by $(w_1^i, w_2^i, \dots, w_{k_i}^i)$ and the associated eigenvalues with λ_i. (Here w_1^i are the eigenvectors; the λ_i are possibly infinite and may be repeated if there are multiple chains with the same eigenvalue, and $\sum_i k_i = N$.) We divide the spectrum into the sets $\mathcal{S} = \{w_j^i : |\lambda_i| < 1\}$, $\mathcal{U} = \{w_j^i : |\lambda_i| > 1\}$, $\mathcal{C}_1 = \{w_j^i : |\lambda_i| = 1, j \le k_1/2\}$, and $\mathcal{C}_2 = \{w_j^i : |\lambda_i| = 1, j > k_1/2\}$. Notice that $|\mathcal{S}| + |\mathcal{U}| + |\mathcal{C}_1| + |\mathcal{C}_2| = N$ (where $|\mathcal{X}|$ denotes the cardinality of a set $\mathcal{X}$), and in fact their union is a basis of $\mathbb{C}^N$ composed of Kronecker chains. Moreover, let S, U, C_1, C_2 be matrices whose columns span $\mathcal{S}$, $\mathcal{U}$, $\mathcal{C}_1$, and $\mathcal{C}_2$, respectively. Then we have the following convergence theorem.

THEOREM 8.1. *Let $A_0 - sE_0$ be a regular $2n \times 2n$ matrix pencil such that for each i with $|\lambda_i| = 1$, 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 $\operatorname{span} S \cup C_1$ *and* $\operatorname{span} U \cup C_1$*, respectively. The convergence is quadratic with rate $l_{\max}/l_{\min}$, where*

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

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

Proof. We can easily obtain a slightly modified version of the Kronecker canonical form as

$$W(A_0 - sE_0)Z = \begin{bmatrix} J_S & & & \\ & J_{C_1} & H & \\ & & J_{C_1} & \\ & & & I \end{bmatrix} - s \begin{bmatrix} I & & & \\ & I & & \\ & & I & \\ & & & J_U \end{bmatrix},$$

where $W, Z \in \mathbb{C}^{N,N}$ are the nonsingular changes of bases that take the pencil to this canonical form, $Z = \begin{bmatrix} S & C_1 & C_2 & U \end{bmatrix}$, J_S is a Jordan matrix containing the stable eigenvalues, J_U is a Jordan matrix containing the inverses of the unstable eigenvalues, J_{C_1} contains the first half of each unimodular Jordan chain, and H is such that

$$\begin{bmatrix} J_{C_1} & H \\ 0 & J_{C_1} \end{bmatrix}$$

is a permutation of the Jordan matrix containing the unimodular eigenvalues. As in [28], 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 via

$$\begin{bmatrix} J_{C_1} & H \\ 0 & J_{C_1} \end{bmatrix}^{2^k} = \begin{bmatrix} J_{C_1}^{2^k} & H_k \\ 0 & J_{C_1}^{2^k} \end{bmatrix}.$$

Clearly, $J_S^{2^k} = \mathcal{O}(l_{\max}^{2^k})$, $J_U^{2^k} = \mathcal{O}(l_{\min}^{-2^k})$. It is proved in [28, 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

$$\begin{bmatrix} I & & \\ & I & \\ & -H_k^{-1} J_{C_1}^{2^k} & \\ 0 & & \end{bmatrix}$$

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 the boundedness of A_k and E_k, we have

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

from which we see that $\Pi_2 \left[\begin{smallmatrix} I \\ -X_{22} \end{smallmatrix}\right]$ converges to a PGR of $\begin{bmatrix} S & C_1 \end{bmatrix}$. The analogous result for the semiunstable subspace follows with a similar argument by considering a Kronecker canonical form with $\widetilde{Z} = \begin{bmatrix} U & C_1 & C_2 & S \end{bmatrix}$. ∎

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 the 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 sensitivity of the Q factor in the QR factorization of U depends on it [26]. In this sense, we show that when X^{Π} has bounded entries, a PGR is a good representation of the subspace, and it can be computed in a numerically stable way from another given good representation.

