

A SIMPLE TLS TREATMENT OF THE PARTIAL EIV-MODEL AS ONE WITH SINGULAR COFACTOR MATRICES, PART II: SURPRISING RESULTS FOR AR(3) PROCESSES

Shahram Jazaeri¹, Burkhard Schaffrin², and Kyle Snow³

¹Department of Civil and Environmental Engineering, Shiraz University, Iran

²D-78 462 Konstanz, Germany; e-mail: baschaffrin@gmail.com

³Polaris Geospatial Solutions, LLC, Westerville, Ohio, USA; e-mail: ksnow6378137@gmail.com

prepared for the
XI Hotine-Marussi Symp., Genova/Italy, Sept. 2026

- Review of the EIV-Model with singular cofactor matrices (Snow, 2012)
 - Algorithm A or WTLS-EIV04 (resp. Alg. 1 of Snow, 2012)
 - Algorithm B or WTLS-EIV-02 (resp. Alg. 3 of Snow, 2012)
 - Algorithm D as a generalization of Mahboub (2012) and Schaffrin (2015)
- Review of the Gauss-Helmert Model as applied to the EIV-Model (Deming, 1931, 1934); Algorithm LESS-GHM and the Neitzel-Schaffrin condition (2016)
- Review of the partial EIV-Model with nonzero correlations (Xu et al., 2012)
 - Algorithm by Wang et al. (2017), later again published by Han et al. (2020)
 - The PEIV-Model as special case of the EIV-Model with singular cofactor matrices (Schaffrin et al., 2014)
- Application: An EIV-Model for an auto-regressive process of the third order, AR(3)
 - Two data sets: Table 5 in Han et al. (2020) and email from Wang (2022).
 - Computational efficiency comparisons
- Conclusions

Review (1/5)

- The EIV-Model with singular cofactor matrices (Snow, 2012; Schaffrin et al., 2014)
- Setup of the *EIV-Model*:

$$\mathbf{y} = \underbrace{(A - E_A)}_{n \times m} \boldsymbol{\xi} + \mathbf{e}_y, \quad \mathbf{e}_A := \text{vec } E_A, \quad \text{rk } A = m < n, \quad nm \times 1$$

$$\begin{bmatrix} \mathbf{e}_y \\ \mathbf{e}_A \end{bmatrix} \sim \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \sigma_0^2 Q = \sigma_0^2 \begin{bmatrix} Q_y & Q_{yA} \\ Q_{Ay} & Q_A \end{bmatrix} \right), \quad Q \text{ potentially singular.}$$

- *Lagrange Target function* for TLS-estimates of $\boldsymbol{\xi}$ if Q is nonsingular, $Q^{-1} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$:

$$\begin{aligned} \Phi(\mathbf{e}_y, \mathbf{e}_A, \boldsymbol{\xi}, \boldsymbol{\lambda}) &:= \mathbf{e}_y^T P_{11} \mathbf{e}_y + 2\mathbf{e}_y^T P_{12} \mathbf{e}_A + \mathbf{e}_A^T P_{22} \mathbf{e}_A + \\ &+ 2\boldsymbol{\lambda}^T [\mathbf{y} - A\boldsymbol{\xi} - \mathbf{e}_y + (\boldsymbol{\xi}^T \otimes I_n) \mathbf{e}_A] \doteq \text{stationary} \end{aligned}$$

- Euler-Lagrange *necessary conditions*:

$$\frac{1}{2} \frac{\partial \Phi}{\partial \mathbf{e}_y} = P_{11} \tilde{\mathbf{e}}_y + P_{12} \tilde{\mathbf{e}}_A - \hat{\boldsymbol{\lambda}} \doteq \mathbf{0} \quad (1)$$

$$\frac{1}{2} \frac{\partial \Phi}{\partial \mathbf{e}_A} = P_{22} \tilde{\mathbf{e}}_A + P_{21} \tilde{\mathbf{e}}_y + (\hat{\boldsymbol{\xi}}^T \otimes I_n) \hat{\boldsymbol{\lambda}} \doteq \mathbf{0} \quad (2)$$

$$\frac{1}{2} \frac{\partial \Phi}{\partial \boldsymbol{\xi}} = -A^T \hat{\boldsymbol{\lambda}} + \tilde{E}_A^T \hat{\boldsymbol{\lambda}} \doteq \mathbf{0} \quad (3)$$

$$\frac{1}{2} \frac{\partial \Phi}{\partial \boldsymbol{\lambda}} = \mathbf{y} - A\hat{\boldsymbol{\xi}} - \tilde{\mathbf{e}}_y + (\hat{\boldsymbol{\xi}}^T \otimes I_n) \tilde{\mathbf{e}}_A \doteq \mathbf{0} \quad (4)$$

with $\otimes$ as Kronecker-Zehfuss product defined by $F \otimes G := [f_{ij} \cdot G]$.

