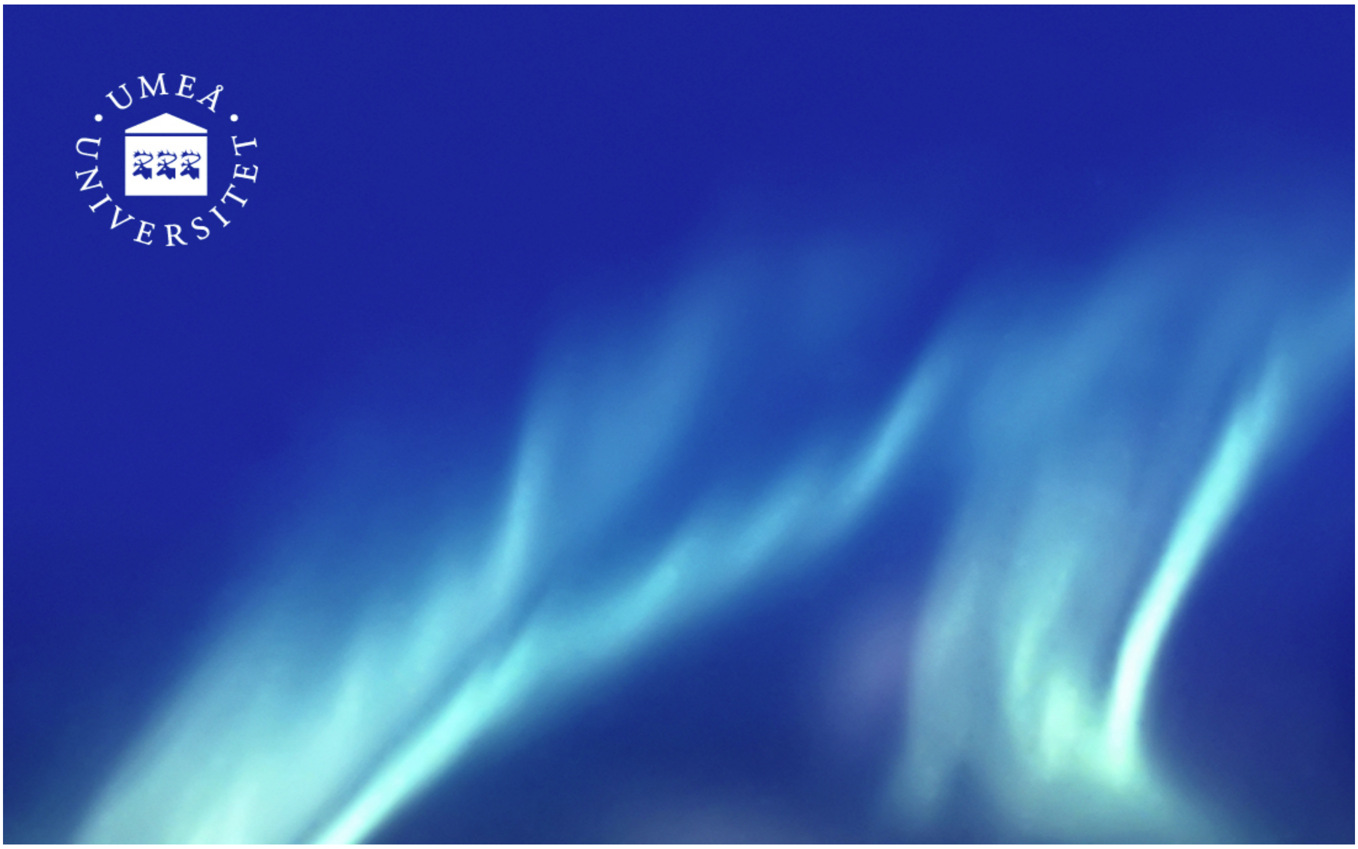

Codimension computations of congruence orbits of matrices, symmetric and skew-symmetric matrix pencils using Matlab

by

Andrii Dmytryshyn, Stefan Johansson, and Bo Kågström

UMINF-13/18

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

Codimension computations of congruence orbits of matrices, symmetric and skew-symmetric matrix pencils using Matlab

Andrii Dmytryshyn* Stefan Johansson* Bo Kågström*

Abstract

Matlab functions to work with the canonical structures for congruence and *congruence of matrices, and for congruence of symmetric and skew-symmetric matrix pencils are presented. A user can provide the canonical structure objects or create (random) matrix example setups with a desired canonical information, and compute the codimensions of the corresponding orbits: if the structural information (the canonical form) of a matrix or a matrix pencil is known it is used for the codimension computations, otherwise they are computed numerically. Some auxiliary functions are provided too. All these functions extend the Matrix Canonical Structure Toolbox.

AMS classification: 15A21, 15A22, 15A24

Keywords: Congruence; *congruence; Symmetric matrix pencils; Skew-symmetric matrix pencils; Orbits; Codimension; MATLAB

1 Introduction

This paper presents software to work with the canonical structures for congruence and *congruence of matrices, as well as congruence of symmetric and skew-symmetric matrix pencils. It also recalls the associated canonical forms and reviews recent theoretical results about the codimension computations. The software includes functions that create canonical structure objects or

*Department of Computing Science and HPC2N, Umeå University, SE-901 87 Umeå, Sweden. E-mails: `andrii@cs.umu.se`, `stefanj@cs.umu.se`, `bokg@cs.umu.se`

(random) matrix example setups with a desired canonical information, the functions that compute the codimensions of the corresponding orbits, and a number of auxiliary functions. If the canonical forms of the matrices or the matrix pencils are known (or specified) we use the associated structural information for the codimension computations. Otherwise, we determine the codimensions numerically by computing the rank and nullity of Kronecker product matrices associated with the problems. These are matrix representation of the tangent space of the associated orbits. Motivations for computing codimensions of these matrix structures can be found in [2, 3, 5, 6, 7, 8, 9, 10]. Analogous functions for matrix orbits up to similarity, matrix pencils up to strict equivalence, controllability and observability pairs up to feedback equivalence are provided by the *Matrix Canonical Structure* (MCS) Toolbox for Matlab[1] [18], while the theoretical backgrounds and motivations are given in [4, 11, 12, 13]. There also exists a Python implementation for computing codimensions of generalized matrix products (see [21]).

Whenever the canonical forms of matrices or matrix pencils are known, the explicit formulas for computing codimensions from the canonical structure information derived in [2, 3, 9, 10] should be applied because this computation is always exact and fast for problems of any sizes.

In this paper, we present new Matlab functions that extend MCS Toolbox with routines for computing codimensions of congruence and *congruence orbits of matrices, and congruence orbits of symmetric and skew-symmetric matrix pencils.

The rest of the paper is organized as follows. Theoretical background is presented in Section 2. Subsections 2.1 and 2.2 are devoted to the computations of the codimensions using the canonical information. Subsection 2.3 explains the numerical codimension computations using the associated Kronecker product matrices. In Section 3, we give detailed instructions on using the implemented functions and illustrate them by several examples. Finally, in the appendices we present a table summary of the main Matlab functions with compendious descriptions and the possible types of canonical blocks.

2 Theoretical background

In this section, we introduce and review theoretical results needed, e.g., canonical forms and the notion of codimension. We include these results

[1]Matlab is a registered trademark of The MathWorks, Inc.

to make the paper self-contained and to introduce the notation used (for more details and proofs see [2, 3, 5, 6, 7, 8, 9, 10, 17, 22]).

2.1 Matrices under congruence and *congruence

Let A be an $n \times n$ matrix over the field of complex numbers, denoted by $\mathbb{C}$ and $GL_n(\mathbb{C})$ be a group of $n \times n$ nonsingular complex matrices. Consider the congruence transformation

$$A \mapsto C^T AC,$$

where $C \in GL_n(\mathbb{C})$. The set of matrices congruent to A forms a manifold in the complex n^2 dimensional space. This manifold is the orbit of A under the action of congruence:

$$\text{orbit}(A) = \{C^T AC : C \in GL_n(\mathbb{C})\}.$$

The vector space

$$T(A) := \{X^T A + AX : X \in \mathbb{C}^{n\times n}\}$$

is the tangent space to the congruence class of A at the point A since

$$(I + \varepsilon X)^T A(I + \varepsilon X) = A + \varepsilon(X^T A + AX) + \varepsilon^2 X^T AX$$

for all n-by-n matrices X and each $\varepsilon \in \mathbb{C}$.

The *dimension of the orbit* of A is the dimension of its tangent space at the point A; it is well-defined because the dimensions of the tangent spaces at every point of the orbit are equal (e.g., see [1]). The *codimension of the orbit* A is the dimension of the normal space of its orbit at the point A which is equal to n^2 minus the dimension of the orbit. Note that it is also equal to the number of linearly independent solutions of the matrix equation

$$X^T A + AX = 0; \tag{1}$$

for more details see [2].

By the *congruence transformation we mean

$$A \mapsto C^* AC,$$

where $C \in GL_n(\mathbb{C})$ and C^* denotes the conjugate transpose of the matrix C. Note that the definitions given above for the congruence orbits remain analogous for the *congruence orbits, but the dimensions and codimensions are defined over $\mathbb{R}$ (*congruence orbits are manifolds over $\mathbb{R}$, not over $\mathbb{C}$).

