Variable projection for affinely structured low-rank approximation in weighted 2-norm

Konstantin Usevich[a,*], Ivan Markovsky[a]

[a] *Vrije Universiteit Brussel, Department ELEC, Pleinlaan 2, B-1050, Brussels, Belgium*

Abstract

The structured low-rank approximation problem for general affine structures, weighted 2-norm and fixed elements is considered. The variable projection priciple is used to reduce the dimensionality of the optimization problem. Algorithms for evaluation of the cost function, the gradient and approximation of the Hessian are developed. For $m \times n$ mosaic Hankel matrices the algorithms have complexity $O(m^2 n)$.

Keywords: low-rank approximation, structured low-rank approximation, variable projection, mosaic Hankel matrices, total least squares, weighted 2-norm, fixed elements, computational complexity
2010 MSC: 15B99, 15B05, 41A29, 49M30, 65F30, 65K05, 65Y20

1. Introduction

An *affine matrix structure* is an affine mapping from a *structure parameter space* $\mathbb{R}^{n_p}$ to a space of matrices $\mathbb{R}^{m \times n}$:

$$\mathscr{S}(p) = S_0 + \sum_{i=1}^{n_p} p_i S_i, \quad S_k \in \mathbb{R}^{m \times n}. \tag{$\mathscr{S}$}$$

Throughout the paper we assume that $m \leq n$. The *structured low-rank approximation* is the problem of finding the best low-rank structure-preserving approximation of a given data matrix [1, 2].

Problem 1 (Structured low-rank approximation)**.** *Given an affine structure $\mathscr{S}$, data vector $p_d \in \mathbb{R}^{n_p}$, norm $\|\cdot\|$ and natural number $r < m$*

$$\underset{\widehat{p} \in \mathbb{R}^{n_p}}{\text{minimize}} \ \|p_d - \widehat{p}\| \quad \text{subject to} \quad \text{rank}\, \mathscr{S}(\widehat{p}) \leq r. \tag{SLRA}$$

In this paper, we consider the case of *weighted* 2*-norm*, given by

$$\|p\|_{\mathrm{W}}^2 := p^\top \mathrm{W} p, \quad \mathrm{W} \in \mathbb{R}^{n_p \times n_p}, \tag{$\|\cdot\|_{\mathrm{W}}^2$}$$

where W is

*Corresponding author
Email addresses: Konstantin.Usevich@vub.ac.be (Konstantin Usevich), Ivan.Markovsky@vub.ac.be (Ivan Markovsky)

- either a symmetric positive definite matrix,
- or a diagonal matrix

$$\mathrm{W} = \mathrm{diag}(w_1, \ldots, w_{n_p}), \quad w_i \in (0; +\infty], \qquad (w \to \mathrm{W})$$

where $+\infty \cdot 0 = 0$ by convention. A *finiteness* constraint $\|p_{\mathrm{d}} - \widehat{p}\|_{\mathrm{W}}^2 < +\infty$ is additionally imposed on (SLRA). Problem (SLRA) with the weighted norm ($\|\cdot\|_{\mathrm{W}}^2$) given by ($w \to \mathrm{W}$) is equivalent to structured low-rank approximation with *elementwise weighted* 2-norm

$$\|p\|_w^2 = \sum_{w_i \neq +\infty} w_i p_i^2,$$

and a set of *fixed values* constraints

$$(p_{\mathrm{d}})_i = \widehat{p}_i \text{ for all } i \text{ with } w_i = +\infty.$$

The structured low-rank approximation problem with weighted 2-norm appears in signal processing, computer algebra, identification of dynamical systems, and other applications. We refer the reader to [1, 2] for an overview. In this paper, we consider general affine structures ($\mathscr{S}$) and, in particular, structures that have the form

$$\mathscr{S}(p) = \Phi \mathscr{H}_{\mathbf{m},\mathbf{n}}(p), \qquad (\Phi\mathscr{H}_{\mathbf{m},\mathbf{n}})$$

where Φ is a full row rank matrix and $\mathscr{H}_{\mathbf{m},\mathbf{n}}$ is a *mosaic Hankel* matrix structure [3]. Many applied problems can be reduced to (SLRA) with the structure ($\Phi\mathscr{H}_{\mathbf{m},\mathbf{n}}$) and weighted norm, defined by ($w \to \mathrm{W}$), see [4, 1].

1.1. Mosaic Hankel structure

A *mosaic Hankel* matrix structure $\mathscr{H}_{\mathbf{m},\mathbf{n}}$ is a map defined by two integer vectors

$$\mathbf{m} = \begin{bmatrix} m_1 & \cdots & m_q \end{bmatrix} \in \mathbb{N}^q \qquad \text{and} \qquad \mathbf{n} = \begin{bmatrix} n_1 & \cdots & n_N \end{bmatrix} \in \mathbb{N}^N \qquad (\mathbf{m}, \mathbf{n})$$

as follows:

$$\mathscr{H}_{\mathbf{m},\mathbf{n}}(p) := \begin{bmatrix} \mathscr{H}_{m_1,n_1}(p^{(1,1)}) & \cdots & \mathscr{H}_{m_1,n_N}(p^{(1,N)}) \\ \vdots & & \vdots \\ \mathscr{H}_{m_q,n_1}(p^{(q,1)}) & \cdots & \mathscr{H}_{m_q,n_N}(p^{(q,N)}) \end{bmatrix}, \qquad (\mathscr{H}_{\mathbf{m},\mathbf{n}})$$

where

$$p = \mathrm{col}(p^{(1,1)}, \ldots, p^{(q,1)}, \ldots, p^{(1,N)}, \ldots, p^{(q,N)}), \qquad p^{(k,l)} \in \mathbb{R}^{m_k+n_l-1}, \qquad (p)$$

is the partition of the parameter vector, and $\mathscr{H}_{m,n} : \mathbb{R}^{m+n-1} \to \mathbb{R}^{m \times n}$ is the *Hankel* structure

$$\mathscr{H}_{m,n}(p) := \begin{bmatrix} p_1 & p_2 & p_3 & \cdots & p_n \\ p_2 & p_3 & \iddots & & p_{n+1} \\ p_3 & \iddots & & & \vdots \\ \vdots & & & & p_{m+n-2} \\ p_m & p_{m+1} & \cdots & p_{m+n-2} & p_{m+n-1} \end{bmatrix}.$$

Note that the number of parameters for ($\mathscr{H}_{\mathbf{m},\mathbf{n}}$) is equal to

$$n_p = N \sum_{k=1}^{q} m_k + q \sum_{l=1}^{N} n_l - Nq,$$

the number of columns is equal to $\sum_{k=1}^{q} m_k$, and the number of rows is $\sum_{l=1}^{N} n_l$. The *mosaic Hankel* structure is a generalization of the *block-Hankel* structure [3], which is defined as

$$\mathscr{H}_{L,K}(C) := \begin{bmatrix} C_1 & C_2 & \cdots & C_K \\ C_2 & \iddots & & C_{K+1} \\ \vdots & \iddots & & \vdots \\ C_L & C_{L+1} & \cdots & C_{L+K-1} \end{bmatrix}, \qquad (\mathscr{H}_{L,K}(C))$$

where C_k are $q \times N$ matrices. Indeed, consider permutation matrices

$$\Phi := \begin{bmatrix} I_L \otimes e_1 & \dots & I_L \otimes e_q \end{bmatrix}^\top,$$
$$\Psi := \begin{bmatrix} I_K \otimes e_1 & \dots & I_K \otimes e_N \end{bmatrix},$$

where $\otimes$ is the Kronecker product. Then the block-Hankel structure ($\mathscr{H}_{L,K}(C)$) can be transformed to a mosaic Hankel structure ($\mathscr{H}_{\mathbf{m},\mathbf{n}}$) by permutation of the rows and columns

$$\Phi \mathscr{H}_{L,K}(C) \Psi = \mathscr{H}_{[L \ \dots \ L],[K \ \dots \ K]}(p), \qquad (\mathscr{H}_{L,K} \leftrightarrow \mathscr{H})$$

where the parameter vector p is defined as (p) with $p_i^{(k,l)} := (C_i)_{k,l}$. The parameter mapping corresponds to an unfolding of the 3-dimensional tensor defined by the matrices C_k.

1.2. Optimization methods and the variable projection

Problem (SLRA) is nonconvex and except for a few special cases (*e.g.* for unstructured matrices and Frobenius norm, for circulant matrices, and for some classes of square matrices, see [5, 1]) has no closed-form solution.

Different optimization methods have been developed for different structures and approximation criteria, see [1] for a historical overview. Many optimization methods use the fact that the rank constraint rank $\widehat{D} \leq r$ is equivalent to existence of a full row rank matrix $R \in \mathbb{R}^{d \times r}$ satisfying $R\widehat{D} = 0$, where $d := m - r$ is the *rank reduction* in (SLRA). Problem (SLRA) then can be rewritten (for weighted 2-norm) as

$$\underset{R \in \mathbb{R}^{d \times m}, \widehat{p} \in \mathbb{R}^{n_p}}{\text{minimize}} \ \|p_{\mathrm{d}} - \widehat{p}\|_{\mathrm{W}}^2 \quad \text{subject to} \quad \operatorname{rank} R = d \quad \text{and} \quad R\mathscr{S}(\widehat{p}) = 0. \qquad (\mathrm{SLRA}_R)$$

Methods for (SLRA_R) include Riemannian SVD [6], structured total least norm approach [7, 8], and variable projection. Note that most of the optimization methods mentioned above were developed for the *structured total least squares* problem, which is the problem (SLRA_R) with an additional contraint

$$R = \begin{bmatrix} X & -I_d \end{bmatrix}, \quad X \in \mathbb{R}^{d \times r}. \qquad (\mathrm{STLS})$$

Most of the methods are designed for 2-norm approximation criteria.

