Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Future Generation Computer Systems 19 (2003) 1231–1242

FGCS
FUTURE GENERATION COMPUTER SYSTEMS

www.elsevier.com/locate/future

New algorithms for the iterative refinement of estimates of invariant subspaces

K. Hüper [a,*], P. Van Dooren [b]

[a] *Department of Mathematics, Würzburg University, Am Hubland, D-97074 Würzburg, Germany*
[b] *Department of Mathematical Engineering, Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium*

Abstract

New methods for refining estimates of invariant subspaces of a non-symmetric matrix are presented. We use global analysis to show local quadratic convergence of our method under mild conditions on the spectrum of the matrix.

Keywords: Non-symmetric eigenvalue problem; Invariant subspace; Riccati equations; Sylvester equations; Global analysis; Local quadratic convergence

1. Refining estimates of invariant subspaces

The computation of invariant subspaces has received a lot of attention in the numerical linear algebra literature. In this paper we present a new algorithm that borrows ideas of two well established methods: the Riccati-/Sylvester-like iteration, and the Jacobi-like iteration. Even though these techniques are traditionally associated with very different eigenvalue algorithms, the algorithm that we derive in this paper makes a nice link between them. Our method refines estimates of invariant subspaces of real non-symmetric matrices which are already "nearly" upper block triangular, but not in condensed form. It uses the Lie group of real unipotent lower block triangular $(n \times n)$-matrices as similarities on a nearly upper block triangular matrix. We develop a class of algorithms based on such similarity transformations, which display local quadratic convergence to an upper block triangular form. The formulation of the algorithms and their convergence analysis are valid for different diagonal block sizes, as long as these blocks have disjoint spectrum, i.e., as long as the corresponding invariant subspaces are well defined. An important special case is the computation of the real Schur form, which groups complex conjugate eigenvalues in 2×2 diagonal blocks. It can be obtained by our method, provided these eigenvalue pairs are disjoint. We always work over $\mathbb{R}$, but the generalization to $\mathbb{C}$ is immediate and we state without proof that all the results from this paper directly apply to the complex case. The outline of this paper is as follows. After introducing some notation we will focus on an algorithm consisting of similarity transformations by unipotent lower block triangular matrices. In order to improve numerical accuracy, we then use orthogonal transformations instead. The convergence properties of the orthogonal algorithm is shown to be an immediate consequence of the former one.

* Corresponding author.
E-mail addresses: hueper@mathematik.uni-wuerzburg.de (K. Hüper), vdooren@csam.ucl.ac.be (P. Van Dooren).

0167-739X/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0167-739X(03)00048-7

2. Lower unipotent block triangular transformations

Let $V \subset \mathbb{R}^{n\times n}$ denote the vector space of real upper block triangular $(n \times n)$-matrices

$$V := \{X \in \mathbb{R}^{n\times n} | X_{ij} = 0_{n_i \times n_j} \forall 1 \leq j < i \leq r\} \quad (1)$$

with square diagonal sub-blocks $X_{ii} \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \ldots, r$ and $\sum_{i=1}^{r} n_i = n$. Let $\mathcal{L}_n$ denote the Lie group of real unipotent lower block triangular $(n \times n)$-matrices partitioned conformably with V:

$$\mathcal{L}_n := \{X \in \mathbb{R}^{n\times n} | X_{kk} = I_{n_k} \forall 1 \leq k \leq r, X_{ij} = 0_{n_i \times n_j} \forall 1 \leq i < j \leq r\}. \quad (2)$$

Given a real upper block triangular matrix $A \in V$, we consider the orbit $M_{\mathcal{L}_n}$ of A under similarity action σ of $\mathcal{L}_n$:

$$\sigma : \mathcal{L}_n \times V \to \mathbb{R}^{n\times n},$$
$$(L, X) \mapsto \sigma(L, X) = LXL^{-1}, \quad (3)$$

$$M_{\mathcal{L}_n} := \{X \in \mathbb{R}^{n\times n} | X = LAL^{-1}, L \in \mathcal{L}_n\}. \quad (4)$$

The following (generic) assumption will be crucial in our analysis.

Assumption 2.1. Let $A \in V$, then its diagonal sub-blocks A_{ii}, $i = 1, \ldots, r$ have mutually disjoint spectra.

Our first result shows that any matrix lying in a sufficiently small neighborhood of A which fulfils Assumption 2.1 is an element of an $\mathcal{L}_n$-orbit of some other matrix, say B, which also fulfils Assumption 2.1. We first show that under Assumption 2.1 the smooth mapping σ satisfies the following lemma.

Lemma 2.1. *The mapping σ defined by (3) is locally surjective around (I, A).*

Proof. Let $\mathfrak{l}_n$ denote the Lie algebra of real lower block triangular $(n \times n)$-matrices

$$\mathfrak{l}_n := \{X \in \mathbb{R}^{n\times n} | X_{kk} = 0_{n_k} \forall 1 \leq k \leq r, X_{ij} = 0_{n_i \times n_j} \forall 1 \leq i < j \leq r\}. \quad (5)$$

It is sufficient to show that the derivative $D\sigma(I, A) : \mathfrak{l}_n \times V \to \mathbb{R}^{n\times n}$ is locally surjective. For arbitrary $l \in \mathfrak{l}_n$ and for arbitrary $a \in V$

$$D\sigma(I, A) \cdot (l, a) = lA - Al + a. \quad (6)$$

We show that for any $h \in \mathbb{R}^{n\times n}$ the linear system

$$lA - Al + a = h \quad (7)$$

has a solution in terms of $l \in \mathfrak{l}_n$ and $a \in V$. By decomposing into upper block triangular and *strictly* lower block triangular parts $h = h_{\text{upp.bl.}} + h_{\text{str.low.bl.}}$ and because $a \in V$ is already upper block triangular it remains to show that the strictly lower block triangular part of (7)

$$(lA - Al)_{\text{str.low.bl.}} = h_{\text{str.low.bl.}} \quad (8)$$

can be solved for $l \in \mathfrak{l}_n$. We partition into "blocks of sub-blocks"

$$l = \begin{bmatrix} l_{11} & 0 \\ l_{\widetilde{21}} & l_{\widetilde{22}} \end{bmatrix}, \qquad A = \begin{bmatrix} A_{11} & A_{\widetilde{12}} \\ 0 & A_{\widetilde{22}} \end{bmatrix},$$
$$h_{\text{str.low.bl.}} = \begin{bmatrix} (h_{11})_{\text{str.low.bl.}} & 0 \\ h_{\widetilde{21}} & (h_{\widetilde{22}})_{\text{str.low.bl.}} \end{bmatrix},$$

accordingly, i.e., $A_{11} \in \mathbb{R}^{n_1 \times n_1}$ and $l_{11} = 0_{n_1}$ as before. Thus one has to solve for $l_{\widetilde{21}}$ and $l_{\widetilde{22}}$. Considering the $(\widetilde{21})$-block of (8) gives

$$l_{\widetilde{21}} A_{11} - A_{\widetilde{22}} l_{\widetilde{21}} = h_{\widetilde{21}}. \quad (9)$$