Review (2/5)

- Leading to: (1) & (2) $\Rightarrow \begin{bmatrix} \tilde{\mathbf{e}}_y \\ \tilde{\mathbf{e}}_A \end{bmatrix} = \begin{bmatrix} Q_y & Q_{yA} \\ Q_{Ay} & Q_A \end{bmatrix} \cdot \begin{bmatrix} I_n \\ -(\hat{\xi} \otimes I_n) \end{bmatrix} \cdot \hat{\lambda} = \begin{bmatrix} Q_y - Q_{yA}(\hat{\xi} \otimes I_n) \\ Q_{Ay} - Q_A(\hat{\xi} \otimes I_n) \end{bmatrix} \cdot \hat{\lambda}$ (5)

(4)
 $\Rightarrow \mathbf{y} - A\hat{\xi} = [I_n | -(\hat{\xi}^T \otimes I_n)] \begin{bmatrix} Q_y & Q_{yA} \\ Q_{Ay} & Q_A \end{bmatrix} \begin{bmatrix} I_n \\ -(\hat{\xi} \otimes I_n) \end{bmatrix} \cdot \hat{\lambda} =: Q_1 \cdot \hat{\lambda} \text{ or } \boxed{\hat{\lambda} = Q_1^{-1}(\mathbf{y} - A\hat{\xi})}$ (A3)

wh. $\boxed{Q_1 := BQB^T}$ (A2), for $B := \underbrace{[I_n | -(\hat{\xi}^T \otimes I_n)]}_{n \times n(m+1)} = B(\hat{\xi})$ (A1), with $\text{rk } B = n$ ("full row rank")

- Generalizing Fang's (2011) TLS solution; Algorithm A or WTLS-EIV04 (Snow, 2012)

(3) $\Rightarrow \mathbf{0} = (A - \tilde{E}_A)^T \cdot \hat{\lambda} = (A - \tilde{E}_A)^T Q_1^{-1}[(\mathbf{y} - \tilde{E}_A \hat{\xi}) - (A - \tilde{E}_A)\hat{\xi}] \Rightarrow$
 $\Rightarrow \boxed{\hat{\xi} = [(A - \tilde{E}_A)^T Q_1^{-1}(A - \tilde{E}_A)]^{-1} \cdot [(A - \tilde{E}_A)^T Q_1^{-1}(\mathbf{y} - \tilde{E}_A \hat{\xi})]}$ (A5)

and: $\boxed{\begin{bmatrix} \tilde{\mathbf{e}}_y \\ \tilde{\mathbf{e}}_A \end{bmatrix} = Q \cdot B^T \cdot \hat{\lambda} = Q \cdot B^T \cdot Q_1^{-1}(\mathbf{y} - A\hat{\xi})}$ (A4)

with: $\boxed{\hat{\sigma}_0^2 := (\mathbf{y} - A\hat{\xi})^T \cdot \hat{\lambda} \cdot (n - m)^{-1}}$ (A6)

- This TLS solution only requires Q_1 to be *nonsingular*, not Q itself! For the initial approx. $\hat{\xi}$, the WOLS estimate $\hat{\xi}_0 := (A^T Q_y^+ A)^{-1} \cdot A^T Q_y^+ \mathbf{y}$ may serve, where Q_y^+ denotes the "pseudo-inverse" of Q_y if singular.

Derivation of the TLS solution (3/5)

- Allowing Q_1 to be singular, due to $\text{rk } BQ < \text{rk } B = n$; Algorithm B (Snow, 2012): Following Neitzel and Schaffrin (2016), a unique TLS solution will exist if and only if

$$\boxed{\text{rk}[BQ \mid A] = n = \text{rk } B; \quad \text{i.e., } B \text{ with full row rank.}} \quad (\text{B1})$$

Both equalities matter, which, however, has been neglected by Hu, Fang, & Zeng (J. of Geodesy 98, 2024:75) in their erroneously claimed “counter examples.”