THEOREM 8.2. *Let* $\Pi^T \left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right] = U \in \mathbb{C}^{N+M,N}$, *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, and thus $\sigma_{\min}(U) = \lambda_{\min}(U^*U) \geq 1$. Given a vector w with $\|w\|_2 = 1$, then $(X^{\Pi}w)_i \leq T\sqrt{N}$ for each i, by the Cauchy–Schwarz inequality, and thus $\sigma_{\max}(U) = \|U\|_2 \leq \sqrt{MNT^2+1}$. ∎

THEOREM 8.3. *Let U, Π, Y^{Π}, Z^{Π}, X^{Π} be as in* (2.1)*, and let $|x^{\Pi}_{i,j}| \leq T$ for each i, j. Then,*

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

Proof. Since multiplying by the orthogonal matrix Π has no effect on the conditioning, we may safely assume $\Pi = I$ and drop the superscripts Π for ease of notation. Let $QDQ^* = Y^*Y + Z^*Z$ and $PEP^* = I + X^*X$ be spectral decompositions, 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

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

and

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

By multiplying the two bounds and noticing that

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

the assertion follows. ∎

Another interesting observation is the following. Given a choice of $(\widetilde{E}, \widetilde{A})$ satisfying (7.1), all other possible choices can be expressed as $(M\widetilde{E}, M\widetilde{A})$ for a suitable nonsingular M. Note that all such M lead to the same $s\widetilde{E}E - \widetilde{A}A$ up to right-handed equivalence. However, not all choices of M, i.e., of the pair satisfying (7.1), 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

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

the first is to be preferred, since the second is close to a singular pencil with the two matrices almost having a common left nullspace. Extending our analogy between

matrix pencils up to right-handed equivalence and subspaces, we may argue that $\kappa\left(\left[\begin{smallmatrix} E^T \\ A^T \end{smallmatrix}\right]\right)$ 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 $s\widetilde{E} - \widetilde{A}$, then it is clear that an orthogonal basis of $\kappa\left(\left[\begin{smallmatrix} \widetilde{E}^T \\ \widetilde{A}^T \end{smallmatrix}\right]\right)$ 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.,

$$\kappa\left(\begin{bmatrix} E^T\widetilde{E}^T \\ A^T\widetilde{A}^T \end{bmatrix}\right). \tag{8.1}$$

In this view, it is not clear that the path chosen in the inverse-free disc algorithm is the best choice. In fact, for very small matrices the graph subspace strategy seems equivalent. We compared the magnitude of (8.1) when $(\widetilde{E}, \widetilde{A})$ are computed via a QR decomposition as in (7.2) or with a PGR and (7.4). 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 (7.2) is lower, and in the other 449 (7.4) 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 8.3.

The next step in a complete stability analysis would be to show that a single step of doubling performed with the strategy of (7.4) is backward stable. However, this result cannot be obtained, not because the error bounds are unsatisfactory but rather because 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

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

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

$$s\widetilde{E_c}E_c - \widetilde{A_c}A_c = s\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

holds in exact arithmetic. However, this would imply that $E_c^{-1}A_c$ is a matrix square root of $\left[\begin{smallmatrix} 0 & \varepsilon \\ 0 & 0 \end{smallmatrix}\right]$, but it is well-known that this matrix does not admit a square root [27].

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 invariant subspace associated with the eigenvalues inside and outside the unit disk, then a backward stability analysis may still be possible, although a challenging task.

9. Numerical results. We have implemented a MATLAB version of a PGR doubling algorithm as in Algorithm 4. We ran the method on the 33 test examples used in [13]. These problems come from the standard `carex` test suite [8], some using the standard parameters, some using different choices in order to create more challenging examples. The exact values of the parameters can be found in [13].

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 γ 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 2 with $T_D = 2$, $T_O = 3$.