By Assumption 2.1, the Sylvester equation (9) has a unique solution $l_{\widetilde{21}}$. In order to prove the result, we proceed analogously with the $(\widetilde{22})$-block of (8), i.e. $l_{\widetilde{22}} A_{\widetilde{22}} - A_{\widetilde{22}} l_{\widetilde{22}} = -l_{\widetilde{21}} A_{\widetilde{12}} + (h_{\widetilde{22}})_{\text{str.low.bl.}}$, and continue inductively ($l := l_{\widetilde{22}}$, $A := A_{\widetilde{22}}$, etc.) by partitioning the remaining diagonal blocks A_{ii}, $i = 2, \ldots, r$ into smaller blocks of sub-blocks. □

Let $A \in \mathbb{R}^{n\times n}$ fulfil Assumption 2.1, then the next lemma characterizes the $\mathcal{L}_n$-orbit of the matrix A.

Lemma 2.2. *$M_{\mathcal{L}_n} := \{X \in \mathbb{R}^{n\times n} | X = LAL^{-1}, L \in \mathcal{L}_n\}$ is diffeomorphic to $\mathcal{L}_n$.*

Proof. The set $M_{\mathcal{L}_n}$ is a smooth manifold because it is the orbit of a semi-algebraic group action, see [10, p. 353]. We will show that the stabilizer subgroup $\text{stab}(A) \subset \mathcal{L}_n$ equals the identity $\{I\}$ in $\mathcal{L}_n$, i.e. that the only solution in terms of $L \in \mathcal{L}_n$ for

$$LAL^{-1} = A \Leftrightarrow [L, A] = 0 \quad (10)$$

is $L = I$. Partition L and A conformably as

$$L = \begin{bmatrix} I_{n_1} & 0 \\ L_{\widetilde{21}} & L_{\widetilde{22}} \end{bmatrix}, \qquad A = \begin{bmatrix} A_{11} & A_{\widetilde{12}} \\ 0 & A_{\widetilde{22}} \end{bmatrix},$$

where $L_{\widetilde{22}} \in \mathcal{L}_{n-n_1}$. The $(\widetilde{21})$-block of $[L, A] = 0$ yields $L_{\widetilde{21}} A_{11} - A_{\widetilde{22}} L_{\widetilde{21}} = 0$, implying $L_{\widetilde{21}} = 0$ by Assumption 2.1 on the spectrum of A. By recursive application of this argument to the $(\widetilde{22})$-block of (10) the result follows. Therefore, $L = I$ implies $\mathrm{stab}(A) = \{I\}$ and hence $M_{\mathcal{L}_n} \cong \mathcal{L}_n / \mathrm{stab}(A) = \mathcal{L}_n$. □

3. Algorithms, main ideas

The algorithms presented in this section for the iterative refinement of invariant subspaces of non-symmetric real matrices are driven by the following ideas. Let the matrix $A \in V$ satisfy Assumption 2.1 and consider an $X \in M_{\mathcal{L}_n}$ sufficiently close to A, i.e.,

$$\|X - A\| < \Delta_\lambda, \tag{11}$$

where $\|Z\| := \sqrt{\mathrm{tr}(ZZ^\top)}$ and Δ_λ denotes the absolute value of the smallest difference of any eigenvalues of two *different* diagonal sub-blocks of A. Obviously,

$$\mathrm{span}\left(\begin{bmatrix} I_{n_1} \\ 0_{(n_2+\cdots+n_r)\times n_1} \end{bmatrix}\right) \tag{12}$$

is then a good approximation of an n_1-dimensional right invariant subspace of X, because by assumption (11) on X, the blocks $\|X_{ji}\|$ are small for all $j > i$. Consider an $L^{(\alpha)} \in \mathcal{L}_n$ of the following partitioned form:

$$L^{(\alpha)} := \begin{bmatrix} I_{n_1} & & & & & \\ & \ddots & & & & \\ & & I_{n_\alpha} & & & \\ & & p^{(\alpha+1,\alpha)} & \ddots & & \\ & & \vdots & & \ddots & \\ & & p^{(r,\alpha)} & & & I_{n_r} \end{bmatrix}, \tag{13}$$

where empty blocks are considered to be zero. We want to compute

$$P^{(\alpha)} := \begin{bmatrix} p^{(\alpha+1,\alpha)} \\ \vdots \\ p^{(r,\alpha)} \end{bmatrix} \in \mathbb{R}^{\bar{n}_\alpha \times n_\alpha},$$

$$\bar{n}_\alpha \doteq n_{\alpha+1} + \cdots + n_r, \tag{14}$$

such that (for $\underline{n}_\alpha \doteq n_1 + \cdots + n_{\alpha-1}$):

$$L^{(\alpha)} X L^{(\alpha)^{-1}} = \begin{bmatrix} I_{\underline{n}_\alpha} & 0 & 0 \\ 0 & I_{n_\alpha} & 0 \\ 0 & P^{(\alpha)} & I_{\bar{n}_\alpha} \end{bmatrix} X \times \begin{bmatrix} I_{\underline{n}_\alpha} & 0 & 0 \\ 0 & I_{n_\alpha} & 0 \\ 0 & -P^{(\alpha)} & I_{\bar{n}_\alpha} \end{bmatrix} = Z, \tag{15}$$

where Z is of the form

$$Z = \begin{bmatrix} Z_{1,1} & \cdots & \cdots & \cdots & \cdots & \cdots & Z_{1,r} \\ \vdots & \ddots & & & & & \vdots \\ \vdots & & Z_{\alpha-1,\alpha-1} & & & & \vdots \\ \vdots & & \vdots & Z_{\alpha,\alpha} & & & \vdots \\ \vdots & & \vdots & 0 & Z_{\alpha+1,\alpha+1} & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_{r,1} & \cdots & Z_{r,\alpha-1} & 0 & Z_{r,\alpha+1} & \cdots & Z_{r,r} \end{bmatrix}, \tag{16}$$

i.e., the blocks below the diagonal block $Z_{\alpha,\alpha}$ are zero. For convenience we first assume without loss of generality that $r = 2$. In this case we want to solve the (21)-block of

$$\begin{bmatrix} I & 0 \\ P^{(1)} & I \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ -P^{(1)} & I \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} \tag{17}$$

in terms of $P^{(1)}$, i.e., we want to solve the matrix valued algebraic Riccati equation

$$P^{(1)} X_{11} + X_{21} - P^{(1)} X_{12} P^{(1)} - X_{22} P^{(1)} = 0. \tag{18}$$

Since (18) is in general not solvable in closed form, several authors have suggested different approaches to solve (18) iteratively; see [4] for Newton-type iterations on the non-compact Stiefel manifold and [8,17] for iterations like

$$P_{i+1} X_{11} - X_{22} P_{i+1} = P_i X_{12} P_i - X_{21}, \quad P_0 = 0. \tag{19}$$

We refer to [5] for a comparison of the approaches of the former three papers and to [16] for quantitative results concerning Newton-type iterations to solve Riccati equations, see also the recent work [6,7].

