NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2004; **11**:591–608 (DOI: 10.1002/nla.361)

On the computation of the multivariate structured total least squares estimator

Ivan Markovsky[1,*,†], Sabine Van Huffel[1] and Alexander Kukush[2]

[1]*K.U. Leuven, ESAT-SCD, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium*
[2]*National Taras Shevchenko University, Vladimirskaya st. 64, 01601, Kiev, Ukraine*

SUMMARY

A multivariate structured total least squares problem is considered, in which the extended data matrix is partitioned into blocks and each of the blocks is Toeplitz/Hankel structured, unstructured, or noise free. Two types of numerical solution methods for this problem are proposed: (i) standard local optimization methods in combination with efficient evaluation of the cost function and its first derivative, and (ii) an iterative procedure proposed originally for the element-wise weighted total least squares problem. The computational efficiency of the proposed methods is compared with this of alternative methods. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: parameter estimation; total least squares; structured total least squares; system identification

1. INTRODUCTION

1.1. Identification of a moving average time series model as an STLS problem

We introduce the structured total least squares (STLS) problem by an example. Consider the moving average time series model

$$\bar{a}(i)x(1) + \bar{a}(i-1)x(2) = \bar{b}(i) \quad \text{for } i = 1, \ldots, m \tag{1}$$

*Correspondence to: I. Markovsky, K.U.Leuven, ESAT-SCD, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium.
†E-mail: ivan.markovsky@esat.kuleuven.ac.be

Contract/grant sponsor: Research Council KUL; contract/grant numbers: GOA-Mefisto 666, IDO/99/003, IDO/02/009
Contract/grant sponsor: Flemish Government FWO; contract/grant numbers: G.0200.00, G.0078.01, G.0407.02, G.0269.02, G.0270.02
Contract/grant sponsor: AWI
Contract/grant sponsor: IWT
Contract/grant sponsor: Belgian Federal Government; contract/grant number: IUAP IV-02; IUAP V-22
Contract/grant sponsor: EU

Received 10 January 2003
Revised 14 July 2003
Accepted 16 September 2003

Published online 5 May 2004

The vector $x := [x(1)\ x(2)]^\top$ of the weights is the *parameter vector of the model*, $\{\bar{a}(i)\}_{i=1}^m$ is the input time series, $\{\bar{b}(i)\}_{i=1}^m$ is the output time series, and $\bar{a}(0)$ is the initial condition. With

$$\bar{A} := \begin{bmatrix} \bar{a}(1) & \bar{a}(0) \\ \bar{a}(2) & \bar{a}(1) \\ \vdots & \vdots \\ \bar{a}(m) & \bar{a}(m-1) \end{bmatrix} \quad \text{and} \quad \bar{b} := \begin{bmatrix} \bar{b}(1) \\ \bar{b}(2) \\ \vdots \\ \bar{b}(m) \end{bmatrix}$$

the time-series model (1) is written as a linear system of equations $\bar{A}x = \bar{b}$, with structured *data matrix* $\bar{C} := [\bar{A}\ \ \bar{b}]$ ($\bar{A}$ Toeplitz and $\bar{b}$ unstructured). The *structure parameter vector* is the vector

$$\bar{p} := [\,\bar{a}(0)\ \ \bar{a}(1)\ \ \cdots\ \ \bar{a}(m)\ \ \bar{b}(1)\ \ \cdots\ \ \bar{b}(m)\,]^\top$$

i.e. there exists a mapping $\mathscr{S} : \mathbb{R}^{2m+1} \to \mathbb{R}^{m\times 3}$ (linear, in the example), such that $\bar{C} = \mathscr{S}(\bar{p})$.

Suppose that we measure the input, the output, and the initial condition with additive noise: $p = \bar{p} + \tilde{p}$. Here $\bar{p}$ is the true value and $\tilde{p}$ is the measurement noise that is assumed to be a realization of a zero mean random vector with known covariance matrix $\sigma^2 I$. The noise level σ^2 is *not* given but is estimated on the way of solving the problem.

We consider the following system identification problem: given the measurements p, find an estimate of the true value of the model parameter vector $\bar{x}$ (i.e. $\bar{A}\bar{x} = \bar{b}$). With $[A\ \ b] := \mathscr{S}(p)$, in general, we have an incompatible system of equations $Ax \approx b$. Thus, the considered identification problem is equivalent to the problem of solving the over-determined system of equations $Ax \approx b$ with structured data matrix $C := [A\ \ b]$.

One can take as an estimate the solution of the least squares (LS) problem

$$\min_{x,\,\Delta b} \|\Delta b\|_2^2 \quad \text{s.t. } Ax = b - \Delta b$$

It is well known, however, that this approach leads to a biased estimate, see Reference [1]. In Reference [2], a *bias corrected least squares* estimator is proposed that leads to a consistent estimator. Another approach that yields a consistent estimator, see Reference [3], is the *total least squares* (TLS) method [4, 5],

$$\min_{x,\,\Delta A,\,\Delta b} \|[\,\Delta A\ \ \Delta b\,]\|_F^2 \quad \text{s.t. } (A - \Delta A)x = b - \Delta b \tag{2}$$

Both the bias corrected LS and the TLS methods, however, ignore the structure in the data matrix C, i.e. the corrected data matrices $[A\ \ b - \Delta b]$, in the LS case, and $[A - \Delta A\ \ b - \Delta b]$, in the TLS case, do not necessarily have the required structure. Taking into account the structure leads to statistically more efficient estimates and also to computationally faster algorithms.

A TLS-like problem, that performs minimization (2) over the class of matrices with the required structure is

$$\min_{x,\,\Delta p} \|\Delta p\|_2^2 \quad \text{s.t. } \mathscr{S}(p - \Delta p) \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

If the noise vector $\tilde{p}$ is normally distributed, then this *structured total least squares* problem, yields the maximum likelihood estimate of $\bar{x}$. Statistical consistency of the STLS estimate

is proven in References [6, 7]. The fact that the STLS estimator is consistent and efficient under mild assumptions, satisfied in many applications, and the possibility to design efficient algorithms by exploiting the structure on the level of the computations makes the STLS problem attractive.

1.2. The multivariate STLS problem

Other applications, e.g. finite impulse response (FIR) model identification, autoregressive moving average (ARMA) model identification, and approximation of a Hankel matrix by a lower rank Hankel matrix (Hankel low rank approximation), can be formulated and solved as STLS problems. For more examples, see References [8, 9]. Different applications, however, result in different structures of the extended data matrix C. Also some applications, e.g. the Hankel low rank approximation problem, require a *multivariate* linear model $AX \approx B$. We define a multivariate STLS problem as one that has a flexible structure specification, covering a wide spectrum of applications.

Consider the multivariate linear errors-in-variables (EIV) model

$$AX \approx B, \quad A = \bar{A} + \tilde{A}, \quad B = \bar{B} + \tilde{B}, \quad \bar{A}\bar{X} = \bar{B} \tag{3}$$

where $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times d}$ are *observations*, and $X \in \mathbb{R}^{n \times d}$ is a *parameter* of interest. We denote the corresponding (non-stochastic) *true values* by bar and *measurement errors* by tilde. Typically the dimensions of the estimated parameter are small compared with the number of measurements, i.e. $nd \ll m$.