Define a direct sum of two complex matrices A and B as follows

$$A \oplus B := \left[\begin{array}{c:c} A & 0 \\ \hdashline 0 & B \end{array}\right].$$

We recall the canonical forms of matrices under congruence and *congruence. These results were proven in [17]. For each positive integer m define the m-by-m unit matrix I_m and the m-by-m matrices

$$J_m(\lambda) := \begin{bmatrix} \lambda & 1 & & 0 \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{bmatrix}, \quad \text{and} \quad \mathit{\Gamma}_m := \begin{bmatrix} 0 & & & & \iddots \\ & & & -1 & \iddots \\ & & 1 & 1 & \\ & -1 & -1 & & \\ 1 & 1 & & & 0 \end{bmatrix},$$

where $J_m(\lambda)$ is a Jordan block associated with the eigenvalue λ. We use the following canonical form of complex matrices for congruence and *congruence.

Theorem 2.1 [17]. *Each* square complex matrix is congruent *to a direct sum, uniquely determined up to permutation of summands, of canonical matrices of three types*

$$J_p(0), \quad \mathit{\Gamma}_q, \quad \textit{and} \quad W_r(\lambda) := \begin{bmatrix} 0 & I_r \\ J_r(\lambda) & 0 \end{bmatrix} \; (\lambda \neq 0, \; \lambda \neq (-1)^{r+1}), \tag{2}$$

where $\lambda \in \mathbb{C}$ is determined up to replacement by λ^{-1}.

Example 1

The 20×20 canonical matrix $\Gamma_3 \oplus W_3(5) \oplus W_4(5) \oplus J_3(0)$, presented as the direct sum of the blocks (2), can be written explicitly as follows (where $\cdot$ denote zeroes):

$$
\left[\begin{array}{ccc|cccccc|cccccccc|ccc}
0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & -1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & 0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 5 & 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 5 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & 5 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 5 & 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0
\end{array}\right].
$$

Theorem 2.2 [17]. *Each* square complex matrix is *congruent *to a direct sum, uniquely determined up to permutation of summands, of canonical matrices of the three types*

$$J_p(0), \quad \mu\Gamma_q \ (|\mu| = 1), \quad \textit{and} \quad {}^*W_r(\lambda) := \begin{bmatrix} 0 & I_r \\ J_r(\lambda) & 0 \end{bmatrix} \ (|\lambda| > 1), \qquad (3)$$

where $\lambda, \mu \in \mathbb{C}$.

Example 2

The 20×20 canonical matrix $\Gamma_3 \oplus i\Gamma_6 \oplus {}^*W_4(5) \oplus J_3(0)$, presented as the direct sum of the blocks (3), can be written explicitly as follows.

$$
\left[\begin{array}{ccc|cccccc|cccccccc|ccc}
0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & -1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & -i & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & i & i & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & 0 & -i & -i & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & 0 & i & i & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 0 & -i & -i & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & i & i & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 5 & 1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0
\end{array}\right].
$$

Using the canonical structure information, the following two theorems provide explicit formulas for computing the codimensions of the congruence and *congruence orbits.

Theorem 2.3 [2]. *Let* $A \in \mathbb{C}^{n\times n}$ *and*

$$A_{\text{can}} = \bigoplus_{l=1}^{a} J_{p_l}(0) \oplus \bigoplus_{j=1}^{b} \mathit{\Gamma}_{q_j} \oplus \bigoplus_{i=1}^{c} W_{r_i}(\lambda_i), \quad p_1 \geq p_2 \geq \cdots \geq p_a \tag{4}$$

be its canonical form for congruence. The codimension of the orbit of A under congruence *(denoted by* $\operatorname{cod}(A)$*) can be computed as the sum*

$$\operatorname{cod}(A) = c_J + c_\Gamma + c_W + c_{JJ} + c_{\Gamma\Gamma} + c_{WW} + c_{W\Gamma} + c_{WJ} + c_{\Gamma J} \tag{5}$$

whose summands correspond to

- *the direct summands of* (4)*:*

$$c_J := \sum_{i=1}^{a} \left\lceil \frac{p_i}{2} \right\rceil, \qquad c_\Gamma := \sum_{i=1}^{b} \left\lfloor \frac{q_i}{2} \right\rfloor, \qquad c_W := \sum_{i=1}^{c} r_i + 2\sum_{j} \left\lceil \frac{r_j}{2} \right\rceil;$$

  where the second sum in c_W *is taken over the blocks* $W_{r_j}((-1)^{r_j})$ *in* A_{can}*;*

- *the pairs of direct summands of* (4) *of the same type:*

$$c_{JJ} := \sum_{\substack{i,j=1\\ i<j}}^{a} \operatorname{inter}(J_{p_i}(0), J_{p_j}(0)),$$

where

$$\operatorname{inter}(J_{p_i}(0), J_{p_j}(0)) := \begin{cases} p_j & \textit{if } p_j \textit{ is even,}\\ p_i & \textit{if } p_j \textit{ is odd and } p_i \neq p_j,\\ p_i + 1 & \textit{if } p_j \textit{ is odd and } p_i = p_j; \end{cases}$$

$$c_{\Gamma\Gamma} := \sum_{i\leqslant j} \min(q_i, q_j),$$

where the sum is taken over all pairs of blocks $(\Gamma_{q_i}, \Gamma_{q_j}), i \leqslant j,$ *in* A_{can} *such that* q_i *and* q_j *have the same parity;*

$$c_{WW} := 2\sum \min(r_i, r_j) + 4\sum \min(r_s, r_t),$$

where the first sum is taken over all pairs of blocks $(W_{r_i}(\lambda_i), W_{r_j}(\lambda_j)), i \leqslant j,$ *in* A_{can} *such that* $\lambda_i \neq \lambda_j$ *and* $\lambda_i\lambda_j = 1$ *or* $\lambda_i = \lambda_j \neq \pm 1$*, and the second sum is taken over all pairs* $(W_{r_s}(\lambda_s), W_{r_t}(\lambda_t)), s \leqslant t,$ *of blocks in* A_{can} *such that* $\lambda_s = \lambda_t = \pm 1$;

- *the pairs of direct summands of* (4) *of different types:*

$$c_{W\Gamma} := 2\sum \min(k, l),$$

where the sum is taken over all pairs $(\Gamma_k, W_l((-1)^{k+1}))$ *of blocks in* A_{can};

$$c_{WJ} := 2N_{odd}\sum_{i=1}^{c} r_i, \quad \textit{and} \quad c_{\Gamma J} := N_{odd}\sum_{i=1}^{b} q_i,$$

where N_{odd} *is the number of* J *blocks with odd size in* A_{can}.

Note that the codimensions of the congruence orbits of matrices can also be obtained by computing the number of independent parameters in the miniversal deformations [7].

Example 3

The codimension of the 20×20 matrix $\Gamma_3 \oplus W_4(5) \oplus W_3(5) \oplus J_3(0)$ from Example 1 can be computed as follows:

$$\begin{aligned}\operatorname{cod}(A) &= c_J + c_\Gamma + c_W + c_{WW} + c_{W\Gamma} + c_{WJ} + c_{\Gamma J}\\ &= 2 + 1 + 7 + 6 + 0 + 14 + 3\\ &= 33.\end{aligned}$$

Theorem 2.4 [3]. *Let $A \in \mathbb{C}^{n\times n}$ and*

$$A_{\text{can}} = \bigoplus_{i=1}^{a} J_{p_l}(0) \oplus \bigoplus_{j=1}^{b} \mu_j \Gamma_{q_j} \oplus \bigoplus_{l=1}^{c} {}^*W_{r_i}(\lambda_i), \quad p_1 \geq p_2 \geq \cdots \geq p_a \tag{6}$$

*be its canonical form for *congruence. The* codimension of the orbit of A under *congruence *(denoted by* $\operatorname{cod}^*(A)$*) can be computed as the sum*

$$\operatorname{cod}^*(A) = c_J + c_{\mu\Gamma} + c_{*W} + c_{JJ} + c_{\mu\Gamma\mu\Gamma} + c_{*W*W} + c_{*W\mu\Gamma} + c_{*WJ} + c_{\mu\Gamma J} \tag{7}$$

whose summands correspond to

- *the direct summands of* (6)*:*

$$c_J := 2\sum_{i=1}^{a} \left\lceil \frac{p_i}{2} \right\rceil, \qquad c_{\mu\Gamma} := \sum_{i=1}^{b} q_i, \qquad c_{*W} := 2\sum_{i=1}^{c} r_i;$$

- *the pairs of direct summands of* (6) *of the same type:*

$$c_{JJ} := \sum_{\substack{i,j=1\\ i<j}}^{a} \operatorname{inter}(J_{p_i}(0), J_{p_j}(0)),$$

where

$$\operatorname{inter}(J_{p_i}(0), J_{p_j}(0)) := \begin{cases} 2p_j & \text{if } p_j \text{ is even,}\\ 2p_i & \text{if } p_j \text{ is odd and } p_i \neq p_j,\\ 2(p_i+1) & \text{if } p_j \text{ is odd and } p_i = p_j; \end{cases}$$

$$c_{\mu\Gamma\mu\Gamma} := 2\sum_{i<j} \min(q_i, q_j),$$

where the sum is taken over all pairs of blocks $(\mu_i \mathit{\Gamma}_{q_i}, \mu_j \mathit{\Gamma}_{q_j}), i < j$, *in* A_{can} *such that: (a)* q_i *and* q_j *have the same parity and* $\mu_i = \pm\mu_j$, *and (b)* q_i *and* q_j *have different parity and* $\mu_i = \pm i\mu_j$;

$$c_{{}^*W{}^*W} := 4\sum \min(r_i, r_j),$$

where the sum is taken over all pairs of blocks $({}^*W_{r_i}(\lambda_i), {}^*W_{r_j}(\lambda_j)), i < j$, *in* A_{can} *such that* $\lambda_i = \lambda_j$;

- *the pairs of direct summands of* (6) *of different types:*

$$c_{{}^*W\mu\mathit{\Gamma}} := 0, \quad c_{{}^*WJ} := 4N_{odd}\sum_{i=1}^{c} r_i, \quad c_{\mu\mathit{\Gamma}J} := 2N_{odd}\sum_{i=1}^{b} q_i,$$

where N_{odd} *is the number of J blocks with odd size in* A_{can}.