A rather natural idea to solve (18) approximately is to ignore the second order term, $-P^{(1)}X_{12}P^{(1)}$, and solve instead the Sylvester equation

$$P^{(1)}X_{11} + X_{21} - X_{22}P^{(1)} = 0, \tag{20}$$

which by Assumption 2.1 has a unique solution.

We now return to the general case where the number r of invariant subspaces to be computed is larger than 2. With Jacobi-like sweep algorithms in mind, it is natural to formulate an algorithm which solves an equation like (20) for $P^{(1)}$, say, along the lines of (13)–(16), then transform X according to $X \mapsto L^{(1)}X(L^{(1)})^{-1}$, do the same for $P^{(2)}$, and so forth. One can show that such an algorithm would be a differentiable map around A. Moreover, local quadratic convergence could be proved by means of analysis.

Instead of solving a Sylvester equation for $P^{(\alpha)}$, i.e., solving for the corresponding block of (15), one can reduce the complexity by solving Sylvester equations of lower dimension in a cyclic manner, i.e., perform the algorithm block wise on each $p^{(ij)} \in \mathbb{R}^{n_i \times n_j}$. The smaller the block sizes n_i, the smaller the Riccati equations one has to solve and the simpler the transformations $L^{(l)}$. The smallest possible sizes n_i one can choose for the diagonal blocks $A_{i,i}$ should group the multiple eigenvalues of A into the diagonal blocks since this is needed by Assumption 2.1. An algorithm for block sizes 1×1 (implying distinct (real) eigenvalues of A) would lead to scalar algebraic Riccati equations which are solvable in closed form. Such an approach would come very close to [2,3,18] where the authors studied Jacobi-type methods for solving the non-symmetric (generalized) eigenvalue problem.

3.1. Formulation of the algorithm

The following algorithm will be analyzed for a matrix A satisfying Assumption 2.1 and an initial matrix $X \in M_{\mathcal{L}_n}$ that is sufficiently close to A. Consider the index set $\mathcal{I} := \{(ij)\}_{i=2,\dots,r;\, j=1,\dots,r-1}$ and fix an ordering, i.e., a surjective map $\beta : \mathcal{I} \to \{1, \dots, \binom{r}{2}\}$. For convenience we rename double indices in the description of the algorithm by simple ones by means of $X_{ij} \mapsto X_{\beta((ij))}$ respecting the ordering β.

Algorithm 3.1 (Sylvester sweep). Given an $X \in M_{\mathcal{L}_n}$. Define

$$\begin{aligned}
X_k^{(1)} &:= L^{(1)}XL^{(1)^{-1}} \\
X_k^{(2)} &:= L^{(2)}X_k^{(1)}L^{(2)^{-1}} \\
&\vdots \\
X_k^{(\binom{r}{2})} &:= L^{(\binom{r}{2})}X_k^{(\binom{r}{2}-1)}(L^{(\binom{r}{2})})^{-1},
\end{aligned}$$

where for $l = 1, \dots, \binom{r}{2}$, the transformation matrix $L^{(l)} \in \mathcal{L}_n$ differs from the identity matrix I_n only by the ijth block, say $p^{(l)}$.

Here $\beta((ij)) = l$ and $p^{(l)} \in \mathbb{R}^{n_j \times n_i}$ solves the Sylvester equation

$$p^{(l)}(X_k^{(l-1)})_{jj} - (X_k^{(l-1)})_{ii}p^{(l)} + (X_k^{(l-1)})_{ij} = 0.$$

The overall algorithm then consists of the following iteration of sweeps.

Algorithm 3.2 (Refinement of estimates of subspaces).

- Let $X_0, \dots, X_k \in M_{\mathcal{L}_n}$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $X_k^{(1)}, \dots, X_k^{(\binom{r}{2})}$ as above (sweep).
- Set $X_{k+1} := X_k^{(\binom{r}{2})}$. Proceed with the next sweep.

For the index set $\mathcal{I} := \{(ij)\}_{i=2,\dots,r;\, j=1,\dots,r-1}$ we propose two particular orderings $\beta_{\text{col}} : \mathcal{I} \to \{1, \dots, \binom{r}{2}\}$, and $\beta_{\text{row}} : \mathcal{I} \to \{1, \dots, \binom{r}{2}\}$, that are best illustrated by the two diagrams in Fig. 1. Obviously, the two orderings are mapped into each other by just transposing the diagrams with respect to the antidiagonal.

3.2. Local convergence analysis

The next result shows that our algorithm is locally a smooth map.

Theorem 3.1. Algorithm 3.2, *i.e., the mapping* $s : M_{\mathcal{L}_n} \to M_{\mathcal{L}_n}$ *is smooth locally around* A.

Proof. The algorithm is a composition of partial algorithmic steps $r_i : M_{\mathcal{L}_n} \to M_{\mathcal{L}_n}$, with $r_i(A) = A$ for all i. It therefore suffices to show smoothness for each r_i around the fixed point A. Typically, for one partial iteration step one has to compute the sub-block p of

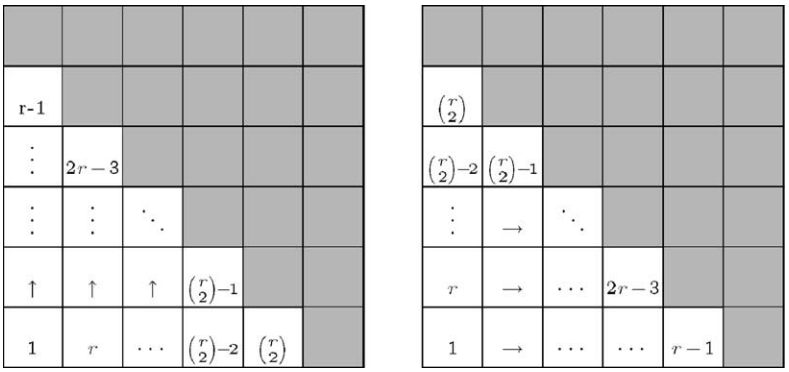

Fig. 1.

the unipotent lower block triangular matrix

$$L = \begin{bmatrix} I & 0 \\ p & I \end{bmatrix}$$

satisfying the equation

$$LXL^{-1} = \begin{bmatrix} I & 0 \\ p & I \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ -p & I \end{bmatrix} = \begin{bmatrix} * & * \\ -pX_{12}p & * \end{bmatrix},$$

i.e., p has to solve the Sylvester equation $pX_{11} + X_{21} - X_{22}p = 0$. By Assumption 2.1 and since X is close to A, the spectra of X_{11} and X_{22} will be disjoint and the solution of this Sylvester equation exists and is unique. Moreover, applying the implicit function theorem to the function $(X, p) \mapsto f(X, p)$, defined by $f(X, p) = pX_{11} + X_{21} - X_{22}p = 0$, implies that $X \mapsto p(X)$ is smooth around A. Hence all partial iteration steps are smooth and the result follows. □

Theorem 3.1 justifies to use calculus for proving higher order convergence of our algorithm. We show next that the first derivative of our algorithm s at the fixed point A vanishes identically implying quadratic convergence if the chosen ordering is either β_{row} or β_{col}.