- Assuming that $\text{rk}[BQ \mid A - \tilde{E}_A] = n$ holds as well under (B1), we can conclude that

$$Q_3 := Q_1 + (A - \tilde{E}_A)S(A - \tilde{E}_A)^T \quad (\text{B2})$$

will be *nonsingular* for any symmetric, positive-definite matrix S , where

$$\boxed{Q_3 \cdot \hat{\lambda} = (y - A\hat{\xi}) + (A - \tilde{E}_A)S \underbrace{[(A - \tilde{E}_A)^T \hat{\lambda}]}_{= \mathbf{0} \text{ due to (3)}} = y - A\hat{\xi} = Q_1 \cdot \hat{\lambda}.} \quad (\text{B3})$$

- Thus, because of (3), $\mathbf{0} = (A - \tilde{E}_A)^T \hat{\lambda} = (A - \tilde{E}_A)^T Q_3^{-1} [(y - \tilde{E}_A \hat{\xi}) - (A - \tilde{E}_A) \hat{\xi}] \Rightarrow$

$$\hat{\xi} = [(A - \tilde{E}_A)^T Q_3^{-1} (A - \tilde{E}_A)]^{-1} \cdot [(A - \tilde{E}_A)^T Q_3^{-1} (y - \tilde{E}_A \hat{\xi})], \quad \text{with} \quad (\text{B5})$$

$$\Rightarrow \begin{bmatrix} \tilde{e}_y \\ \tilde{e}_A \end{bmatrix} = QB^T \cdot \hat{\lambda} = QB^T \cdot Q_3^{-1} (y - A\hat{\xi}) \quad (\text{B4}), \quad \text{and} \quad \hat{\sigma}_0^2 := (y - A\hat{\xi})^T \cdot \hat{\lambda} \cdot (n - m)^{-1}. \quad (\text{B6})$$

Review (4/5)

- Modifying Mahboub's TLS solution (Mahboub, 2012), Algorithm D (Schaffrin, 2015)

$$(B3) \Rightarrow A^T Q_3^{-1}(\mathbf{y} - A\hat{\boldsymbol{\xi}}) = A^T \hat{\boldsymbol{\lambda}} \stackrel{(3)}{=} \tilde{E}_A^T \hat{\boldsymbol{\lambda}} = \text{vec}(\hat{\boldsymbol{\lambda}}^T \tilde{E}_A) = (I_m \otimes \hat{\boldsymbol{\lambda}}^T) \tilde{\mathbf{e}}_A \quad (D1)$$

and, by using $\tilde{\mathbf{e}}_A$ from (5),

$$A^T Q_3^{-1}(\mathbf{y} - A\hat{\boldsymbol{\xi}}) = (I_m \otimes \hat{\boldsymbol{\lambda}}^T)[Q_{Ay} \hat{\boldsymbol{\lambda}} - Q_A(\hat{\boldsymbol{\xi}} \otimes \hat{\boldsymbol{\lambda}})] = (I_m \otimes \hat{\boldsymbol{\lambda}}^T)[Q_{Ay} \hat{\boldsymbol{\lambda}} - Q_A(I_m \otimes \hat{\boldsymbol{\lambda}})\hat{\boldsymbol{\xi}}] \quad (D2)$$

from which the following equation can be obtained:

$$\boxed{[A^T Q_3^{-1} A - (I_m \otimes \hat{\boldsymbol{\lambda}})^T Q_A (I_m \otimes \hat{\boldsymbol{\lambda}})] \hat{\boldsymbol{\xi}} = A^T Q_3^{-1} \mathbf{y} - (I_m \otimes \hat{\boldsymbol{\lambda}})^T Q_{Ay} \hat{\boldsymbol{\lambda}}} \quad (D3)$$

Starting with the WOLS solution $\hat{\boldsymbol{\xi}}_0 := (A^T Q_y^+ A)^{-1} A^T Q_y^+ \mathbf{y}$ and $\tilde{E}_A := 0$, Algorithm D would follow with (A1) to get B , with (A2) to Q_1 , then with (B2) to get Q_3 if Q_1 turned out to be singular, and finally with (B3) to get $\hat{\boldsymbol{\lambda}}$, which would further provide:

- the residual vector via (B4),
 - the estimated WTLS parameter vector via (D3), and
 - the estimated variance component via (B6), until convergence.
- Note that Algorithm D can, for the first time, tolerate a *singular* matrix Q_1 here by extending it to Q_3 while adding a term which requires the choice of a suitable matrix S (just as in the case of algorithm B). Most likely, it will be possible to apply the same trick to Algorithm A as well.
- What is remarkable, however, seems that the exchange of (B5) through (D3) to gather $\hat{\boldsymbol{\xi}}$ makes algorithm D, on average, superior to Algorithm B. *But why is it so?*

Review (5/5)

- The EIV-Model as nonlinear Gauss-Helmert Model; Algorithm LESS-GHM (Deming, 1931, 1934); by interpreting the *functional* part of the EIV-Model as follows:

$$\underset{n \times 1}{\mathbf{w}} := \mathbf{y} - A\boldsymbol{\xi} = [I_n \mid -(\boldsymbol{\xi}^T \otimes I_n)] \begin{bmatrix} \mathbf{e}_y \\ \mathbf{e}_A \end{bmatrix} =: B(\boldsymbol{\xi}) \cdot \mathbf{e}, \quad \mathbf{e} \in \mathbb{R}^{n(m+1) \times 1}, \quad \text{rk } A = m < n,$$

and by combining it with the *stochastic* part:

$$\mathbf{e} \sim (\mathbf{0}, Q), \quad Q_A \text{ possibly singular, but } Q_y \text{ not,}$$

the resulting model will be *nonlinear* and of type *Gauss-Helmert*. Its typical LESS (= LEast-Squares Solution) is based on iteratively linearizing the nonlinear function $B(\boldsymbol{\xi})$; see, e.g., Schaffrin and Snow (2010), or chapter 4 in Snow (2012), for more details. For instance, by choosing $\mathbf{e}_A := \mathbf{0}$ and $\hat{\boldsymbol{\xi}}_0 := (A^T Q_y^{-1} A)^{-1} A^T Q_y^{-1} \mathbf{y}$, and consequently $B_0 := [I_n \mid -(\hat{\boldsymbol{\xi}}_0^T \otimes I_n)]$ and $Q_1^0 := B_0 Q B_0^T$ as initial values along with $\mathbf{w}_0 := \mathbf{y} - A\hat{\boldsymbol{\xi}}_0$, the LESS in the first iteration step would be:

$$\hat{\boldsymbol{\xi}}_1 := [A^T (Q_1^0)^{-1} A]^{-1} A^T (Q_1^0)^{-1} \mathbf{w}_0, \text{ with } \tilde{\mathbf{e}}_1 := Q B_0^T \cdot (Q_1^0)^{-1} \mathbf{w}_1, \text{ where } \mathbf{w}_1 := \mathbf{y} - A\hat{\boldsymbol{\xi}}_1, \text{ etc.}$$

- The inverse will always exist when the *sufficient* conditions $\text{rk}[B_j Q] = n = \text{rk } B_j$ hold. A set of *sufficient and necessary* conditions according to Neitzel and Schaffrin (2016) would be

$$\text{rk} [B_j Q \mid A] = n = \text{rk } B_j \quad \text{for all iteration cycles } j,$$

where B_j has to have *full row rank*. All statements that claim otherwise (such as those of Hu, Fang, & Zeng, *J. of Geodesy*, 98, 2004:75) are definitely erroneous.

The Weighted TLS solution for the Partial EIV-Model—special case

- Model setup with *nonrandom elements* in matrix A (Xu et al., 2012):

$$\mathbf{y} = \underbrace{(A - E_A)}_{n \times m} \boldsymbol{\xi} + \mathbf{e}_y = (\boldsymbol{\xi}^T \otimes I_n) \cdot \underbrace{\text{vec}(A - E_A)}_{=: \boldsymbol{\mu}_A} + \mathbf{e}_y = (\boldsymbol{\xi}^T \otimes I_n) \cdot \underbrace{\boldsymbol{\mu}_A}_{nm \times 1} + \mathbf{e}_y,$$

where $\boldsymbol{\mu}_A$ is split into $\boldsymbol{\mu}_A := \boldsymbol{\alpha} + G \cdot \boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_a := \mathbf{a} - \mathbf{e}_a$, with the $nm \times 1$ vector $\boldsymbol{\alpha}$ consisting of the *nonrandom* elements of A and zeros elsewhere. The $t \times 1$ vector $\mathbf{a}$ lists a basis for the *random* elements, which, by the given $nm \times t$ matrix G , generates all the random elements of A as a combination of $\mathbf{a}$. In addition, it is assumed that:

$$\begin{bmatrix} \mathbf{e}_y \\ \mathbf{e}_a \end{bmatrix} \sim \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \sigma_0^2 Q' = \sigma_0^2 \begin{bmatrix} Q_y & Q_{ya} \\ Q_{ay} & Q_a \end{bmatrix} \right) = \sigma_0^2 \begin{bmatrix} p'_{11} & p'_{12} \\ p'_{21} & p'_{22} \end{bmatrix}^{-1} \quad \text{symmetric, positive-definite}$$

- Lagrange target function for TLS-estimate of $\boldsymbol{\xi}$:

$$\Phi(\boldsymbol{\mu}_a, \boldsymbol{\xi}) := \left[\left[\mathbf{y} - (\boldsymbol{\xi}^T \otimes I_n)(\boldsymbol{\alpha} + G\boldsymbol{\mu}_a) \right]^T \middle| (\mathbf{a} - \boldsymbol{\mu}_a)^T \right] \begin{bmatrix} p'_{11} & p'_{12} \\ p'_{21} & p'_{22} \end{bmatrix} \begin{bmatrix} \mathbf{y} - (\boldsymbol{\xi}^T \otimes I_n)(\boldsymbol{\alpha} + G\boldsymbol{\mu}_a) \\ \mathbf{a} - \boldsymbol{\mu}_a \end{bmatrix} \doteq \text{stationary}$$

- Euler-Lagrange necessary conditions:

$$\begin{aligned} \frac{1}{2} \frac{\partial \Phi}{\partial \mu_a} = & -G^T (\hat{\xi} \otimes I_n) p'_{11} [\mathbf{y} - (\hat{\xi}^T \otimes I_n)(\alpha + G\hat{\mu}_a)] - G^T (\hat{\xi} \otimes I_n) p'_{12} (\mathbf{a} - \hat{\mu}_a) - \\ & - p'_{21} [\mathbf{y} - (\hat{\xi}^T \otimes I_n)(\alpha + G\hat{\mu}_a)] - p'_{22} (\mathbf{a} - \hat{\mu}_a) \doteq 0 \end{aligned} \quad (6)$$

$$\frac{1}{2} \frac{\partial \Phi}{\partial \hat{\xi}} = \text{etc. (cf. Wang et al., 2017, resp. Han et al., 2020).} \quad (7)$$

Note that Han et al. (2020) independently invented a similar algorithm that provides the same results as that of Wang et al. (2017) at all iterations, only less efficiently so.

- Model setup with *quasi-random* elements of matrix A that have zero variances and zero covariances, and thus turn out to be *nonrandom*, indeed (Schaffrin, 2015). So, the cofactor matrix Q_A may be *singular*, and the cross-correlations in Q_{yA} may be nonzero. Suitable algorithms with reduced “bookkeeping” would include Algorithms B and D as well as Deming’s Algorithm LESS-GHM, which works with the classical (nonlinear) Gauss-Helmert Model and its iterative linearization.

EIV-Model for the auto-regressive process AR(3)

- Application to an *auto-regressive process* of the third order, AR(3); setting up the EIV-Model, assuming the collection of $n + 3$ observations:

$$\left. \begin{array}{l} y_4 - e_4 = (y_3 - e_3) \cdot \xi_1 + (y_2 - e_2) \cdot \xi_2 + (y_1 - e_1) \cdot \xi_3 \\ \vdots \\ y_{n+2} - e_{n+2} = (y_{n+1} - e_{n+1}) \cdot \xi_1 + (y_n - e_n) \cdot \xi_2 + (y_{n-1} - e_{n-1}) \cdot \xi_3 \\ y_{n+3} - e_{n+3} = (y_{n+2} - e_{n+2}) \cdot \xi_1 + (y_{n+1} - e_{n+1}) \cdot \xi_2 + (y_n - e_n) \cdot \xi_3 \end{array} \right\} \Rightarrow \boxed{\begin{array}{l} \mathbf{y} = (\mathbf{A} - \mathbf{E}_A) \cdot \boldsymbol{\xi} + \mathbf{e}_y \\ \mathbf{y} \in \mathbb{R}^{n \times 1}, \quad \boldsymbol{\xi} \in \mathbb{R}^{3 \times 1} \end{array}}$$