The *variable projection* approach was proposed in [9] for separable non-linear least squares problems. Variable projection was first applied for some special cases of (SLRA$_R$) [10, 11]. In the variable projection aproach, (SLRA$_R$) is rewritten as

$$\underset{R\in\mathbb{R}^{d\times m},\operatorname{rank} R=d}{\text{minimize}} f(R), \quad \text{where} \tag{OUTER}$$

$$f(R) := \Big(\min_{\widehat{p}\in\mathbb{R}^{n_p}} \|p_{\mathrm{d}} - \widehat{p}\|^2_{\mathrm{W}} \quad \text{subject to} \quad R\mathscr{S}(\widehat{p}) = 0 \Big). \tag{$f(R)$}$$

The inner minimization problem ($f(R)$) is a linear *least-norm problem* [12] and has a closed form solution (see also [1]). Therefore (SLRA) is reduced to optimization of the function ($f(R)$) on a space of dimension dm, which is typically much smaller than the dimension n_p of the eliminated variable $\widehat{p}$.

The cost function $f(R)$ depends only on the subspace spanned by the rows of the argument R, *i.e.* $f(R_1) = f(R_2)$ for rowspan $R_1 =$ rowspan R_2. Therefore, $f(R)$ can be considered as a function defined on the Grassmann manifold [13] of all d-dimensional subspaces of $\mathbb{R}^n$, and the problem (OUTER) is optimization on a Grassmann manifold. The optimization problem on a Grassmann manifold can be either transformed to an optimization problem on an Euclidean space (see [14, Sec. 2]), or can be solved by iterations in the tangent space (see, for example, [10, 13]). Therefore, standard optimization routines can be used to minimize (OUTER).

The computation of the cost function has complexity $O(n^3)$ if the inner minimization problem ($f(R)$) is solved by general-purpose methods (*e.g.* the QR decomposition [15]). For analytic computation of derivatives of $f(R)$, which can speed up the convergence of local optimization methods, only algorithms with complexity $O(n^3)$ were proposed in the case of general affine structure [16].

In [11, 17] it was shown that the variable projection approach leads to efficient (with complexity $O(mn)$) local optimization methods for solution of (SLRA) (with (STLS) restriction on R) with 2-norm for structures of type $\begin{bmatrix} C^{(1)} & \cdots & C^{(q)} \end{bmatrix}^\top$, where each block $C^{(l)}$ is block-Toeplitz, block-Hankel or unstructured, and only whole blocks can be fixed.

1.3. Main results and composition of the paper

In this paper we show that the cost function $f(R)$, its gradient and an approximation of its Hessian can be evaluated in $O(m^2 n)$ operations for structure ($\Phi\mathscr{H}_{\mathbf{m},\mathbf{n}}$) and elementwise weighted 2-norm (that allows fixed values constraints). If the weights are block-wise (or only whole blocks are fixed), the cost function and the gradient can be evaluated in $O(mn)$ operations, as in [18].

We develop algorithms for evaluation of $f(R)$ and its gradient for general affine structure and arbitrary weighted 2-norm. The structure of $f(R)$ is derived in a similar way to [11], but in contrast to [11, 17] we do not use a probabilistic interpretation of (SLRA). Instead, we show how the matrix structure ($\mathscr{S}$) is mapped to the structure of the cost function. In addition, our definition of weighted 2-norm incorporates fixed values constraints, which also simplifies the derivation of the cost function structure.

We provide an explicit derivation of the approximation of the Hessian of $f(R)$, for general affine structure and weighted 2-norm. This derivation was omitted in [17] and other papers, but was used in the computational routines in [18]. We provide two variants of the approximation of the Hessian, based on two different representations of $f(R)$ as a sum of squares.

The paper is organized as follows. In Section 2 we review some basic properties of affine structures and weighted 2-norm. Section 3 covers the derivation of $f(R)$ for general affine structure and weighted 2-norm. For blocked structures the cost function is expressed via cost functions for the blocks. Based on results of Section 3, we develop general-purpose algorithms for evaluation of the cost function and its derivatives in Section 4. Finally, in Section 5 we specialize the algorithms for mosaic Hankel structures and derive their computational complexity.

1.4. Notation

Some of the notation used in the paper is summarized in Table 1.

$\mathbf{0}$	—	vector of zeros
$\mathbf{1}$	—	vector of ones
e_i	—	ith unit vector
$\mathrm{col}(v_1, \ldots, v_N)$	—	concatenation of vectors $v_1, \ldots, v_N$
$\mathrm{blkdiag}\,(A_1, \ldots, A_N)$	—	block-diagonal matrix composed of matrices $A_1, \ldots, A_N$
vec	—	column-wise vectorization of a matrix
W^{-1}	—	either inverse of W if W is positive definite, or $\mathrm{diag}(\gamma_1, \ldots \gamma_{n_p})$, where $\gamma_k := \begin{cases} w_k^{-1}, & w_k < +\infty, \\ 0, & w_k = +\infty, \end{cases}$ if W is given by ($w \to \mathrm{W}$)
L_{A}	—	a right Cholesky factor of a positive semi-definite matrix A (a lower-triangular matrix that satisfies $L_{\mathrm{A}}^\top L_{\mathrm{A}} = \mathrm{A}$), L_{A} is unique if A is positive definite

Table 1: Notation

2. Affine structures and weighted 2-norms

2.1. Basic properties

Affine structures can be defined in an equivalent to ($\mathscr{S}$) *vector form*

$$\mathrm{vec}\,\mathscr{S}(p) = \mathrm{vec}\,S_0 + \mathbf{S}_{\mathscr{S}}\,p, \qquad (\mathrm{vec}\,\mathscr{S})$$

where

$$\mathbf{S}_{\mathscr{S}} := \begin{bmatrix} \mathrm{vec}\,S_1 & \cdots & \mathrm{vec}\,S_{n_p} \end{bmatrix}, \qquad (\mathbf{S}_{\mathscr{S}})$$

is the matrix representation of the linear part of ($\mathscr{S}$) in the basis of elementary matrices $\{e_i e_j^\top\}_{i,j=1}^{m,n}$.

Example 1. 2×3 *Hankel matrices* $\mathscr{H}_{2,3}(p)$ *can be represented as* ($\mathscr{S}$) *with*

$$S_0 = \mathbf{0},\ S_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ S_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},\ S_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this case

$$\mathbf{S}_{\mathscr{S}} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & 1 & & \\ & & 1 & \\ & & 1 & \\ & & & 1 \end{bmatrix},$$

where blank elements denote zeros.

Using the representation (vec $\mathscr{S}$), we show that the Frobenius norm is a weighted 2-norm.

Note 1. *Let* ($\mathscr{S}$) *be an injective map (which corresponds to linearly independent* $\{S_k\}_{k=1}^{n_p}$ *or* ($\mathbf{S}_{\mathscr{S}}$) *with full column rank). Then*

$$\|\mathscr{S}(\widehat{p}) - \mathscr{S}(p_d)\|_{\mathrm{F}}^2 = \|\mathbf{S}_{\mathscr{S}}(\widehat{p} - p_d)\|_2^2 = \|\widehat{p} - p_d\|_{\mathrm{W}}^2,$$

where $\mathrm{W} := \mathbf{S}_{\mathscr{S}}^\top \mathbf{S}_{\mathscr{S}}$ *is the Gramian of the system of vectors* $\{S_k\}_{k=1}^{n_p}$.

Example 2. *In Example 1, the Gramian* $\mathbf{S}_{\mathscr{S}}^\top \mathbf{S}_{\mathscr{S}}$ *is diagonal, and the weighted norm corresponding to the Frobenius norm is given by* ($w \to \mathrm{W}$) *with* $w = \mathrm{col}(1, 2, 2, 1)$.

It is not difficult to show that the Frobenius norm in Note 1 can be replaced by any weighted Frobenius norm (weighted 2-norm of vec $(\mathscr{S}(\widehat{p}) - \mathscr{S}(p_{\mathrm{d}}))$).

2.2. From weighted 2*-norm to* 2*-norm*

Any (SLRA) problem with weighted 2-norm can be reduced to an unweighted (SLRA) problem. Consider a change of parameters

$$\widehat{p} = p_{\mathrm{d}} - L_{\mathrm{W}^{-1}}^\top \Delta p. \qquad (\Delta p \to \widehat{p})$$

Then the transformed structure

$$\mathscr{S}_\Delta(\Delta p) := \mathscr{S}(p_{\mathrm{d}}) - \left(\mathscr{S}\left(L_{\mathrm{W}^{-1}}^\top \Delta p\right) - S_0\right), \qquad (\mathscr{S}_\Delta)$$

for any Δp satisfies

$$\mathscr{S}_\Delta(\Delta p) = \mathscr{S}(\widehat{p}). \qquad (\mathscr{S}_\Delta \leftrightarrow \mathscr{S})$$

Moreover,

$$\|\widehat{p} - p_{\mathrm{d}}\|_{\mathrm{W}}^2 = \Delta p^\top L_{\mathrm{W}^{-1}} \mathrm{W} L_{\mathrm{W}^{-1}}^\top \Delta p,$$

and $\|\widehat{p} - p_{\mathrm{d}}\|_{\mathrm{W}}^2 = \|\Delta p\|_2^2$ is W is positive definite. In general, the following result holds.