Theorem 3.2. Algorithm 3.2 *converges locally quadratically fast if ordering* β_{row} *or* β_{col} *is chosen.*

Proof. We will show that the first derivative $\mathrm{D}s(A)$ of the algorithm s at the fixed point A vanishes identically if β_{col} or β_{row} is chosen. By the chain rule we therefore have to compute $\mathrm{D}r_{ij}(A)$ for all $i > j$ with $2 \le i \le l$ and $1 \le j \le m-1$. To be more precise, we have to study the effect of applying the linear map $\mathrm{D}r_{ij}(A) : T_A M_{\mathcal{L}_n} \to T_A M_{\mathcal{L}_n}$ to those tangent vectors $[l, A] \in T_A M_{\mathcal{L}_n}$ onto which the "earlier" linear maps $\mathrm{D}r_{pq}(A)$ have already been applied to

$$\mathrm{D}s(A) \cdot [l, A] = \mathrm{D}r_{\text{last}}(A) \cdots \mathrm{D}r_{\text{first}}(A) \cdot [l, A], \quad l \in \mathfrak{l}_n.$$

Notice that A is not only a fixed point of s but also one of each individual r_{ij}. Without loss of generality we make the simplifying assumption that the partitioning consists of 5×5 blocks. Typically, an $r_{ij}(X) = L_{ij}XL_{ij}^{-1}$ looks like

$$r_{ij}(X) = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & p_{ij} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \cdot X \cdot \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & -p_{ij} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}. \tag{21}$$

Therefore, $\mathrm{D}r_{ij}(A) \cdot [l, A] = \mathrm{D}(L_{ij}XL_{ij}^{-1}) \cdot [l, X]|_{X=A} = [L'_{ij}, A] + [l, A]$, where $L'_{ij} := \mathrm{D}L_{ij}(A) \cdot [l, A]$, and

typically

$$L'_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p'_{ij} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with $p'_{ij} := \mathrm{D}p_{ij}(X)\cdot[l, X]|_{X=A}$. We already know that p_{ij} solves a Sylvester equation, namely $p_{ij}(X)X_{jj} + X_{ij} - X_{ii}p_{ij}(X) = 0$, with $p_{ij}(X)|_{X=A} = 0$. Taking the derivative of this Sylvester equation acting on $[l, X]$ evaluated at $X = A$ gives

$$p'_{ij}(A)A_{jj} + [l, A]_{ij} - A_{ii}p'_{ij}(A) = 0. \tag{22}$$

An easy computation verifies that the commutator $[L'_{ij}, A]$ is of the following form:

$$[L'_{ij}, A] = \begin{bmatrix} 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & p'_{ij}A_{jj} - A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

i.e., it differs from zero only by the (ij)th block as well as by the blocks right to it and above it. By (22), we therefore obtain for the derivative of the (ij)th partial step r_{ij}:

$$\mathrm{D}r_{ij}(A)\cdot[l, A] = \underbrace{\begin{bmatrix} 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 \\ 0 & p'_{ij}A_{jj} - A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{[L'_{ij}, A]} + \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & [l, A]_{ij} & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{[l, A]}.$$

That is, by (22) the first derivative annihilates the (ij)th block, altering those blocks which are above or to the right to this (ij)th block, but it leaves invariant all the remaining blocks. Both ordering strategies then imply that after a whole iteration step all those blocks of the tangent vector $[l, A]$ lying below the main block diagonal are eliminated. We therefore can conclude that $\mathrm{D}r_{ij}(A)\cdot[l, A]$ is upper block triangular. Moreover, it follows from the proof of Lemma 2.2 that $\mathrm{D}r_{ij}(A)\cdot[l, A] = 0$ as well. Essentially, Assumption 2.1 implies that the only Lie algebra element of $\mathfrak{l}_n$ commuting with A into a upper block triangular matrix like A itself, is the zero matrix. The theorem then follows from a Taylor-type argument [14]:

$$\|X_{k+1} - A\| \leq \sup_{Z\in\bar{U}} \|\mathrm{D}^2 s(Z)\| \cdot \|X_k - A\|^2. \qquad \square$$

Quite naturally one might ask if the two orderings β_{row} and β_{col} are the only possible ones ensuring quadratic convergence. The answer is no, because "mixtures" of both strategies also suffice. As a corollary of Theorem 3.2 we obtain the following result.

Corollary 3.1. Algorithm 3.2 *is quadratic convergent if the ordering is specified by the following two rules. The integers* $1, \ldots, \binom{r}{2}$ *to be filled in are*: (i) *strictly increasing across each row, and* (ii) *are strictly increasing up each column.*

Remark 3.1. These possible orderings are related to Young tableaux, or to be more precise, to standard tableaux; see [9] for the connections between geometry of flag manifolds, representation theory of $\mathcal{GL}_n$, and calculus of tableaux.

It easily follows that for $r = 3$, β_{col} and β_{row} are the only two possible orderings ensuring quadratic convergence. For $r = 4$ there are already eight possible orderings together with their "conjugate" counterparts.

We did not comment yet on orderings which are definitely *not* leading to quadratic convergence. Generically, this is the case for any ordering which does not respect Corollary 3.1, but when the fixed point matrix A has some specific zero blocks above the diagonal, quadratic convergence may be recovered. For a more detailed discussion on this and for illustrative examples, we refer to [14,15].

4. Orthogonal transformations

For numerical reasons it makes more sense to use orthogonal transformations instead of unipotent

lower triangular ones. We therefore reformulate Algorithm 3.2 in terms of orthogonal transformations. The convergence analysis for this new algorithm will greatly benefit from the calculations we already did.