As in the case of congruence orbits, the codimensions of the *congruence orbits of matrices can be obtained by computing the number of independent parameters over $\mathbb{R}$ in the miniversal deformations [8].

Example 4

The codimension of the 20×20 matrix $\Gamma_3 \oplus i\Gamma_6 \oplus {}^*W_4(5) \oplus J_3(0)$ from Example 2 can be computed as follows:

$$\begin{aligned}\operatorname{cod}^*(A) &= c_J + c_{\mu\mathit{\Gamma}} + c_{{}^*W} + c_{\mu\mathit{\Gamma}\mu\mathit{\Gamma}} + c_{{}^*W\mu\mathit{\Gamma}} + c_{{}^*WJ} + c_{\mu\mathit{\Gamma}J} \\ &= 4 + 9 + 8 + 6 + 0 + 16 + 18 \\ &= 61.\end{aligned}$$

The canonical structure of a complex matrix A under congruence given in Theorem 2.3 can be expressed as a direct sum of the blocks, as follows:

$$C^T AC \equiv \mathbb{J} \oplus \mathbb{\Gamma} \oplus \mathbb{W}(\lambda_1) \oplus \cdots \oplus \mathbb{W}(\lambda_t), \quad \det C \neq 0, \quad \lambda_i \neq \lambda_j \text{ if } i \neq j,$$

where

$$\begin{aligned}\mathbb{J} &:= \bigoplus_{i=1}^{a} J_{p_i}(0), \\ \mathbb{\Gamma} &:= \bigoplus_{j=1}^{b} \Gamma_{q_j}, \\ \mathbb{W}(\lambda_i) &:= \bigoplus_{k=1}^{c_i} W_{r_k}(\lambda_i).\end{aligned}$$

Note that $r_1^{(i)}, \dots, r_{c_i}^{(i)}$ are indices (half of the sizes) of the canonical summands associated with the eigenvalue λ_i. We assume that the sequences $p_1, \dots, p_a$, $q_1, \dots, q_b$, and $r_1^{(i)}, \dots, r_{c_i}^{(i)}$, for every $i = 1, \dots, t$, decrease monotonically. By analogy with the Jordan canonical form we call the following set of partitions the Segre characteristics associated with a matrix up to congruence.

$$\begin{aligned} \mathcal{J} &:= (p_1, \dots, p_a), \\ \mathit{\Gamma} &:= (q_1, \dots, q_b), \\ \mathcal{W}(\lambda_i) &:= (r_1^{(i)}, \dots, r_{c_i}^{(i)}), \quad i = 1, \dots, t \end{aligned}$$

Similarly, the canonical structure of a complex matrix A under *congruence given in Theorem 2.4 can be expressed as a direct sum of the blocks as follows.

$$\begin{aligned} C^*AC \equiv \mathbb{J} \oplus \mathbb{\Gamma}(\mu_1) \oplus \dots \oplus \mathbb{\Gamma}(\mu_s) \oplus {}^*\mathbb{W}(\lambda_1) \oplus \dots \oplus {}^*\mathbb{W}(\lambda_t), \\ \det C \neq 0, \quad \lambda_i \neq \lambda_j \text{ if } i \neq j, \text{ and } \quad \mu_k \neq \mu_m \text{ if } k \neq m, \end{aligned}$$

where

$$\begin{aligned} \mathbb{J} &:= \bigoplus_{k=1}^{a} J_{p_k}(0), \\ \mathbb{\Gamma}(\mu_j) &:= \bigoplus_{l=1}^{b_j} \mu_j \Gamma_{q_l}, \\ {}^*\mathbb{W}(\lambda_i) &:= \bigoplus_{m=1}^{c_i} {}^*W_{r_m}(\lambda_i). \end{aligned}$$

Let $r_1^{(i)}, \dots, r_{c_i}^{(i)}$ be indices (half of the sizes) of canonical summands associated with the eigenvalue λ_i and $q_1^{(j)}, \dots, q_{b_j}^{(j)}$ be the sizes of canonical summands associated with μ_j. We assume that the sequences $p_1, \dots, p_a$, $q_1^{(j)}, \dots, q_{b_j}^{(j)}$, for every $j = 1, \dots, s$, and $r_1^{(i)}, \dots, r_{c_i}^{(i)}$, for every $i = 1, \dots, t$, decrease monotonically. By analogy with the Jordan canonical form we call the following set of partitions the Segre characteristics associated with a matrix up to *congruence.

$$\begin{aligned} \mathcal{J} &:= (p_1, \dots, p_a), \\ \mathit{\Gamma}(\mu_j) &:= (q_1^{(j)}, \dots, q_{b_j}^{(j)}), \quad j = 1, \dots, s \\ {}^*\mathcal{W}(\lambda_i) &:= (r_1^{(i)}, \dots, r_{c_i}^{(i)}), \quad i = 1, \dots, t \end{aligned}$$

2.2 Skew-symmetric and symmetric matrix pencils under congruence

Let A and B be symmetric (or both skew-symmetric) $n \times n$ matrices over $\mathbb{C}$. Consider a matrix pencil $A - sB$ and the structure preserving congruence transformation

$$A - sB \mapsto C^T(A - sB)C, \quad \text{where } C \in GL_n(\mathbb{C}). \tag{8}$$

The set of matrix pencils congruent to $A-sB$ forms a manifold in the complex n^2+n (or, respectively, n^2-n) dimensional space. This manifold is the orbit of $A - sB$ under the action of congruence

$$\text{orbit}(A - sB) = \{C^T(A - sB)C : C \in GL_n(\mathbb{C})\}. \tag{9}$$

The vector space

$$T(A - sB) := \{(X^TA + AX) - s(X^TB + BX) : X \in \mathbb{C}^{n\times n}\} \tag{10}$$

is the tangent space to the congruence class of $A - sB$ at the point $A - sB$ since

$$(I + \varepsilon X)^T(A - sB)(I + \varepsilon X) = A - sB + \varepsilon((X^TA + AX) - s(X^TB + BX)) \\ +\varepsilon^2(X^TAX - sX^TBX)$$

for all n-by-n matrices X and each $\varepsilon \in \mathbb{C}$.

Recall that $A - sB$ is a symmetric (or a skew-symmetric) $n \times n$ matrix pencil. The *dimension of the orbit* of $A - sB$ is the dimension of its tangent space at the point $A - sB$. The *codimension of the orbit* $A - sB$ is the dimension of the normal space of its orbit at the point $A - sB$ which is equal to $n^2 + n$ (or $n^2 - n$ for a skew-symmetric pencil) minus the dimension of the orbit. Note that it is also equal to the number of linearly independent solutions of the following system of matrix equations

$$\begin{aligned} X^TA + AX &= 0, \\ X^TB + BX &= 0, \end{aligned} \tag{11}$$

plus n (or minus n). For more details see [9, 10].

We recall the canonical forms of symmetric and skew-symmetric matrix pencils under congruence, that were proven in [22]. These canonical forms are "symmetrized" or "skew-symmetrized" analogies of the Kronecker canonical forms for matrix pencils under the strict equivalence [15]. Following [22] we call them Kronecker canonical forms for symmetric and skew-symmetric matrix pencils.

For each positive integer m define the $m \times m$ matrices

$$\Lambda_m(\lambda) := \begin{bmatrix} 0 & & & \lambda \\ & & \lambda & 1 \\ & \iddots & \iddots & \\ \lambda & 1 & & 0 \end{bmatrix} \quad \text{and} \quad \Delta_m := \begin{bmatrix} 0 & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & 0 \end{bmatrix}.$$

For each non-negative integer define the $m \times (m+1)$ matrices

$$F_m := \begin{bmatrix} 1 & 0 & & 0 \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix} \quad \text{and} \quad G_m := \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ 0 & & 0 & 1 \end{bmatrix}.$$

Moreover, define the direct sum of matrix pencils as follows:

$$(A - sB) \oplus (C - sD) = (A \oplus C) - s(B \oplus D).$$

Theorem 2.5 [22]. *Every* complex symmetric matrix pencil is congruent *to a direct sum, determined uniquely up to permutation of summands, of pencils of the form*

$$H_p(\lambda) := \Lambda_p(\lambda) - s\Delta_p, \quad \lambda \in \mathbb{C}, \tag{12}$$

$$K_q := \Delta_q - s\Lambda_q(0), \tag{13}$$

$$M_r := \begin{bmatrix} 0 & G_r^T \\ G_r & 0 \end{bmatrix} - s \begin{bmatrix} 0 & F_r^T \\ F_r & 0 \end{bmatrix}. \tag{14}$$