Proposition 1. *Any problem* (SLRA) *with weighted norm* $\|\cdot\|_{\mathrm{W}}$ *is equivalent to*

$$\underset{\Delta p \in \mathbb{R}^{n_p}}{\textit{minimize}} \ \|\Delta p\|_2^2 \quad \textit{subject to} \quad \textit{rank}\, \mathscr{S}_\Delta(\Delta p) \leq r. \qquad (\mathrm{SLRA}_\Delta)$$

Proof. The proof is given in Appendix A. □

3. Variable projection

3.1. Variable projection for weighted 2*-norm*

In this subsection we derive an explicit expression for the cost function ($f(R)$) for general affine structure and weighted 2-norm. Consider the change of variables ($\Delta p \to \widehat{p}$). Then, by properties of the vectorization operator and ($\mathscr{S}_\Delta \leftrightarrow \mathscr{S}$), we have

$$\mathrm{vec}(R\mathscr{S}(\widehat{p})) = (I_d \otimes R)\,\mathrm{vec}\,\mathscr{S}(\widehat{p}) = (I_d \otimes R)\left(\mathrm{vec}\,\mathscr{S}(p_{\mathrm{d}}) - \mathbf{S}_{\mathscr{S}} L_{\mathrm{W}^{-1}}^\top \Delta p\right).$$

The optimization problem in $(f(R))$ can be rewritten as

$$\begin{aligned} &\underset{\Delta p}{\text{minimize}} \ \|\Delta p\|_2^2 \\ &\text{subject to } \ G(R)L_{\mathrm{W}^{-1}}^{\top}\Delta p = \nu(R), \end{aligned} \tag{LN}$$

where

$$\nu(R) := (I_n \otimes R)\,\mathrm{vec}\,\mathscr{S}(p_d) = \mathrm{vec}\,(R\mathscr{S}(p_d))\,, \tag{$\nu(R)$}$$
$$G(R) = G_{\mathscr{S}}(R) := (I_n \otimes R)\mathbf{S}_{\mathscr{S}} = \begin{bmatrix}\mathrm{vec}\,(RS_1) & \cdots & \mathrm{vec}\,\left(RS_{n_p}\right)\end{bmatrix}. \tag{$G_{\mathscr{S}}(R)$}$$

Example 3. *In Example 1, for* $R = \begin{bmatrix} r_1 & r_2 \end{bmatrix}$

$$G(R) = \begin{bmatrix} r_1 & r_2 & & \\ & r_1 & r_2 & \\ & & r_1 & r_2 \end{bmatrix}.$$

Problem (LN) is a linear *least-norm problem* in a standard form [12, Ch. 6], and has a closed-form solution, which we summarize in the following proposition.

Proposition 2. *Let* $G(R)$ *be of full row rank*[1]*. Then the solution of* (LN) *exists and is given by*

$$\Delta p^*(R) = L_{\mathrm{W}^{-1}}G^{\top}(R)\Gamma^{-1}(R)\nu(R), \tag{$\Delta p^*(R)$}$$
$$f(R) = \|\Delta p^*(R)\|_2^2 = \nu^{\top}(R)\Gamma^{-1}(R)\nu(R), \tag{$\nu^{\top}\Gamma^{-1}\nu$}$$

where

$$\Gamma(R) := \begin{bmatrix} \Gamma_{1,1}(R) & \cdots & \Gamma_{1,n}(R) \\ \vdots & & \vdots \\ \Gamma_{n,1}(R) & \cdots & \Gamma_{n,n}(R) \end{bmatrix} \in \mathbb{R}^{nd\times nd}, \quad \Gamma_{i,j}(R) := R\mathrm{V}_{i,j}R^{\top}, \tag{$\Gamma(R)$}$$

where the matrices $\mathrm{V}_{i,j}$ *are* $d \times d$ *blocks of the matrix* V

$$\mathrm{V} = \begin{bmatrix} \mathrm{V}_{1,1} & \cdots & \mathrm{V}_{1,n} \\ \vdots & & \vdots \\ \mathrm{V}_{n,1} & \cdots & \mathrm{V}_{n,n} \end{bmatrix}, \qquad \mathrm{V}_{i,j} \in \mathbb{R}^{m\times m},$$

and the matrix V *is defined as*

$$\mathrm{V} = \mathrm{V}_{\mathscr{S},\mathrm{W}} := \mathbf{S}_{\mathscr{S}}\mathrm{W}^{-1}\mathbf{S}_{\mathscr{S}}^{\top}. \tag{V}$$

Proof. Note that by definition

$$\Gamma(R) = (I_n \otimes R)\mathrm{V}(I_n \otimes R^{\top}) = G(R)\mathrm{W}^{-1}G^{\top}(R). \tag{$G\mathrm{W}^{-1}G^{\top}$}$$

Then $(\Delta p^*(R))$ coincides with the solution of the least-norm prolem (LN). □

[1]If $G(R)$ is not of full row rank but the problem is feasible (the constraint in (OUTER) is consistent), the solution is given by $f(R) = \nu^{\top}(R)\Gamma(R)^{\dagger}\nu(R)$.

Example 4. *In Example 2,*

$$\mathrm{V}_{\mathscr{S}} = \begin{bmatrix} 1 & & & & & \\ & 2 & 2 & & & \\ & 2 & 2 & & & \\ & & & 2 & 2 & \\ & & & 2 & 2 & \\ & & & & & 1 \end{bmatrix}$$

Examples 3 and 4 are generalized in Section 5.1.

The structure of Γ is determined by the structure of V. For example, if V is block-sparse, block-banded or block-Toeplitz, then Γ has the same structural property.

Note 2. *The representation of type* ($\Gamma(R)$) *was already derived in [11, 17] from statistical considerations. Here we derive it through a series of linear algebraic transformations.*

3.2. Variable projection for blocked matrices

We consider basic transformations of the structures and derive the form of ($f(R)$) for these transformed structures. First, we consider striped and layered block matrices.

Lemma 1 (Striped structure)**.** *Let* $p = \mathrm{col}(p^{(1)}, \ldots, p^{(N)})$*, with* $p_{(l)} \in \mathbb{R}^{n_p^{(l)}}$ *and*

$$\mathscr{S}(p) = \begin{bmatrix} \mathscr{S}^{(1)}(p^{(1)}) & \cdots & \mathscr{S}^{(N)}(p^{(N)}) \end{bmatrix},$$

be the striped structure, $\mathscr{S}_l : \mathbb{R}^{n_p^{(l)}} \to \mathbb{R}^{m \times n_l}$*, and* $\mathrm{W} = \mathrm{blkdiag}\left(\mathrm{W}^{(1)}, \ldots, \mathrm{W}^{(N)}\right)$*. Then*

1. *The cost function* ($f(R)$) *for the striped structure is equal to the sum*

$$f(R) = \sum_{l=1}^{N} f_l(R), \qquad \text{(striped } f\text{)}$$

   where f_l *is the cost function* ($f(R)$) *for the structure* $\mathscr{S}_l$ *and the weight matrix* $\mathrm{W}^{(l)}$*.*

2. *The correction* $\Delta p^*(R)$ *is the concatenation*

$$\Delta p^*(R) = \mathrm{col}\left(\Delta p^{(1)*}(R), \ldots, \Delta p^{(N)*}(R)\right) \qquad \text{(striped } \Delta p^*\text{)}$$

   of the corrections ($\Delta p^*(R)$) *for the structures* $\mathscr{S}^{(l)}$ *and the weighted matrices* $\mathrm{W}^{(l)}$*.*

Proof. The inner minimzation problem (OUTER) can be expressed as

$$\underset{\Delta p^{(l)} \in \mathbb{R}^{n_p^{(l)}}}{\text{minimize}} \sum_{l=1}^{N} \|\Delta p^{(l)}\|_2^2 \quad \text{subject to} \quad R\mathscr{S}_{\Delta}^{(l)}(\Delta p^{(l)}) = 0, \quad \text{for } l = 1, \ldots, N.$$

This sum of squares is therefore minimized by (striped Δp^*), and its norm is given by (striped f). □

Lemma 2 (Layered structure)**.** *Let* $p = \mathrm{col}(p^{(1)}, \ldots, p^{(q)})$*, with* $p^{(k)} \in \mathbb{R}^{n_p^{(k)}}$ *and*

$$\mathscr{S}(p) = \begin{bmatrix} \mathscr{S}^{(1)}(p^{(1)}) \\ \vdots \\ \mathscr{S}^{(q)}(p^{(q)}) \end{bmatrix},$$

be a layered structure, where $\mathscr{S}^{(k)} : \mathbb{R}^{n_p^{(k)}} \to \mathbb{R}^{m_k \times n}$*.*

1. *The G matrix is*
$$G(R) = \begin{bmatrix} G^{(1)}(R^{(1)}) & \cdots & G^{(q)}(R^{(q)}) \end{bmatrix},$$
where $R = \begin{bmatrix} R^{(1)} & \cdots & R^{(q)} \end{bmatrix}$ is the partition of R into $R^{(k)} \in \mathbb{R}^{d \times m_k}$.
2. *The Γ matrix is equal to the sum*
$$\Gamma(R) = \sum_{k=1}^{q} \Gamma^{(k)}(R^{(k)})$$
of the matrices ($\Gamma(R)$) *corresponding to* (SLRA) *with $\mathscr{S}^{(k)}$ and $\mathrm{W}^{(k)}$.*
3. *Let $\mathrm{V}^{(k)}$ be the matrix* (V) *for the structure $\mathscr{S}^{(k)}$ and the weight matrix $\mathrm{W}^{(k)}$. Then the matrix* (V) *for $\mathscr{S}$ and W is given by*
$$\mathrm{V}_{i,j} = \mathrm{blkdiag}\left(\mathrm{V}_{i,j}^{(1)}, \ldots, \mathrm{V}_{i,j}^{(q)}\right).$$
4. *The correction $\Delta p^*(R)$ can be expressed as*
$$\Delta p^*(R) = \mathrm{col}\left(\Delta p^{(1)*}(R), \ldots, \Delta p^{(N)*}(R)\right),$$
where $\Delta p^{(k)*}(R) := L_{(\mathrm{W}^{(k)})^{-1}}(G^{(k)})^\top \big(\Gamma(R)\big)^{-1} \nu(R)$.