For convenience we assume for a while that $r = 5$. Given

$$L = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & p & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix},$$

a quite natural idea is to use instead of L the orthogonal Q-factor from L after performing Gram–Schmidt, i.e., $L = RQ$, to the rows of sub-blocks of L. We have with $N_l \doteq (I + p^\top p)^{-1/2}$ and $N_r \doteq (I + pp^\top)^{-1/2}$ the factorization:

$$L = R \cdot Q = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ & N_l & 0 & p^\top N_r & 0 \\ & & I & 0 & 0 \\ & & & N_r^{-1} & 0 \\ & & & & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & N_l & 0 & -N_l p^\top & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & N_r p & 0 & N_r & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}. \quad (23)$$

Before we proceed to formulate the orthogonal version of Algorithm 3.2 we need some preliminaries. Namely we have to fix the manifold such an algorithm is "living" on. Consider an "Iwasawa Decomposition" [13] of the Lie group $\mathcal{L}_n$. The set of orthogonal matrices Q coming from an RQ-decomposition as in (23) do in general not generate an orthogonal group with the ordinary matrix product as group operation. To see this we look at the simple 2×2 case

$$\begin{bmatrix} 1 & 0 \\ p & 1 \end{bmatrix} = \begin{bmatrix} (1+p^2)^{-1/2} & p(1+p^2)^{-1/2} \\ 0 & (1+p^2)^{1/2} \end{bmatrix} \cdot \begin{bmatrix} (1+p^2)^{-1/2} & -(1+p^2)^{-1/2} p \\ (1+p^2)^{-1/2} p & (I+p^2)^{-1/2} \end{bmatrix}.$$

Obviously, the set of orthogonal Q-matrices does include

$$\overline{Q} := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \text{but does not include}$$

$$\widetilde{Q} := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \overline{Q}^2.$$

Note that $\lim_{p \to \pm\infty} L \notin \mathcal{L}_2$. Nevertheless, we are able to construct at least locally the space on which an orthogonal version of Algorithm 3.2 can be defined. This construction will allow us to use a similar analysis to prove quadratic convergence. Consider an arbitrary element $L \in \mathcal{L}_n$ in a sufficiently small neighborhood $U_{\mathcal{L}_n}(I_n)$ of the identity I_n in $\mathcal{L}_n$, such that L can be parameterized by exponential coordinates of the second kind, cf. [19, p. 86]. Let

$$L = L_{\binom{r}{2}} \cdots L_1 = R_{\binom{r}{2}} Q_{\binom{r}{2}} \cdots R_1 Q_1. \quad (24)$$

Here the L_i are defined as in (21). Each L_i, for $i = 1, \ldots, \binom{r}{2}$, is represented as $L_i = \mathrm{e}^{l_i}$ with, e.g., using β_{row} as an ordering,

$$l_1 = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & \ddots & \vdots \\ p_1 & 0 & \cdots & 0 \end{bmatrix}, \quad l_2 = \begin{bmatrix} 0 & \cdots & & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & & & \ddots & \vdots \\ 0 & p_2 & 0 & \cdots & 0 \end{bmatrix}, \ldots \quad (25)$$

We can therefore study the map

$$\sigma : \mathcal{L}_n \supset U_{\mathcal{L}_n}(I_n) \to \mathcal{SO}_n,$$
$$L \mapsto Q_{\binom{r}{2}}(L) \cdot \cdots \cdot Q_1(L). \quad (26)$$

Note that $Q_i(I_n) = I_n$ for all $i = 1, \ldots, \binom{r}{2}$. The following series of lemmata characterizes the mapping σ.

Lemma 4.1. *The mapping σ defined by (26) is smooth.*

Proof. See the explicit form of the Q_i given as in (23). □

Lemma 4.2. *The mapping σ defined by (26) is an immersion at I_n.*

Proof. We have to show that the derivative $\mathrm{D}\sigma(I_n) : \mathfrak{l}_n \to \mathfrak{so}_n$ is injective. For arbitrary $l = \sum_{i=1}^{\binom{r}{2}} l_i \in \mathfrak{l}_n$ the following holds true:

$$\mathrm{D}\sigma(I_n)\cdot l = \sum_{i=1}^{\binom{r}{2}} \mathrm{D}Q_i(I_n)\cdot l_i = \sum_{i=1}^{\binom{r}{2}} (l_i - l_i^\top) = l - l^\top, \tag{27}$$

where we have used $(\mathrm{d}/\mathrm{d}\varepsilon)(I+\varepsilon^2 p^\top p)^{-1/2}|_{\varepsilon=0} = 0$ and $(\mathrm{d}/\mathrm{d}\varepsilon)(I+\varepsilon^2 pp^\top)^{-1/2}|_{\varepsilon=0} = 0$. Eq. (27) implies injectivity in an obvious manner. □

We can now apply the *immersion theorem*, cf. [1, p. 199].

Lemma 4.3. *The mapping σ as defined by (26) is a diffeomorphism of $U_{\mathcal{L}_n}(I_n)$ onto the image $\sigma(U_{\mathcal{L}_n}(I_n))$.*

Consider the isospectral manifold

$$M_{\mathcal{SO}_n} := \{X \in \mathbb{R}^{n\times n} | X = QAQ^\top,\ Q \in \mathcal{SO}_n\} \tag{28}$$

with $A \in V$ as above fulfilling Assumption 2.1. Define

$$\alpha : \sigma(U_{\mathcal{L}_n}(I_n)) \to M_{\mathcal{SO}_n}, \qquad Q \mapsto QAQ^\top. \tag{29}$$

Lemma 4.4. *The mapping α defined as in (29) is smooth.*

Proof. The result follows by the explicit construction of an arbitrary Q by using exponential coordinates of the second kind. □

Lemma 4.5. *The mapping α defined as in (29) is an immersion at I_n.*

Proof. We have to show that the derivative $\mathrm{D}\alpha(I_n) : T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n)) \to T_A M_{\mathcal{SO}_n}$ is injective. Arbitrary elements of the tangent space $T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n))$ have the form

$$\sum_{i=1}^{\binom{r}{2}} (l_i - l_i^\top) = l - l^\top,$$

whereas those of the tangent space $T_A M_{\mathcal{SO}_n}$ look like $[l - l^\top, A]$. To show injectivity of $\mathrm{D}\alpha(I_n) : T_{I_n}\sigma(U_{\mathcal{L}_n}(I_n)) \to T_A M_{\mathcal{SO}_n}$, defined by $l - l^\top \mapsto [l - l^\top, A]$, we partition $l - l^\top$ conformably with A, i.e.,

$$A = \begin{bmatrix} A_{11} & \cdots & A_{rr} \\ & \ddots & \vdots \\ & & A_{rr} \end{bmatrix},$$

$$l - l^\top = \begin{bmatrix} 0 & -p_{21}^\top & \cdots & -p_{r1}^\top \\ p_{21} & \ddots & & \vdots \\ \vdots & & \ddots & -p_{r,r-1}^\top \\ p_{r1} & \cdots & p_{r,r-1} & 0 \end{bmatrix}.$$

Note that $[l - l^\top, A]_{r1} = p_{r1}A_{11} - A_{rr}p_{r1}$. Assume the converse, i.e.,

$$[l - l^\top, A] = [\tilde{l} - \tilde{l}^\top, A] \tag{30}$$

holds for some $\tilde{l} \neq l$ with

$$\tilde{l} := \begin{bmatrix} 0 & & & \\ \tilde{p}_{21} & \ddots & & \\ \vdots & & \ddots & \\ \tilde{p}_{r1} & \cdots & \tilde{p}_{r,r-1} & 0 \end{bmatrix} \in \mathfrak{l}_n.$$