We assume that there is an *a priori* known affine function $\mathscr{S} : \mathbb{R}^{n_p} \to \mathbb{R}^{m \times (n+d)}$,

$$\mathscr{S}(p) = S_0 + \sum_{l=1}^{n_p} S_l p_l \quad \text{for all} \quad p \in \mathbb{R}^{n_p}$$

with $n_p \geqslant md$, such that

$$C := [A \;\; B] = \mathscr{S}(p)$$

for some structure parameter vector $p \in \mathbb{R}^{n_p}$. The true data matrix $\bar{C} := [\bar{A} \;\; \bar{B}]$ also satisfies the affine function $\mathscr{S}$, i.e. $\bar{C} = \mathscr{S}(\bar{p})$, for some unknown parameter vector $\bar{p} \in \mathbb{R}^{n_p}$. The vector p is a noisy measurement of $\bar{p}$, i.e. $p = \bar{p} + \tilde{p}$, where $\tilde{p}$ is a zero mean random vector with a covariance matrix $\sigma^2 I$. The function $\mathscr{S}$ defines the structure in the problem.

The STLS problem for the structured EIV model (3) is defined as

$$\min_{X, \Delta p} \|\Delta p\|_2^2 \quad \text{s.t.} \quad \mathscr{S}(p - \Delta p) \begin{bmatrix} X \\ -I_d \end{bmatrix} = 0 \tag{4}$$

The STLS estimate $\hat{X}$ of $\bar{X}$ is defined as a global minimum point of the optimization problem (4).

Apart from the assumption that $\mathscr{S}$ is affine, in the derivation of the algorithms, we require that the structure in the problem is such that the data matrix can be partitioned into blocks $C = [C_1 \; \cdots \; C_q]$ and each of the blocks C_i is either Hankel structured, Toeplitz structured, unstructured, or noise free. We confine to the type of structure for which the STLS estimator is proven to be consistent [7].

1.3. Computation of the STLS estimate

The main difficulty in the numerical solution of the STLS problem is its non-convexity, i.e. the presence of local minima. We derive algorithms that perform *local* optimization starting from a given initial approximation. Thus there is no guarantee that a global minimum point is found.

With sample size m much larger than the number nd of parameters in the estimate X, however, (4) has a unique solution; see Reference [7]. Due to the consistency results, this implies also more accurate estimation of the true parameter $\bar{X}$. Correspondingly, our main objective is to derive algorithms that can deal with large sample sizes.

Standard algorithms [10] for constrained optimization can be applied to the STLS problem. The algorithms are usually compared with respect to the local convergence rate. In our case, however, most important is the computational efficiency of the algorithm, measured by the increase of the required amount of computations (or computation time) as a function of m. In this respect, special purpose algorithms can significantly outperform the straightforward application of the standard optimization methods.

One approach, see References [11–13], to derive special purpose algorithms is to apply an iterative procedure, in which the constraint of (4) is linearized around the current approximation point and an equality constrained least squares problem is solved. Due to the structure of the involved matrices, significant speedup can be achieved. In Reference [14] the equality constrained least squares problem is efficiently solved via the Generalized Schur Algorithm. The resulting algorithms for solving the STLS problem have computational cost linear in m. Unfortunately the developed algorithms are bound to particular structures and univariate STLS problems. For example, in Reference [15], A must be Toeplitz (or Hankel) and b unstructured, while in Reference [16], $[A\ \ b]$ must be Toeplitz (or Hankel). New algorithms are needed for other structures and multivariate STLS problems, and their development is non-trivial.

The contribution of the present paper is a derivation of efficient numerical methods for the solution of a multivariate STLS problem with arbitrary combination of Toeplitz/Hankel structured, unstructured, and noise-free blocks in the data matrix. We note that currently the method of Reference [12] is the only one in the literature that can deal with multivariate problems. Although the problem class being considered is a very general one, restricting to particular cases, the asymptotic computational efficiency of the derived algorithms as $m \to \infty$ is comparable to or better than that of the best currently available algorithms.

Unlike the methods mentioned above, which solve the STLS problem in its original formulation (4), the proposed methods solve an equivalent optimization problem, derived by analytically minimizing (4) over Δp, for a fixed X. A similar approach, using a different parameterization of the structure, is taken in the derivation of the so-called *constrained total least squares* (CTLS) problem [17]. However, Reference [17] is restricted to univariate problems and does not use the best optimization techniques in terms of computational efficiency and robustness (very good initial estimates are needed). Another STLS problem formulation is based on the Riemannian singular value decomposition [18], where the derived equivalent problem is interpreted as a non-linear singular value decomposition problem.

The equivalent problem is an unconstrained, non-convex, and non-differentiable optimization problem. Since the number of decision variables nd is fixed and much smaller than m, the main computational effort for applying standard optimization techniques is in the cost function evaluation. We describe how the cost function and its first derivative can be

evaluated efficiently under our assumptions. As a result, the computational cost of the standard optimization solvers is linear in m.

Alternatively to the use of standard local optimization methods, we describe an iterative procedure for the solution of the equivalent problem. The proposed iterative method is essentially different from the standard optimization methods. It is similar to the one used for the solution of the *element-wise weighted total least squares* (EW-TLS) problem [19, 20]. We compare numerically the efficiency of the proposed methods and the methods of References [12, 15, 16].

Standard notation used in the paper is: $\mathbb{R}$ for the set of the real numbers, $\mathbb{N}$ for the set of the natural numbers, $\mathbf{E}Z$ for the expectation of the random vector or matrix Z, $N(0, V)$ for the zero mean normal distribution with covariance matrix V, $\|x\|$ for the Euclidean norm of the vector x, and $\|A\|_F$ for the Frobenius norm of the matrix A. For any matrix $A \in \mathbb{R}^{m\times n}$, we denote by a_i its ith row transposed, i.e. $A^\top =: [a_1 \cdots a_m]$, and by a the vector $[a_1^\top \cdots a_m^\top]^\top =: \mathrm{vec}(A^\top)$. Accordingly vec^{-1} is defined by $A = \mathrm{vec}^{-1}(a)$.

The paper is structured as follows. In Section 2, we derive the equivalent optimization problem. In Section 3, we define the considered class of structures and identify useful properties that hold in this case. In Section 4, we introduce the proposed algorithms for solving the equivalent optimization problem and in Section 5, we describe their implementation. In Section 6, we compare numerically the efficiency of the proposed algorithms and the algorithms of References [12, 15, 16]. Section 7 gives conclusions and directions for future work.

2. DERIVATION OF AN EQUIVALENT OPTIMIZATION PROBLEM

The first step towards the solution of the STLS problem is the elimination of the correction Δp by analytically minimizing it. For a fixed X, consider the solution of (4) as a function of X, i.e. consider the function

$$f_0(X) := \min_{\Delta p} \|\Delta p\|^2 \quad \text{s.t. } \mathscr{S}(p - \Delta p)X_{\text{ext}} = 0$$

where X_{ext} is the *extended parameter* $X_{\text{ext}} := \begin{bmatrix} X \\ -I \end{bmatrix}$. The STLS problem (4) is equivalent to the unconstrained minimization of f_0,

$$\min_X f_0(X) \tag{5}$$

Next, we obtain the cost function f_0. Denote the residual $AX - B$ by R

$$R(X) := AX - B = CX_{\text{ext}}$$

and let r be the vectorized $R^\top$, i.e.

$$r(X) := \mathrm{vec}(R^\top(X)) = \mathrm{vec}([\, r_1(X) \ \cdots \ r_m(X)\,]) = \begin{bmatrix} r_1(X) \\ \vdots \\ r_m(X) \end{bmatrix} \in \mathbb{R}^{md\times 1}$$

We use similar notation for the random part $\tilde{R} = R - \mathbf{E}R = \tilde{A}X - \tilde{B} = \tilde{C}X_{\text{ext}}$ of the residual.