Proof. It is sufficient to prove statement 1, which follows from
$$\mathrm{vec}(R\mathscr{S}(p)) = \sum_{k=1}^{q} \mathrm{vec}\left(R^{(k)}\mathscr{S}^{(k)}(p^{(k)})\right) = \sum_{k=1}^{q} \left(G^{(k)}(R^{(k)})p^{(k)} + \mathrm{vec}\, S_0^{(k)}\right).$$
The other statements follow from (V) and ($G\mathrm{W}^{-1}G^\top$). □

Next we examine the effect of left multiplication of the structure by a full-row rank matrix.

Lemma 3 (Multiplication by Φ)**.** *Let*
$$\mathscr{S}(p) = \Phi\mathscr{S}'(p)$$
be an affine structure, where $\mathscr{S}' : \mathbb{R}^{n_p} \to \mathbb{R}^{m' \times n}$ and $\Phi \in \mathbb{R}^{m \times m'}$ is a full-row-rank matrix. Then,
$$f_{\mathscr{S}}(R) = f_{\mathscr{S}'}(R\Phi).$$

Proof. This property is easily verified by rewriting the constraint in ($f(R)$) as
$$R\mathscr{S}(\widehat{p}) = R\Phi(\widehat{p}) = 0,$$
where $R\Phi$ is a full row rank matrix. □

4. Cost function and derivatives evaluation

In this section, we develop algorithms for computing the cost function ($f(R)$) and its derivatives using the representation ($\nu^\top \Gamma^{-1} \nu$), for the general affine structure $\mathscr{S}$. The algorithms can be specialized for a specific class of structures by deriving the form of $\mathrm{V}_{i,j}$, as it will be done in Section 5 for the mosaic Hankel structure.

In Section 4.1, we provide algorithms for evaluation of ($f(R)$), computation of the optimal $\widehat{p}$ in ($f(R)$), and computation of the gradient of ($f(R)$).

In Sections 4.2 and 4.3, we consider the following approximation of Hessian of ($f(R)$). Let $f(R) = \|F(R)\|_2^2$, where $F(R) \in \mathbb{R}^{n_s}$ with $n_s \gg dm$. Then $J_F^\top J_F$ is an approximation of the Hessian, where J_F is the Jacobian of F. This approximation is frequently used in methods of solution of nonlinear least-squares problems, *e.g.* in the Levenberg-Marquardt algorithm [19]. In Section 4.2, we consider the Jacobian $F(\cdot) = \Delta p^*(\cdot)$, where $\Delta p^*(\cdot)$ is defined in ($\Delta p^*(R)$). In Section 4.3 we consider $F(\cdot) = (L_\Gamma^\top)^{-1}\nu(\cdot)$, where L_Γ is the right Cholesky factor of Γ.

We will frequently use the following notation

$$y_\nu := \Gamma^{-1}\nu(R), \tag{y_ν}$$

for the solution of the system $\Gamma y_\nu = \nu(R)$.

4.1. Cost function, correction and gradient

From (y_ν) and ($\nu^\top \Gamma^{-1} \nu$), the cost function ($f(R)$) can be represented as

$$f(R) = y_\nu^\top(R)\nu(R) = \nu(R)^\top y_\nu(R), \tag{$y_\nu^\top \nu$}$$

and can be computed by Algorithm 1.

Algorithm 1 (Cost function evaluation)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output:* $f(R)$.

1. *Compute* ($\nu(R)$).
2. *Compute* ($\Gamma(R)$) *using* $\mathrm{V}_{i,j}$.
3. *Compute* (y_ν).
4. *Compute the cost function as* ($y_\nu^\top \nu$).

The term $\Delta p^*(R)$ in ($\Delta p^*(R)$) can be evaluated by Algorithm 2.

Algorithm 2 (Correction computation)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output:* $\Delta p^*(R)$.

1. *Perform steps 1–3 of Algorithm 1.*
2. *Multiply* y_ν *by* $L_{\mathrm{W}^{-1}} G^\top(R)$ *on the left.*

The optimal $\widehat{p}$ in ($f(R)$) can be computed by combining Algorithm 2 and ($\Delta p \to \widehat{p}$).

Algorithm 3 (Computation of $\widehat{p}^*$)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output:* $\widehat{p}^*(R)$.

1. *Perform steps 1–3 of Algorithm 1.*
2. *Multiply* y_ν *by* $\mathrm{W}^{-1} G^\top(R)$ *on the left.*

Instead of the gradient ∇f, for convenience, we use the *matrix gradient* $\nabla_{d\times m} f \in \mathbb{R}^{d\times m}$, defined as

$$\mathrm{vec}(\nabla_{d\times m} f) := \nabla f.$$

Proposition 3. *Let $y_\nu = \mathrm{col}(y_\nu^{(1)}, \cdots, y_\nu^{(n)})$ be the partition of y_ν into $y_\nu^{(k)} \in \mathbb{R}^d$, and*

$$Y_\nu := \begin{bmatrix} y_\nu^{(1)} & \cdots & y_\nu^{(n)} \end{bmatrix}, \quad y_\nu = \mathit{vec}\, Y_\nu. \tag{Y_ν}$$

Then the matrix gradient is given by

$$\nabla_{d\times m}(f) = 2Y_\nu \mathscr{S}^\top(p) - 2\sum_{i,j=1}^{m} y_\nu^{(j)} (y_\nu^{(i)})^\top R \mathrm{V}_{i,j}. \tag{$\nabla_{d\times m}$}$$

Proof. The proof is given in Appendix A. □

From Proposition 3, the gradient can be computed by Algorithm 4.

Algorithm 4 (Gradient evaluation)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. Output: $\nabla_{d\times m} f(R)$.

1. *Perform steps 1–3 of Algorithm 1.*
2. *Compute the first term $2Y_\nu \mathscr{S}^\top(p_d)$ of the gradient ($\nabla_{d\times m}$).*
3. *Compute the second term of the gradient ($\nabla_{d\times m}$).*

4.2. Approximation of the Hessian: Jacobian of Δp^*

The following proposition gives an expression to compute the Jacobian of ($\Delta p^*(R)$).

Proposition 4.

- *The Jacobian of ($\Delta p^*(R)$) is given by*

$$\frac{\partial \Delta p^*}{\partial R_{ij}} = L_{\mathrm{W}^{-1}} \left(G^\top(R) \Gamma^{-1} z_{ij} + \frac{\partial G^\top}{\partial R_{ij}} y_\nu \right), \tag{$\partial \Delta p^* / R_{ij}$}$$

  where

$$z_{ij} := \frac{\partial \nu}{\partial R_{ij}} - \frac{\partial \Gamma}{\partial R_{ij}} y_\nu. \tag{z_{ij}}$$

- *The vector z_{ij} can be expressed as*

$$z_{ij} = \mathrm{row}_j(\mathscr{S}(p_d)) \otimes e_i - \left((z_j^{(1)}) \otimes e_i + z_{ij}^{(2)} \right),$$

  where

$$z_j^{(1)} := \mathrm{col} \left(\sum_{k=1}^{n} e_j^\top \mathrm{V}_{1,k} R^\top y_\nu^{(k)}, \ldots, \sum_{k=1}^{n} e_j^\top \mathrm{V}_{n,k} R^\top y_\nu^{(k)} \right), \tag{$z_j^{(1)}$}$$

$$z_{ij}^{(2)} := \sum_{k=1}^{n} (y_\nu^{(k)})_i \, \mathrm{col}\left(R\mathrm{V}_{1,k} e_j, \ldots, R\mathrm{V}_{n,k} e_j \right). \tag{$z_{ij}^{(2)}$}$$

Proof. The proof is given in Appendix A. □

Proposition 4 leads to the following algorithm.

Algorithm 5 (Jacobian evaluation)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output: Jacobian of* $\Delta p^*(R)$.