Looking at the $(r1)$-block of (30) implies $(p_{r1} - \tilde{p}_{r1})A_{11} - A_{rr}(p_{r1} - \tilde{p}_{r1}) = 0$, which by Assumption 2.1 implies in turn that $p_{r1} = \tilde{p}_{r1}$. Now we use induction on the sub-diagonal blocks going from the lower left corner block of (30) to the first sub-diagonal blocks. Applying recursively the same arguments on the $(r-1, 1)$-block of (30), as well as on the $(r2)$-block of (30), then implies $p_{r2} = \tilde{p}_{r2}$ and $p_{r-1,1} = \tilde{p}_{r-1,1}$. Finally, we get $[l - l^\top, A] = [\tilde{l} - \tilde{l}^\top, A] \Rightarrow l = l^\top$, a contradiction. Therefore, $\mathrm{D}\alpha(I_n)$ is injective, hence α is an immersion at I_n. □

Consequently, we have the following lemma.

Lemma 4.6. *The composition mapping $\alpha \circ \sigma : U_{\mathcal{L}_n}(I_n) \to M_{\mathcal{SO}_n}$ is a diffeomorphism of $U_{\mathcal{L}_n}(I_n)$ onto the image $(\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n))$.*

4.1. The algorithm

The following algorithm will be analyzed for A satisfying Assumption 2.1. We denote in the sequel $M := (\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n))$. We are given an

$X \in M$ close to A and we consider the index set $\mathcal{I} := \{(ij)\}_{i=2,\dots,r;\, j=1,\dots,r-1}$ and fix an ordering β. For convenience we again rename double indices in the description of the algorithm by simple ones by means of $X_{ij} \mapsto X_{\beta((ij))}$ respecting the ordering β.

Algorithm 4.1 (Orthogonal Sylvester sweep). Given an $X \in (\alpha \circ \sigma)(U_{\mathcal{L}_n}(I_n)) = M$, define

$$
\begin{aligned}
X_k^{(1)} &:= Q_1 X Q_1^\top \\
X_k^{(2)} &:= Q_2 X_k^{(1)} Q_2^\top \\
&\vdots \\
X_k^{(\binom{r}{2})} &:= Q_{\binom{r}{2}} X_k^{(\binom{r}{2}-1)} Q_{\binom{r}{2}}^\top,
\end{aligned}
$$

where for $l = 1, \dots, \binom{r}{2}$ the transformation matrix $Q_l \in \mathcal{SO}_n$ differs from the identity matrix I_n only by four sub-blocks. Namely, the

$$
\begin{aligned}
&(jj)\text{th block equals } (I + p^\top p)^{-1/2}, \\
&(ji)\text{th block equals } -(I + p^\top p)^{-1/2} p^\top, \\
&(ij)\text{th block equals } (I + pp^\top)^{-1/2} p, \\
&(ii)\text{th block equals } (I + pp^\top)^{-1/2}.
\end{aligned}
$$

Here $\beta((ij)) = l$ and $p_l \in \mathbb{R}^{n_j \times n_i}$ solves the Sylvester equation

$$
p^{(l)} (X_k^{(l-1)})_{jj} - (X_k^{(l-1)})_{ii} p^{(l)} + (X_k^{(l-1)})_{ij} = 0.
$$

The overall algorithm consists of the following iteration of orthogonal sweeps.

Algorithm 4.2 (Orthogonal refinement of subspace estimates).

- Let $X_0, \dots, X_k \in M$ be given for $k \in \mathbb{N}_0$.
- Define the recursive sequence $X_k^{(1)}, \dots, X_k^{(\binom{r}{2})}$ as above (sweep).
- Set $X_{k+1} := X_k^{(\binom{r}{2})}$. Proceed with the next sweep.

4.2. Local convergence analysis

Analogous to Theorem 3.1 we have the following theorem.

Theorem 4.1. Algorithm 4.2, *i.e., the mapping* $s : M \to M$ *is smooth locally around* A.

Proof. The algorithm is a composition of partial algorithmic steps r_i. Smoothness of these partial algorithmic steps follows from the smoothness of each p_i already shown. □

Theorem 4.2. Algorithm 4.2 *converges locally quadratically fast if for working off the partial algorithmic steps an ordering is chosen which respects* Corollary 3.1.

Proof. We will compute $\mathrm{D}r_{ij}(A)$ for all $i > j$ with $2 \le i \le l$ and $1 \le j \le m-1$. Without loss of generality we may assume that the partitioning consists of 5×5 blocks. Typically, a transformation matrix Q_{ij} for $r_{ij}(X) = Q_{ij} X Q_{ij}^\top$ looks like

$$
Q_{ij}(X) = \begin{bmatrix}
I & 0 & 0 & 0 & 0 \\
0 & S_{ij}(X) & 0 & -S_{ij}(X) p_{ij}^\top(X) & 0 \\
0 & 0 & I & 0 & 0 \\
0 & T_{ij}(X) p_{ij}(X) & 0 & T_{ij}(X) & 0 \\
0 & 0 & 0 & 0 & I
\end{bmatrix}, \tag{31}
$$

where $S_{ij}(X) := (I + p^\top(X) p(X))^{-1/2}$, and $T_{ij}(X) := (I + p(X) p^\top(X))^{-1/2}$. Moreover, $S_{ij}(A) = I_{n_i}$ and $T_{ij}(A) = I_{n_j}$. An arbitrary $\Omega \in \mathfrak{so}_n / (\mathfrak{so}_{n_1} \oplus \cdots \oplus \mathfrak{so}_{n_r})$ looks like

$$
\Omega = \begin{bmatrix}
0 & -\Omega_{21}^\top & \cdots & -\Omega_{r1}^\top \\
\Omega_{21} & \ddots & & \vdots \\
\vdots & & \ddots & -\Omega_{r,r-1}^\top \\
\Omega_{r1} & \cdots & \Omega_{r,r-1} & 0
\end{bmatrix}.
$$

The derivative of one partial algorithmic step acting on $[\Omega, A] \in T_A M$ is as $\mathrm{D}r_{ij}(A) \cdot [\Omega, A] = [Q'_{ij}, A] + [\Omega, A]$, where $Q'_{ij} := \mathrm{D}Q_{ij}(A) \cdot [\Omega, A]$, and typically

$$
Q'_{ij} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & S'_{ij}(A) & 0 & -(p_{ij}^\top)'(A) & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & p'_{ij}(A) & 0 & T'_{ij}(A) & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

with $p'_{ij}(A) := \mathrm{D}p_{ij}(X) \cdot [\Omega, X]|_{X=A}$. We already know that p_{ij} solves a Sylvester equation, namely