Example 5

The 13×13 canonical symmetric matrix pencil $H_6(7) \oplus M_3$, presented as the direct sum of the pencils (14)–(13), can be written explicitly as follows:

$$
\left[\begin{array}{cccccc|cccc:ccc}
0&0&0&0&0&7&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&0&7&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&7&1&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&7&1&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&7&1&0&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
7&1&0&0&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\hline
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&1&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&1&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&0&1\\
\hdashline
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&-1&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&-1&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&-1&0&0&0
\end{array}\right]
- s \left[\begin{array}{cccccc|cccc:ccc}
0&0&0&0&0&1&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&0&1&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&1&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&1&0&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&1&0&0&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
1&0&0&0&0&0&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\hline
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&1&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&1&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&0&1\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&0&0&0&0&0\\
\hdashline
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&-1&0&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&-1&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&0&0&-1&0&0&0&0
\end{array}\right].
$$

Theorem 2.6 [22]. *Every* complex skew-symmetric matrix pencil is congruent *to a direct sum, determined uniquely up to permutation of summands, of pencils of the form*

$$
{}^{S}H_p(\lambda) := \begin{bmatrix} 0 & J_p(\lambda) \\ -J_p(\lambda)^T & 0 \end{bmatrix} - s \begin{bmatrix} 0 & I_p \\ -I_p & 0 \end{bmatrix}, \quad \lambda \in \mathbb{C}, \tag{15}
$$

$$
{}^{S}K_q := \begin{bmatrix} 0 & I_q \\ -I_q & 0 \end{bmatrix} - s \begin{bmatrix} 0 & J_q(0) \\ -J_q(0)^T & 0 \end{bmatrix}, \tag{16}
$$

$$
{}^{S}M_r := \begin{bmatrix} 0 & G_r \\ -G_r^T & 0 \end{bmatrix} - s \begin{bmatrix} 0 & F_r \\ -F_r^T & 0 \end{bmatrix}. \tag{17}
$$

EXAMPLE 6

The 13×13 canonical skew-symmetric matrix pencil ${}^{S}H_3(7) \oplus {}^{S}M_3$, presented as the direct sum of the pencils (17)–(16), can be written explicitly as follows:

$$
\left[\begin{array}{ccc:ccc|ccc:cccc}
0 & 0 & 0 & 7 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & 0 & 7 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & 0 & 0 & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hdashline
-7 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-1 & -7 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & -1 & -7 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & -1 & 0 & 0 & 0 & 0
\end{array}\right]
- s \left[\begin{array}{ccc:ccc|ccc:cccc}
0 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & 0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hdashline
-1 & 0 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & -1 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & -1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hdashline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right].
$$

We write the superscripts S in parentheses, i.e., ${}^{(S)}H, {}^{(S)}K, {}^{(S)}M$, when we refer to the corresponding symmetric and skew-symmetric canonical blocks, e.g., ${}^{(S)}H$ refers to both H blocks for symmetric matrix pencils and ${}^{S}H$ blocks for skew-symmetric matrix pencils.

Note also that the indices of ${}^{(S)}H, {}^{(S)}K$, and ${}^{(S)}M$ in Theorems 2.5 and 2.6 do not always coincide with the dimensions of the matrices. Moreover, the blocks ${}^{(S)}H_p(\lambda)$ correspond to Jordan structures $J_p(\lambda) - sI_p$ associated with finite eigenvalues, the blocks ${}^{(S)}K_q$ correspond to Jordan structures

$I_q - sJ_q(0)$ associated with the infinite eigenvalue and the blocks ${}^{(S)}M_r$ are associated with the singular Kronecker blocks $G_r - sF_r$.

Theorem 2.7 [10]. *Let* $A - sB$ *be a* complex symmetric matrix pencil *and*

$$(A - sB)_{can} = \bigoplus_{i=1}^{a} H_{p_i}(\lambda_i) \oplus \bigoplus_{j=1}^{b} K_{q_j} \oplus \bigoplus_{l=1}^{c} M_{r_l}, \tag{18}$$

be its canonical form for congruence. Then the codimension of the orbit of $A - sB$ under congruence *(denoted by* $\operatorname{cod}(A - sB)$*) can be computed as the sum*

$$\operatorname{cod}(A - sB) = c_H + c_K + c_M + c_{HH} + c_{KK} + c_{MM} + c_{HK} + c_{HM} + c_{KM} \tag{19}$$

whose summands correspond to

- *the direct summands of* (18)*:*

$$c_H := \sum_{i=1}^{a} p_i, \qquad c_K := \sum_{i=1}^{b} q_i, \qquad c_M := 2(c + \sum_{i=1}^{c} r_i);$$

- *the pairs of direct summands of* (18) *of the same type:*

$$c_{HH} := \sum_{\substack{i \leqslant j \\ \lambda_i = \lambda_j}} \min(p_i, p_j), \qquad c_{KK} := \sum_{i \leqslant j} \min(q_i, q_j),$$

$$c_{MM} := \sum_{j \leqslant i} \left(2 \max(r_i, r_j) + \varepsilon_{ij}\right), \qquad \textit{in which } \varepsilon_{ij} := \begin{cases} 2 & \textit{if } r_i = r_j, \\ 1 & \textit{if } r_i \neq r_j; \end{cases}$$

- *the pairs of direct summands of* (18) *of different types:*

$$c_{HK} := 0, \qquad c_{HM} := \sum_{i,j} p_i, \qquad c_{KM} := \sum_{i,j} q_i.$$

Example 7

The codimension of the 13×13 symmetric matrix pencil $H_6(7) \oplus M_3$ from Example 5 can be computed as follows:

$$\operatorname{cod}(A - sB) = c_H + c_M + c_{HM} = 6 + 8 + 6 = 20.$$

Theorem 2.8 [9]. *Let $A - sB$ be a* complex skew-symmetric matrix *pencil and*

$$(A - sB)_{\mathrm{can}} = \bigoplus_{i=1}^{a} {}^{S}H_{p_i}(\lambda_i) \oplus \bigoplus_{j=1}^{b} {}^{S}K_{q_j} \oplus \bigoplus_{l=1}^{c} {}^{S}M_{r_l}, \tag{20}$$

be its canonical form for congruence. The codimension of the orbit of $A-sB$ under congruence *(denoted by* $\operatorname{cod}(A - sB)$*) can be computed as the sum*

$$\operatorname{cod}(A-sB) = c_{s_H} + c_{s_K} + c_{s_M} + c_{s_H s_H} + c_{s_K s_K} + c_{s_M s_M} + c_{s_H s_K} + c_{s_H s_M} + c_{s_K s_M}, \tag{21}$$

whose summands correspond to

- *the direct summands of* (20)*:*

$$c_{s_H} := \sum_{i=1}^{a} p_i, \qquad c_{s_K} := \sum_{i=1}^{b} q_i, \qquad c_{s_M} := 0;$$

- *the pairs of direct summands of* (20) *of the same type:*

$$c_{s_H s_H} := 4 \sum_{\substack{i \leqslant j \\ \lambda_i = \lambda_j}} \min(p_i, p_j), \qquad c_{s_K s_K} := 4 \sum_{i \leqslant j} \min(q_i, q_j),$$

$$c_{s_M s_M} := \sum_{j \leqslant i} \left(2 \max(r_i, r_j) + \varepsilon_{ij}\right), \quad \textit{in which } \varepsilon_{ij} := \begin{cases} 2 & \textit{if } r_i = r_j, \\ 1 & \textit{if } r_i \neq r_j; \end{cases}$$

- *the pairs of direct summands of* (20) *of different types:*

$$c_{s_H s_K} := 0, \qquad c_{s_H s_M} := 2 \sum_{i,j} p_i, \qquad c_{s_K s_M} := 2 \sum_{i,j} q_i.$$

Example 8

The codimension of the 13×13 skew-symmetric matrix pencil ${}^{S}H_3(7) \oplus {}^{S}M_3$ from Example 6 can be computed as follows:

$$\operatorname{cod}(A - sB) = c_{s_H} + c_{s_M} + c_{s_H s_M} = 3 + 0 + 6 = 9.$$

The canonical structure can be expressed as the direct sum of the ${}^{(S)}H$, ${}^{(S)}K$, and ${}^{(S)}M$ blocks as follows.

$$C^T(A - sB)C \equiv {}^{(S)}\mathbb{H}(\lambda_1) \oplus \cdots \oplus {}^{(S)}\mathbb{H}(\lambda_t) \oplus {}^{(S)}\mathbb{K} \oplus {}^{(S)}\mathbb{M},$$

where $\det C \neq 0$, $\lambda_i \neq \lambda_j$ if $i \neq j$, and $A - sB$ is a symmetric (or skew-symmetric) matrix pencil, and

$$\begin{aligned} {}^{(S)}\mathbb{H}(\lambda_i) &:= \bigoplus_{j=1}^{a_i} {}^{(S)}H_{p_j}(\lambda_i), \\ {}^{(S)}\mathbb{K} &:= \bigoplus_{l=1}^{b} {}^{(S)}K_{q_l}, \\ {}^{(S)}\mathbb{M} &:= \bigoplus_{m=1}^{c} {}^{(S)}M_{r_m}. \end{aligned}$$

Let $p_1^{(i)}, \dots, p_{a_i}^{(i)}$ be the indices of canonical summands associated with the eigenvalue λ_i. We assume that the sequences $q_1, \dots, q_b$, $r_1, \dots, r_c$, and $p_1^{(i)}, \dots, p_{a_i}^{(i)}$, for every $i = 1, \dots, t$, decrease monotonically. By analogy with Jordan canonical form we call the following set of partitions the Segre characteristics associated with a symmetric (or skew-symmetric) matrix pencil.