Due to the assumption that $\mathscr{S}$ is affine, the constraint of (4) is linear in Δp:

$$\mathscr{S}(p-\Delta p)X_{\text{ext}}=0 \Leftrightarrow CX_{\text{ext}}=\sum_{l=1}^{n_p} S_l X_{\text{ext}}\Delta p_l \Leftrightarrow R^\top(X)=\sum_{l=1}^{n_p}(S_l X_{\text{ext}})^\top \Delta p_l$$

$$\Leftrightarrow \operatorname{vec}(R^\top(X))=\sum_{l=1}^{n_p}\operatorname{vec}((S_l X_{\text{ext}})^\top)\Delta p_l \Leftrightarrow r(X)=G(X)\Delta p$$

where

$$G(X):=[\ \operatorname{vec}((S_1 X_{\text{ext}})^\top)\ \cdots\ \operatorname{vec}((S_{n_p} X_{\text{ext}})^\top)\ ]\in\mathbb{R}^{md\times n_p}$$

Thus we have to solve the following problem:

$$\min_{\Delta p}\Delta p^\top\Delta p \quad \text{s.t. } G(X)\Delta p=r(X) \tag{6}$$

Note that for the feasibility of (6), the constraint $G(X)\Delta p=r(X)$ has to be solvable. Assuming that $G(X)$ is full rank, at least md parameters are needed, i.e. $n_p\geqslant md$. Under this condition, (6) is a least-norm problem and its solution is given by

$$\Delta p_{\min}(X)=G^\top(X)(G(X)G^\top(X))^{-1}r(X)$$

so that

$$f_0(X)=\Delta p_{\min}^\top(X)\Delta p_{\min}(X)=r^\top(X)(\underbrace{G(X)G^\top(X)}_{\Gamma(X)})^{-1}r(X)=:r^\top(X)\Gamma^{-1}(X)r(X) \tag{7}$$

Note 1 (*Relation to the EW-TLS problem* [*20*])
We can write f_0 as

$$f_0(X)=\sum_{i,j=1}^{m} r_i^\top(X)M_{ij}(X)r_j(X)$$

where $M_{ij}(X)\in\mathbb{R}^{d\times d}$ is the (i,j)th block of the matrix $M(X):=\Gamma^{-1}(X)$. The cost function of the EW-TLS problem [20] is of the same type but $M_{ij}(X)=0$ for $i\neq j$; equivalently the matrix $\Gamma(X)$ is block diagonal.

Note 2 (*Relation to the CTLS problem* [*17*])
The CTLS problem considers the same optimization problem as defined in (5) but restricted to univariate problems ($d=1$) and using a different weight matrix Γ due to a different parameterization of the structure.

Next, we show that the weight matrix Γ is up to the scale factor σ^2 equal to the covariance matrix $V_{\tilde{r}}$ of the centred residual $\tilde{r}$. We have $\tilde{r}(X)=\operatorname{vec}(\tilde{R}(X))=G(X)\tilde{p}$, so that

$$V_{\tilde{r}}(X)=\mathbf{E}\tilde{r}(X)\tilde{r}^\top(X)=G(X)\mathbf{E}\,\tilde{p}\,\tilde{p}^\top G^\top(X)=\sigma^2\Gamma(X) \tag{8}$$

3. PROPERTIES OF THE WEIGHT MATRIX Γ

For the derivation of the cost function f_0 of the equivalent minimization problem (5), only the assumption that $\mathscr{S}$ is an affine function was used. Now we give the following additional assumptions:

(i) $S_l(i,j) = \begin{cases} 1 & \text{if } T(i,j) = l \\ 0 & \text{otherwise} \end{cases}$, where $T \in \{0,1,\ldots,n_p\}^{m\times(n+d)}$ is a known matrix;

(ii) $T = [T_1 \cdots T_q]$, where $T_k \in \{0,1,\ldots,n_p\}^{m\times n_k}$ has one of the following structures: `T`—Toeplitz, `H`—Hankel, `U`—unstructured, or `F`—noise free ($T_k = 0$);

Assumption (i) allows at most one element of p to enter the (i,j)th entry of the data matrix $\mathscr{S}(p)$:

$$[\mathscr{S}(p)]_{ij} = \begin{cases} S_0(i,j) + p(T(i,j)) & \text{if } 1 \leqslant T(i,j) \leqslant n_p \\ S_0(i,j) & \text{if } T(i,j) = 0 \end{cases}$$

In the latter case, the entry $[\mathscr{S}(p)]_{ij}$ is not modified by any of the structure parameters p_i. (Clearly we do not modify the noise-free entries.) Assumption (ii) further restricts $\mathscr{S}(p)$ to be a block matrix of which the blocks are structured with one of the four predefined structures.

Under assumptions (i) and (ii), the specification of the structure describing function $\mathscr{S}$ is given by the matrix S_0 and an array

$$\mathscr{T} \in \{\{\mathtt{T},\mathtt{H},\mathtt{U},\mathtt{F}\} \times \mathbb{N}\}^q \tag{9}$$

that describes the structure of the blocks $\{T_k\}_{k=1}^q$; $\mathscr{T}_k$ specifies the block T_k by giving its type $\mathscr{T}_k(1)$ and the number of columns $n_k = \mathscr{T}_k(2)$. For example, $\mathscr{T}_1 = \{[\mathtt{T}\ \ 4]\}$ defines that $T = [T_1]$, with T_1 a Toeplitz matrix with 4 columns. Due to assumption (ii), the matrix T is completely described by its first $s+1$ rows, where

$$s := \max_{k\in\{1,\ldots,q\}} \{\mathscr{T}_k(2):\ \mathscr{T}_k(1) = \mathtt{T} \text{ or } \mathscr{T}_k(1) = \mathtt{H}\} - 1$$

The entries in the Toeplitz (Hankel) structured submatrices C_k are equal along the diagonals (antidiagonals). Thus the elements in the first row of C appear at most down the first $s+1$ rows. This property and the constant s are extensively used later on.

In terms of the measurement errors matrix $\tilde{C}$, our assumptions imply stationarity in a wide sense and s-dependence of the sequence $\{\tilde{c}_i\}_{i=1}^m$ of its row. A centred sequence $\{v_i\}$ of random vectors is called *stationary in a wide sense* if $\mathbf{E}v_i v_{i+k}^\top$, for all i and j, depends only on k and does not depend on i. A sequence of random vectors $\{v_i\}$ is called *s-dependent*, $s \geqslant 1$, if for each i, the two sequences $\{v_1,\ldots,v_i\}$ and $\{v_{i+s+1}, v_{i+s+2},\ldots\}$ are independent from each other. Assumptions (i) and (ii) are the basic assumptions from Reference [7] for consistency of the STLS estimator.

By definition, the weight matrix $\Gamma(X)$ is a positive semidefinite matrix. Under assumptions (i) and (ii), however, it has useful additional structure. To show this, define the covariance matrix $V_{\tilde{c}} := \mathbf{E}\tilde{c}\tilde{c}^\top$ of $\tilde{c} = \mathrm{vec}(\tilde{C}^\top)$ and let $V_{\tilde{c},ij} \in \mathbb{R}^{(n+d)\times(n+d)}$, $i,j = 1,\ldots,m$, be the (i,j)th block of $V_{\tilde{c}}$. We have $\tilde{r}_i(X) = X_{\mathrm{ext}}^\top \tilde{c}_i$, so that $\sigma^2\Gamma = V_{\tilde{r}}(X)$ consists of the blocks

$$\sigma^2\Gamma_{ij}(X) = \mathbf{E}\tilde{r}_i(X)\tilde{r}_j^\top(X) = X_{\mathrm{ext}}^\top \mathbf{E}\tilde{c}_i\tilde{c}_j^\top X_{\mathrm{ext}} = X_{\mathrm{ext}}^\top V_{\tilde{c},ij} X_{\mathrm{ext}} \in \mathbb{R}^{d\times d}$$