where $\mathbf{y} := [y_4, \dots, y_{n+3}]_{n \times 1}^T$, $\mathbf{A} := \begin{bmatrix} y_3 & y_4 & y_5 & \cdots & y_{n+2} \\ y_2 & y_3 & y_4 & \cdots & y_{n+1} \\ y_1 & y_2 & y_3 & \cdots & y_n \end{bmatrix}_{n \times 3}^T$, similarly $\mathbf{e}_y$, E_A , & ξ . Moreover,

$$D\{\mathbf{e}_y\} = \sigma_0^2 \underset{n \times n}{Q_y}, \quad D\{\mathbf{e}_A\} = \sigma_0^2 \underset{3n \times 3n}{Q_A}, \quad \text{and} \quad C\{\mathbf{e}_y, \mathbf{e}_A\} = \begin{bmatrix} 0 & \sigma_4^2 & 0 & 0 & \dots \\ 0 & 0 & \sigma_5^2 & 0 & \dots \\ 0 & 0 & 0 & \sigma_6^2 & \dots \\ \dots & & & \ddots & \dots \end{bmatrix}$$

must be defined, too. Here, we let Q_y be *diagonal and invertible*; but Q_A will *not be diagonal*, while Q_1 will be *invertible*, so that the use of Q_3 may not be necessary, but possible.

Two datasets for an *auto-regressive process* of type AR(3)

- Two datasets have been used for the AR(3) EIV-Model
 - (i) The “*Han data*” were taken from *Table 5* of Han et al. (2020), published in the *Survey Review* 52(371):126–133 along with an algorithm based on the Partial EIV-Model of Xu et al. (2012). Unfortunately, this algorithm turned out to be identical in each iteration step with that of Wang et al. (2017) that had been published — in Chinese language — three years earlier. So, we only list *Wang’s algorithm* in our experiments. Note that $Q_y := I_n$, and similarly Q_A , resp. Q_{yA} .
 - (ii) A different set of data with $n + 3 = 36$, the “*Wang data*” were provided to us by L. Wang in August 2022 by email. His experiments with this dataset on the basis of a PEIV-Model again are documented in Wang et al. (2017) using his original algorithm.

Computational efficiency comparisons (1/4)

- A comparison of the efficiency of the above algorithms, using the WOLSE for initial values.
 - by Wang et al. (2017), “*Wang’s algorithm*”
 - by Snow (2012), algorithms A and B
 - by Schaffrin et al. (2012), algorithm D
 - by Deming (1931, 1934), “*Deming’s algorithm*”

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	1.179081	0.041900	-0.214448	12.4197	33	2.0E-20	2.7
A	1.066619	0.376566	-0.436110	12.9746	130	1.8E-19	20.1
B	1.179081	0.041900	-0.214448	12.4197	110	7.8E-20	17.9
D	1.179081	0.041900	-0.214448	12.4197	58	4.5E-28	8.0
Deming	1.066619	0.376566	-0.436110	12.9746	81	7.2E-19	18.0

$\xi_0 = [0.635059, 0.327809, 0.041087]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

For Han’s data, the five algorithms provide two different local estimates, TLSE-1 and TLSE-2, with slightly different TSSR’s, namely 12.4197 for algorithms “Wang,” B, and D; and 12.9746 for algorithms A and “Deming.” Which of the two estimates might be more plausible?

Computational efficiency comparisons (2/4)

- Use all 5 algorithms with TLSE-1 & TLSE-2 for init. values; look for unique solutions.

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	1.179081	0.041900	-0.214448	12.419737	9	4.0E-20	2.2
A	1.179081	0.041900	-0.214448	12.419737	9	5.7E-20	3.1
B	1.179081	0.041900	-0.214448	12.419737	99	1.5E-19	33.6
D	1.179081	0.041900	-0.214448	12.419737	14	7.5E-28	4.1
Deming	1.179081	0.041900	-0.214448	12.419737	38	2.6E-19	15.6

$\xi_0 = [1.179081, 0.041900, -0.214448]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

All the TSSR's-1 and the TLSE's-1 do coincide; but algorithm B needs exceptionally more time to converge. The estimate $\hat{\xi}_2$ is awfully small. What could be the reason?