1. *Compute steps 1–3 of Algorithm 1.*
2. ***for*** $j = 1, \ldots, m$ ***do***
3. *Compute* ($z_j^{(1)}$).
4. ***for*** $j = 1, \ldots, d$ ***do***
5. *Compute* ($z_{ij}^{(2)}$) *and* (z_{ij}).
6. *Set* $x \leftarrow G^\top \Gamma^{-1} z_{ij}$.
7. *Set* $x \leftarrow x + \frac{\partial G^\top}{\partial R_{ij}} y_\nu$.
8. *Set* $\frac{\partial \Delta p^*}{\partial R_{ij}} \leftarrow L_{\mathrm{W}^{-1}} x$.
9. ***end for***
10. ***end for***

4.3. Using Cholesky factorization

In this section, we show that the cost function and an approximation of the Hessian can be computed using the Cholesky factorization of Γ

$$\Gamma = L_\Gamma^\top L_\Gamma.$$

The Cholesky factorization yields a numerically reliable way [15] to solve the system of equations $\Gamma u = v$ using the following identity

$$\Gamma^{-1} = L_\Gamma^{-1} (L_\Gamma^\top)^{-1}.$$

Algorithm 6 (Solve system $\Gamma u = v$)**.**
Input: Γ, v. *Output:* u.

1. *Compute the Cholesky factor* L_Γ *of* Γ.
2. *Solve* $L_\Gamma^\top F_\nu = v$ *and* $L_\Gamma u = F_\nu$ *by backward substitution.*

Moreover, the cost function can be represented as

$$f(R) = \|(L_{\Gamma(R)}^\top)^{-1} \nu(R)\|_2^2, \qquad (\|L_\Gamma^{\top -1} \nu\|_2^2)$$

which leads to a more efficient algorithm for the cost function evaluation than Algorithm 1.

Algorithm 7 (Cost function evaluation using Cholesky factorization)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output:* $f(R)$.

1. *Compute* ($\nu(R)$).
2. *Compute* ($\Gamma(R)$) *using* $\mathrm{V}_{i,j}$.
3. *Compute the Cholesky factor* L_Γ.
4. *Solve* $L_\Gamma^\top F_\nu(R) = \nu(R)$.
5. *Compute* $f(R) = \|F_\nu(R)\|_2^2$.

In addition, equation ($\|L_\Gamma^{\top\,-1}\nu\|_2^2$) defines a representation of the cost function as a sum of squares, which is more compact and easier to compute than ($\nu^\top\Gamma^{-1}\nu$). Indeed, $\Delta p^*(R) \in \mathbb{R}^{n_p}$, but $F_\nu(R) \in \mathbb{R}^{nd}$. However, the elements of the Jacobian of F_ν

$$\frac{\partial F_\nu}{\partial R_{ij}} = (L_{\Gamma(R)}^\top)^{-1}\frac{\partial s}{\partial R_{ij}} + \frac{\partial (L_{\Gamma(R)}^\top)^{-1}}{\partial R_{ij}}\nu(R),$$

cannot be computed analytically due to the need of differentiation of $(L_\Gamma^\top)^{-1}$. For this purpose the *pseudo-Jacobian* can be used

$$\frac{\partial^{(\mathrm{p})} F_\nu}{\partial R_{ij}} := \left((L_{\Gamma(R)}^\top)^{-1}\frac{\partial \nu(R)}{\partial R_{ij}} - \frac{1}{2}(L_{\Gamma(R)}^\top)^{-1}\frac{\partial \Gamma}{\partial R_{ij}}\Gamma^{-1}\nu(R)\right). \qquad (\partial^{(\mathrm{p})}F_\nu/\partial R_{ij})$$

In [20] it was shown that for functions of the form ($\nu^\top\Gamma^{-1}\nu$), the pseudo-Jacobian yields the stationary points of $f(R)$ and gives an approximation of the Hessian of f.

It is easy to see that ($\partial^{(\mathrm{p})}F_\nu/\partial R_{ij}$) and ($z_{ij}$) differ only by a constant factor in one summand. Therefore,

$$\begin{aligned} &\frac{\partial^{(\mathrm{p})} F_\nu}{R_{ij}} = (L_{\Gamma(R)}^\top)^{-1} z_{ij}^{(\mathrm{p})}, \\ &z_{ij}^{(\mathrm{p})} := \mathrm{row}_j(\mathscr{S}(p_{\mathrm{d}})) \otimes e_i - \frac{1}{2}\left((z_j^{(1)}) \otimes e_i + z_{ij}^{(2)}\right). \end{aligned} \qquad (z_{ij}^{(\mathrm{p})})$$

The resulting algorithm is Algorithm 8.

Algorithm 8 (Pseudo-Jacobian evaluation)**.**
Input: $\mathscr{S}$, W, $\mathrm{V}_{i,j}$, p_d, R. *Output: pseudo-Jacobian.*

1. *Compute steps 1–2 of gradient evaluation using Algorithm 6 for solution of system of equation.*
2. ***for*** $j = 1, \ldots, m$ ***do***
3. *Compute* ($z_j^{(1)}$).
4. ***for*** $j = 1, \ldots, d$ ***do***
5. *Compute* ($z_{ij}^{(2)}$) *and* ($z_{ij}^{(\mathrm{p})}$).
6. *Multiply* ($z_{ij}^{(\mathrm{p})}$) *on the left by* $(L_{\Gamma(R)}^\top)^{-1}$, *using the precomputed Cholesky factor.*
7. ***end for***
8. ***end for***

5. Weighted mosaic Hankel structured low-rank approximation problem

In this section, we establish the form of $\Gamma(R)$, $\mathrm{V}_{i,j}$ and $f(R)$ for the structure ($\Phi\mathscr{H}_{\mathbf{m},\mathbf{n}}$) and weight matrix

$$\mathrm{W} = \mathrm{blkdiag}\left(\mathrm{W}^{(1,1)}, \ldots, \mathrm{W}^{(q,N)}\right). \qquad (\mathrm{blkdiag}\,\mathrm{W})$$

In view of Lemmae 1–3, we can consider only scalar Hankel structure.

5.1. Scalar Hankel matrices

Let

$$J_m = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ & & \ddots & 1 \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{J_m}$$

be the right shift matrix for a row vector, *i.e.*

$$\begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix} J_m = \begin{bmatrix} 0 & x_1 & \cdots & x_{m-1} \end{bmatrix}.$$

Lemma 4. *For scalar Hankel structure* ($\mathbf{S}_{\mathscr{S}}$) *has the following form*

$$\mathbf{S}_{\mathscr{H}_{m,n}} := \begin{bmatrix} \begin{bmatrix} I_m & 0 \end{bmatrix} J_{n_p}^0 \\ \begin{bmatrix} I_m & 0 \end{bmatrix} J_{n_p}^1 \\ \vdots \\ \begin{bmatrix} I_m & 0 \end{bmatrix} J_{n_p}^{n-1} \end{bmatrix} \tag{$\mathbf{S}_{\mathscr{H}_{m,n}}$}$$

and the matrix $G_{\mathscr{H}_{m,n}}$ *has the structure*

$$G_{\mathscr{H}_{m,n}}(R) = \begin{bmatrix} R_1 & R_2 & \dots & R_m & & \\ & R_1 & R_2 & \dots & R_m & \\ & & \ddots & & & \ddots \\ & & & R_1 & R_2 & \dots & R_m \end{bmatrix} \in \mathbb{R}^{nd \times n_p}, \tag{$G_{\mathscr{H}_{m,n}}$}$$

where $R = \begin{bmatrix} R_1 & \dots & R_m \end{bmatrix} \in \mathbb{R}^{m \times d}$, *with* $R_i \in \mathbb{R}^d$.

Proof. Indeed, for a vector $p = \begin{bmatrix} p_1 & \dots & p_{n_p} \end{bmatrix}^\top$, we have

$$\begin{bmatrix} I_m & 0 \end{bmatrix} J_{n_p}^k p = \begin{bmatrix} p_{k+1} & \dots & p_{k+m} \end{bmatrix}. \tag{$[I\ 0]J^k$}$$

Therefore, (vec $\mathscr{S}$) holds with $S_0 = 0$.

By substituting ($\mathbf{S}_{\mathscr{H}_{m,n}}$) into ($G_{\mathscr{S}}(R)$) the equality ($G_{\mathscr{H}_{m,n}}$) can be verified. □

We next consider the case of Hankel low-rank approximation with weighted 2-norm.

Proposition 5. *For Hankel structure and a weight matrix* W *the matrix* (V) *is equal to*

$$\mathrm{V}_{i,j} = (\mathrm{W}^{-1})_{i,j}^{m \times m},$$

where $(\mathrm{W}^{-1})_{i,j}^{m \times m}$ *is a* $m \times m$ *submatrix of* W^{-1}, *starting from element* (i, j).

Proof. From (V) and ($\mathbf{S}_{\mathscr{H}_{m,n}}$), we have that

$$\mathrm{V}_{i,j} = \begin{bmatrix} I_m & 0 \end{bmatrix} J_{n_p}^{i-1} \mathrm{W}^{-1} (J_{n_p}^\top)^{j-1} \begin{bmatrix} I_m \\ 0 \end{bmatrix}. \tag{$V_{i,j}(\mathscr{H}_{m,n})$}$$

From ($[I\ 0]J^k$) one can verify that ($V_{i,j}(\mathscr{H}_{m,n})$) corresponds to selecting an $m \times m$ submatrix of W^{-1}. □

We will call a symmetric matrix s-*banded* (s-*block-banded*) if all upper diagonals (resp. upper block diagonals) starting from s-th are zero.