Due to the stationarity of $\{\tilde{c}_i\}$, $V_{\tilde{c},ij} = V_{\tilde{c},i-j}$ is a function of the difference $i-j$ only, and due to the s-dependence of $\{\tilde{c}_i\}$, $V_{\tilde{c},ij} = 0$ for $|i-j| \geqslant s+1$. Consequently, $\Gamma_{ij}(X) = \Gamma_{i-j}(X)$, and $\Gamma_{ij}(X) = 0$ for $|i-j| \geqslant s+1$. Thus $\Gamma(X)$ has the block banded Toeplitz structure,

$$\Gamma(X) = \begin{bmatrix} \Gamma_0 & \Gamma_{-1} & \cdots & \Gamma_{-s} & & 0 \\ \Gamma_1 & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \Gamma_{-s} \\ \Gamma_s & \ddots & \ddots & \ddots & \ddots & \vdots \\ & \ddots & \ddots & \ddots & \ddots & \Gamma_{-1} \\ 0 & & \Gamma_s & \cdots & \Gamma_1 & \Gamma_0 \end{bmatrix} \tag{10}$$

where $\Gamma_k(X) = \Gamma_{-k}^{\top}(X)$, for $k = 0, 1, \ldots, s$.

In order to save notation, we will occasionally drop the explicit dependence of r and Γ on X.

4. PROPOSED NUMERICAL ALGORITHMS

We consider numerical methods for the solution of the optimization problem (5). One approach is to use standard algorithms for local optimization. The choice of the optimization method is inspired by the need to use as much as possible the specific features of the problem. Due to the non-differentiability of the cost function, a natural candidate is the Nelder–Mead simplex algorithm [21]. If the initial approximation is far from the discontinuities, however, more efficient methods such as Quasi–Newton exist that can further exploit the derivative information. Even further improvement is achieved by taking into account the least-squares nature of the problem. The cost function can be written as

$$r^{\top}\Gamma^{-1}r = (\Gamma^{-1/2}r)^{\top}(\Gamma^{-1/2}r) \tag{11}$$

where $\Gamma^{-1/2}$ is the Cholesky factor of Γ^{-1}, and one can exploit special methods for non-linear least squares problems, e.g. the Gauss–Newton and the Levenberg–Marquardt method [22]. The main computational effort in solving the problem by local optimization methods is in the cost function and the derivative evaluation. Crucial for the efficiency is the special structure of Γ.

Another approach for the solution of (5) is an iterative procedure for solving the first order optimality condition $f_0'(X) = 0$. The method is first proposed in Reference [19] for the univariate EW-TLS problem, then developed for more general EW-TLS problems in Reference [20], and recently generalized for the STLS problem in Reference [7]. The derivative $f_0'(X)$ is, see Reference [23, p. 19],

$$f_0'(X) = 2\sum_{i,j=1}^{m} a_j r_i^{\top}(X) M_{ij}(X) - 2\sum_{i,j=1}^{m} [\, I \;\; 0 \,] \frac{V_{\tilde{c},ij}}{\sigma^2} \begin{bmatrix} X \\ -I \end{bmatrix} N_{ji}(X) \tag{12}$$

where

$$M(X) := \Gamma^{-1}(X), \quad N(X) := \Gamma^{-1}(X) r(X) r^{\top}(X) \Gamma^{-1}(X)$$

and $M_{ij} \in \mathbb{R}^{d\times d}$, $N_{ij} \in \mathbb{R}^{d\times d}$ are the corresponding (i,j)th blocks of M and N.

We approach a solution of the equation $f_0'(X)=0$ by organizing an iterative procedure. Let $\{X^{(l)}\}$, $l=0,1,\ldots$ be the sequence of approximations produced by the iterative procedure starting from a given initial approximation $X^{(0)}$. On the lth step, the following linear equation:

$$F(X^{(l+1)},X^{(l)}) := \sum_{i,j=1}^{m} a_j(a_i^\top X^{(l+1)} - b_i^\top)M_{ij}(X^{(l)}) - \sum_{i,j=1}^{m} [\,I \;\; 0\,]\frac{V_{\tilde{c},ij}}{\sigma^2}\begin{bmatrix} X^{(l+1)} \\ -I \end{bmatrix} N_{ji}(X^{(l)}) = 0$$

is solved for the approximation $X^{(l+1)}$ on the next step. The proposed iterative algorithm is

1. Find an initial approximation $X^{(0)}$, e.g. the TLS estimate, and let $k:=0$.
2. Repeat
   (2a) Solve the linear system $F(X^{(l+1)},X^{(l)})=0$ for $X^{(l+1)}$ and let $l:=l+1$. Until $\|X^{(l)}-X^{(l-1)}\|_F/\|X^{(l)}\|_F<\varepsilon$.
3. The computed STLS estimator is $\hat{X}:=X^{(l)}$.

In References [20, 24] conditions are established under which a similar iterative algorithm for the EW-TLS problem has local convergence. For a fixed sample size, the convergence of the algorithm for the EW-TLS problem is linear and as $m\to\infty$ the convergence rate tends to the quadratic.

5. IMPLEMENTATION OF THE ALGORITHMS

The input data for the algorithms is the data matrix $C:=\mathscr{S}(p)$ and the structure description $\mathscr{T}$. First, we show how the compressed structure information $\mathscr{T}$ is used in the computations. Then, we describe the evaluation of the cost function, the derivative f_0', and the implementation of the proposed iterative algorithm.

For the computation of the cost function, we need the matrices $\{W_{\tilde{c},k}:=V_{\tilde{c},k}/\sigma^2\}_{k=0}^{s}$, which in turn can be constructed from the structure describing matrix $T(1{:}s+1,:)$. The first $s+1$ rows of T are constructed by subsequently reading the rows of $\mathscr{T}$ and filling in the corresponding blocks of $T(1{:}s+1,:)$ with consecutive natural numbers according to the block-type specification. For example, with $\mathscr{T}=\{[\mathtt{T}\ 3],\ [\mathtt{H}\ 2],\ [\mathtt{U}\ 2],\ [\mathtt{F}\ 1]\}$, $s=2$ and we have

$$T(1:s+1,:)=\left[\begin{array}{ccc|cc|cc|c} 3 & 2 & 1 & 6 & 7 & 10 & 13 & 0 \\ 4 & 3 & 2 & 7 & 8 & 11 & 14 & 0 \\ 5 & 4 & 3 & 8 & 9 & 12 & 15 & 0 \end{array}\right]$$

The structured noise matrix $\tilde{C}$ is related to the parameter noise vector $\tilde{p}$ as follows:

$$\tilde{C}=\sum_{l=1}^{n_p} S_l\tilde{p}_l = [\,\tilde{p}(T(i,j))]_{i=1,\ldots,m}^{j=1,\ldots,n+d}$$

so that

$$W_{\tilde{c},k}(i,j)=\mathbf{E}\tilde{c}_{1i}\tilde{c}_{kj}/\sigma^2=\mathbf{E}\,\tilde{p}(T(1,i))\,\tilde{p}(T(k,j))/\sigma^2=\delta(T(1,i)-T(k,j))$$

where δ is the delta function, $\delta(i)=0$, for $i\neq 0$, and $\delta(0)=1$.

Now, we consider the evaluation of the cost function, i.e. given X, we aim to compute $f_0(X) := r^\top(X)\Gamma^{-1}(X)r(X)$. The weight matrix Γ is symmetric, positive definite, block banded, and Toeplitz, see (10), with block entries $\Gamma_k := X_{\text{ext}}^\top W_{\tilde{c},k} X_{\text{ext}}$, for $k = 0, 1, \ldots, s$. For given X, and with $\{W_{\tilde{c},k}\}_{k=0}^s$, constructed as explained above, the sequence $\{\Gamma_k(X)\}_{k=0}^s$ is readily computable. Then $\Gamma(X)$ can be constructed, and from the solution of the system $\Gamma(X)y_r(X) = r(X)$, the cost function is found as $f_0(X) = r^\top(X)y_r(X)$.