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	1.066619	0.376566	-0.436110	12.974643	15	6.8E-20	1.30
A	1.066619	0.376566	-0.436110	12.974643	16	1.9E-19	2.61
B	1.066619	0.376566	-0.436110	12.974643	43	1.1E-19	7.41
D	1.066619	0.376566	-0.436110	12.974643	18	5.8E-28	2.63
Deming	1.066619	0.376566	-0.436110	12.974643	75	6.0E-19	18.38

$\xi_0 = [1.066619, 0.376566, -0.43611]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

All TSSR's-2 and TLSE's-2 coincide with reasonable times. The TLSE-2 looks more meaningful than TLSE-1 at first sight even though TSSR-2 is slightly bigger.

Computational efficiency comparisons (3/4)

- Analyzing “Wang’s data,” using WOLS estimates for the initial values

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	0.977246	-1.016296	1.042873	37.842832	52	7.4E-21	5.57
A	1.051499	0.575898	-0.619968	34.702215	90	1.0E-19	15.99
B	1.055186	0.864275	-0.911266	32.396447	40	1.5E-19	7.46
D	0.150127	0.718933	0.13483	40.657354	258	5.1E-28	35.53
Deming	1.055186	0.864275	-0.911266	32.396447	24	2.0E-19	5.84

$\xi_0 = [0.299361, 0.532447, 0.169688]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

The five algorithms provide four different local estimates, TLSE-1 to TLSE-4, those two with the smallest TSSR-1 = 32.3964 come from Algorithm B and from Deming’s algorithm, TSSR-2 = 34.7002 from Algorithm A, TSSR-3 = 37.8428 from Wang’s algorithm, and TSSR-4 = 40.6573 from Algorithm D. Which of the four estimates might be the most plausible? And why did Algorithm D take so long to converge?

Computational efficiency comparisons (3/4 cont'd)

- Analyzing “Wang’s data,” using TLSE-1 or TLSE-2 for the initial values:

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	1.055186	0.864275	-0.911266	32.396447	8	1.4E-20	1.35
A	1.055186	0.864275	-0.911266	32.396447	6	1.8E-19	1.11
B	1.055186	0.864275	-0.911266	32.396447	37	1.8E-19	7.25
D	1.055186	0.864275	-0.911266	32.396447	9	1.1E-26	1.47
Deming	1.055186	0.864275	-0.911266	32.396447	26	2.8E-19	6.12

$\xi_0 = [1.055186, 0.864275, -0.911266]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	1.051499	0.575898	-0.619968	34.702215	8	3.7E-20	0.70
A	1.051499	0.575898	-0.619968	34.702215	4	1.8E-19	0.83
B	1.051499	0.575898	-0.619968	34.702215	57	4.1E-20	9.33
D	1.051499	0.575898	-0.619968	34.702215	13	2.9E-27	1.92
Deming	1.051499	0.575898	-0.619968	34.702215	48	4.4E-19	9.83

$\xi_0 = [1.051499, 0.575898, -0.619968]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

All the TSSR's-1 and TLSE's-1 as well as all the TSSR's-2 and TLSE's-2 do coincide; all the algorithms converge in reasonable time to meaningful estimates. At first sight, TLSE-2 may appear superior to TLSE-1, but the TSSR-2 is only the second smallest!

Computational efficiency comparisons (4/4)

- Analyzing “Wang’s data;” using TLSE-3 for the initial values; TLSE-4 seems unlikely to matter:

Alg.	$\hat{\xi}_1$	$\hat{\xi}_2$	$\hat{\xi}_3$	TSSR	Iter.	mod. ck.	t [s]
Wang	0.977246	-1.016296	1.042873	37.842832	15	1.5E-20	1.35
A	0.977246	-1.016296	1.042873	37.842832	14	1.2E-19	2.21
B	1.055186	0.864275	-0.911266	32.396447	89	5.8E-20	13.45
D	0.804878	0.051626	0.148681	40.737315	102	3.7E-28	12.78
Deming	0.977246	-1.016296	1.042873	37.842832	38	2.0E-19	7.61

$\xi_0 = [0.977246, -1.016296, 1.042873]^T$, conv. criterion: 1.E-10, t is for 1,000 runs

- Three algorithms converge to the same TLSE-3 that served for initial values; but algorithms B and D did not. In fact, B converged to TSSR-1 and TLSE-1.
- It turns out that only TSSR’s-1 and TSSR’s-2 may serve as meaningful solutions for “Wang’s data.” At this stage, it cannot be decided whether one solution is superior to the other; but TLSE-3 and TLSE-4 appear unsuitable if we look for the “best choice.”