Corollary 1. 1. *If* W^{-1} *is* s*-banded then* V *and* Γ_{V} *are* $(m+s-1)$*-block-banded (with* $d \times d$ *blocks).*

2. *If* W^{-1} *is Toeplitz then* V *and* Γ *are block-Toeplitz.*
3. *For the case* ($w \to \mathrm{W}$) V *and* Γ *are* m*-block-banded. In particular,*
   (a) *For* $\mathrm{W} = I_d$ *(ordinary* 2*-norm),*

$$\mathrm{V}_{i,j} = \mathrm{V}_{i-j} := \begin{cases} (J_m^\top)^{j-i}, & \text{for } j \geq i, \\ (J_m)^{i-j}, & \text{for } j < i. \end{cases}$$

   (b) *For* $\mathrm{W} = \operatorname{diag}(w_1, \ldots, w_{n_p})$ *(elementwise weights and fixed values),*

$$\mathrm{V}_{i,j} = \begin{cases} \operatorname{diag}(\gamma_i, \ldots, \gamma_{i+m-1})(J_m^\top)^{j-i}, & \text{for } j \geq i, \\ (\mathrm{V}_{j,i})^\top, & \text{for } j < i, \end{cases}$$

   where $\gamma_i := w_i^{-1}$.

5.2. The main theorem

By Lemma 1, we can consider only the layered Hankel structure

$$\mathscr{H}_{\mathbf{m},[n]}(p) = \begin{bmatrix} \mathscr{H}_{m_1,n}(p^{(1)}) \\ \vdots \\ \mathscr{H}_{m_q,n}(p^{(q)}) \end{bmatrix}, \quad \text{where } \mathbf{m} = \begin{bmatrix} m_1 & \cdots & m_q \end{bmatrix}. \qquad (\mathscr{H}_{\mathbf{m},[n]})$$

Theorem 1. *For structure* ($\mathscr{H}_{\mathbf{m},[n]}$) *and* $\mathrm{W} = \operatorname{blkdiag}\left(\mathrm{W}^{(1)}, \ldots, \mathrm{W}^{(q)}\right)$*, with* $(\mathrm{W}^{(k)})^{-1}$ *being* $b^{(k)}$*-banded*

1. *The matrix* Γ *is* μ*-block banded with* $d \times d$ *blocks, with* $\mu := \max_k\{m_k + b^{(k)} - 1\}$.
2. *If* $\mathrm{C}_{\mathrm{W}^{(k)}}$ *are all Toeplitz, then* Γ *is block-Toeplitz.*
3. *For the case* ($w \to \mathrm{W}$) *and* $\mathrm{W}^{(k)} = \operatorname{diag} w^{(k)}$, $w^{(k)} \in \mathbb{R}^{n_p^{(k)}}$ *the matrices* $\mathrm{V}_{\mathscr{H}_{\mathbf{m},n}}$ *and* $\Gamma_{\mathscr{H}_{\mathbf{m},n}}(R)$ *are block-banded with block bandwidth*

$$\mu := \max\{m_l\}_{l=1}^q.$$

   In particular,
   (a) *the matrices* $\mathrm{V}_{i,j}$ *for* $j \geq i$ *can be expressed as*

$$\mathrm{V}_{i,j} = \operatorname{diag}(\gamma_i^{(1)}, \ldots, \gamma_{i+m_1-1}^{(1)}, \ldots, \gamma_i^{(q)}, \ldots, \gamma_{i+m_q-1}^{(q)})(J_{\mathbf{m}}^\top)^{j-i}, \qquad (\mathrm{V}_{i,j}(\mathscr{H}_{\mathbf{m},[n]}))$$

   where $\gamma^{(k)} = (w^{(k)})^{-1}$ *and* $J_{\mathbf{m}} := \operatorname{blkdiag}\left(J_{m_1}, \ldots, J_{m_q}\right)$*;*
   (b) *if the weights are constant for the blocks of* ($\mathscr{H}_{\mathbf{m},[n]}$)*,* i.e.*,*

$$w = \operatorname{col}\left(w_1 \mathbf{1}_{n_p^{(1)}}, \ldots, w_q \mathbf{1}_{n_p^{(q)}}\right),$$

   then $\mathrm{V}_{\mathscr{H}_{\mathbf{m},n}}$ *is block-Toeplitz,* i.e.*,* $(\mathrm{V}_{\mathscr{H}_{\mathbf{m},n}})_{i,j} = \mathrm{V}_{j-i}$*, and*

$$\Gamma_{i,j}(R) = \Gamma_{j-i}, \quad \Gamma_k = R\mathrm{V}_k R^\top,$$

   where

$$\mathrm{V}_k = \operatorname{diag}(\gamma_1 I_{m_1}, \ldots, \gamma_q I_{m_q})(J_{\mathbf{m}}^\top)^k \text{ and } \quad \gamma_l := w_l^{-1}.$$

Proof. Theorem 1 follows from Lemma 2 and Corollary 1. □

5.3. Complexity of the algorithms for the mosaic Hankel structure

In this section we use Theorem 1 and the fact that the Γ matrix is μ-block-banded. We consider only the case of layered Hankel structure ($\mathscr{H}_{\mathbf{m},[n]}$) and count only the number of multiplications. The number of additions is typically less than the number of multiplications.

Lemma 5. *The complexity of the steps in the Algorithm 6 is given by $O(d^3\mu^2 n)$ for Step 1 and $O(d^2\mu n)$ for Step 2.*

Proof. The $\Gamma \in \mathbb{R}^{nd\times nd}$ is (μd)-banded, and the complexity of both steps is given in [15]. □

Theorem 2. *The complexity of cost function evaluation using Algorithm 7 and gradient evaliation using Algorithm 4 is $(d^3\mu mn)$.*

Proof. The proof is given in Appendix A. □

Note 3. *Due to Lemma 1, for mosaic Hankel structure $\mathscr{H}_{\mathbf{m},\mathbf{n}}$ with $\mathbf{n} = \begin{bmatrix} n_1 & \cdots & n_N \end{bmatrix}$ the complexity of cost function and gradient evaluation is $O(d^3\mu mn)$.*

Note 4. *The computation on the Step 2 in the Algorithm 2 has complexity $O(mnd)$.*

Theorem 3. *Jacobian/pseudo-Jacobian have the computational complexity $O(d^3m^2n)$.*

Proof. The proof is given in Appendix A. □

Next, we show that the complexity is linear in m for the cost function and the gradient in the case of block-wise weights (Theorem 1, p. 3b). In the cost function evaluation, the most expensive step is the Cholesky factorization, which can be performed in linear in m number of operations in this case [21]. The computation of the gradient can be also simplified due to block-Toeplitz structure of Γ.

Proposition 6. *Let $y_\nu = \Gamma^{-1}s(R)$, and*

$$y_\nu = \mathit{vec}\, Y_\nu, \quad Y_\nu := \begin{bmatrix} y_\nu^{(1)} & \cdots & y_\nu^{(n)} \end{bmatrix}$$

be the partition of y_ν into a sequence of subvectors of length d. Under the conditions of Theorem 1, p. 3b, the gradient ($\nabla_{d\times m}$) can be simplified to

$$\nabla_{d\times m}(f) = 2Y_\nu \mathscr{S}^\top(p_d) - 2\Big(N_0 R \mathrm{V}_0 + \sum_{0<k\le \min(n,\mu)} \big(N_k R \mathrm{V}_k^\top + N_k^\top R \mathrm{V}_k)\big)\Big),$$

where $N_k := Y_\nu (J_n^\top)^k (Y_\nu)^\top$, where J_n is defined in (J_m).

Proof. The proof is given in Appendix A. □

Using Proposition 6, the following theorem can be proved.

Theorem 4. *Under the conditions of Theorem 1, p. 3b the complexity of cost function and gradient evaluation is equal to $O(d^3mn)$.*

Proof. The proof is given in Appendix A. □

Note 5. *Consider mosaic Hankel matrices ($\mathscr{H}_{\mathbf{m},\mathbf{n}}$) and block-wise weights, which are constant along the block rows of the mosaic Hankel matrix,* i.e.

$$w = \mathrm{col}\left(w_1 \mathbf{1}_{n_p^{(1,1)}}, \ldots, w_q \mathbf{1}_{n_p^{(q,1)}}, \ldots, w_1 \mathbf{1}_{n_p^{(1,N)}}, \ldots, w_q \mathbf{1}_{n_p^{(q,N)}}.\right)$$

Let the $\Gamma^{(l)}$ be the Γ matrix for the structure $\mathscr{H}_{\mathbf{m},[n_l]}$, as in Lemma 1. Then all $\Gamma^{(l)}$ are submatrices of $\Gamma^{(l_{max})}$, where $n_{l_max} \ge n_l$ for all l. Therefore, only Cholesky factorization of $\Gamma^{(l_{max})}$ needs to be computed.

5.4. Numerical examples

First, we consider a family of 2×2 mosaic Hankel matrices

$$\mathscr{H}_{[m_1\ m_2],[n_1\ n_2]}, m_1 = 20, m_2 = 22, \begin{bmatrix} n_1 & n_2 \end{bmatrix} = \begin{bmatrix} 250, 255 \end{bmatrix} + 250k, k = \{0, \ldots, 10\}. \quad \text{(EX1)}$$

In the case of 2-norm ($w = \mathbf{1}$) we compute the cost function and its derivatives using the SLRA package [4]. Hereafter, we use the term "element-wise variant" for the algorithms that treat weights as different values (implemented in the `WLayeredHankelStructure` C++ class). We use the term "block-wise variant" for the algorithms that utilize Theorem 4 (block-Toeplitz structure of Γ), and is implemented in the `LayeredHankelStructure` C++ class. In these examples,we consider only rank reduction by 1 ($r = m - 1$).