$p_{ij}(X)X_{jj}+X_{ij}-X_{ii}p_{ij}(X)=0$, with $p_{ij}(X)|_{X=A}=0$. Taking the derivative of this Sylvester equation acting on $[\Omega, A]$ gives

$$p'_{ij}(A)A_{jj}+[\Omega, A]_{ij}-A_{ii}p'_{ij}(A)=0. \tag{32}$$

An easy computation verifies that the commutator $[Q'_{ij}, A]$ is of the following form:

$$[Q'_{ij}, A]=\begin{bmatrix} 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & p'_{ij}A_{jj}-A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

i.e., the (ij)th block equals $p'_{ij}A_{jj}-A_{ii}p'_{ij}$ and columns of blocks to the left as well as rows of blocks below are zero. By (32), we therefore obtain for the derivative of the (ij)th partial step r_{ij}:

$$\mathrm{D}r_{ij}(A)\cdot[\Omega, A]=\underbrace{\begin{bmatrix} 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & p'_{ij}A_{jj}-A_{ii}p'_{ij} & * & * & * \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{[Q'_{ij},A]}+\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & [\Omega, A]_{ij} & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{[\Omega,A]}.$$

That is, by (32) the first derivative annihilates the (ij)th block, altering eventually those blocks which are above, to the right, or a combination of both, to this (ij)th block, but it leaves invariant all the remaining blocks. Apparently, all ordering strategies respecting Corollary 3.1 ensure that after a whole iteration step all those blocks lying below the main block diagonal are eliminated. We therefore can conclude that $\mathrm{D}r_{ij}(A)\cdot[\Omega, A]$ is strictly upper block triangular. Again we can even conclude more, namely $\mathrm{D}r_{ij}(A)\cdot[\Omega, A]=0$. Following the argumentation in the proof of Lemma 2.2, essentially, Assumption 2.1 ensures that the only element of $\mathfrak{so}_n/(\mathfrak{so}_{n_1}\oplus\cdots\oplus\mathfrak{so}_{n_r})$, which commutes with A into a upper block triangular matrix, is the zero matrix. This can also be seen from the fact that the above Ω is of the type $l-l^{\top}$, where $l\in\mathfrak{l}_n$. The result then follows. □

5. Computational aspects

In this section we look at computational aspects of the basic recurrences described in Section 3. We first focus on the complexity of the solution of the Sylvester equation needed both in the Sylvester sweep and in the orthogonal Sylvester sweep algorithm. Using the notation of those sections, we want to solve the equation

$$PX_{11}+X_{21}-X_{22}P=0, \tag{33}$$

where we will assume $X_{ij}\in\mathbb{R}^{n_i\times n_j}$ and $n_2\geq n_1$. The recommended method here is to use the Hessenberg–Schur method described in [11]. In this method one computes orthogonal similarity transformations U_i, $i=1,2$ such that $S\doteq U_1^{\mathrm{T}}X_{11}U_1$ is in real Schur form (i.e. an upper block triangular matrix with 1×1 or 2×2 blocks on diagonal) and $H\doteq U_2^{\mathrm{T}}X_{22}U_2$ is in Hessenberg form (i.e. a matrix with zeros below the first sub-diagonal). Defining also $F\doteq U_2^{\mathrm{T}}X_{21}U_1$ and $Z\doteq U_2^{\mathrm{T}}PU_1$ we obtain an equivalent equation in Z:

$$ZS+F-HZ=0, \tag{34}$$

which is easier to solve. The special forms of H and S indeed yield simple recurrences for the columns z_k, $k=1,\ldots,n_1$ of Z. If $s_{k+1,k}=0$ then z_k is obtained from previously computed columns using

$$(H-s_{k,k}I)z_k=f_k+\sum_{j=1}^{k-1}s_{j,k}z_j \tag{35}$$

(which can be solved in n_2^2 flops) and if $s_{k+1,k}\neq0$ then columns z_k and z_{k+1} are similarly computed from

$$H[z_k|z_{k+1}]-[z_k|z_{k+1}]\begin{bmatrix} s_{k,k} & s_{k,k+1} \\ s_{k+1,k} & s_{k+1,k+1} \end{bmatrix}=[f_k|f_{k+1}]+\sum_{j=1}^{k-1}[s_{j,k}z_j|s_{j,k+1}z_j]$$

(which can be solved in $6n_2^2$ flops). It is shown in [11] that the overall algorithm, including the back transformation $P=U_2ZU_1^{\mathrm{T}}$ requires $(5/3)n_2^3+10n_1^3+$

$5n_2^2n_1+(5/2)n_1^2n_2$ flops. It is also shown in that paper that this method is weakly stable, which essentially guarantees that the computed solution is as accurate as what could be expected from any algorithm solving (33). Let us point out here that the computation of the solutions of each Sylvester equation of the Sylvester sweep is negligible in comparison to the application of the similarity transformations, provided the sizes n_j of the blocks are small compared to n, the dimension of the matrix they operate on. Indeed, one similarity transform implying a matrix P of dimension $n_i \times n_j$ requires $4n \cdot n_i \cdot n_j$ flops (for multiplying P with an $n_j \times n$ matrix from the right and with an $n \times n_i$ matrix from the left). A total Sylvester sweep therefore requires

$$\sum_i \sum_j 4n(n_i \cdot n_j) \leq 4n \sum_i n_i \sum_j n_j = 4n^3$$

flops. For the more concrete case where all $n_i = n_j = 2$ (i.e. when we calculate a Schur-like form) the complexity can be calculated more precisely and is $2n^3$ flops. Clearly, the computation of the solutions of the $n^2/8$ Sylvester equations of size 2×2 is of the order of n^2 flops and hence an order of magnitude less.

For the orthogonal Sylvester sweep one has first to construct the orthogonal transformation Q described in (23). A simple and reliable procedure for reasonable sized matrices P is to use its singular value decomposition. Assuming $P \in \mathbb{R}^{n_i \times n_j}$ with $n_i \leq n_j$, we have $P = U[\,T \quad 0\,]V^\top$, where $U \in \mathbb{R}^{n_i \times n_i}$, $V \in \mathbb{R}^{n_j \times n_j}$ and $T = \mathrm{diag}\{\tan\theta_1, \ldots, \tan\theta_{n_j}\}$ is diagonal and positive. It then follows that the four non-trivial blocks of Q are given by:

$$N_l = VCV^\top, \qquad N_lP = V[\,S \quad 0\,]U^\top,$$
$$N_rP = U\begin{bmatrix} S \\ 0 \end{bmatrix}V^\top, \qquad N_r = U\begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}U^\top,$$

where

$$C = \mathrm{diag}\{\cos\theta_1, \ldots, \cos\theta_{n_j}\},$$
$$S = \mathrm{diag}\{\sin\theta_1, \ldots, \sin\theta_{n_j}\}.$$

The computational cost of constructing these transformations is again negligible with respect to their application. For $n_i = n_j = 2$ each one essentially amounts to a 4×4 row transformation on a $4 \times n$ block row of A and 4×4 column transformation on an $n \times 4$ block column of A. This requires $32n$ flops and since there are $n^2/8$ of those per sweep, the orthogonal sweep requires about $4n^3$ flops, i.e., about the double of a standard Sylvester sweep. Since both algorithms are quadratically convergent, one can say that these algorithms require $O(n^3)$ flops to converge (when starting from a form that is almost block diagonal). We point out that the standard *QR* algorithm is of the same order of complexity.