We compare the results with the original SDA [14], the inverse-free sign method [4], the MATLAB command `care(...,'factor')`, the method in [13] based on the periodic Schur decomposition, and the palindromic doubling algorithm (PDA) of [33]. The `care` command from MATLAB is based on the QZ algorithm, which is backward stable but not structure-preserving. It was used with the option `'factor'`, which returns (up to some row scaling) a basis of the Lagrangian subspace $\mathcal{U}$ rather than directly the Riccati solution X. The periodic Schur method is in theory both backward stable and structure-preserving, but as we see, in finite precision in some cases the orthogonal structure is well preserved while the symplectic structure is not. The PDA method is a new type of doubling algorithm, which enforces the weaker 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 that of the matrix to be inverted in SDA. There are problems for which PDA is unstable, but they are in general different from those for which SDA is unstable.

The periodic Schur method uses the URV decomposition implemented in FORTRAN in the library HAPACK which has not yet been adapted to the current version of MATLAB. Therefore, we did not run new tests for this method but present the error results published in [13] instead, which use the same expressions for the errors.

We remark that apart from SDA, all other methods directly compute a basis for the Lagrangian subspace without going through the Riccati solution X: the `'factor'` switch of `care` has already been discussed, and PDA and the inverse-free sign method (essentially) both perform an iteration on the Hamiltonian, then extract the Lagrangian subspace as the kernel of the resulting matrix. As discussed in the introduction, this is the more stable choice, and in many applications the relevant quantities can be computed without ever forming X.

In Figure 9.1 we present the residual of the computed Lagrangian subspace, according to the formula

$$r_S = \frac{\left\|\mathcal{H}U - UU^T\mathcal{H}U\right\|_2}{\|\mathcal{H}\|_2}. \tag{9.1}$$

In Figure 9.2, for the sake of completeness, we also present the corresponding results

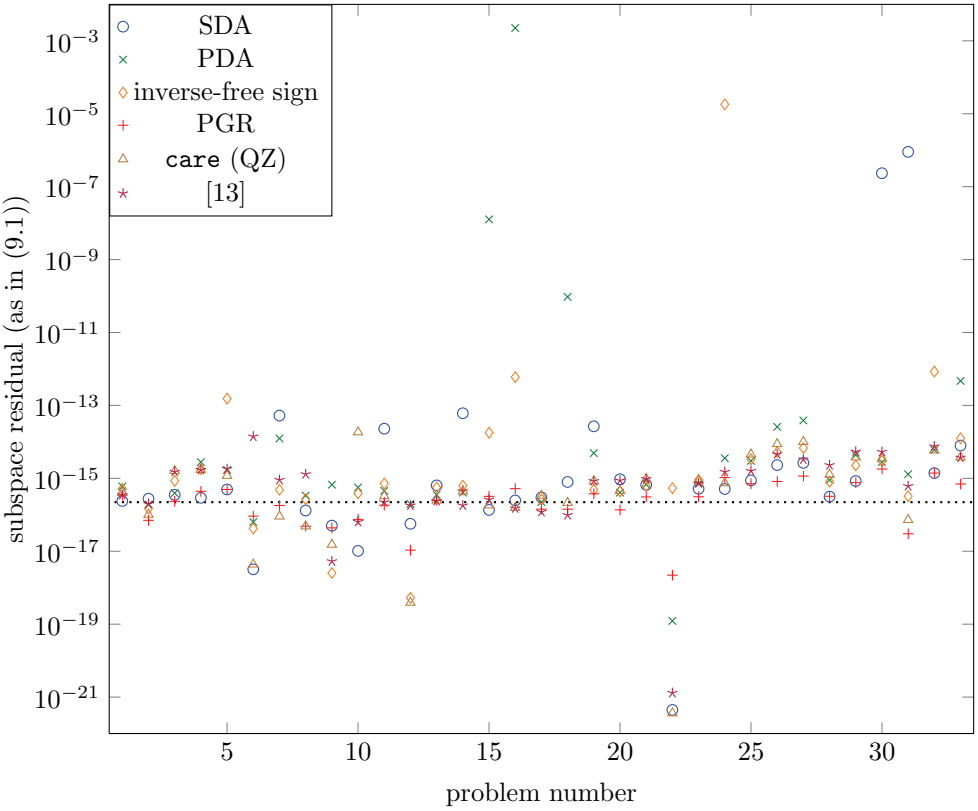

FIG. 9.1. *Subspace (relative) residual for the* 33 *problems in* [13].