$$\begin{aligned} {}^{(S)}\mathcal{H}(\lambda_i) &= (p_1^{(i)}, \dots, p_{a_i}^{(i)}), \quad i = 1, \dots, t, \\ {}^{(S)}\mathcal{K} &= (q_1^{(\infty)}, \dots, q_{a_\infty}^{(\infty)}), \\ {}^{(S)}\mathcal{M} &= (r_1, \dots, r_c). \end{aligned}$$

2.3 Numerical computations of codimensions

Let us recall that the tangent space to the similarity orbit of an $n \times n$ matrix A at the point A, is of the form $T_A = XA - AX$, where X is an $n \times n$ matrix and thus it is associated with the following homogeneous matrix equation:

$$XA - AX = 0. \tag{22}$$

Using $\operatorname{vec}(X)$, which denotes the n^2-long ordered stack of the columns of X from left to right, we can rewrite the equation (22) in the following form (e.g., see [11, 19, 20])

$$(A^T \otimes I_n) \operatorname{vec}(X) - (I_n \otimes A) \operatorname{vec}(X) = 0,$$

or equivalently

$$\left[A^T \otimes I_n - I_n \otimes A\right] \operatorname{vec}(X) = 0. \tag{23}$$

The $n^2 \times n^2$ Kronecker product matrix in (23) is a matrix representation of the tangent space of the similarity orbit of A at the point A. The codimension of the orbit of A is equal to the nullity of the matrix in (23). This method of computing codimensions was developed in [12, 14] and used in [18] for the codimensions of matrices, matrix pencils, and controllability/observability pairs. It is used in this paper too.

By P we denote the $n^2 \times n^2$ permutation matrix that can "transpose" $n \times n$ matrices, i.e., $\operatorname{vec}(X^T) = P \operatorname{vec}(X)$ for any $n \times n$ matrix X.

Numerical computation of the codimension of the congruence orbit of A is done analogously to the similarity case. For example, the equation (1) rewritten as the following system of equations:

$$\begin{aligned} YA - AX &= 0, \\ Y &= -X^T, \end{aligned} \tag{24}$$

leads to the Kronecker product matrix

$$\begin{bmatrix} A^T \otimes I_n & -I_n \otimes A \\ I_{n^2} & P \end{bmatrix}. \tag{25}$$

Note that the numerical codimension computations are based on calculating the rank of the matrix (25). Thus we can decrease the computational cost and the storage requirements by rewriting (25) as the $n^2 \times n^2$ matrix which is a matrix representation of the tangent space to the congruence orbit of A at the point A:

$$\left[A^T \otimes I_n + (I_n \otimes A)P\right]. \tag{26}$$

Numerical computation of the codimension of the *congruence orbit of A differs from the previous cases. As it was mentioned in Section 2.1, we must compute the codimension over the field of real numbers because the *congruence orbit of a matrix is a manifold over $\mathbb{R}$ (not over $\mathbb{C}$), see [3, 8] for more details.

For a matrix $A \in \mathbb{C}^{n \times n}$ let $\operatorname{Re}(A)$ and $\operatorname{Im}(A)$ be its real and imaginary parts, i.e.,

$$\operatorname{Re}(A) = \frac{A + \bar{A}}{2} \quad \text{and} \quad \operatorname{Im}(A) = \frac{A - \bar{A}}{2i}.$$

By considering separately the real and imaginary parts of A, X, and Y we obtain the following system of equations:

$$\begin{aligned} \operatorname{Re}(YA + AX) &= 0, \\ \operatorname{Im}(YA + AX) &= 0, \\ \operatorname{Re}(Y) &= \operatorname{Re}(X^T), \\ \operatorname{Im}(Y) &= -\operatorname{Im}(X^T). \end{aligned} \tag{27}$$

By constructing the Kronecker product matrix of size $4n^2 \times 4n^2$ associated with (27), we obtain

$$\begin{bmatrix} I_n \otimes \operatorname{Re}(A^T) & \operatorname{Re}(A) \otimes I_n & -I_n \otimes \operatorname{Im}(A^T) & -\operatorname{Im}(A) \otimes I_n \\ I_n \otimes \operatorname{Im}(A^T) & \operatorname{Im}(A) \otimes I_n & I_n \otimes \operatorname{Re}(A^T) & \operatorname{Re}(A) \otimes I_n \\ I_{n^2} & P & 0 & 0 \\ 0 & 0 & I_{n^2} & -P \end{bmatrix}.$$

As before, we reduce the size of the matrix and obtain the following $2n^2 \times 2n^2$ matrix

$$\begin{bmatrix} I_n \otimes \operatorname{Re}(A^T) + (\operatorname{Re}(A) \otimes I_n)P & -I_n \otimes \operatorname{Im}(A^T) + (\operatorname{Im}(A) \otimes I_n)P \\ I_n \otimes \operatorname{Im}(A^T) + (\operatorname{Im}(A) \otimes I_n)P & I_n \otimes \operatorname{Re}(A^T) - (\operatorname{Re}(A) \otimes I_n)P \end{bmatrix}. \tag{28}$$

The nullity of the matrix (28) is equal to the codimension of the *congruence orbit of A.

For the cases of square skew-symmetric and symmetric matrix pencils under congruence the system (11) can be rewritten in the following form

$$\begin{aligned} YA + AX &= 0, \\ YB + BX &= 0, \\ Y &= X^T. \end{aligned} \tag{29}$$

Analogously to (23) the $3n^2 \times 2n^2$ Kronecker product matrix associated with (29) is

$$\begin{bmatrix} A^T \otimes I_n & I_n \otimes A \\ B^T \otimes I_n & I_n \otimes B \\ I_{n^2} & -P \end{bmatrix}. \tag{30}$$

As before, reducing the size we obtain the following $2n^2 \times n^2$ matrix which is a matrix representation of the tangent space to the congruence orbit of $A - sB$ at the point $A - sB$:

$$\begin{bmatrix} A^T \otimes I_n + (I_n \otimes A)P \\ B^T \otimes I_n + (I_n \otimes B)P \end{bmatrix}. \tag{31}$$

The nullities of (31) plus (or minus) n, i.e., the size of the $n \times n$ matrix pencils, is equal to the codimensions of the congruence orbits of symmetric (or skew-symmetric) matrix pencils, see [9, Theorem 3] and [10, Theorem 3] for more details.

3 Implementation in the toolbox

In this section, we present the new Matlab functions in MCS Toolbox based on the results presented and discussed in Section 2. We illustrate them by several examples.

We follow the naming convention of MCS Toolbox [18]. The prefixes of the functions correspond to the type of setup they can be used for, i.e., *cm* is used for matrices under congruence, *scm* for matrices under *congruence, *sp* for symmetric matrix pencils, and *ssp* for skew-symmetric matrix pencils.

For each type of problem setup, there exist functions to create structure objects that represent canonical structures, to create new, possibly random, matrix example setups, to compute their codimensions, and a number of auxiliary functions.

3.1 Creating structure objects for matrices under congruence and *congruence

Structure objects of matrices under the congruence or *congruence transformations can be created with the *cmstruct* and *scmstruct* functions:

```
cmstruct(gblocks, wblocksv, eigv, zjblocks)
scmstruct(sgblocksv, muv, swblocksv, eigv, zjblocks)
```

The argument *gblocks* defines the Γ blocks for congruence (see (2)) of the structure and is given as a row-vector with the indices of the blocks. The corresponding argument *sgblocksv* defines $\mu\Gamma$ blocks for *congruence (see (3)) and must be a cell-array of row-vectors, each containing the indices of blocks with the same associated parameter μ. The parameters associated with each row-vector in *sgblocksv* are specified in the second argument, *muv*. The length of *muv* must be the same as *sgblocksv* and the absolute value of each element must be equal to 1 (see (3)). Note that the values of the parameters must be specified since the codimension depends on them.

The arguments *wblocksv* and *swblocksv* define the W and $^*W$ blocks, respectively, generated by Jordan blocks (see (2) and (3)) and must be cell-arrays of row-vectors. Each row-vector contains the canonical block indices associated with the same eigenvalue. The eigenvalues associated with each row-vector in *wblocksv* (or *swblocksv*) are specified in the argument, *eigv*. The length of *eigv* must be the same as *wblocksv* (or *swblocksv*), and the eigenvalues must satisfy the restrictions stated in (2) or (3).

The argument *zjblocks* specifies the indices of J blocks as defined in (2) and (3), i.e., the sizes of the Jordan blocks with zero eigenvalue.