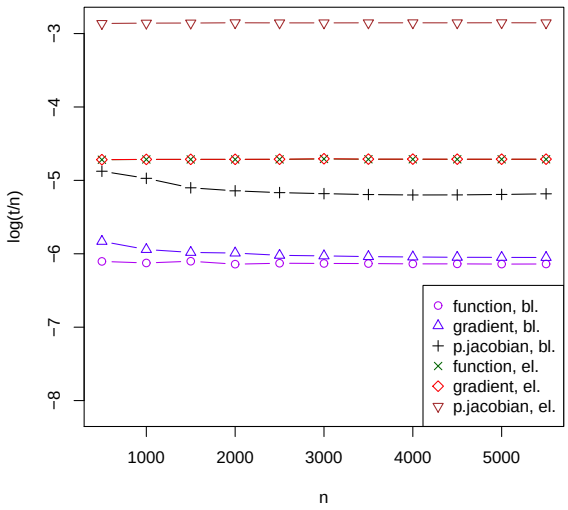

Figure 1: Logarithm of computation times divided by n for cost function, gradient and pseudo-Jacobian, depending on n. "el" denotes element-wise variant and "bl" denotes block-wise variant.

Fig. 1 shows the time needed for computation of the cost function, gradient and pseudo-Jacobian for (EX1), with $k = \{0, \ldots, 10\}$. The computation time is divided by n and plotted in logarithmic scale. In all cases the computation time is bounded by a linear function in n. This empirical observation agrees with Theorem 2 and Theorem 4.

Next, we show the computation time for variying m, on examples of scalar Hankel matrix. In Fig. 2 and Fig. 3 the computational time is plotted for element-wise and block-wise variants, as explained above. From Fig. 3 one can see that the computation time for the cost function and gradient in the element-wise variant the computation time grows quadratically in m, which is in agreement with Theorem 2.

For the block-wise variant, Fig. 2 shows the computational time for the cost function and the gradient is growing faster than linearly in m (in contrast to Theorem 4). The computation of the pseudo-Jacobian is also growing faster than quadratically in m (in contrast to Theorem 3). This effect can be expected because in the current version of the software products by $V_{k,l}$ matrices are implemented as products by precomputed matrices, and not as elementwise products.

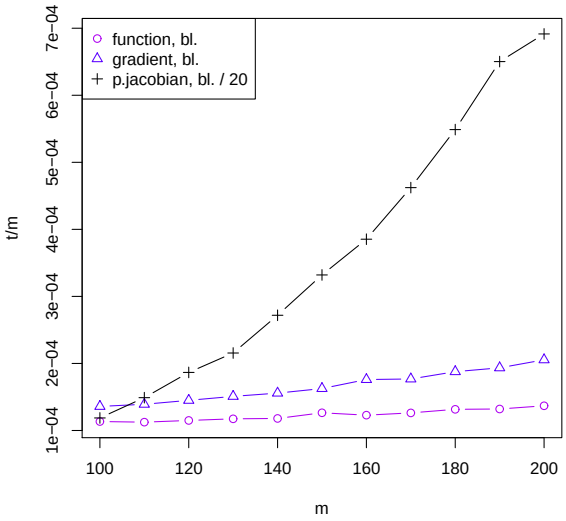

Figure 2: Computation time divided by m for the cost function, gradient and pseudo-Jacobian, for structure $\mathscr{H}_{m,2000}$ and block-wise variant. The computation time for the pseudo-Jacobian is divided by 20.

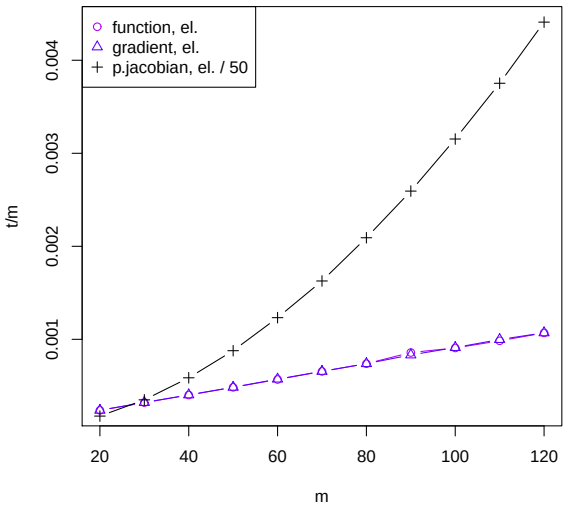

Figure 3: Computation time divided by m for the cost function, gradient and pseudo-Jacobian, for structure $\mathscr{H}_{m,2000}$ and element-wise variant. The computation time for the pseudo-Jacobian is divided by 50.

6. Conclusions

In this paper we considered the structured low-rank approximation problem for general affine structure, weighted 2-norm and fixed values constraints. We used the variable projection principle, which has many advantages when applied to (SLRA). The Γ matrix in $(\nu^\top \Gamma^{-1} \nu)$ is structured and its structure is determined by the original matrix structure. For the mosaic Hankel

structure ($\mathscr{H}_{\mathbf{m},\mathbf{n}}$), the Γ matrix is block-banded/block-Toeplitz, depending on the structure of the weight matrix. This allows us to evaluate the cost function $f(R)$ and its derivatives can be evaluated in $O(m^2n)$ flops ($O(mn)$ for the cost function and the gradient if Γ is block-Toeplitz). The approach can be applied for other matrix structures $\mathscr{S}$, where the structure of Γ (*e.g.* sparseness) can be exploited for efficient computations.

Whenever possible, we considered the most general cases (structures, weights) and developed the algorithms for the general affine structure. We showed how the cost functions for the blocked structures (layered, striped) can be expressed through the cost functions of the blocks. This allowed us to reduce the mosaic Hankel case to the scalar Hankel case and helped to simplify the derivations of the algorithms and their complexities (compared to [11, 17]). We also considered the fixed values constraints as a part of the weighted norm, which also simplified the treatment of this case.

Acknowledgements

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 258581 "Structured low-rank approximation: Theory, algorithms, and applications".

Appendix A. Proofs

Proof of Proposition 1. All admissible $\widehat{p}$ for (SLRA) can be parametrized as ($\Delta p \to \widehat{p}$). Indeed, if W^{-1} is positive definite, then $L_{\mathrm{W}^{-1}}^{\top}$ is nonsingular and ($\Delta p \to \widehat{p}$) is an invertible affine transformation $\mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$. If W is given by ($w \to \mathrm{W}$), then $L_{\mathrm{W}^{-1}} = \operatorname{diag}(\sqrt{\gamma_1}, \ldots, \sqrt{\gamma_{n_p}})$ and the parametrization ($\Delta p \to \widehat{p}$) keeps $\widehat{p}_k = (p_{\mathrm{d}})_k$ for fixed values $k : w_k = +\infty$ and runs over all possible $\widehat{p}_k$ for other k.

After substitution of ($\Delta p \to \widehat{p}$) into ($\mathscr{S}$) and ($\| \cdot \|_{\mathrm{W}}^2$) one obtains

$$\mathscr{S}(\widehat{p}) = \mathscr{S}_{\Delta}(\Delta p).$$

For positive definite W we have

$$\|\Delta p\|_2^2 = \|\widehat{p} - p_{\mathrm{d}}\|_{\mathrm{W}}^2,$$

and therefore (SLRA) with weighted norm $\| \cdot \|_{\mathrm{W}}$ is equivalent to (SLRA_{Δ}). If W is given by ($w \to \mathrm{W}$), then

$$\|\Delta p\|_2^2 = \|\widehat{p} - p_{\mathrm{d}}\|_{\mathrm{W}}^2 + \sum_{\{k:\ w_k = +\infty\}} (\Delta p_k)^2, \tag{A.1}$$

and (SLRA) is equivalent to

$$\underset{\Delta p \in \mathbb{R}^{n_p}}{\text{minimize}} \ \ (A.1) \ \ \text{subject to} \ \ \operatorname{rank} \mathscr{S}_{\Delta}(\Delta p) \leq r, \tag{A.2}$$

It is straightforward to see that $\mathscr{S}_{\Delta}(\Delta p)$ does not depend on the entries corresponding to fixed values. By checking the first-order optimality conditions one can see that all local minima of (A.2) satisfy $\Delta p_k = 0$ for $w_k = +\infty$. The local minima of (A.2) therefore coincide with the local minima of (SLRA_{Δ}). □

Proof of Proposition 3. First, note that if the differential of f is represented as

$$df(R,H) = \text{tr}(AH^\top),$$

then $\nabla_{d\times m}(f) = A$.