using the Riccati residual

$$r_R = \frac{\left\|H + F^T X + XF - XGX\right\|_2}{\|H\|_2 + \|F^T X\|_2 + \|XF\|_2 + \|XGX\|_2}. \tag{9.2}$$

As discussed in the introduction, this residual measure may be in itself an ill-conditioned function of the Hamiltonian matrix $\mathcal{H}$, when X has large norm. For this reason, the residuals in Figure 9.2 sometimes vary wildly even when the results of the different methods on the same experiment are indistinguishable according to the error measure (9.1). Therefore, the results in Figure 9.2 are less conclusive.

To check how well the Lagrangian property is perserved, in Figure 9.3 we present the value of $\|U^* J U\|_2$, where U is an orthonormal basis for the computed subspace. This value should be exactly zero, since invariant subspaces of $2N \times 2N$ Hamiltonian matrices associated with N eigenvalues inside the open unit disk are Lagrangian in exact arithmetic. Values significantly larger than machine precision indicate further errors in the computed subspace that are not revealed by residual checking and may appear even when using backward stable algorithms, if they do not preserve structure. Algorithms SDA and PGR return subspaces in the forms (1.1) and (1.5), respectively, with $X = X^*$. Thus this residual is exactly zero, even when computed in IEEE arithmetic. Therefore, they are not reported in the figure. For the URV-based method of [13], the residual is reported in that article only for two experiments in which the