Another way to create the structure objects *cmstruct* and *scmstruct* is to input the names of the blocks with the corresponding sets of parameters (indices, eigenvalues, etc.):

```
cmstruct('gblock',[q1,q2,...], ...
      'wblock',{[r1,r2,...],eig1}, ...
              'wblock',{[r1,r2,...],eig2}, ...
                            'zjblock',[p1,p2,...])
scmstruct('sgblock',{[q1,q2,...],mu1}, ...
  'swblock',{[r1,r2,...],eig1},'zjblock',[p1,p2,...])
```

Example 9

A canonical structure object for the matrix with the following canonical form under congruence $\Gamma_2 \oplus W_4(5) \oplus W_3(5) \oplus J_1(0)$, with one Γ block of size 2×2, one W block of size 8×8, one W block of size 6×6, both with the eigenvalue equal to 5, and one block $J_1(0)$ of size 1×1, is created by one of the following two equivalent calls.

```
>> cmstr = cmstruct([2],[4 3],5,[1])
>> cmstr = cmstruct('gblock',[2], ...
              'wblock',{[4 3],5},'zjblock',[1])
```

Both of them create the described object and display the following

```
cmstr =


        G = (2)         (2x2)
     W(5) = (4   3)     (14x14)
     J(0) = (1)         (1x1)
```

Any of the calls

```
>> cmstr = cmstruct([],{[3][1]},[3 7],[2])
>> cmstr = cmstruct('wblock'{[3],3}, ...
              'wblock',{[1],7},'zjblock',[2])
```

returns an object with the canonical structure $W_3(3) \oplus W_1(7) \oplus J_2(0)$, with no Γ blocks:

```
cmstr =

       W(3) = (3)     (6x6)
       W(7) = (1)     (2x2)
       J(0) = (2)     (2x2)
```

EXAMPLE 10

In this example we illustrate the *congruence case. The calls

```
>> scmstr = scmstruct({[3][2]},[1 i], ...
                         {[2][3 1]},[5 7],[2])
>> scmstr = scmstruct('sgblock',{[3],1}, ...
       'sgblock',{[2],i},'swblock',{[2],5}, ...
              'swblock',{[3 1],7},'zjblock',[2])
```

both return the canonical structure $\Gamma_3 \oplus i\Gamma_2 \oplus {}^*W_2(5) \oplus {}^*W_3(7) \oplus {}^*W_1(7) \oplus J_2(0)$.

```
scmstr =

        SG(1) = (3)        (3x3)
     SG(0+1i) = (2)        (2x2)
        SW(5) = (2)        (4x4)
        SW(7) = (3  1)     (8x8)
         J(0) = (2)        (2x2)
```

3.2 Creating structure objects for skew-symmetric and symmetric matrix pencils

Structure objects for symmetric and skew-symmetric matrix pencils are created with the functions *spstruct* and *sspstruct*, respectively:

```
spstruct(mblocks, hblocksv, eigv, kblocks)
sspstruct(smblocks, shblocksv, eigv, skblocks)
```

The arguments *mblocks* and *smblocks* define symmetric M blocks and skew-symmetric ${}^{S}M$ blocks, respectively (see (14) and (17)) of the structure and are given as row-vectors with the canonical block indices.

The arguments *hblocksv* and *shblocksv* define the symmetric H blocks and skew-symmetric ${}^{S}H$ blocks, respectively, corresponding to Jordan structures associated with finite eigenvalues (as defined in (12) and (15)). Each of them must be either a row-vector with the canonical block indices corresponding to the same eigenvalue or a cell-array of row-vectors, each containing the canonical block indices for the same associated eigenvalue. The eigenvalues associated with each row-vector in *hblocksv* (or *shblocksv*) are specified in an optional second argument, *eigv*. The length of *eigv* must be the same as *hblocksv* (or *shblocksv*). ${}^{(S)}H$ blocks generated by Jordan blocks with an unspecified eigenvalue can be given in *hblocksv* (or *shblocksv*) with *NaN* as the corresponding eigenvalue in *eigv*. If the argument *eigv* is not given, all eigenvalues are assumed to be unspecified.

The fourth arguments *kblocks* and *skblocks* specify the indices of the K blocks and ${}^{S}K$ blocks, respectively, as defined in (13) and (16), i.e., the sizes of the Jordan structures corresponding to blocks with the infinite eigenvalue.

An alternative way to create the *spstruct* and *sspstruct* objects is to input the names of the blocks with the corresponding sets of parameters (analogously to the matrices under congruence and *congruence):

```
spstruct('mblock',[r1,r2,...], ...
    'hblock',{[p1,p2,...], eig1},'kblock',[q1,q2,...])
sspstruct('smblock',[r1,r2,...], ...
    'shblock',{[p1,p2,...], eig1},'skblock',[q1,q2,...])
```

Example 11

A canonical structure object for the skew-symmetric matrix pencil with the following canonical form under congruence ${}^{S}M_2 \oplus$

${}^SH_3(5) \oplus {}^SH_2(5) \oplus {}^SK_1$, with one singular block of size 5×5, one ${}^SH$ block of size 6×6, one ${}^SH$ block of size 4×4 , both corresponding to the eigenvalue equal to 5, and one ${}^SM$ block of size 2×2 associated with the infinite eigenvalue, is created by one of the following commands:

```
>> sspstr = sspstruct([2],[3 2],5,[1])
>> sspstruct('smblock',[2], ...
     'shblock',{[3 2],5},'skblock',[1])
```

```
sspstr =

        SM = (2)        (5x5)
     SH(5) = (3   2)    (10x10)
        SK = (1)        (2x2)
```

Any of the calls

```
>> sspstr = sspstruct([],{[1][2]},[3 7],[2])
>> sspstruct('shblock',{[1],3}, ...
               'shblock', {[2],7},'skblock',[2])
```

returns an object with the canonical structure, ${}^SH_1(3) \oplus {}^SH_2(7) \oplus {}^SK_2$, with no singular blocks, but with the blocks generated by Jordan blocks corresponding to different eigenvalues ($3, 7$, and ∞):

```
sspstr =

     SH(3) = (1)    (2x2)
     SH(7) = (2)    (4x4)
        SK = (2)    (4x4)
```

Example 12

The major difference between defining symmetric and skew-symmetric pencils is the following: in the symmetric case indices are equal to the sizes of blocks associated to the Jordan blocks while in the skew-symmetric case they are equal to a half of the actual size of the blocks (see ${}^{(S)}H$ and ${}^{(S)}K$ blocks, Theorems 2.5 and 2.6).

Any of the calls

```
>>spstr = spstruct([3],{[4][2]},[5 7],[2])
>>spstruct('mblock',[3],'hblock',{[4],5}, ...
                    'hblock',{[2],7},'kblock',[2])
```

returns an object with the canonical structure, $M_3 \oplus H_4(5) \oplus H_2(7) \oplus K_2$, with one M block and three blocks associated with three different eigenvalues ($5, 7$, and ∞):

```
spstr =

       M = (3)     (7x7)
    H(5) = (4)     (4x4)
    H(7) = (2)     (2x2)
       K = (2)     (2x2)
```

3.3 Displaying structures in different notations

Canonical structures are displayed as any other data type in Matlab by entering the name of the variable without a trailing semi-colon. Structure objects can be displayed using four different notations, namely Segre characteristics, Weyr characteristics, indices, and block structure notation.

To view or change the used notation, the *mcsdisplay* function exists.

```
                [arg] = mcsdisplay
[arg, ispersistent] = mcsdisplay
                        mcsdisplay(arg)
                        mcsdisplay(arg, persistent)
```

Without any arguments, *mcsdisplay* returns the currently used notation. By providing the argument *arg*, a new notation is set. Valid values of *arg* are the following.

'segre' Displays the structure as integer partitions in Segre characteristics, which means that each integer in an integer partition corresponds to the index of a canonical block and all integers are ordered as a monotonically decreasing integer sequence. Each block type (e.g., H blocks, K blocks, J blocks, …) has its own integer partition. Further, the partition has a prefix identifier and a suffix that specifies the total block size. See Example 13.

'weyr' This notation is similar to 'segre' except that the integer partitions are in Weyr characteristics. For ${}^{(S)}M$ blocks, the partition $\mathcal{X} = (k_0, \ldots, k_r)$ means that there are k_0 blocks with the indices larger or equal to 0, k_1 blocks with the indices larger or equal to 1, etc. For the rest of the blocks (with unknown, finite, infinite, or zero eigenvalues) the partition $\mathcal{X} = (k_1, \ldots, k_r)$ instead means that there are k_1 blocks with the indices greater or equal to 1, etc. Notice that the partitions are still monotonically decreasing.

'sizes' This notation is similar to 'segre' except that the indices are displayed in the order they were created.

'block' Displays the structures in canonical block notation. See Example 13.

A second input argument, *persistent*, tells if the setting should be persistent between sessions or not. A persistent setting is saved in the Matlab environment and read the next time Matlab is started. A second output argument reports if the currently used notation is persistent.

Example 13

The canonical structure of the skew-symmetric matrix pencil ${}^{S}M_0 \oplus {}^{S}M_3 \oplus 3\,{}^{S}H_1(\alpha)$ will be displayed in Matlab as follows using the four different notations:

```
>> sspstr = sspstruct([0 3],[1 1 1]);
>> mcsdisplay('segre')
>> sspstr

sspstr =

            SM = (3   0)        (8x8)
       SH(NaN) = (1   1   1)    (6x6)
```

```
>> mcsdisplay('weyr')
>> sspstr

sspstr =

             SM = (2   1   1   1)     (8x8)
        SH(NaN) = (3)                 (6x6)

 >> mcsdisplay('sizes')
 >> sspstr

sspstr =

             SM = (0   3)         (8x8)
        SH(NaN) = (1   1   1)     (6x6)

>> mcsdisplay('block')
>> sspstr

sspstr =

         SM3 + SM0 + 3SH1(NaN)
```

3.4 Block sizes and substructures

The function *get* returns either an array with the corresponding indices or a cell-array of such arrays. A cell-array output is used when several integer partitions corresponding to different parameters need to be returned. The arrays returned are all in analogy with how the structures have been created. The admissible arguments for *get* are all the block names, see Appendix B.

Another important function is *size*. It returns the sizes of the matrices that represent the corresponding structure object.