The properties of $\Gamma(X)$ can be exploited in the solution of the system $\Gamma(X)y_r(X) = r(X)$. The LAPACK solver `DPBSV.F` is based on the banded Cholesky factorization [25] and has computational cost $O(d^3s^2m)$ floating point operations (flops). It ignores the Toeplitz structure of Γ and as a result the cost for this function increases quadratically with respect to the bandwidth s. The function `MB02GD.F` from the SLICOT library [26] uses simultaneously the band and the Toeplitz structure of Γ and has computational cost $O(d^3sm)$ flops. For the purpose of the simulation study of the algorithms, see Section 6, we use an m-file implementation of the banded Cholesky factorization.

In the case when a non-linear least squares optimization is used, instead of the cost function $f_0(X)$, one has to evaluate the vector $\Gamma^{-1/2}(X)r(X)$; see (11). It can be computed from the Cholesky factorization of $\Gamma(X)$ by back substitution only, so that it is cheaper than computing $f_0(X)$.

Next, we consider the computation of the derivative $f_0'(X)$, given in (12). Let y_r be the solution of $\Gamma y_r = r$, and let $y_r^\top =: [y_{r,1}^\top \cdots y_{r,m}^\top]$, where $y_{r,i} \in \mathbb{R}^{d\times 1}$. The first sum in (12) becomes

$$\sum_{i,j=1}^{m} a_j r_i^\top M_{ij} = A^\top Y_r \quad \text{where } Y_r := \text{vec}^{-1}(y_r) := \begin{bmatrix} y_{r,1}^\top \\ \vdots \\ y_{r,m}^\top \end{bmatrix} \tag{13}$$

The second sum in (12) can be written as

$$\sum_{i,j=1}^{m} [\, I \;\; 0 \,] \frac{V_{\tilde{c},ij}}{\sigma^2} \begin{bmatrix} X \\ -I \end{bmatrix} N_{ji} = \sum_{k=-s}^{s} (W_{\tilde{a},k}X - W_{\tilde{a}\tilde{b},k}) N_k^\top$$

where

$$W_{\tilde{c},k} =: \begin{bmatrix} W_{\tilde{a},k} & W_{\tilde{a}\tilde{b},k} \\ W_{\tilde{a}\tilde{b},k}^\top & W_{\tilde{b},k} \end{bmatrix}, \; k = -s, \ldots, s \quad \text{and} \quad N_k := \sum_{i=1}^{m-k} y_{r,i+k} y_{r,i}^\top, \; N_k = N_{-k}^\top, \; k = 0, \ldots, s \tag{14}$$

Thus the evaluation of the derivative $f_0'(X)$ uses the solution of $\Gamma(X)y_r(X) = r(X)$, already computed for the cost function evaluation.

The steps described above and the required number of flops are summarized in Algorithm 1.

Algorithm 1 (*Cost function and first derivative evaluation*)
Input: A, B, X, $\{W_{\tilde{c},k}\}_{k=0}^s$. flops per step

1. $\Gamma_k = X^\top W_{\tilde{a},k} X - X^\top W_{\tilde{a}\tilde{b},k} - (X^\top W_{\tilde{a}\tilde{b},k})^\top + W_{\tilde{b},k}$, for $k = 0, 1, \ldots, s$, $(s+1)(n^2d + 2nd^2 + 3d^2)$
2. $r = \text{vec}((AX - B)^\top)$, $m(n+1)d$
3. Solve (via banded Cholesky factorization) the system $\Gamma y_r = r$, where Γ is given in (10), $md(s^2d^2 + 7sd + 2)$

4. $f_0 = r^\top y_r$. md

   If only the cost function evaluation is required, output f_0 and stop.

5. $Y_r = \mathrm{vec}^{-1}(y_r)$, where vec^{-1} is defined in (13) 0
6. $N_k = \sum_{i=1}^{m-k} y_{r,i+k} y_{r,i}^\top$, for $k = 0, 1, \ldots, s$, $msd^2 - s(s+1)d^2/2$
7. $f_0' = 2A^\top Y_r - 2\sum_{k=-s}^{s} (W_{\tilde{a},k} X - W_{\tilde{a}\tilde{b},k}) N_k^\top$. $mnd + (2s+1)(n^2 d + nd + nd^2)$

Output f_0, f_0' and stop.

Algorithm 1 requires $O(md(s^2d^2+8sd+n)+sn^2d+snd^2)$ flops for a cost function evaluation and $O(md(s^2d^2+8sd+2n)+3sn^2d+3snd^2)$ for cost function and first derivative evaluation. Note that the flop counts depend on the structure through s. For any structure, however, $s \leqslant n+d$, where the worst case is achieved for $\mathscr{T} = \{[\mathtt{H}\ \ n+d]\}$ and $\mathscr{T} = \{[\mathtt{T}\ \ n+d]\}$.

Next, we consider the proposed iterative method. First we describe the implementation for the univariate case. Given an approximation $x^{(k)} \in \mathbb{R}^n$ on the current iteration step, we form the matrix $\Gamma(x^{(k)})$ and the residual vector $r(x^{(k)})$. Let us take $M = \Gamma^{-1}$ and $N = (\Gamma^{-1} r)(\Gamma^{-1} r)^\top$. The approximation $x^{(k+1)}$, on the next iteration step, is obtained from the solution of the following system:

$$\sum_{i,j=1}^{m} (a_j a_i^\top M_{ij} - W_{\tilde{a},ij} N_{ij}) x^{(k+1)} = \sum_{i,j=1}^{m} (a_j b_i^\top M_{ij} - W_{\tilde{a}\tilde{b},ij} N_{ij})$$

or equivalently

$$\left(A^\top Y_a - \sum_{k=-s}^{s} W_{\tilde{a},k} N_k \right) x^{(k+1)} = A^\top y_b - \sum_{k=-s}^{s} W_{\tilde{a}\tilde{b},k} N_k \tag{15}$$

where $Y_a := \Gamma^{-1} A$, $y_b := \Gamma^{-1} b$, and $W_{\tilde{a},k}$, $W_{\tilde{a}\tilde{b},k}$, N_k are defined in (14).

Algorithm 2 (*MVK1*)
Input: A, b, $\mathscr{T}$, ε. flops per step

1. Compute an initial approximation x, e.g. the TLS or the LS estimate.
2. Form $\{W_{\tilde{c},k}\}_{k=0}^{s}$ from the given $\mathscr{T}$.
3. Repeat
   - 3.1. $\Gamma_k = x^\top W_{\tilde{a},k} x - 2x^\top W_{\tilde{a}\tilde{b},k} + W_{\tilde{b},k}$, for $k = 0, 1, \ldots, s$, $(s+1)(n^2+2n+3)$
   - 3.2. Solve (via banded Cholesky factorization) the system $\Gamma Y_c = [A\ \ b]$, where Γ is given in (10), $m(s^2 + (4n+7)s + 2n + 2)$
   - 3.3. $y_r = Y_a x - y_b$, where $Y_c = [Y_a\ \ y_b]$, $m(n+1)$
   - 3.4. $N_k = \sum_{i=1}^{m-k} y_{r,i+k} y_{r,i}$, for $k = 0, 1, \ldots, s$, $ms - s(s+1)/2$
   - 3.5. $G = A^\top Y_a - W_{\tilde{a},0} N_0 - 2\sum_{k=1}^{s} W_{\tilde{a},k} N_k$, $mn + n^2(2s+3)$
   - 3.6. $h = A^\top y_b - W_{\tilde{a}\tilde{b},0} N_0 - 2\sum_{k=1}^{s} W_{\tilde{a}\tilde{b},k} N_k$, $m + n(2s+3)$
   - 3.7. Set $x_{\mathrm{prev}} := x$ and solve $Gx = h$. $2n^3/3$

Until $\|x - x_{\mathrm{prev}}\|/\|x\| < \varepsilon$.
Output $\hat{x} = x$ and stop.