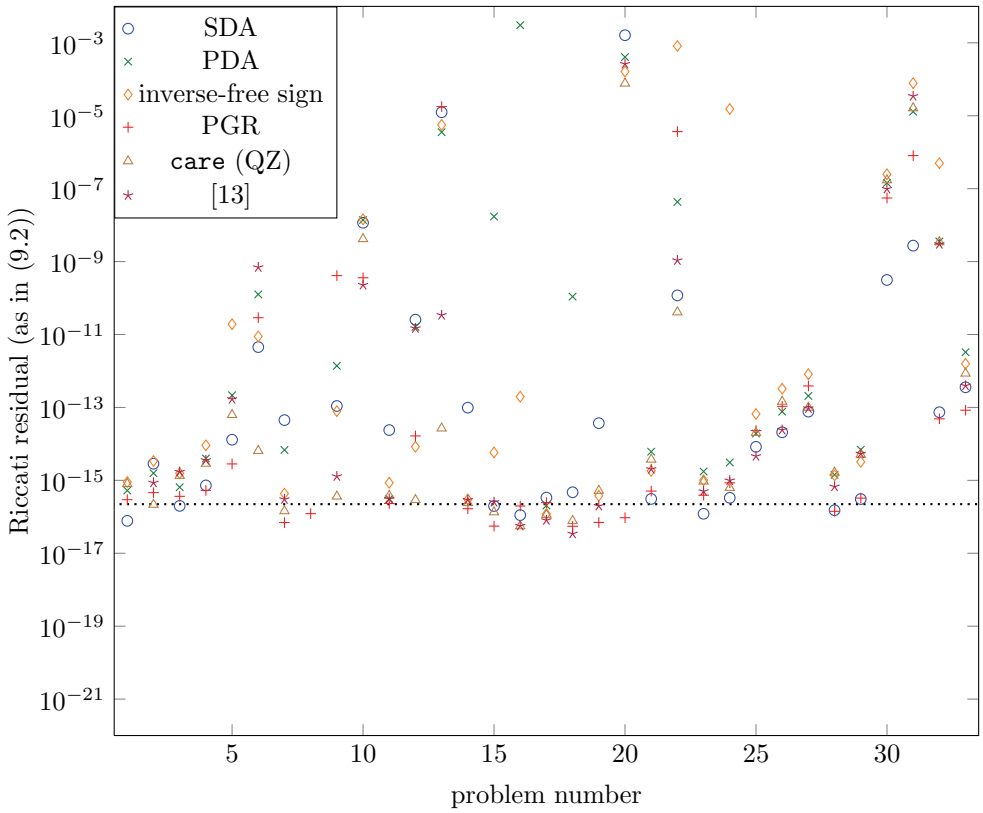

FIG. 9.2. *Riccati (relative) residual for the* 33 *test problems.*

deviation from the Lagrangian property is particularly significant. For this reason there are only two data points in the figure for this method; nevertheless, they are sufficient to prove that the algorithm suffers from the loss of Lagrangian structure.

The results clearly show that doubling algorithms with PGRs can compute invariant subspaces of the same quality as the backward stable algorithms based on orthogonal transformations, while preserving structure exactly. All the other methods, on the contrary, do not reach both these goals on all the experiments.

We conclude this section with some remarks on the computational cost and the number of optimization steps needed. As stated before, with this implementation the computational cost of the kth step of doubling is $\frac{19}{3}N^3 + N^2(2\xi_1^{(k)} + \xi_2^{(k)}) + o(N^3)$, where $\xi_1^{(k)}$ and $\xi_2^{(k)}$ are, respectively, the number of optimizations steps in Algorithms 1 and 2. Table 9.1 presents the values of

$$\Xi_1 = \sum \xi_1^{(k)}, \qquad \Xi_2 = \sum \xi_2^{(k)},$$

where the sum is taken over all the doubling steps needed along the algorithm. The number is always comparable with the dimension n of the problem, and in many cases it is exactly zero. These results show that the overhead due to the optimization procedure of Algorithms 1 and 2 is very small in practice and cheaper in comparison than the cost of one additional step of doubling.