EXAMPLE 14

Assume we have a canonical structure object of the skew-symmetric matrix pencil representing ${}^{S}M_0 \oplus {}^{S}M_1 \oplus 2{}^{S}H_1(\alpha) \oplus {}^{S}K_1$. The structure can be created as follows:

```
>> sspstr = sspstruct([0 1],[1 1],NaN,[1]);
```

The indices of the ${}^{S}M$ blocks can be obtained with any of the two analogous calls:

```
>> mysmblocks = get(sspstr,'smblock');
>> mysmblocks = sspstr.get('smblock');
```

EXAMPLE 15

Given the same structure as in Example 14, a cell array containing the canonical information of the ${}^{S}H$ blocks can be extracted by

```
>>myshblocks = get(sspstr,'shblock');
```

or

```
>>myshblocks = sspstr.get('shblock');
```

EXAMPLE 16

The sizes of matrices from the structure in Example 14 can be computed by *size* using the following call

```
>>size(sspstr)
ans =

    10    10
```

or equivalently

```
>>sspstr.size
ans =

    10    10
```

3.5 Conversion between structure objects

New structures can also be created by converting structure objects of different classes. This can be done by providing a structure object of one class as an input argument to the constructor function of another class. The admissible conversions are the following:

- from a skew-symmetric matrix pencil structure object to a corresponding matrix pencil structure object;
- from a symmetric matrix pencil structure object to a corresponding matrix pencil structure object.

The command *pstruct* works analogously to the commands *spstruct* and *sspstruct*, see [18] for the details.

EXAMPLE 17

The following commands

```
>> sspstr = sspstruct([2],[2 1],4,[3]);
>> pstr = pstruct(sspstr);
```

convert the canonical structure of a skew-symmetric matrix pencil object representing ${}^{S}M_2 \oplus {}^{S}H_2(4) \oplus {}^{S}H_1(4) \oplus {}^{S}K_3$, to a Kronecker structure object representing $L_2 \oplus L_2^T \oplus J_2(4) \oplus J_2(4) \oplus J_1(4) \oplus J_1(4) \oplus J_3(\infty) \oplus J_3(\infty)$. The following use of *pstruct* would create an equivalent Kronecker structure:

```
>> pstr = pstruct([2],[2],[2 2 1 1],4,[3 3]);
```

The call for the conversion of a symmetric matrix pencil structure object to a corresponding matrix pencil structure object is:

```
>> pstr = pstruct(spstruct(. . .));
```

3.6 Creating example data setups for matrices under congruence and *congruence

Matrices in the canonical form under congruence and *congruence can be created using the function *ccf* (congruence canonical form). The input arguments of *ccf* are the structure objects *cmstruct* or *scmstruct*.

```
cmstr  = cmstruct(gblocks, wblocksv, eigv, zjblocks)
scmstr = scmstruct(sgblocksv, muv, swblocksv, ...
                                         eigv, zjblocks)
```

The calls to create the setup which is a matrix A are the following

```
A = ccf(cmstr)
A = ccf(scmstr)
```

Alternatively, we may use the calls

```
A = cmstr.ccf
A = scmstr.ccf
```

Example 18

The following call to *ccf*

```
>> cmstr=cmstruct([3],{[1] [2]}, [7 3])
>> A = ccf(cmstr)
```

returns a 9×9 matrix A with the canonical structure $\Gamma_3 \oplus W_1(7) \oplus W_2(3)$ under congruence:

```
A =

     0     0     1     0     0     0     0     0     0
     0    -1    -1     0     0     0     0     0     0
     1     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     7     0     0     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     3     1     0     0
     0     0     0     0     0     0     3     0     0
```

The call above can also be done in the following two equivalent ways:

```
>> [A] = ccf(cmstruct([3],{[1][2]},[7 3]))
```

or

```
>> cmstr = cmstruct([3],{[1][2]},[7 3])
>> [A] = cmstr.ccf
```

EXAMPLE 19

The following call to *ccf* for a *scmstruct* object

```
>> scmstr=scmstruct({[3][2]},[-1 i],[2],[3])
>> A = ccf(scmstr)
```

returns a 9×9 matrix A with the canonical structure $-\Gamma_3 \oplus i\Gamma_2 \oplus {}^*W_2(3)$ under *congruence:

```
A =

     0     0    -1     0     0     0     0     0     0
     0     1     1     0     0     0     0     0     0
    -1    -1     0     0     0     0     0     0     0
     0     0     0     0    -i     0     0     0     0
     0     0     0     i     i     0     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     3     1     0     0
     0     0     0     0     0     0     3     0     0
```

The following calls are equivalent to the one made above.

```
>> [A] = ccf(scmstruct({[3][2]},[-1 i],[2],[3]))
```

and

```
>> scmstr = scmstruct({[3][2]},[-1 i],[2],[3])
>> [A] = scmstr.ccf
```

It is also possible to generate a random matrix with a desired structure under congruence or *congruence. The *ccf* function can return the two matrices B and C such that

$$C^T AC = B \quad \text{(respectively, } C^* BC = A\text{)},$$

where A is in the specified canonical form and C is a random orthogonal matrix:

```
[B, C] = ccf(cmstr)
[B, C] = ccf(scmstr)
```

The *ccf* function can also return the three matrices B, C, and A defined above:

```
[B, C, A] = ccf(cmstr)
[B, C, A] = ccf(scmstr)
```

3.7 Creating example data setups for symmetric and skew-symmetric matrix pencils

Symmetric and skew-symmetric matrix pencils in canonical form are created using the function *kcf* (Kronecker canonical form). The input arguments of *kcf* are the objects of *spstruct* and *sspstruct*.

```
spstr = spstruct(mblocks,hblocksv,eigv,kblocks)
sspstr = sspstruct(smblocks,shblocksv,eigv,skblocks)
```

The calls to create a matrix pencil $A - sB$ which corresponds to a desired setup, are as follows:

```
[A, B] = kcf(spstruct)
[A, B] = kcf(sspstruct)
```

Alternatively, we may use the calls

```
[A, B] = spstr.kcf
[A, B] = sspstr.kcf
```

Example 20

The following call to *kcf* for a *spstruct* object

```
>> spstr = spstruct([1],{[2][4]},[0 3]);
>> [A,B] = kcf(spstr)
```

returns a 9×9 symmetric matrix pencil $A - sB$ with the canonical structure $M_1 \oplus H_2(0) \oplus H_4(3)$:

```
A =
     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     3
     0     0     0     0     0     0     0     3     1
     0     0     0     0     0     0     3     1     0
     0     0     0     0     0     3     1     0     0
B =
     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     1     0     0     0
```

For *spstruct*, the two calls equivalent to the one above are

```
>> [A,B] = kcf(spstruct([1],{[2][4]},[0 3]))
```

and

```
>> spstr = kcf(spstruct([1],{[2][4]},[0 3]))
>> [A,B] = spstr.kcf
```

Example 21

The following call to *kcf* for an *sspstruct* object

```
>> sspstr = sspstruct([1],{[1][2]},[0 3]);
>> [A,B] = kcf(sspstr)
```

returns a 9×9 skew-symmetric matrix pencil $A - sB$ with the canonical structure ${}^{S}M_1 \oplus {}^{S}H_1(0) \oplus {}^{S}H_2(3)$:

```
A =
     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
    -1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     3     1
     0     0     0     0     0     0     0     0     3
     0     0     0     0     0    -3     0     0     0
     0     0     0     0     0    -1    -3     0     0
B =
     0     1     0     0     0     0     0     0     0
    -1     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0    -1     0     0     0     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
     0     0     0     0     0    -1     0     0     0
     0     0     0     0     0     0    -1     0     0
```

For *sspstruct*, the two call equivalent to the one above are

```
>> [A,B] = kcf(sspstruct([1],{[1][2]},[0 3]))
```

and

```
>> sspstr = sspstruct([1],{[1][2]},[0 3])
>> [A,B] = sspstr.kcf
```

It is also possible to generate a random matrix pencil with a desired structure. The function *kcf* can return the three matrices Q, S, and T such that

$$Q^T(A - sB)Q = S - sT,$$

where (A, B) is in the specified canonical form and Q is an orthogonal matrix:

```
[S, T, Q] = kcf(spstr)
[S, T, Q] = kcf(sspstr)
```

The function *kcf* can also return the five matrices S, T, Q, A, and B defined above:

```
[S, T, Q, A, B] = kcf(spstr)
[S, T, Q, A, B] = kcf(sspstr)
```

3.8 Codimension and distance functions

The function *codim* is used to determine the codimension of the corresponding orbit (i.e., congruence or *congruence orbit) of a matrix or matrix pencil canonical structure object. The codimensions are computed from the canonical structured information as stated in Theorems 2.3 and 2.4 for matrices and Theorems 2.7 and 2.8 for matrix pencils. The corresponding functions for the four types of setups are:

```
codim(cmstr)                codim(spstr)
codim(scmstr)               codim(sspstr)
```

The codimentions of matrix and pencil orbits can also be computed numerically using the results of Section 2.3. In this case, for matrices up to congruence (or *congruence) the function has the prefix *cm* (or *scm*) and operates on the setup

$$A, \quad \text{where } A \in \mathbb{C}^{n\times n}.$$

Note that beside taking a complex matrix A as an argument, the following functions can operate on a structure object *cmstruct* (or *scmstruct*), then the canonical structure information is used for the computations (i.e., they work the same as *codim*).

```
cmcodim(A)                  scmcodim(A)
cmcodim(A, tol)             scmcodim(A, tol)
cmcodim(cmstruct)           scmcodim(scmstruct)
```

The analogous functions for the symmetric (or skew-symmetric) matrix pencil functions have the prefix *sp* (or *ssp*) and operates on the setup

$$A - sB, \quad \text{where } A, B \in \mathbb{C}^{n\times n}.$$

Note that as in the matrix case the following functions for matrix pencils can take a structure object *spstruct* (or *sspstruct*) as an input.

```
spcodim(A, B)              sspcodim(A, B)
spcodim(A, B, tol)         sspcodim(A, B, tol)
spcodim(spstr)             sspcodim(sspstr)
```

In the x*codim* functions above an optional rank tolerance parameter, *tol*, can be specified. If not, the default tolerance for *rank* is used.