From $(y_\nu^\top \nu)$ the differential is given by

$$df(R,H) = 2y_\nu^\top d\nu(R,H) - y_\nu^\top d\Gamma(R,H)y_\nu. \qquad (df(R,H))$$

The first term in $(df(R,H))$ is equal to

$$2y_\nu^\top d\nu(R,H) = \text{tr}\left(2Y_\nu(H\mathscr{S}(p_\text{d}))^\top\right) = \text{tr}\left(2Y_\nu\mathscr{S}^\top(p_\text{d})H^\top\right).$$

The second term in $(df(R,H))$ is equal to

$$\begin{aligned} y_\nu^\top d\Gamma(R,H)y_\nu &= \sum_{i,j=1}^n (y_\nu^{(i)})^\top d\Gamma_{i,j}(R,H)y_\nu^{(j)}, \\ &= \sum_{i,j=1}^n (y_\nu^{(i)})^\top (H\text{V}_{i,j}R^\top + R\text{V}_{i,j}H^\top)y_\nu^{(j)} \\ &= \sum_{i,j=1}^n \text{tr}\left(\left(y_\nu^{(i)}(y_\nu^{(j)})^\top R\text{V}_{i,j}^\top + y_\nu^{(j)}(y_\nu^{(i)})^\top R\text{V}_{i,j}\right)H^\top\right), \end{aligned}$$

which in combination with $V_{i,j} = V_{j,i}^\top$ yields $(\nabla_{d\times m})$. □

Proof of Proposition 4. The equation $(\partial\Delta p^*/R_{ij})$ can be obtained by differentiation of $(\Delta p^*(R))$. The first summand in (z_{ij}) can be expressed as

$$\frac{\partial\nu}{\partial R_{ij}} = \text{vec}\left(e_i e_j^\top \mathscr{S}(p_\text{d})\right) = \text{row}_j(\mathscr{S}(p_\text{d})) \otimes e_i.$$

The second summand is

$$\frac{\partial\Gamma}{\partial R_{ij}}y_\nu = (a_1,\ldots,a_n),$$

where

$$\begin{aligned} a_l &= \sum_{k=1}^n \frac{\partial\Gamma_{l,k}}{\partial R_{ij}}y_\nu^{(k)} = \sum_{k=1}^n (e_i e_j^\top \text{V}_{l,k}R^\top + R\text{V}_{l,k}e_j e_i^\top)y_\nu^{(k)} \\ &= \sum_{k=1}^n (e_j^\top \text{V}_{l,k}R^\top y_\nu^{(k)})e_i + (y_\nu^{(k)})_i R\text{V}_{l,k}e_j. \end{aligned}$$

□

Proof of Theorem 2. First, we provide the complexities for each individual step of Algorithm 7.

1. Computation of $(\nu(R))$ amounts to multiplication of a $d \times m$ matrix R by $m \times n$ matrix $\mathscr{S}(p)$, which has computational complexity $O(dmn)$.
2. We need to compute μn matrices $\Gamma_{i,j}$. By Theorem 1 each V_{ij} has $\leq m$ nonzero elements and $\Gamma_{i,j}$ can be computed with $2d^2m$ multiplications, therefore the total computational complexity is $2d^2m\mu n$.

3. By Lemma 5 the complexity of this step is $O(d^3\mu^2 n)$.
4. By Lemma 5 the complexity of this step is $O(d^2\mu n)$.
5. The number of multiplication needed for computing a norm of a vector is nd.

Second, we provide the complexities for the steps of Algorithm 4.

1. By the above derivations the complexity of this step $O(d^3\mu^2 n)$.
2. The multiplication of a $d \times n$ matrix by $n \times m$ matrix has complexity nmd.
3. We need to compute $2\mu n$ products $y_\nu^{(j)}(y_\nu^{(i)})^\top R\mathrm{V}_{i,j}$. By Theorem 1, the matrix $\mathrm{V}_{i,j}$ has only one nonzero diagonal and the product $u = (y_\nu^{(i)})^\top R\mathrm{V}_{i,j}$ can be computed in dm steps. The product $y_\nu^{(j)}u$ also takes dm steps. Therefore the total computational complexity of this step is $O(d\mu mn)$. □

Proof of Theorem 3. For each $(z_j^{(1)})$, due to block-bandedness of V, we need to compute $2\mu n$ products of the form

$$e_j^\top \mathrm{V}_{l,k} R^\top y_\nu^{(k)} = b_{j,k,l}^\top y_\nu^{(k)}.$$

By Theorem 1, $e_j^\top \mathrm{V}_{l,k} R^\top = b_{j,k,l}^\top$, corresponds to elementwise multiplication of a column of R by elements from γ_t, which takes d multiplications. Another d multiplications are needed to compute the inner product of $b_{j,k,l}$ and $y_\nu^{(k)}$. The computation of $(z_j^{(1)})$ is repeated m times, which lead to $O(d\mu mn)$ number of multiplications.

For each $(z_{ij}^{(2)})$, we need to compute $2\mu n$ products of the form $(y_\nu^{(k)})_i R\mathrm{V}_{1,k}e_j$, where each product, as in the previous step, has also complexity $2d$. The computation of $(z_{ij}^{(2)})$ is repeated md times, which leads to $O(d^2\mu mn)$ complexity.

For pseudo-Jacobian, we need to solve md times the banded system with the Cholesky factor $L_R^\top$ and z_{ij}, which has complexity $O(d^3\mu mn)$. For Jacobian, we also need to multiply each $\Gamma^{-1}z_{ij}$ from the left by $G^\top R$, which has complexity nmd by Note 4. Therefore, this step has additional complexity $O(d^2m^2n)$. □

Proof of Proposition 6. The second term in $(\nabla_{d\times m})$ can be represented as

$$\sum_{k=-M}^{M} \sum_{j-i=k} y_\nu^{(j)}(y_\nu^{(i)})^\top R\mathrm{V}_{i,j}.$$

In this case, $\mathrm{V}_{i,j} = \mathrm{V}_{j-i}$. It is easy to see that for $j > i$

$$\sum_{j-i=k} y_\nu^{(j)}(y_\nu^{(i)})^\top = \sum_{i=1}^{n-k} y_\nu^{(i+k)}(y_\nu^{(i)})^\top = Y_\nu(J_n^\top)^k(Y_\nu)^\top = N_k.$$

□

Proof of Theorem 4. Since only μ matrices Γ_k need to be computed, step 2 of Algorithm 1 can be performed in $2d^2m\mu$ flops. For Cholesky factorization (step 3) the algorithms exploiting the Toeplitz structure [21] can be used. Their complexity is $O(d^3\mu n)$.

Here we use Proposition 6. Each N_k can be computed in d^2n operations the computation of the product $N_kR\mathrm{V}_k^\top$ has complexity $O(d^2m)$ due to the sizes of the involved matrices. Therefore the total computational complexity of Step 3 of Algorithm 4 if $O(d^2\mu n)$. □

References

[1] I. Markovsky, Structured low-rank approximation and its applications, Automatica 44 (2008) 891–909.

[2] I. Markovsky, Low Rank Approximation: Algorithms, Implementation, Applications, Communications and Control Engineering, Springer, 2012.

[3] G. Heinig, Generalized inverses of Hankel and Toeplitz mosaic matrices, Linear Algebra Appl. 216 (1995) 43–59.

[4] I. Markovsky, K. Usevich, Software for weighted structured low-rank approximation, Technical Report 339974, ECS, Univ. of Southampton, http://eprints.soton.ac.uk/339974/, 2012.

[5] S. Rump, Structured perturbations part I: Normwise distances., SIAM J. Matrix Anal. Appl. 25 (2003) 1–30.

[6] B. De Moor, Structured total least squares and L_2 approximation problems, Linear Algebra Appl. 188–189 (1993) 163–207.

[7] J. Rosen, H. Park, J. Glick, Total least norm formulation and solution of structured problems, SIAM J. Matrix Anal. Appl. 17 (1996) 110–126.

[8] N. Mastronardi, P. Lemmerling, S. Van Huffel, Fast structured total least squares algorithm for solving the basic deconvolution problem, SIAM J. Matrix Anal. Appl. 22 (2000) 533–553.

[9] G. H. Golub, V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, SIAM Journal on Numerical Analysis 10 (1973) pp. 413–432.

[10] J. Manton, R. Mahony, Y. Hua, The geometry of weighted low-rank approximations, IEEE Trans. Signal Proc. 51 (2003) 500–514.

[11] A. Kukush, I. Markovsky, S. Van Huffel, Consistency of the structured total least squares estimator in a multivariate errors-in-variables model, J. Statist. Plann. Inference 133 (2005) 315–358.

[12] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[13] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, NJ, 2008.

[14] K. Usevich, I. Markovsky, Structured low-rank approximation as a rational function minimization, in: Proc. of the 16th IFAC Symposium on System Identification, Brussels, 2012.

[15] G. Golub, C. Van Loan, Matrix Computations, Johns Hopkins University Press, third edition, 1996.

[16] R. Borsdorf, Structured Matrix Nearness Problems: Theory and Algorithms, Ph.D. thesis, The University of Manchester, 2012.

[17] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, Exact and Approximate Modeling of Linear Systems: A Behavioral Approach, SIAM, 2006.

[18] I. Markovsky, S. Van Huffel, High-performance numerical algorithms and software for structured total least squares, J. Comput. Appl. Math. 180 (2005) 311–331.

[19] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, SIAM J. Appl. Math. 11 (1963) 431–441.

[20] P. Guillaume, R. Pintelon, A Gauss–Newton-like optimization algorithm for "weighted" nonlinear least-squares problems 44 (1996) 2222–2228.

[21] S. Van Huffel, V. Sima, A. Varga, S. Hammarling, F. Delebecque, High-performance numerical software for control, IEEE Control Systems Magazine 24 (2004) 60–76.