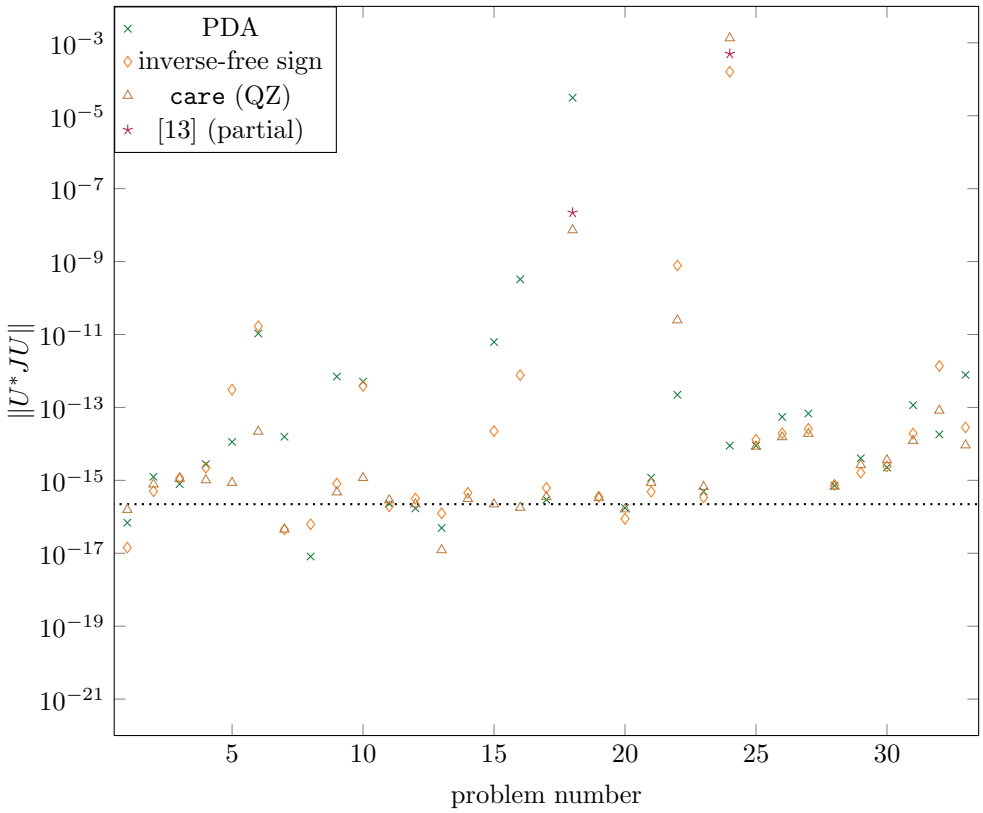

FIG. 9.3. *Loss of Lagrangianity in the* 33 *test problems.*

TABLE 9.1
Number of optimization steps needed in the algorithm.

Problem	n	S_1	S_2
1	2	0	0
2	2	0	0
3	4	1	1
4	8	0	0
5	9	4	1
6	30	29	14
7	2	0	0
8	2	0	0
9	2	2	1
10	2	3	1
11	2	0	0

Problem	n	S_1	S_2
12	2	0	0
13	2	0	0
14	2	0	0
15	2	0	1
16	2	0	1
17	2	0	0
18	2	4	1
19	3	0	0
20	3	0	0
21	4	0	1
22	4	3	3

Problem	n	S_1	S_2
23	4	0	0
24	4	0	0
25	77	0	0
26	237	0	0
27	397	0	0
28	8	0	0
29	64	0	0
30	21	0	0
31	21	14	20
32	100	0	0
33	60	0	30

10. Conclusions and challenges. As a main result of this paper we have shown that doubling algorithms can be performed in a structure-preserving fashion, without the need for inverting ill-conditioned matrices, and that the accuracy of the computed invariant subspaces is of quality equal to that of the modern algorithms based on orthogonal transformations. We have formulated all results for complex matrices, but all the results hold in a similar way for real matrices.

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 us 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 γ in the Cayley transform corresponds to scaling at the first step only. Moreover, as argued in the previous section, the value of γ is usually chosen not to minimize the number of iterations but rather 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 γ can be sought, focusing on convergence speed.
- Can we obtain stronger bounds on the number of optimization steps needed during Algorithms 1 and 2?
- The presented results can be adapted to doubling algorithms for several nonsymmetric entrywise-positive equations as studied in [12, 24]. It would be interesting to analyze if the entrywise 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, 18, 35]. 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 cost and more use of BLAS level-3 arithmetic.

Acknowledgments. The authors are grateful to David Speyer [44] for pointing out the connection to Plücker coordinates, which led to a clearer presentation of the results in section 2. We also thank two referees and the handling editor for their helpful comments that improved the presentation.

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*, Ph.D. dissertation, Fakultät für Mathematik, TU Chemnitz-Zwickau, Chemnitz, 1997.
[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, P. Losse, V. Mehrmann, and H. Xu, *Robust formulas for optimal H_∞ controllers*, Automatica, 47 (2011), pp. 1639–2646.
[6] 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.
[7] P. Benner, R. Byers, V. Mehrmann, and H. Xu, *A robust numerical method for the γ-iteration in H_∞-control*, Linear Algebra Appl., 425 (2007), pp. 548–570.

[8] 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*, Technical report SPC 95-22, Forschergruppe Scientific Parallel Computing, Fakultät für Mathematik, TU Chemnitz-Zwickau, 1995.

[9] P. Benner, A. Laub, and V. Mehrmann, *A Collection of Benchmark Examples for the Numerical Solution of Algebraic Riccati Equations* II: *The Discrete-Time Case*, Technical report SPC 95-23, Forschergruppe Scientific Parallel Computing, Fakultät für Mathematik, TU Chemnitz-Zwickau, 1995.