Algorithm 2 requires $O(m(s^2 + 4ns) + 3n^2 s + 2n^3/3)$ flops per iteration.

In the multivariate case M_{ij} are $d \times d$ matrices, so that $X^{(k+1)}$ cannot be extracted out of the sums, as we did in the univariate case. Vectorizing the equation $F(X^{(k+1)}, X^{(k)})$, we have

$$\left(\sum_{i=1}^{m}\left(\sum_{j=1}^{m} M_{ji} \otimes a_j\right) \otimes a_i^\top\right) x - \operatorname{vec}\left(\sum_{i,j=1}^{m} a_j b_i^\top M_{ij}\right) = \left(\sum_{k=-s}^{s} N_k \otimes W_{\tilde{a},k}\right) x - \operatorname{vec}\left(\sum_{k=-s}^{s} W_{\tilde{a}\tilde{b},k} N_k^\top\right)$$

where $x := \operatorname{vec}(X^{(k+1)})$ and in order to save notation, we do not show the dependence of M_{ij} and N_{ij} on $X^{(k)}$. Next we specify how to compute the sums involving M_{ij} without computing the inverse matrix $M = \Gamma^{-1}$.

For the second sum, we have

$$\sum_{i,j=1}^{m} a_j b_i^\top M_{ij} = A^\top Y_b, \quad Y_b := \operatorname{vec}^{-1}(y_b) := \begin{bmatrix} y_{b,1}^\top \\ \vdots \\ y_{b,m}^\top \end{bmatrix}, \quad y_b =: \begin{bmatrix} y_{b,1} \\ \vdots \\ y_{b,m} \end{bmatrix}, \quad y_{b,i} \in \mathbb{R}^{d \times 1}$$

with y_b being the solution of the system $\Gamma y_b = \operatorname{vec}(B^\top)$. The sums $\sum_{j=1}^{m} M_{ji} \otimes a_j$, $i = 1, \ldots, m$ are found from the solution of the system $\Gamma Y_{\mathscr{A}} = \mathscr{A}$, where

$$\mathscr{A}^\top := \left[\begin{array}{ccc|c|ccc} a_1 & & 0 & & a_m & & 0 \\ & \ddots & & \cdots & & \ddots & \\ 0 & & a_1 & & 0 & & a_m \end{array}\right] \in \mathbb{R}^{nd \times md} \tag{16}$$

Let

$$Y_{\mathscr{A}}^\top =: [\, Q_1 \ \cdots \ Q_m \,] \quad \text{with } Q_i \in \mathbb{R}^{nd \times d}$$

One can check by inspection that $\sum_{j=1}^{m} M_{ji} \otimes a_j = Q_i$. Thus the second sum can be computed by

$$\sum_{i=1}^{m}\left(\sum_{j=1}^{m} M_{ji} \otimes a_j\right) \otimes a_i^\top = \sum_{i=1}^{m} Q_i \otimes a_i^\top$$

In addition to $Y_{\mathscr{A}}$ and y_b, we have to compute the solution y_r of the system $\Gamma y_r = r$, needed for the evaluation of the matrix N. In total, a system $\Gamma Y = [\mathscr{A} \ \operatorname{vec}(B^\top) \ r]$ with $n+2$ right-hand sides should be solved in order to assemble the system

$$\left(\sum_{i=1}^{m} Q_i \otimes a_i^\top - \sum_{k=-s}^{s} N_k \otimes W_{\tilde{a},k}\right) x = \operatorname{vec}\left(A^\top Y_b - \sum_{k=-s}^{s} W_{\tilde{a}\tilde{b},k} N_k^\top\right)$$

giving the approximation on the next iteration step.

Algorithm 3 (*MVK2*)
Input: A, B, $\mathscr{T}$, ε. flops per step

1. Compute an initial approximation X, e.g. the TLS or the LS estimate.
2. Form $\{W_{\tilde{c},k}\}_{k=0}^{s}$ from the given $\mathscr{T}$.

3. Repeat
   - 3.1. $\Gamma_k = X^\top W_{\tilde{a},k} X - X^\top W_{\tilde{a}\tilde{b},k} - (X^\top W_{\tilde{a}\tilde{b},k})^\top + W_{\tilde{b},k}$, for $k = 0, 1, \ldots, s$, $(s+1)(n^2d + 2nd^2 + 3d^2)$
   - 3.2. $r = \mathrm{vec}((AX - B)^\top)$, $m(n+1)d$
   - 3.3. Solve (via banded Cholesky factorization) the system $\Gamma Y = [\mathscr{A}\ \ \mathrm{vec}(B^\top)\ \ r]$, where Γ is given in (10), and $\mathscr{A}$ is given in (16), $md(s^2d^2 + (4n+11)sd + 2n + 4)$
   - 3.4. $N_k = \sum_{i=1}^{m-k} y_{r,i+k} y_{r,i}$, for $k = 0, 1, \ldots, s$, where $Y = [Y_{\mathscr{A}}\ \ y_b\ \ y_r]$, $mnd + (2s+1)(n^2d + nd + nd^2)$
   - 3.5. $G = \sum_{i=1}^{m} Q_i \otimes a_i^\top - \sum_{k=-s}^{s} N_k \otimes W_{\tilde{a},k}$, where $Y_{\mathscr{A}}^\top = [Q_1 \cdots Q_m]$, $(1+m)n^2d^2 + (2s+1)n^2d^2$
   - 3.6. $h = \mathrm{vec}(A^\top Y_b - \sum_{k=-s}^{s} W_{\tilde{a}\tilde{b},k} N_k^\top)$, where $Y_b = \mathrm{vec}^{-1}(y_b)$ $(1+m)nd + (2s+1)nd^2$
   - 3.7. Set $X_{\mathrm{prev}} := X$ and solve $Gx = h$, $2n^3d^3/3$
   - 3.8. $X = \mathrm{vec}^{-1}(x)$. 0

Until $\|X - X_{\mathrm{prev}}\|_F / \|X\|_F < \varepsilon$.

Output $\hat{X} = X$ and stop.

Algorithm 3 requires $O(md(s^2d^2 + 4nsd + n^2d) + 2sn^2d^2 + 2n^3d^3/3)$ flops per iteration.

6. COMPARISON OF THE ALGORITHMS

In this section, we compare numerically the statistical and computational efficiency of the STLS solution methods. In Section 6.1, we compare the best currently available algorithms from the literature [12, 15, 16], with the proposed algorithms, described in Section 4. In Section 6.2, we check the achievable accuracy of the algorithms by a benchmark problem with analytically known solution. All experiments are carried out in MATLAB 5 running on a PC i686.

6.1. Comparison with the algorithms of References [12, 15, 16]

For the optimization-based methods, described in Section 4, an option is the choice of the optimization method. We use the following functions from MATLAB's Optimization Toolbox:

- `fminsearch`—implements the Nelder–Mead simplex algorithm (labelled with NM),
- `fminunc`—implements the BFGS Quasi–Newton method (labelled with QN),
- `lsqnonlin`—implements the Levenberg–Marquardt method (labelled with LM).