We should mention that the methods described in this paper extend to the generalized eigenvalue problem in a straightforward manner. Instead of one Riccati or one Sylvester equation one has then to solve a system of two coupled ones. Everything is similar under an equivalent assumption on the spectra of the sub-blocks.

If the considered matrix is symmetric, our method is related to [12]. There, the so-called approximate Givens (or Jacobi) transformations are developed which essentially approximate an exact rotation to zero out a matrix entry. It is not clear though that our algorithm has an interpretation as a Jacobi-type method in the general non-symmetric case.

The appeal of our method is to be found in time-varying eigenvalue problems (as occurring, e.g. in tracking problems of slowly varying phenomena). At each step one then has a nearly block triangular matrix to start with, which needs to be updated in order to reduce it again to block triangular form via similarity transformations. One should also point out that in terms of efficient implementation on parallel architectures, our method should have much the same properties as Jacobi-like methods.

References

[1] R. Abraham, J.E. Marsden, T. Ratiu, Manifolds, Tensor Analysis, and Applications, 2nd ed., Springer, New York, 1988.

[2] A. Bunse-Gerstner, H. Fassbender, On the generalized Schur decomposition of a matrix pencil for parallel computation, SIAM J. Sci. Statist. Comput. 12 (4) (1991) 911–939.

[3] J.-P. Charlier, P. Van Dooren, A Jacobi-like algorithm for computing the generalized Schur form of a regular pencil, J. Comput. Appl. Math. 27 (1989) 17–36.

[4] F. Chatelin, Simultaneous Newton's iteration for the eigenproblem, Computing (Suppl. 5, Defect Correction Methods, Oberwolfach, 1983) (1984) 67–74.

[5] J. Demmel, Three methods for refining estimates of invariant subspaces, Computing 38 (1987) 43–57.

[6] L. Dieci, A. Papini, Continuation of eigendecompositions, J. Comput. Appl. Math. 19 (7) (2003) 1225–1237.
[7] L. Dieci, M. Friedman, Continuation of invariant subspaces, Preprint, 2001.
[8] J.J. Dongarra, C.B. Moler, J.H. Wilkinson, Improving the accuracy of computed eigenvalues and eigenvectors, SIAM J. Numer. Anal. 20 (1) (1983) 23–45.
[9] W. Fulton, Young Tableaux, LMS Student Texts, vol. 35, Cambridge University Press, Cambridge, 1997.
[10] C.G. Gibson, Singular Points of Smooth Mappings, Pitman, Boston, 1979.
[11] G. Golub, S. Nash, C. Van Loan, A Hessenberg–Schur method for the problem $AX + XB = C$, IEEE Trans. Automat. Contr. AC-24 (1979) 909–913.
[12] J. Götze, Orthogonale Matrixtransformationen, Parallele Algorithmen, Architekturen und Anwendungen, Oldenbourg, München, 1995 (in German).
[13] S. Helgason, Differential Geometry, Lie Groups, and Symmetric Spaces, Academic Press, Boston, 1978.
[14] K. Hüper, A calculus approach to matrix eigenvalue algorithms, Habilitation Address, Würzburg University, Germany, 2002.
[15] K. Hüper, A dynamical system approach to matrix eigenvalue algorithms, in: J. Rosenthal, D.S. Gilliam (Eds.), Mathematical Systems Theory in Biology, Communication, and Finance, IMA, vol. 134, Springer, New York, 2003, pp. 257–274.
[16] M.T. Nair, Computable error estimates for Newton's iterations for refining invariant subspaces, Indian J. Pure Appl. Math. 21 (12) (1990) 1049–1054.
[17] G.W. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, SIAM Rev. 15 (4) (1973) 727–764.
[18] G.W. Stewart, A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix, SIAM J. Sci. Statist. Comput. 6 (4) (1986) 853–864.
[19] V.S. Varadarajan, Lie Groups, Lie Algebras, and their Representations, Springer, New York, 1984.

K. Hüper was born in Karlsruhe, Germany, in 1962. He received his Diploma Physicist and the Doctor rer.nat. degrees from Munich University of Technology, Germany, in 1990 and 1996, respectively. In 2002, he finished his Habilitation thesis in Mathematics. From 1990 to 1995, he was a teaching and research assistant at the Institute for Circuit Theory and Signal Processing, Munich University of Technology. In 1995, he was a teaching and research assistant at the Institute for Algorithms and Cognitive Systems, University of Karlsruhe, Germany. In 1996, he spent 2 months as a visiting associate at the Research School of Information Sciences and Engineering in the Institute of Advanced Studies, Australian National University, Canberra. In 2000, he was a postdoctoral researcher at the Department of Mathematics and the Institute for Systems and Robotics, University of Coimbra, Portugal. From 1996 to 2002, he was teaching and research assistant at the Institute of Mathematics, Würzburg University, Germany. Since 2003, he is Privatdozent (Senior Lecturer) at the Institute of Mathematics, Würzburg University, Germany. His research interests are in the area of Lie theory and geometric optimization on manifolds. His most recent interests are the application of dynamical systems methods to matrix algorithms, interpolation problems on manifolds, and the mathematics behind nuclear-magnetic-resonance experiments. He was a Co-guest Editor of a special issue of *Systems and Control Letters*. Dr. Hüper is member of the AMS and the German GAMM.

P. Van Dooren received his engineering degree in computer science and the doctoral degree in applied sciences, both from the Katholieke Universiteit te Leuven, Belgium, in 1974 and 1979, respectively. He held research and teaching positions at the Katholieke Universiteit te Leuven (1974–1979), the University of Southern California (1978–1979), Stanford University (1979–1980), the Australian National University (1984), Philips Research Laboratory Belgium (1980–1991), the University of Illinois at Urbana-Champaign (1991–1994), and the Universite Catholique de Louvain (1980–1991, 1994 to now), where he is currently a professor of Mathematical Engineering. Dr. Van Dooren received his Householder Award in 1981 and the Wilkinson Prize of Numerical Analysis and Scientific Computing in 1989. He is an Associate Editor of Journal of Computational and Applied Mathematics, Numerische Mathematik, SIAM Journal on Control and Optimization, Applied Mathematics Letters, Linear Algebra and its Applications, Journal of Numerical Algorithms, the Electronic Transactions of Numerical Analysis, Applied and Computational Control, Signals and Systems, Mathematics of Control, Signals and the SIAM Journal on Matrix Analysis and Applications. His main interests lie in the areas of numerical linear algebra, systems and control theory, digital signal processing, and parallel algorithms.