[10] 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.

[11] J. Brenner and L. Cummings, *The Hadamard maximum determinant problem*, Amer. Math. Monthly, 79 (1972), pp. 626–630.

[12] 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.

[13] D. Chu, X. Liu, and V. Mehrmann, *A numerical method for computing the Hamiltonian Schur form*, Numer. Math., 105 (2007), pp. 375–412.

[14] 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.

[15] 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.

[16] A. Çivril and M. Magdon-Ismail, *On selecting a maximum volume sub-matrix of a matrix and related problems*, Theoret. Comput. Sci., 410 (2009), pp. 4801–4811.

[17] G.B. Dantzig and M.N. Thapa, *Linear Programming*, Springer Ser. Oper. Res. 1, Springer-Verlag, New York, 1997.

[18] J.W. Demmel, I. Dumitriu, and O. Holtz, *Fast linear algebra is stable*, Numer. Math., 108 (2007), pp. 59–91.

[19] 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.

[20] H. Fassbender, *Symplectic Methods for the Symplectic Eigenproblem*, Kluwer Academic/Plenum Publishers, New York, 2000.

[21] F.R. Gantmacher, *The Theory of Matrices*, vols. 1, 2, Chelsea Publishing, New York, 1959.

[22] G.H. Golub and C.F. Van Loan, *Matrix computations*, in Johns Hopkins Studies in the Mathematical Sciences, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[23] S.A. Goreinov, I.V. Oseledets, D.V. Savostyanov, E.E. Tyrtyshnikov, and N.L. Zamarashkin, *How to find a good submatrix*, in Matrix Methods: Theory, Algorithms and Applications, World Scientific, Hackensack, NJ, 2010, pp. 247–256.

[24] 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.

[25] J. Harris, *Algebraic Geometry*, Grad. Texts Math. 133, Springer-Verlag, New York, 1995.

[26] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[27] N.J. Higham, *Functions of Matrices. Theory and Computation*, SIAM, Philadelphia, 2008.

[28] 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.

[29] M. Kimura, *Convergence of the doubling algorithm for the discrete-time algebraic Riccati equation*, Internat. J. Systems Sci., 19 (1988), pp. 701–711.

[30] D.E. Knuth, *Semioptimal bases for linear dependencies*, Linear Multilinear Algebra, 17 (1985), pp. 1–4.

[31] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, Oxford, UK, 1995.

[32] A.J. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.

[33] 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 Appl., 434 (2011), pp. 2269–2284.

[34] 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.

[35] A.N. Malyshev, *Calculation of invariant subspaces of a regular linear matrix pencil*, Sibirsk. Mat. Zh., 30 (1989), pp. 76–86, 217.

[36] 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.
[37] V.L. Mehrmann, *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Berlin, 1991.
[38] V. Mehrmann and F. Poloni, *A generalized structured doubling algorithm for optimal control problems*, Numer. Linear Algebra Appl., to appear; also available online from http://www.matheon.de.
[39] 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.
[40] C.-T. Pan, *On the existence and computation of rank-revealing LU factorizations*, Linear Algebra Appl., 316 (2000), pp. 199–222.
[41] 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.
[42] T. Penzl, *LYAPACK Users' Guide: A MATLAB Toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problems, and Linear Quadratic Optimal Control Problems*, Technical report, Technical University Chemnitz, SFB 393, 2000.
[43] P.H. Petkov, N.D. Christov, and M.M. Konstantinov, *Computational Methods for Linear Control Systems*, Prentice-Hall, Hertfordshire, UK, 1991.
[44] D. Speyer, *"Best" local chart for an element of $gr(n, 2n)$, MathOverflow*, http://mathoverflow.net/questions/58766 (17 March 2011).
[45] S.-F. Xu, *Sensitivity analysis of the algebraic Riccati equations*, Numer. Math., 75 (1996), pp. 121–134.