Example 22

The codimensions calculated in Examples 3, 4, 8, and 7 can be computed as follows.

```
>> codim(cmstruct([3],[4 3],[5],[3]))
ans = 33
>> codim(scmstruct({[3][6]},[1 i],[4],5,[3]))
ans = 61
>> codim(spstruct([3],[6],[7]))
ans = 20
>> codim(sspstruct([3],[3],[7]))
ans = 9
```

Appendix A: Summary of Matlab functions

Matrix up to congruence	
cmstr = *cmstruct (gblocks, wblocks, eigv, zjblocks)*	Returns a new object representing the canonical structure of a matrix under congruence.
[B, C, A] = *ccf* (cmstr) [B, C, A] = cmstr.*ccf*	Returns a matrix A in the canonical form under congruence and matrices B, C, such that $C^T AC = B$.
size(cmstr) cmstr.*size*	Returns the size of the matrix.
cmcodim (A, B, tol) *cmcodim*(cmstr) *codim* (cmstr) cmstr.*codim*	Computes the codimension of the congruence orbit of a matrix either numerically or from the canonical structure information.

Matrix up to *congruence	
scmstr = *scmstruct (sgblocks, muv, swblocks, eigv, zjblocks)*	Returns a new object representing the canonical structure of a matrix under *congruence.
[B, C, A] = *ccf* (scmstr) [B, C, A] = scmstr.*ccf*	Returns a matrix A in the canonical form under *congruence and matrices B, C, such that $C^* AC = B$.
size(scmstr) scmstr.*size*	Returns the size of the matrix.
scmcodim(A, B, tol) *scmcodim*(scmstr) *codim*(scmstr) scmstr.*codim*	Computes the codimension of the *congruence orbit of a matrix either numerically or from the canonical structure information.

Symmetric matrix pencils up to congruence	
spstr = *spstruct(mblocks, hblocks, eigv, kblocks)*	Returns a new object representing the canonical structure of a symmetric matrix pencil.
[S, T, Q, A, B] = *kcf* (spstr) [S, T, Q, A, B] = spstr.*kcf*	Returns a symmetric matrix pencil $A - sB$ in the canonical form and matrices S, T, and Q such that $Q^T(A - sB)Q = S - sT$.
size(spstr) spstr.*size*	Returns the size of the matrix pencil.
spcodim(A, B, tol) *spcodim*(spstr) *codim*(spstr) spstr.*codim*	Computes the codimension of the congruence orbit of a symmetric matrix pencil either numerically or from the canonical structure information.
Skew-symmetric matrix pencils up to congruence	
sspstr = *sspstruct (smblocks, shblocks, eigv, skblocks)*	Returns a new object representing the canonical structure of a skew-symmetric matrix pencil.
[S, T, Q, A, B] = *kcf* (sspstr) [S, T, Q, A, B] = sspstr.*kcf*	Returns a skew-symmetric matrix pencil $A - sB$ in the canonical form and matrices S, T, and Q such that $Q^T(A - sB)Q = S - sT$.
size(sspstr) sspstr.*size*	Returns the size of the matrix pencil.
sspcodim(A, B, tol) *sspcodim*(sspstr) *codim*(sspstr) sspstr.*codim*	Computes the codimension of the congruence orbit of a skew-symmetric matrix pencil either numerically or from the canonical structure information.

Appendix B: Summary of canonical blocks

Canonical blocks for matrices under congruence

wblock	W blocks with admissible eigenvalues.
gblock	Γ blocks.
zjblock	Jordan blocks with zero eigenvalues.

Canonical blocks for matrices under *congruence

swblock	${}^*W$ blocks with admissible eigenvalues.
sgblock	$\mu\Gamma$ blocks with admissible parameters.
zjblock	Jordan blocks with zero eigenvalues.

Canonical blocks for symmetric matrix pencils

hblock	H blocks with specified or unspecified eigenvalues.
kblock	K blocks, i.e., blocks associated with an infinite eigenvalue.
mblock	M blocks.

Canonical blocks for skew-symmetric matrix pencils

shblock	${}^SH$ blocks with specified or unspecified eigenvalues.
skblock	${}^SK$ blocks, i.e., blocks associated with an infinite eigenvalue.
smblock	${}^SM$ blocks.

Acknowledgements

Supported by the Swedish Research Council (VR) under grant A0581501, and by eSSENCE, a strategic collaborative e-Science programme funded by the Swedish Research Council.

References

[1] W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry,* Academic Press, New York, 1975.

[2] F. De Terán, F.M. Dopico, The solution of the equation $XA + AX^T = 0$ and its application to the theory of orbits, *Linear Algebra Appl.* 434 (2011) 44–67.

[3] F. De Terán, F.M. Dopico, The equation $XA + AX^* = 0$ and the dimension of *congruence orbits, *Electr. J. Linear Algebra* 22 (2011) 448–465.

[4] J. Demmel, A. Edelman, The dimension of matrices (matrix pencils) with given Jordan (Kronecker) canonical forms, *Linear Algebra Appl.* 230 (1995) 61–87.

[5] A. Dmytryshyn, *Miniversal Deformations of Pairs of Skew-symmetric Forms*, Master Thesis, Kiev National University, Kiev, 2010, arXiv:1104.2492.

[6] A. Dmytryshyn, Miniversal deformations of pairs of symmetric forms, Manuscript, 2011, arXiv:1104.2530.

[7] A.R. Dmytryshyn, V. Futorny, V.V. Sergeichuk, Miniversal deformations of matrices of bilinear forms, *Linear Algebra Appl.* 436(7) (2012) 2670–2700.

[8] A.R. Dmytryshyn, V. Futorny, V.V. Sergeichuk, Miniversal deformations of matrices of sesquilinear forms, Manuscript, 2011, arXiv:1105.2160.

[9] A. Dmytryshyn, B. Kågström, V.V. Sergeichuk, Skew-symmetric matrix pencils: codimension counts and the solution of a pair of matrix equations, *Linear Algebra Appl.*, 438(8) (2013) 3375–3396.

[10] A. Dmytryshyn, B. Kågström, V.V. Sergeichuk, Symmetric matrix pencils: codimension counts and the solution of a pair of matrix equations, Report UMINF 13.12, Department of Computing Science, Umeå University, 2013.

[11] A. Edelman, E. Elmroth, B. Kågström, A geometric approach to perturbation theory of matrices and matrix pencils. Part I: Versal deformations, *SIAM J. Matrix Anal. Appl.* 18 (3) (1997) 653–692.

[12] A. Edelman, E. Elmroth, B. Kågström, A geometric approach to perturbation theory of matrices and matrix pencils. Part II: A stratification-enhanced staircase algorithm, *SIAM J. Matrix Anal. Appl.* 20 (1999) 667–669.

[13] E. Elmroth, S. Johansson, B. Kågström, Stratification of controllability and observability pairs theory and use in applications, *SIAM J. Matrix Anal. Appl.* 31 (2) (2009) 203–226.

[14] E. Elmroth, P. Johansson, B. Kågström, Bounds for the Distance Between Nearby Jordan and Kronecker Structures in a Closure Hierarchy, *Journal of Mathematical Sciences* 114 (6) (2003) 1765–1779.

[15] F.R. Gantmacher. *The Theory of Matrices.* Chelsea, New York, 1959.

[16] M.I. Garcia-Planas, V.V. Sergeichuk, Simplest miniversal deformations of matrices, matrix pencils, and contragredient matrix pencils, *Linear Algebra Appl.* 302–303 (1999) 45–61.

[17] R.A. Horn, V.V. Sergeichuk, Canonical forms for complex matrix congruence and *congruence, *Linear Algebra Appl.* 416 (2006) 1010–1032.

[18] P. Johansson, *Matrix Canonical Structure Toolbox*, Report UMINF 06.15, Department of Computing Science, Umeå University, 2006.

[19] S. Johansson, *Reviewing the Closure Hierarchy of Orbits and Bundles of System Pencils and Their Canonical Forms*, Report UMINF 09.02, Department of Computing Science, Umeå University, 2009.

[20] S. Johansson, *Tools for Control System Design: Stratification of Matrix Pencils in Systems and Periodic Riccati Differential Equation Solvers*, PhD Thesis UMINF 09.04, Department of Computing Science, Umeå University, 2009.

[21] B. Kågström, L. Karlsson, D. Kressner, Computing codimensions and generic canonical forms for generalized matrix products, *Electron. J. Linear Algebra* 22 (2011) 277–309.

[22] R.C. Thompson, Pencils of complex and real symmetric and skew matrices, *Linear Algebra Appl.* 147 (1991) 323–371.