- While working with five algorithms to solve a nonlinear system of equations for the best estimates of the parameters that should most suitably describe an auto-regressive process of type AR(3) — using two different datasets that have been analyzed before with Total Least-Squares estimation within an EIV-Model — it became apparent that multiple local optima may result from different algorithms, depending on the choice of the initial values before iteration.
- In one case (“Wang’s data”) the five algorithms produced four TSSR’s and consequently also four TLSE’s; with the TSSR’s fluctuating between 32.3964 and 40.6573, it is still difficult to determine if the best TLSE comes with the smallest TSSR.
- More research is desirable to determine whether some of the algorithms are more reliable than others; candidates may include algorithms A or D, for instance. It may also be of interest to find out why Snow’s algorithms A and B oftentimes converge to different TLSE’s, although they handle the same provided equations, just in different order.

References

- Deming, W. E. (1931). XI. The application of least squares. *Philosophical Magazine*, 11(68), 146–158.
- Deming, W. E. (1934). LXVII. On the application of least squares II. *Philosophical Magazine*, 17(114), 804–829.
- Fang, X. (2011). Weighted Total Least Squares Solutions for Applications in Geodesy, Publication No. 294, PhD diss., Dept. of Geodesy and Geoinformatics, Leibniz University Hannover, Germany.
- Han, J., Zhang, S., Li, Y., & Zhang, X. (2020). A general partial errors-in-variables model and a corresponding weighted total least-squares algorithm. *Survey Review*, 52(371), 126–133.
- Jazaeri, S., Schaffrin, B., & Snow, K. (2023). A simple TLS-treatment of the partial EIV-model as one with singular cofactor matrices I: The case of a Kronecker product for $Q_A = Q_0 \otimes Q_x$. In J. T. Freymueller & L. Sánchez (Eds.), *X Hotine-Marussi Symposium on Mathematical Geodesy* (pp. 83–91). Springer Intl. Publishing, IAG Symp. Vol. 155.
- Mahboub, V. (2012). On weighted total least-squares for geodetic transformations. *J. of Geodesy*, 86, 359–367.
- Neitzel, F., & Schaffrin, B. (2016). On the Gauss–Helmert model with a singular dispersion matrix where BQ is of smaller rank than B [Mathematical Modeling and Computational Methods]. *J. of Computational and Applied Mathematics*, 291, 458–467.
- Schaffrin, B. (2015). Adjusting the Errors-In-Variables model: Linearized least-squares vs. nonlinear total least-squares. In N. Sneeuw, P. Novák, M. Crespi, & F. Sansò (Eds.), *VIII Hotine-Marussi Symposium on Mathematical Geodesy* (pp. 301–307). Springer International Publishing, IAG Symp. Vol 142.
- Schaffrin, B., Neitzel, F., Uzun, S., & Mahboub, V. (2012). Modifying Cadzow's algorithm to generate the optimal TLS-solution for the structured EIV-Model of a similarity transformation. *J. of Geodetic Science*, 2, 98–106.
- Schaffrin, B., & Snow, K. (2010). Total Least-Squares regularization of Tykhonov type and an ancient racetrack in Corinth. *Linear Algebra and its Applications*, 432(8), 2061–2076.
- Schaffrin, B., Snow, K., & Neitzel, F. (2014). On the Errors-In-Variables model with singular dispersion matrices. *Journal of Geodetic Science*, 4(1), 28–36.
- Snow, K. (2012). Topics in Total Least-Squares Adjustment within the Errors-In-Variables Model: Singular Cofactor Matrices and Prior Information, Report No. 502, PhD diss., Div. of Geodetic Science, School of Earth Sciences, The Ohio State University, Columbus, Ohio, USA.
- Wang, L., Xu, G., & Wen, G. (2017). A method for partial EIV model with correlated observations (in Chinese). *Cehui Xuebao/Acta Geodaetica et Cartographica Sinica*, 46(8), 978–987.
- Xu, P., Liu, J., & Shi, C. (2012). Total least-squares adjustment in partial Errors-In-Variables models: algorithm and statistical analysis. *J. of Geodesy*, 86, 661–675.