Besides the results for the STLS estimator computed with the various solution methods, we show in the comparison the ones for the LS estimate, computed in MATLAB by the command `A\b`, and the TLS estimate, computed via the singular value decomposition [4, 5].

The simulation setup is as follows. A true data matrix $\bar{C} = [\bar{A}\ \ \bar{B}]$ with a desired structure $\mathscr{T}$ is generated, such that $\bar{A}\bar{x} = \bar{b}$, for some true value $\bar{x}$ of the model parameter vector x. The measurements available for the estimation are $p = \bar{p} + \tilde{p}$, where $\tilde{p} \sim N(0, \sigma^2 I)$. The estimation is repeated $N = 100$ times with different noise realizations and the average result is reported. The initial approximation for all algorithms computing the STLS estimate is the TLS estimate.

Table I. Left: average relative error of estimation $\bar{e}$ in per cents as a function of m. Right: average mega flop counts as a function of m.

m	LS	TLS	STLS	m	LS	TLS	STLN2	NM	QN	LM	MVK1
20	2.6	1	0.2	20	0.6	6.2	7.9	319	29	31	6
40	0.8	0.3	0.02	40	1.2	21.6	15.3	622	45	49	9
60	0.6	0.1	0.006	60	1.8	46.2	22.7	958	81	66	11
80	0.5	0.09	0.002	80	2.4	80.7	30.1	1333	131	88	15
100	0.5	0.06	0.001	100	3.0	124.1	37.5	1657	216	120	19

Table II. Left: average relative error of estimation $\bar{e}$ in per cents as a function of n. Right: average mega flop counts as a function of n.

n	LS	TLS	STLS	n	LS	TLS	STLN1	QN	LM	MVK1
2	1.3	1.3	1.0	2	3	125	167	49	71	21
4	2.3	2.3	1.5	4	8	217	234	127	178	94
8	4.1	4.1	3.1	8	21	421	451	507	619	400
16	5.7	5.4	3.9	16	66	919	750	2235	3021	1446
32	8.9	9.2	5.8	32	219	2355	1877	19568	20643	8478

First we compare the proposed algorithms with the algorithm `stln2` from Reference [16] (labelled below STLN2). The structure of the data matrix is Toeplitz with $n = 2$ and $d = 1$, i.e. $\mathscr{T} = \{[\mathtt{T}\ 3]\}$, and $\sigma = 0.015$. We use the experiment to show also the asymptotic properties of the estimators. Thus the sample size m is varied from $m = 20$ to $m = 100$ with a step of 20 samples.

Table I left shows the average relative error of estimation $\bar{e} = 1/N \sum_{l=1}^{N} \|\hat{x}^{(l)} - \bar{x}\|/\|\bar{x}\|$ in per cents, where $\hat{x}^{(l)}$ is the estimate on the lth repetition of the experiment. The various STLS algorithms have (approximately) equal value of $\bar{e}$ for all m (in the table the column STLS), which indicates convergence to the same minimum point. Table I right shows the required amount of computations, measured by the average flop counts (without those for the computation of the initial approximation). For small n, as in the considered simulation, the most efficient, from the STLS solvers, is the proposed iterative algorithm MVK1, followed by STLN2.

Next we compare the proposed algorithms with the algorithm `stln1` from Reference [16] (labelled STLN1). The simulation setup is as the one described above but now the structure is: A Toeplitz, b unstructured, i.e. $\mathscr{T} = \{[\mathtt{T}\ n],\ [\mathtt{U}\ 1]\}$, and $\sigma = 0.05$. In this experiment, we fix $m = 100$ and vary n from 2 to 32, in order to illustrate the behaviour of the methods for n/m growing. The NM algorithm is excluded from the comparison because in this experiment its computation is too expensive.

The results are given in Table II. For larger n (and for fixed m) the computational efficiency of algorithm STLN1 outperforms this of the proposed methods. The reasons are: (i) ignoring the Toeplitz structure of Γ in the implementation of the proposed methods affects the efficiency when $m \gg n$, and (ii) both the optimization-based algorithms and MVK1 solve on each iteration

Table III. Left: average relative error of estimation $\bar{e}$ in per cents as a function of d. Right: average mega flop counts as a function of d.

d	LS	TLS	STLS	d	LS	TLS	STLNB	QN	LM	MVK2
1	0.454	0.449	0.411	1	1	22	1720	30	34	7
2	0.475	0.476	0.443	2	2	30	2887	123	121	30
4	0.567	0.564	0.512	4	2	48	5962	492	733	151
6	0.571	0.568	0.515	6	3	70	9693	1120	2530	447

step an unstructured linear system of equations with n equations and n unknowns, which results in computational complexity $O(n^3)$. The theoretical computational complexity of STLN1 [15] is $O(n^2)$ in n per iteration.

The last experiment in this subsection deals with a multivariate STLS problem and compares the proposed algorithms with the algorithm of [12] (labelled STLNB). The simulation setup is as described above but the structure of the data matrix is: A Toeplitz with $m=40$, $n=2$, and B unstructured with d ranging from 1 to 6, i.e. $\mathscr{T}=\{[\mathtt{T}\ 2],\ [\mathtt{U}\ d]\}$, $\sigma=0.02$. The NM algorithm is excluded from the comparison because in this experiment its computation is also too expensive. Table III shows the results. The big difference between the flop counts obtained with STLN1 and STLN2, and those obtained with STLNB is due to the implementation of STLNB, which is not efficient.

6.2. Benchmark test

In Reference [18, Section IV C] an STLS problem with known analytical solution is given. The problem is with $n=1$, $d=1$, and $\mathscr{S}(\hat{p})$—Toeplitz. In this case

$$\underbrace{\begin{bmatrix} \hat{p}(1) & \hat{p}(0) \\ \hat{p}(2) & \hat{p}(1) \\ \vdots & \vdots \\ \hat{p}(n_p-1) & \hat{p}(n_p-2) \end{bmatrix}}_{\mathscr{S}(\hat{p})} \begin{bmatrix} x \\ -1 \end{bmatrix} = 0 \ \Rightarrow\ \hat{p}(l) = \hat{p}(0)\left(\frac{1}{x}\right)^l \quad \text{for } l=0,\ldots,n_p-1$$

so that the STLS problem

$$\min_{x,\,\hat{p}} \|p-\hat{p}\|_2^2 \quad \text{s.t. } \mathscr{S}(\hat{p}) \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

can be written as

$$\min_{\alpha,\beta} \sum_{l=0}^{n_p-1} (p(l)-\alpha\beta^l)^2 \tag{17}$$

where $\alpha := p(0)$ and $\beta := 1/x$. Eliminating α from the first order optimality condition of (17), the following equation is obtained

$$H(\beta) := \left(\sum_{l=1}^{n_p-1} l\,p(l)\beta^{l-1}\right)\left(\sum_{l=0}^{n_p-1}\beta^{2l}\right) - \left(\sum_{l=1}^{n_p-1} l\beta^{2l-1}\right)\left(\sum_{l=0}^{n_p-1} p(l)\beta^l\right) = 0 \tag{18}$$

Table IV. Benchmark test.

	NM	QN	LM	MVK1
$\lvert H(\hat{\beta})\rvert$	5.1768e−08	2.5288e−11	3.7324e−09	7.1054e−15
# flops	22590	15720	19260	2556

The left-hand side $H(\beta)$ of (18) is a polynomial in β of degree $3n_p - 4$. The solution $\hat{\beta}$ of the STLS problem (17) is the root of H for which the cost function is minimal. The optimal value for α is $\hat{\alpha} = \sum_{l=0}^{n_p-1} p(l)\hat{\beta}^l / \sum_{l=0}^{n_p-1} \hat{\beta}^{2l}$.

We use equation (18) to check the accuracy of the numerical solutions found by the optimization algorithms. The numerical solutions are computed with the highest possible accuracy, i.e. the stopping criterion is $\|x^{(k-1)} - x^{(k)}\| / \|x^{(k-1)}\| < \varepsilon$, where ε is the machine epsilon. Table IV shows $|H(\beta)|$ when β is substituted with the computed STLS solution, and the corresponding flop count. The data for the test is $p = [6\ 5\ 4\ 3\ 2\ 1]^\top$ and the initial approximation for the algorithms is the TLS estimate.

The result shows that the MVK1 algorithm achieves better numerical accuracy than the optimization-based algorithms. MVK1 is based on the first order optimality condition and does not use cost function evaluations. There is a loss of accuracy in the cost function evaluation because the original data C is squared in the computation of f_0. Note that the QN method has 4 more accurate digits than the NM method. This is due to the use of information for the first derivative in addition to the cost function.

7. CONCLUSIONS

We have proposed efficient numerical methods for the computation of the STLS estimator. The structure of the data matrix is specified block-wise, where each of the blocks is Toeplitz/Hankel structured, unstructured, or noise free. The solution methods are based on an equivalent unconstrained optimization problem, in which the correction Δp is eliminated. The cost function of the equivalent problem is $f_0(X) = r^\top \Gamma^{-1} r$ where the weight matrix Γ is proportional to the covariance matrix $V_{\tilde{r}}$ of the centred residual $\tilde{r}$. Under our structure assumptions Γ is a block banded Toeplitz matrix.

The proposed numerical methods are (i) standard optimization methods in combination with an efficient cost function and first derivative evaluation, and (ii) a new iterative method similar to the one proposed in References [19, 20]. Both approaches have computational cost linear in the sample size m. The efficient implementation is possible due to exploitation of the banded structure of the matrix Γ.

We numerically compared the proposed methods with the ones of References [12, 15, 16]. Future work aims to generalize the approach for block Toeplitz/Hankel structured matrices. We are looking for specific problems that can benefit from the algorithms. The numerical efficiency of the proposed methods can be improved when they are specialized to particular STLS problems.

ACKNOWLEDGEMENTS

Dr S. Van Huffel is a full professor and I. Markovsky is a research assistant at the Katholieke Universiteit Leuven, Belgium. I. Markovsky is supported by a K.U. Leuven doctoral scholarship. Dr A. Kukush is supported by a postdoctoral research fellowship of the Belgian office for Scientific, Technical and Cultural Affairs, promoting Scientific and Technical Collaboration with Central and Eastern Europe. Our research is supported by Research Council KUL: GOA-Mefisto 666, IDO/99/003, and IDO/02/009 (predictive computer models for medical classification problems using patient data and expert knowledge), several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0200.00 (damage detection in composites by optical fibers), G.0078.01 (structured matrices), G.0407.02 (support vector machines), G.0269.02 (magnetic resonance spectroscopic imaging), G.0270.02 (non-linear Lp approximation), research communities (ICCoS, ANMMM); AWI: Bil. Int. Collaboration Hungary/Poland; IWT: PhD Grants, Belgian Federal Government: DWTC (IUAP IV-02 (1996-2001) and IUAP V-22 (2002-2006): Dynamical Systems and Control: Computation, Identification & Modelling); EU: NICONET, INTERPRET, PDT-COIL, MRS/MRI signal processing (TMR); Contract Research/agreements: Data4s, IPCOS.

REFERENCES

1. Levin MJ. Estimation of a system pulse transfer function in the presence of noise. *IEEE Transactions on Automatic Control* 1964; **9**:229–235.
2. Stoica P, Söderström T. Bias correction in least-squares identification. *International Journal of Control* 1982; **35**(3):449–457.
3. Aoki M, Yue P. On *a priori* error estimates of some identification methods. *IEEE Transactions on Automatic Control* 1970; **15**(5):541–548.
4. Golub GH, Van Loan CF. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis* 1980; **17**:883–893.
5. Van Huffel S, Vandewalle J. *The Total Least Squares Problem*: *Computational Aspects and Analysis*. SIAM: Philadelphia, 1991.
6. Aoki M, Yue P. On certain convergence questions in system identification. *SIAM Journal on Control and Optimization* 1970; **8**(2):239–256.
7. Kukush A, Markovsky I, Van Huffel S. Consistency of the structured total least squares estimator in a multivariate model. *Technical Report 02-192*, Department EE, K.U. Leuven, 2002.
8. De Moor B. Structured total least squares and L_2 approximation problems. *Linear Algebra and Its Applications* 1993; **188–189**:163–207.
9. De Moor B, Roorda B. L_2-optimal linear system identification structured total least squares for SISO systems. In *The Proceedings of the Conference on Decision and Control*, Peshkin M, McClamroch NH, Yin G (eds), 1994; 2874–2879.
10. Gill PE, Murray M, Wright MH. *Practical Optimization*. Academic Press: New York, 1999.
11. Rosen JB, Park H, Glick J. Total least norm formulation and solution of structured problems. *SIAM Journal on Matrix Analysis and Applications* 1996; **17**:1996.
12. Van Huffel S, Park H, Rosen JB. Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Transactions on Signal Processing* 1996; **44**(10):2464–2474.
13. Lemmerling P. Structured total least squares: analysis, algorithms and applications. *Ph.D. Thesis*, ESAT/SISTA, K.U. Leuven, 1999.
14. Mastronardi N. Fast and reliable algorithms for structured total least squares and related matrix problems. *Ph.D. Thesis*, ESAT/SISTA, K.U. Leuven, 2001.
15. Mastronardi N, Lemmerling P, Van Huffel S. Fast structured total least squares algorithm for solving the basic deconvolution problem. *SIAM Journal on Matrix Analysis and Applications* 2000; **22**:533–553.
16. Lemmerling P, Mastronardi N, Van Huffel S. Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms* 2000; **23**:371–392.
17. Abatzoglou T, Mendel J, Harada G. The constrained total least squares technique and its application to harmonic superresolution. *IEEE Transactions on Signal Processing* 1991; **39**:1070–1087.
18. De Moor B. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing* 1994; **42**(11):3104–3113.
19. Premoli A, Rastello ML. The parametric quadratic form method for solving tls problems with elementwise weighting. In *Total Least Squares and Errors-in-Variables Modeling*: *Analysis*, *Algorithms and Applications*, Van Huffel S, Lemmerling P (eds). Kluwer: Dordrecht, 2002; 67–76.

20. Markovsky I, Rastello M-L, Premoli A, Kukush A, Van Huffel S. The element-wise weighted total least squares problem. *Technical Report 02-48*, Department EE, K.U. Leuven, 2002.
21. Nelder JA, Mead R. A simplex method for function minimization. *Computer Journal* 1965; **7**:308–313.
22. Marquardt D. An algorithm for least squares estimation of non-linear parameters. *SIAM Journal on Applied Mathematics* 1963; **11**:431–441.
23. Markovsky I, Van Huffel S, Kukush A. On the computation of the structured total least squares estimator. *Technical Report 02-203*, Department EE, K.U. Leuven, 2002.
24. Kukush A, Markovsky I, Van Huffel S. About the convergence of the computational algorithm for the EW-TLS estimator. *Technical Report 02-49*, Department EE, K.U. Leuven, 2002.
25. Golub GH, Van Loan CF. *Matrix Computations* (3rd edn). Johns Hopkins University Press: Baltimore, 1996.
26. Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A. SLICOT—a subroutine library in systems and control theory. In *Applied and Computational Control*, *Signal and Circuits*, Datta BN (ed.). vol. 1, Chapter 10. Birkhauser: Boston, 1999; 499–539.

